

PCE Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: September 11, 2016

H. Ananthakrishnan  
Packet Design  
S. Sivabalan  
Cisco  
C. Barth  
R. Torvi  
Juniper Networks  
I. Minei  
Google, Inc  
E. Crabbe  
March 10, 2016

PCEP Extensions for MPLS-TE LSP Path Protection with stateful PCE  
draft-ananthakrishnan-pce-stateful-path-protection-01

#### Abstract

A stateful Path Computation Element (PCE) is capable of computing as well as controlling via Path Computation Element Protocol (PCEP) Multiprotocol Label Switching Traffic Engineering Label Switched Paths (MPLS LSP). Furthermore, it is also possible for a stateful PCE to create, maintain, and delete LSPs. This document describes PCEP extension to associate two or more LSPs to provide end-to-end path protection.

#### Status of This Memo

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

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

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

This Internet-Draft will expire on September 11, 2016.

#### Copyright Notice

Copyright (c) 2016 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 . . . . .	2
2. Terminology . . . . .	3
3. PCEP Extensions . . . . .	4
3.1. Path Protection Association Type . . . . .	4
3.2. Path Protection Association TLV . . . . .	5
4. Operation . . . . .	6
4.1. PCE Initiated LSPs . . . . .	6
4.2. PCC Initiated LSPs . . . . .	6
4.3. State Synchronization . . . . .	7
4.4. Error Handling . . . . .	7
5. IANA considerations . . . . .	7
5.1. Association Type . . . . .	7
5.2. PPAG TLV . . . . .	7
5.3. PCEP Errors . . . . .	8
6. Security Considerations . . . . .	8
7. Acknowledgments . . . . .	9
8. References . . . . .	9
8.1. Normative References . . . . .	9
8.2. Information References . . . . .	10
Authors' Addresses . . . . .	11

## 1. Introduction

[RFC5440] describes PCEP for communication between a Path Computation Client (PCC) and a PCE or between one a pair of PCEs. A PCE computes paths for MPLS-TE LSPs based on various constraints and optimization criteria.

Stateful pce [I-D.ietf-pce-stateful-pce] specifies a set of extensions to PCEP to enable stateful control of paths such as MPLS TE LSPs between and across PCEP sessions in compliance with [RFC4657]. It includes mechanisms to effect LSP state synchronization between PCCs and PCEs, delegation of control of LSPs to PCEs, and PCE control of timing and sequence of path computations within and across PCEP sessions and focuses on a model where LSPs are configured on the PCC and control over them is delegated to the PCE.

Furthermore, a mechanism to dynamically instantiate LSPs on a PCC based on the requests from a stateful PCE or a controller using stateful PCE is specified in [I-D.ietf-pce-pce-initiated-lsp].

Path protection refers to a paradigm in which the working LSP is protected by one or more protection LSP(s). When the working LSP fails, protection LSP(s) is/are activated. When the working LSPs are computed and controlled by the PCE, there is benefit in a mode of operation where protection LSPs are as well.

This document specifies a stateful PCEP extension to associate two or more LSPs for the purpose of setting up path protection. The proposed extension covers the following scenarios:

1. A protection LSP is initiated on a PCC by a stateful PCE which retains the control of the LSP. The PCE is responsible for computing the path of the LSP and updating the PCC with the information about the path.
2. A PCC initiates a protection LSP and retains the control of the LSP. The PCC computes the path and updates the PCE with the information about the path as long as it controls the LSP.
3. A PCC initiates a protection LSP and delegates the control of the LSP to a stateful PCE. The PCE may compute the path for the LSP and update the PCC with the information about the path as long as it controls the LSP.

Note that protection LSP can be established prior to the failure (in which case the LSP is said to be in standby mode) or post failure of the corresponding working LSP according to the operator choice/policy.

## 2. Terminology

The following terminologies are used in this document:

AGID: Association Group ID.

ERO: Explicit Route Object.

LSP: Label Switched Path.

PCC: Path Computation Client.

PCE: Path Computation Element

PCEP: Path Computation Element Protocol.

PPAG: Path Protection Association Group.

TLV: Type, Length, and Value.

### 3. PCEP Extensions

#### 3.1. Path Protection Association Type

LSPs are not associated by listing the other LSPs with which they interact, but rather by making them belong to an association group referred to as "Path Protection Association Group" (PPAG) in this document. All LSPs join a PPAG individually. PPAG is based on the generic Association object used to associate two or more LSPs specified in [I-D.ietf-pce-association-group]. A member of a PPAG can take the role of working or protection LSP. This document defines a new association type called "Path Protection Association Type" of value TBD1. A PPAG can have one working LSP and/or one or more protection LSPs. The source and destination of all LSPs within a PPAG MUST be the same.

The format of the Association object used for PPAG is specified in [I-D.ietf-pce-association-group] and replicated in this document for easy reference in Figure 1 and Figure 2.

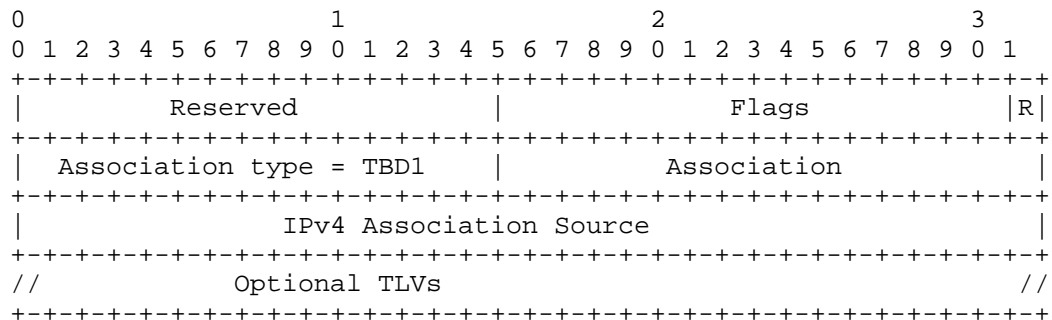


Figure 1: PPAG IPv4 ASSOCIATION Object format

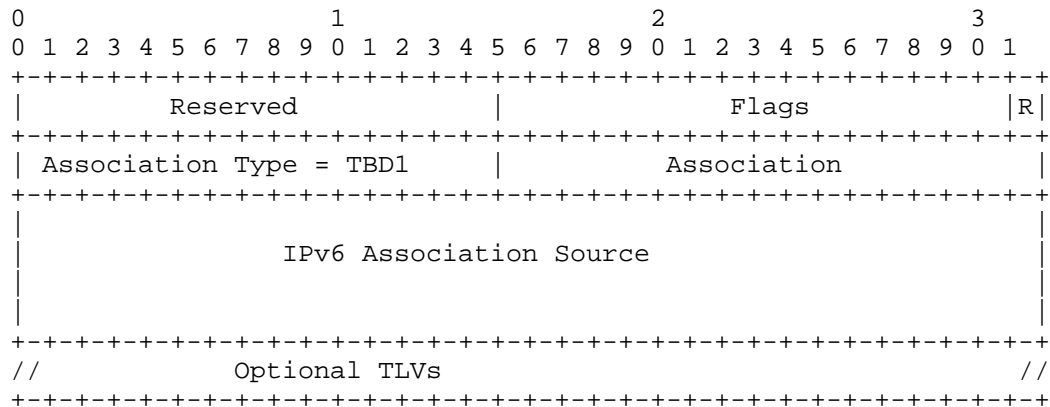


Figure 2: PPAG IPv6 ASSOCIATION Object format

This document defines a new Association type, the Path Protection Association type, value will be assigned by IANA (TBD1).

### 3.2. Path Protection Association TLV

The Path Protection Association TLV is an optional TLV for use with the Path Protection Association Object Type. The Path Protection Association TLV MUST NOT be present more than once. If it appears more than once, only the first occurrence is processed and any others MUST be ignored.

The Path Protection Association TLV follows the PCEP TLV format of [RFC5440].

The type (16 bits) of the TLV is to be assigned by IANA. The length field is 16 bit-long and has a fixed value of 4.

The value comprises a single field, the Path Protection Association Flags (32 bits), where each bit represents a flag option.

The format of the Path Protection Association TLV (Figure 3) is as follows:

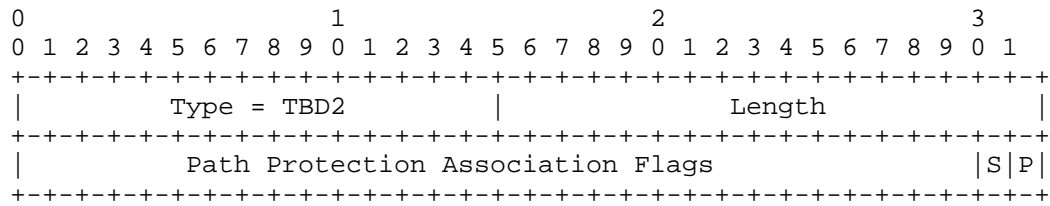


Figure 3: Path Protection Association TLV format

P (PROTECTION-LSP 1 bit) - Indicates whether the LSP associated with the PPAG is working or protection LSP. If this flag is set, the LSP is a protection LSP.

S (STANDBY 1 bit)- When the P flag is set, the S flag indicates whether the protection LSP associated with the PPAG is in standby mode. The S flag is ignored if the P flag is not set.

If the Path Protection Association TLV is missing, it means the LSP is the working LSP.

#### 4. Operation

##### 4.1. PCE Initiated LSPs

A PCE can create/update working and protection LSPs independently. As specified in [I-D.ietf-pce-association-group], Association Groups can be created by both PCE and PCC.

A PCE can remove a protection LSP from a PPAG as specified in [I-D.ietf-pce-association-group].

##### 4.2. PCC Initiated LSPs

A PCC can associate a set of LSPs under its control for path protection purpose. Similarly, the PCC can remove on or more LSPs under its control from the corresponding PPAG. In both cases, the PCC must report the change in association to PCE(s) via PCRpt message.

A stateless PCC can request protection to a PCE thorough PCReq message.

### 4.3. State Synchronization

During state synchronization, a PCC MUST report all the existing path protection association groups as well as any path protection flags to PCE(s). Following the state synchronization, the PCE MUST remove all stale path protection associations.

### 4.4. Error Handling

All LSPs (working or protection) within a PPAG MUST have the same source and destination. If a PCE attempts to add an LSP to a PPAG and the source and/or destination of the LSP is/are different from the LSP(s) in the PPAG, the PCC MUST send PCErr with Error-Type= TBD3 (Path Protection Association Error) and Error-Value = 1 (End points mismatch).

There MUST be only one working LSP within a PPAG. If a PCEP Speaker attempts to add another working LSP, the PCEP peer MUST send PCErr with Error-Type=TBD3(Path Protection Association Error) and Error-Value = 2 (Attempt to add another working LSP).

## 5. IANA considerations

### 5.1. Association Type

This document defines a new association type for path protection as follows:

Association Type Value	Association Name	Reference
TBD1 (Suggested value - 1)	Path Protection Association	This document

### 5.2. PPAG TLV

This document defines a new TLV for carrying additional information of LSPs within a path protection association group as follows:

TLV Type Value	TLV Name	Reference
TBD2 (suggested Value - 29)	Path Protection Association Group TLV	This document

This document requests that a new sub-registry, named "Path protection Association Group TLV Flag Field", is created within the "Path Computation Element Protocol (PCEP) Numbers" registry to manage the Flag field in the Path Protection Association Group TLV. New values are to be assigned by Standards Action [RFC5226]. Each bit should be tracked with the following qualities:

Each bit should be tracked with the following qualities:

- o Bit number (count from 0 as the most significant bit)
- o Name flag
- o Reference

Bit Number	Name	Reference
31	P - PROTECTION-LSP	This document
30	S - STANDBY	This document

Table 1: PPAG TLV

### 5.3. PCEP Errors

This document defines new Error-Type and Error-Value related to path protection association as follows:

Error-Type	Meaning
TBD3 (suggested value - 25)	Path Protection Association error:  Error-value=1: End-Points mismatch Error-value=2: Attempt to add another working LSP

### 6. Security Considerations

The same security considerations apply in head end as described in [I-D.ietf-pce-pce-initiated-lsp]



## 7. Acknowledgments

We would like to thank Jeff Tantsura, Dhruv Dhody and Zhangxian for their contributions to this document.

## 8. References

### 8.1. Normative References

- [I-D.ietf-pce-association-group]  
Minei, I., Crabbe, E., Sivabalan, S., Ananthakrishnan, H., Zhang, X., and Y. Tanaka, "PCEP Extensions for Establishing Relationships Between Sets of LSPs", draft-ietf-pce-association-group-00 (work in progress), November 2015.
- [I-D.ietf-pce-pce-initiated-lsp]  
Crabbe, E., Minei, I., Sivabalan, S., and R. Varga, "PCEP Extensions for PCE-initiated LSP Setup in a Stateful PCE Model", draft-ietf-pce-pce-initiated-lsp-05 (work in progress), October 2015.
- [I-D.ietf-pce-stateful-pce]  
Crabbe, E., Minei, I., Medved, J., and R. Varga, "PCEP Extensions for Stateful PCE", draft-ietf-pce-stateful-pce-13 (work in progress), December 2015.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<http://www.rfc-editor.org/info/rfc2119>>.
- [RFC2205] Braden, R., Ed., Zhang, L., Berson, S., Herzog, S., and S. Jamin, "Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification", RFC 2205, DOI 10.17487/RFC2205, September 1997, <<http://www.rfc-editor.org/info/rfc2205>>.
- [RFC3209] Awduche, D., Berger, L., Gan, D., Li, T., Srinivasan, V., and G. Swallow, "RSVP-TE: Extensions to RSVP for LSP Tunnels", RFC 3209, DOI 10.17487/RFC3209, December 2001, <<http://www.rfc-editor.org/info/rfc3209>>.
- [RFC4090] Pan, P., Ed., Swallow, G., Ed., and A. Atlas, Ed., "Fast Reroute Extensions to RSVP-TE for LSP Tunnels", RFC 4090, DOI 10.17487/RFC4090, May 2005, <<http://www.rfc-editor.org/info/rfc4090>>.

- [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, <<http://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, <<http://www.rfc-editor.org/info/rfc5089>>.
- [RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, DOI 10.17487/RFC5226, May 2008, <<http://www.rfc-editor.org/info/rfc5226>>.
- [RFC5440] Vasseur, JP., Ed. and JL. Le Roux, Ed., "Path Computation Element (PCE) Communication Protocol (PCEP)", RFC 5440, DOI 10.17487/RFC5440, March 2009, <<http://www.rfc-editor.org/info/rfc5440>>.
- [RFC5511] Farrel, A., "Routing Backus-Naur Form (RBNF): A Syntax Used to Form Encoding Rules in Various Routing Protocol Specifications", RFC 5511, DOI 10.17487/RFC5511, April 2009, <<http://www.rfc-editor.org/info/rfc5511>>.

## 8.2. Information References

- [RFC2702] Awduche, D., Malcolm, J., Agogbua, J., O'Dell, M., and J. McManus, "Requirements for Traffic Engineering Over MPLS", RFC 2702, DOI 10.17487/RFC2702, September 1999, <<http://www.rfc-editor.org/info/rfc2702>>.
- [RFC3031] Rosen, E., Viswanathan, A., and R. Callon, "Multiprotocol Label Switching Architecture", RFC 3031, DOI 10.17487/RFC3031, January 2001, <<http://www.rfc-editor.org/info/rfc3031>>.
- [RFC3346] Boyle, J., Gill, V., Hannan, A., Cooper, D., Awduche, D., Christian, B., and W. Lai, "Applicability Statement for Traffic Engineering with MPLS", RFC 3346, DOI 10.17487/RFC3346, August 2002, <<http://www.rfc-editor.org/info/rfc3346>>.
- [RFC3630] Katz, D., Kompella, K., and D. Yeung, "Traffic Engineering (TE) Extensions to OSPF Version 2", RFC 3630, DOI 10.17487/RFC3630, September 2003, <<http://www.rfc-editor.org/info/rfc3630>>.

- [RFC4655] Farrel, A., Vasseur, J., and J. Ash, "A Path Computation Element (PCE)-Based Architecture", RFC 4655, DOI 10.17487/RFC4655, August 2006, <<http://www.rfc-editor.org/info/rfc4655>>.
- [RFC4657] Ash, J., Ed. and J. Le Roux, Ed., "Path Computation Element (PCE) Communication Protocol Generic Requirements", RFC 4657, DOI 10.17487/RFC4657, September 2006, <<http://www.rfc-editor.org/info/rfc4657>>.
- [RFC5305] Li, T. and H. Smit, "IS-IS Extensions for Traffic Engineering", RFC 5305, DOI 10.17487/RFC5305, October 2008, <<http://www.rfc-editor.org/info/rfc5305>>.
- [RFC5394] Bryskin, I., Papadimitriou, D., Berger, L., and J. Ash, "Policy-Enabled Path Computation Framework", RFC 5394, DOI 10.17487/RFC5394, December 2008, <<http://www.rfc-editor.org/info/rfc5394>>.
- [RFC5557] Lee, Y., Le Roux, J.L., King, D., and E. Oki, "Path Computation Element Communication Protocol (PCEP) Requirements and Protocol Extensions in Support of Global Concurrent Optimization", RFC 5557, DOI 10.17487/RFC5557, July 2009, <<http://www.rfc-editor.org/info/rfc5557>>.

#### Authors' Addresses

Hariharan Ananthakrishnan  
Packet Design  
1 South Almaden Blvd, #1150,  
San Jose, CA, 95113  
USA

EMail: [hari@packetdesign.com](mailto:hari@packetdesign.com)

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

EMail: [msiva@cisco.com](mailto:msiva@cisco.com)

Colby Barth  
Juniper Networks  
1194 N Mathilda Ave,  
Sunnyvale, CA, 94086  
USA

EMail: cbarth@juniper.net

Raveendra Torvi  
Juniper Networks  
1194 N Mathilda Ave,  
Sunnyvale, CA, 94086  
USA

EMail: rtorvi@juniper.net

Ina Minei  
Google, Inc  
1600 Amphitheatre Parkway  
Mountain View, CA, 94043  
USA

EMail: inaminei@google.com

Edward Crabbe

EMail: edward.crabbe@gmail.com

PCE Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: August 31, 2018

H. Ananthakrishnan  
Packet Design  
S. Sivabalan  
Cisco  
C. Barth  
R. Torvi  
Juniper Networks  
I. Minei  
Google, Inc  
E. Crabbe  
Individual Contributor  
D. Dhody  
Huawei Technologies  
February 27, 2018

PCEP Extensions for MPLS-TE LSP Path Protection with stateful PCE  
draft-ananthakrishnan-pce-stateful-path-protection-05

## Abstract

A stateful Path Computation Element (PCE) is capable of computing as well as controlling via Path Computation Element Protocol (PCEP) Multiprotocol Label Switching Traffic Engineering Label Switched Paths (MPLS LSP). Furthermore, it is also possible for a stateful PCE to create, maintain, and delete LSPs. This document describes PCEP extension to associate two or more LSPs to provide end-to-end path protection.

## 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 31, 2018.

## Copyright Notice

Copyright (c) 2018 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. Requirements Language . . . . .	4
2. Terminology . . . . .	4
3. PCEP Extensions . . . . .	5
3.1. Path Protection Association Type . . . . .	5
3.2. Path Protection Association TLV . . . . .	5
4. Operation . . . . .	6
4.1. State Synchronization . . . . .	6
4.2. PCC Initiated LSPs . . . . .	6
4.3. PCE Initiated LSPs . . . . .	7
4.4. Session Termination . . . . .	7
4.5. Error Handling . . . . .	7
5. Other considerations . . . . .	8
6. IANA considerations . . . . .	8
6.1. Association Type . . . . .	8
6.2. PPAG TLV . . . . .	8
6.3. PCEP Errors . . . . .	9
7. Security Considerations . . . . .	10
8. Manageability Considerations . . . . .	10
8.1. Control of Function and Policy . . . . .	10
8.2. Information and Data Models . . . . .	10
8.3. Liveness Detection and Monitoring . . . . .	10
8.4. Verify Correct Operations . . . . .	10
8.5. Requirements On Other Protocols . . . . .	10
8.6. Impact On Network Operations . . . . .	11
9. Acknowledgments . . . . .	11
10. References . . . . .	11
10.1. Normative References . . . . .	11
10.2. Information References . . . . .	12
Authors' Addresses . . . . .	13

## 1. Introduction

[RFC5440] describes PCEP for communication between a Path Computation Client (PCC) and a PCE or between one a pair of PCEs as per [RFC4655]. A PCE computes paths for MPLS-TE LSPs based on various constraints and optimization criteria.

Stateful pce [RFC8231] specifies a set of extensions to PCEP to enable stateful control of paths such as MPLS TE LSPs between and across PCEP sessions in compliance with [RFC4657]. It includes mechanisms to effect LSP state synchronization between PCCs and PCEs, delegation of control of LSPs to PCEs, and PCE control of timing and sequence of path computations within and across PCEP sessions and focuses on a model where LSPs are configured on the PCC and control over them is delegated to the PCE. Furthermore, a mechanism to dynamically instantiate LSPs on a PCC based on the requests from a stateful PCE or a controller using stateful PCE, is specified in [RFC8281].

Path protection [RFC4427] refers to a paradigm in which the working LSP is protected by one or more protection LSP(s). When the working LSP fails, protection LSP(s) is/are activated. When the working LSPs are computed and controlled by the PCE, there is benefit in a mode of operation where protection LSPs are as well.

This document specifies a stateful PCEP extension to associate two or more LSPs for the purpose of setting up path protection. The proposed extension covers the following scenarios:

- o A PCC initiates a protection LSP and retains the control of the LSP. The PCC computes the path itself or makes a request for path computation to a PCE. After the path setup, it reports the information and state of the path to the PCE. This includes the association group identifying the working and protection LSPs. This is the passive stateful mode [RFC8051].
- o A PCC initiates a protection LSP and delegates the control of the LSP to a stateful PCE. During delegation the association group identifying the working and protection LSPs is included. The PCE computes the path for the protection LSP and update the PCC with the information about the path as long as it controls the LSP. This is the active stateful mode [RFC8051].
- o A protection LSP could be initiated by a stateful PCE, which retains the control of the LSP. The PCE is responsible for computing the path of the LSP and updating to the PCC with the information about the path. This is the PCE Initiated mode [RFC8281].

Note that protection LSP can be established (signaled) prior to the failure (in which case the LSP is said to be in standby mode [RFC4427]) or post failure of the corresponding working LSP according to the operator choice/policy.

[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 that is equally applicable to stateful PCE (active and passive modes) and stateless PCE.

This document specifies a PCEP extension to associate one working LSP with one or more protection LSPs using the generic association mechanism.

This document describes a PCEP extension to associate protection LSPs by creating Path Protection Association Group (PPAG) and encoding this association in PCEP messages for stateful PCEP sessions.

### 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. Terminology

The following terminologies are used in this document:

ERO: Explicit Route Object.

LSP: Label Switched Path.

PCC: Path Computation Client.

PCE: Path Computation Element

PCEP: Path Computation Element Protocol.

PPAG: Path Protection Association Group.

TLV: Type, Length, and Value.



### 3. PCEP Extensions

#### 3.1. Path Protection Association Type

LSPs are not associated by listing the other LSPs with which they interact, but rather by making them belong to an association group referred to as "Path Protection Association Group" (PPAG) in this document. All LSPs join a PPAG individually. PPAG is based on the generic Association object used to associate two or more LSPs specified in [I-D.ietf-pce-association-group]. A member of a PPAG can take the role of working or protection LSP. This document defines a new association type called "Path Protection Association Type" of value TBD1. A PPAG can have one working LSP and/or one or more protection LSPs. The source, destination and Tunnel ID (as carried in LSP-IDENTIFIERS TLV [RFC8231], with description as per [RFC3209]) of all LSPs within a PPAG MUST be the same. As per [RFC3209], TE tunnel is used to associate a set of LSPs during reroute or to spread a traffic trunk over multiple paths.

The format of the Association object used for PPAG is specified in [I-D.ietf-pce-association-group].

This document defines a new Association type, the Path Protection Association type, value will be assigned by IANA (TBD1).

This Association-Type is dynamic in nature and created by the PCC or PCE for the LSPs belonging to the same TE tunnel (as described in [RFC3209]) originating at the same head node and terminating at the same destination. These associations are conveyed via PCEP messages to the PCEP peer. Operator-configured Association Range MUST NOT be set for this association-type and MUST be ignored.

#### 3.2. Path Protection Association TLV

The Path Protection Association TLV is an optional TLV for use with the Path Protection Association Object Type. The Path Protection Association TLV MUST NOT be present more than once. If it appears more than once, only the first occurrence is processed and any others MUST be ignored.

The Path Protection Association TLV follows the PCEP TLV format of [RFC5440].

The type (16 bits) of the TLV is to be assigned by IANA. The length field is 16 bit-long and has a fixed value of 4.

The value comprises a single field, the Path Protection Association Flags (32 bits), where each bit represents a flag option.

The format of the Path Protection Association TLV (Figure 1) is as follows:

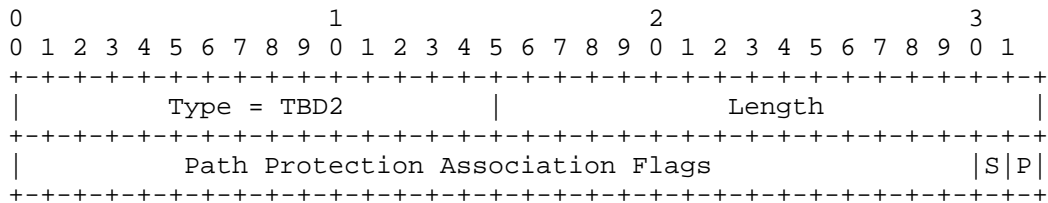


Figure 1: Path Protection Association TLV format

P (PROTECTION-LSP 1 bit) - Indicates whether the LSP associated with the PPAG is working or protection LSP. If this flag is set, the LSP is a protection LSP.

S (STANDBY 1 bit)- When the P flag is set, the S flag indicates whether the protection LSP associated with the PPAG is in standby mode. The S flag is ignored if the P flag is not set.

Unassigned bits are considered reserved. They MUST be set to 0 on transmission and MUST be ignored on receipt.

If the TLV is missing, it is considered that the LSP is the working LSP (i.e. P bit is unset).

#### 4. Operation

LSPs are associated with other LSPs with which they interact by adding them to a common association group via ASSOCIATION object. All procedures and error-handling for the ASSOCIATION object is as per [I-D.ietf-pce-association-group].

##### 4.1. State Synchronization

During state synchronization, a PCC MUST report all the existing path protection association groups as well as any path protection flags to PCE(s) as per [I-D.ietf-pce-association-group].

##### 4.2. PCC Initiated LSPs

A PCC can associate a set of LSPs under its control for path protection purpose. Similarly, the PCC can remove one or more LSPs under its control from the corresponding PPAG. In both cases, the PCC must report the change in association to PCE(s) via PCRpt message. A PCC can also delegate the working and protection LSPs to

a stateful PCE, where PCE would control the LSPs. The stateful PCE could update the paths and attributes of the LSPs in the association group via PCUpd message. A PCE could also update the association to PCC via PCUpd message. These procedures are described in [I-D.ietf-pce-association-group].

#### 4.3. PCE Initiated LSPs

A PCE can create/update working and protection LSPs independently. As specified in [I-D.ietf-pce-association-group], Association Groups can be created by both PCE and PCC. Further, a PCE can remove a protection LSP from a PPAG as specified in [I-D.ietf-pce-association-group]. The PCE uses PCUpd or PCInitiate message to communicate the association information to the PCC.

#### 4.4. Session Termination

As per [I-D.ietf-pce-association-group] the association information is cleared along with the LSP state information. When a PCEP session is terminated, after expiry of State Timeout Interval at PCC, the LSP state associated with that PCEP session is reverted to operator-defined default parameters or behaviors as per [RFC8231]. Same procedure is also followed for the association information. On session termination at the PCE, when the LSP state reported by PCC is cleared, the association information is also cleared as per [I-D.ietf-pce-association-group]. Where there are no LSPs in a association group, the association is considered to be deleted..

#### 4.5. Error Handling

All LSPs (working or protection) within a PPAG MUST belong to the same TE Tunnel (as described in [RFC3209]) and have the same source and destination. If a PCEP speaker attempts to add an LSP to a PPAG and the Tunnel ID (as carried in LSP-IDENTIFIERS TLV [RFC8231], with description as per [RFC3209]) or source or destination of the LSP is different from the LSP(s) in the PPAG, the PCC MUST send PCErr with Error-Type= 29 (Early allocation by IANA) (Association Error) [I-D.ietf-pce-association-group] and Error-Value = TBD3 (Tunnel ID or End points mismatch for Path Protection Association).

There MUST be only one working LSP within a PPAG. If a PCEP Speaker attempts to add another working LSP, the PCEP peer MUST send PCErr with Error-Type=29 (Early allocation by IANA) (Association Error) [I-D.ietf-pce-association-group] and Error-Value = TBD4 (Attempt to add another working LSP for Path Protection Association).

## 5. Other considerations

The working and protection LSPs are typically resource disjoint (e.g., node, srlg disjoint). This ensures that a single failure will not affect both the working and protection LSPs. The disjoint requirement for a group of LSPs is handled via another association type called "Disjointness Association", as described in [I-D.ietf-pce-association-diversity]. The diversity requirements for the the protection LSP are also handled by including both ASSOCIATION object identifying both the protection association group and disjoint association group for the group of LSPs.

## 6. IANA considerations

### 6.1. Association Type

This document defines a new association type, originally defined in [I-D.ietf-pce-association-group], for path protection. IANA is requested to make the assignment of a new value for the sub-registry "ASSOCIATION Type Field" (request to be created in [I-D.ietf-pce-association-group]), as follows:

Association Type Value	Association Name	Reference
TBD1	Path Protection Association	This document

### 6.2. PPAG TLV

This document defines a new TLV for carrying additional information of LSPs within a path protection association group. IANA is requested to make the assignment of a new value for the existing "PCEP TLV Type Indicators" registry as follows:

TLV Type Value	TLV Name	Reference
TBD2	Path Protection Association Group TLV	This document

This document requests that a new sub-registry, named "Path protection Association Group TLV Flag Field", is created within the "Path Computation Element Protocol (PCEP) Numbers" registry to manage

the Flag field in the Path Protection Association Group TLV. New values are to be assigned by Standards Action [RFC8126]. Each bit should be tracked with the following qualities:

Each bit should be tracked with the following qualities:

- o Bit number (count from 0 as the most significant bit)
- o Name flag
- o Reference

Bit Number	Name	Reference
31	P - PROTECTION-LSP	This document
30	S - STANDBY	This document

Table 1: PPAG TLV

### 6.3. PCEP Errors

This document defines new Error-Type and Error-Value related to path protection association. IANA is requested to allocate new error values within the "PCEP-ERROR Object Error Types and Values" sub-registry of the PCEP Numbers registry, as follows:

Error-Type	Meaning	Reference
29	Association error Error-value=TBD3: Tunnel ID or End points mismatch for Path Protection Association	[I-D.ietf-pce-association-group] This document
	Error-value=TBD4: Attempt to add another working LSP for Path Protection Association	This document

## 7. Security Considerations

The security considerations described in [RFC8231], [RFC8281], and [RFC5440] apply to the extensions described in this document as well. Additional considerations related to associations where a malicious PCEP speaker could be spoofed and could be used as an attack vector by creating associations is described in [I-D.ietf-pce-association-group]. Thus securing the PCEP session using Transport Layer Security (TLS) [RFC8253], as per the recommendations and best current practices in [RFC7525], is RECOMMENDED.

## 8. Manageability Considerations

### 8.1. Control of Function and Policy

Mechanisms defined in this document do not imply any control or policy requirements in addition to those already listed in [RFC5440], [RFC8231], and [RFC8281].

### 8.2. Information and Data Models

[RFC7420] describes the PCEP MIB, there are no new MIB Objects for this document.

The PCEP YANG module [I-D.ietf-pce-pcep-yang] supports associations.

### 8.3. 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], [RFC8231], and [RFC8281].

### 8.4. Verify Correct Operations

Mechanisms defined in this document do not imply any new operation verification requirements in addition to those already listed in [RFC5440], [RFC8231], and [RFC8281].

### 8.5. Requirements On Other Protocols

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

## 8.6. 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], [RFC8231], and [RFC8281].

## 9. Acknowledgments

We would like to thank Jeff Tantsura and Xian Zhang for their contributions to 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>>.
- [RFC3209] Awduche, D., Berger, L., Gan, D., Li, T., Srinivasan, V., and G. Swallow, "RSVP-TE: Extensions to RSVP for LSP Tunnels", RFC 3209, DOI 10.17487/RFC3209, December 2001, <<https://www.rfc-editor.org/info/rfc3209>>.
- [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>>.
- [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>>.
- [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-04 (work in progress), August 2017.

## 10.2. Information References

[RFC4427] Mannie, E., Ed. and D. Papadimitriou, Ed., "Recovery (Protection and Restoration) Terminology for Generalized Multi-Protocol Label Switching (GMPLS)", RFC 4427, DOI 10.17487/RFC4427, March 2006, <<https://www.rfc-editor.org/info/rfc4427>>.

[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>>.

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

[RFC7420] Koushik, A., Stephan, E., Zhao, Q., King, D., and J. Hardwick, "Path Computation Element Communication Protocol (PCEP) Management Information Base (MIB) Module", RFC 7420, DOI 10.17487/RFC7420, December 2014, <<https://www.rfc-editor.org/info/rfc7420>>.

[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>>.

[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-06 (work in progress), January 2018.

[I-D.ietf-pce-association-diversity]  
Litkowski, S., Sivabalan, S., Barth, C., and D. Dhody,  
"Path Computation Element communication Protocol extension  
for signaling LSP diversity constraint", draft-ietf-pce-  
association-diversity-03 (work in progress), February  
2018.

#### Authors' Addresses

Hariharan Ananthakrishnan  
Packet Design  
1 South Almaden Blvd, #1150,  
San Jose, CA, 95113  
USA

EMail: [hari@packetdesign.com](mailto:hari@packetdesign.com)

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

EMail: [msiva@cisco.com](mailto:msiva@cisco.com)

Colby Barth  
Juniper Networks  
1194 N Mathilda Ave,  
Sunnyvale, CA, 94086  
USA

EMail: [cbarth@juniper.net](mailto:cbarth@juniper.net)

Raveendra Torvi  
Juniper Networks  
1194 N Mathilda Ave,  
Sunnyvale, CA, 94086  
USA

EMail: rtorvi@juniper.net

Ina Minei  
Google, Inc  
1600 Amphitheatre Parkway  
Mountain View, CA, 94043  
USA

EMail: inaminei@google.com

Edward Crabbe  
Individual Contributor

EMail: edward.crabbe@gmail.com

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

EMail: dhruv.ietf@gmail.com

PCE Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: September 19, 2016

H. Chen  
Huawei Technologies  
M. Toy  
Comcast  
L. Liu  
Fujitsu  
V. Liu  
China Mobile  
March 18, 2016

PCE Hierarchical SDNs  
draft-chen-pce-h-sdns-00

Abstract

This document presents extensions to the Path Computation Element Communication Protocol (PCEP) for supporting a hierarchical SDN control system, which comprises multiple SDN controllers controlling a network with a number of domains.

Status of This Memo

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

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

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

This Internet-Draft will expire on September 19, 2016.

Copyright Notice

Copyright (c) 2016 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 . . . . .	4
2. Terminology . . . . .	4
3. Conventions Used in This Document . . . . .	6
4. Requirements . . . . .	6
5. Overview of Hierarchical SDN Control System . . . . .	6
6. Extensions to PCEP . . . . .	9
6.1. Capability Discovery . . . . .	9
6.2. New Messages for Hierarchical SDN Control System . . . . .	10
6.2.1. Contents of Messages . . . . .	12
6.2.2. Individual Encoding of Messages . . . . .	24
6.2.3. Group Encoding of Messages . . . . .	25
6.2.4. Embedded Encoding of Messages . . . . .	26
6.2.5. Mixed Encoding of Messages . . . . .	27
6.3. Controller Relation Discovery . . . . .	27
6.3.1. Using Open Message . . . . .	27
6.3.2. Using Discovery Message . . . . .	29
6.4. Connections and Accesses Advertisement . . . . .	30
6.5. Tunnel Creation . . . . .	30
6.5.1. Computing Path in Two Rounds . . . . .	31
6.5.2. Computing Path in One Round . . . . .	32
6.5.3. Creating Tunnel along Path . . . . .	34
6.6. Objects and TLVs . . . . .	36
6.6.1. CRP Objects . . . . .	36
6.6.2. LOCAL-CONTROLLER Object . . . . .	37
6.6.3. REMOTE-CONTROLLER Object . . . . .	38
6.6.4. CONNECTION and ACCESS Object . . . . .	40
6.6.5. NODE Object . . . . .	47
6.6.6. TUNNEL Object . . . . .	53
6.6.7. STATUS Object . . . . .	54
6.6.8. LABEL Object . . . . .	54
6.6.9. INTERFACE Object . . . . .	55
7. Security Considerations . . . . .	56
8. IANA Considerations . . . . .	56
9. Acknowledgement . . . . .	56
10. References . . . . .	56
10.1. Normative References . . . . .	56
10.2. Informative References . . . . .	57
Appendix A. Details on Embedded Encoding of Messages . . . . .	58
A.1. Message for Controller Relation Discovery . . . . .	58
A.2. Message for Connections and Accesses Advertisement . . . . .	60
A.3. Request for Computing Path Segments . . . . .	60

A.4. Reply for Computing Path Segments . . . . .	61
A.5. Request for Removing Path Segments . . . . .	61
A.6. Reply for Removing Path Segments . . . . .	62
A.7. Request for Keeping Path Segments . . . . .	62
A.8. Reply for Keeping Path Segments . . . . .	63
A.9. Request for Creating Tunnel Segment . . . . .	63
A.10. Reply for Creating Tunnel Segment . . . . .	64
A.11. Request for Removing Tunnel Segment . . . . .	64
A.12. Reply for Removing Tunnel Segment . . . . .	65

## 1. Introduction

A domain is a collection of network elements within a common sphere of address management or routing procedure which are operated by a single organization or administrative authority. Examples of such domains include IGP (OSPF or IS-IS) areas and Autonomous Systems.

For scalability, security, interoperability and manageability, a big network is organized as a number of domains. For example, a big network running OSPF as routing protocol is organized as a number of OSPF areas. A network running BGP is organized as multiple Autonomous Systems, each of which has a number of IGP areas.

The concepts of Software Defined Networks (SDN) have been shown to reduce the overall network CapEx and OpEx, whilst facilitating the deployment of services and enabling new features. The core principles of SDN include: centralized control to allow optimized usage of network resources and provisioning of network elements across domains.

For a network with a number of domains, it is natural to have multiple SDN controllers, each of which controls a domain in the network. To achieve a centralized control on the network, a hierarchical architecture of controllers is a good fit. At top level of the hierarchy, it is a parent controller that is not a child controller. The parent controller controls a number of child controllers. Some of these child controllers are not parent controllers. Each of them controls a domain. Some other child controllers are also parent controllers, each of which controls multiple child controllers, and so on.

This document presents extensions to the Path Computation Element Communication Protocol (PCEP) for supporting a hierarchical SDN control system, which comprises multiple SDN controllers controlling a network with a number of domains.

## 2. Terminology

The following terminology is used in this document.

ABR: Area Border Router. Router used to connect two IGP areas (Areas in OSPF or levels in IS-IS).

ASBR: Autonomous System Border Router. Router used to connect together ASes of the same or different service providers via one or more inter-AS links.

BN: Boundary Node. A boundary node is either an ABR in the context of inter-area Traffic Engineering or an ASBR in the context of inter-AS Traffic Engineering. A Boundary Node is also called an Edge Node.

Entry BN of domain(n): a BN connecting domain(n-1) to domain(n) along the path found from the source node to the BN, where domain(n-1) is the previous hop (or upstream) domain of domain(n). An Entry BN is also called an in-BN or in-edge node.

Exit BN of domain(n): a BN connecting domain(n) to domain(n+1) along the path found from the source node to the BN, where domain(n+1) is the next hop (or downstream) domain of domain(n). An Exit BN is also called a out-BN or out-edge node.

Source Domain: For a tunnel from a source to a destination, the domain containing the source is the source domain for the tunnel.

Destination Domain: For a tunnel from a source to a destination, the domain containing the destination is the destination domain for the tunnel.

Source Controller: A controller controlling the source domain.

Destination Controller: A controller controlling the destination domain.

Parent Controller: A parent controller is a controller that communicates with a number of child controllers and controls a network with multiple domains through the child controllers. A PCE can be enhanced to be a parent controller.

Child Controller: A child controller is a controller that communicates with one parent controller and controls a domain in a network. A PCE can be enhanced to be a child controller.

Exception list: An exception list for a domain contains the nodes in the domain and its adjacent domains that are on the shortest path tree (SPT) that the parent controller is building.

GTID: Global Tunnel Identifier. It is used to identify a tunnel in a network.

PID: Path Identifier. It is used to identify a path for a tunnel in a network.

Inter-area TE LSP: a TE LSP that crosses an IGP area boundary.

Inter-AS TE LSP: a TE LSP that crosses an AS boundary.

LSP: Label Switched Path

LSR: Label Switching Router

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.

PCE(i): a PCE with the scope of domain(i).

TED: Traffic Engineering Database.

This document uses terminology defined in [RFC5440].

### 3. 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 [RFC2119].

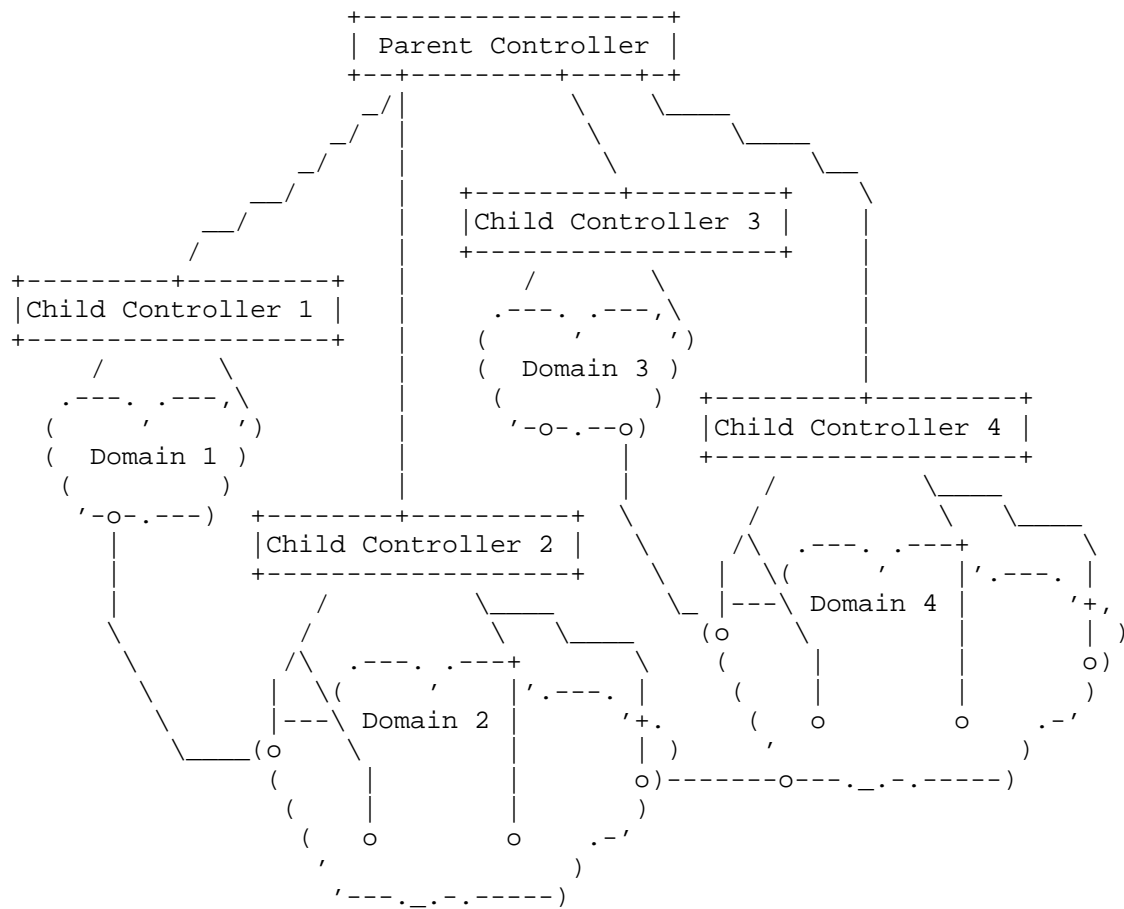
### 4. Requirements

This section summarizes the requirements for Hierarchical SDN Control System (need more text here).

### 5. Overview of Hierarchical SDN Control System

The Figure below illustrates a hierarchical SDN control system. There is one Parent Controller and four Child Controllers: Child Controller 1, Child Controller 2, Child Controller 3 and Child Controller 4.





The parent controller communicates with these four child controllers and controls them, each of which controls (or is responsible for) a domain. Child controller 1 controls domain 1, Child controller 2 controls domain 2, Child controller 3 controls domain 3, and Child controller 4 controls domain 4.

One level of hierarchy of controllers is illustrated in the figure above. There is one parent controller at top level, which is not a child controller. Under the parent controller, there are four child controllers, which are not parent controllers.

In a general case, at top level there is one parent controller that is not a child controller, there are some controllers that are both parent controllers and child controllers, and there are a number of child controllers that are not parent controllers. This is a system

of multiple levels of hierarchies, in which one parent controller controls or communicates with a first number of child controllers, some of which are also parent controllers, each of which controls or communicates with a second number of child controllers, and so on.

Considering one parent controller and its child controllers, each of the child controllers controls a domain and has the topology information on the domain, the parent controller does not have the topology information on any domain controlled by a child controller normally. This is called parent without domain topology.

In some special cases, the parent controller has the topology information on a region consisting of the domains controlled by its child controllers. In other words, the parent controller has the topology information on the domains controlled by its child controllers and the topology/inter-connections among these domains. This is called parent with domain topology.

The parent controller receives requests for creating end to end tunnels from users or applications. For each request, the parent controller is responsible for obtaining a path for the tunnel and creating the tunnel along the path.

For parent without domain topology, the parent controller asks each of its related child controllers to compute path segments from an entry boundary node to exit boundary nodes in the domain it controls or path segments from an exit boundary node in its domain to entry boundary nodes of other adjacent domains just using the inter-domain links attached to the exit boundary node. The details of the segments are hidden from the parent, which sees each of the segments as a link from a boundary node to another boundary node with a cost. The parent controller builds a shortest path tree (SPT) using the path segments computed as links to get the end to end path and then creates the tunnel along the path by asking its related child controllers.

The end to end path does not have any details from the parent's point of view. It can be considered as a sequence of domains containing the shortest path. Along this sequence of domains, the details of the end to end path can be obtained. And then the tunnel along the path with details can be created.

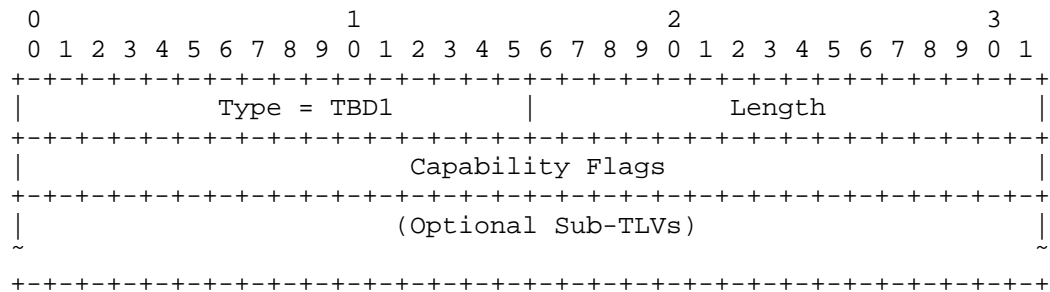
For parent with domain topology, the parent controller computes a path for the tunnel using the topology information on the domains controlled by its child controllers. And then it creates the tunnel along the path computed through asking its related child controllers.

## 6. Extensions to PCEP

This section describes the extensions to PCEP for a Hierarchical SDN Control System (HSCS). The extensions include the definition of a new flag in the RP object, a global tunnel identifier (GTID), a path identifier (PID), a list of path segments and an exception list in the PCReq and PCRep message.

### 6.1. Capability Discovery

During a PCEP session establishment between two PCEP speakers (PCE or PCC), each of them advertises its capabilities for HSCS through the Open Message with the Open Object containing a new TLV to indicate its capabilities for HSCS. This new TLV is called HSCS capability TLV. It has the following format.



The type of the TLV is TBD1. It has a length of 4 octets plus the size of optional Sub-TLVs. The value of the TLV comprises a capability flags field of 32 bits, which are numbered from the most significant as bit zero. Each bit represents a capability.

- o PC (Parent Controller - 1 bit): Bit 0 is used as PC flag. It is set to 1 indicating a parent controller.
- o CC (Child Controller - 1 bit): Bit 1 is used as PC flag. It is set to 1 indicating a child controller.
- o PS (Path Segments - 1 bit): Bit 2 is used as PS flag. It is set to 1 indicating support for computing path segments for HSCS
- o TS (Tunnel Segment - 1 bit): Bit 3 is used as TS flag. It is set to 1 indicating support for creating tunnel segment for HSCS

- o ET (End to end Tunnel - 1 bit): Bit 4 is used as ET flag. It is set to 1 indicating support for creation and maintenance of end to end LSP tunnels

## 6.2. New Messages for Hierarchical SDN Control System

This section describes the contents and semantics of the new messages, and presents a few of different encodings for the messages.

There are a number of new messages for supporting HSCS. These new messages can be encoded in a few of ways as follows:

- o To use a new type at top level for each of the new messages. This is called individual encoding.
- o To use a new type at top level for each group of the new messages and a option/operation/sub-type value for every message in the group. This is called group encoding.
- o To use/re-use existing messages and a value of options/operations for each new message in an existing message. This is called embedded encoding.
- o To combine the ways above. This is called mixed encoding.

Various types of messages for supporting HSCS are listed below. Note that many new messages may not be needed for some procedures/options. For example, four messages Request and Reply for Removing Path Segments and Request and Reply for Keeping Path Segments are not needed if path segments computed are not stored/remembered by a child controller. But in this case, the path segment in each domain along the end to end path computed needs to be re-computed when a tunnel along the path is set up.

**Message for Controller Relation Discovery:** It is a message exchanged between a parent controller and a child controller for discovering their parent-child relation.

**Message for Connections and Accesses Advertisement:** It is a message that a child controller sends its parent controller to describe the connections from the domain it controls to its adjacent domains and the access points in the domain to be accessible outside of the domain.

**Request for Computing Path Segments:** It is a message that a parent controller sends a child controller to request the child controller for computing path segments in the domain the child controller controls.

Reply for Computing Path Segments: It is a message that a child controller sends a parent controller to reply the parent controller for a request message for computing path segments after receiving the request message from the parent controller for computing path segments and computing path segments as requested, which normally contains the path segments computed.

Request for Removing Path Segments: It is a message that a parent controller sends a child controller to request the child controller for removing the path segments computed by the child controller and stored in the child controller.

Reply for Removing Path Segments: It is a message that a child controller sends a parent controller to reply the parent controller for a request message for removing a set of path segments after receiving the request message from the parent controller for removing path segments and removing the path segments as requested, which normally contains a status of removing path segments.

Request for Keeping Path Segments: It is a message that a parent controller sends a child controller to request the child controller for keeping a set of path segments computed by the child controller and stored in the child controller.

Reply for Keeping Path Segments: It is a message that a child controller sends a parent controller to reply the parent controller for a request message for keeping path segments after receiving the request message from the parent controller for keeping path segments and keeping the path segments as requested, which normally contains a status of keeping path segments.

Request for Creating Tunnel Segment: It is a message that a parent controller sends a child controller to request the child controller for creating tunnel segments related to the domain the child controller controls.

Reply for Creating Tunnel Segment: It is a message that a child controller sends a parent controller to reply the parent controller for a request message for creating tunnel segment after receiving the request message from the parent controller for creating tunnel segment and creating tunnel segment as requested, which normally contains a status of creating tunnel segment and a label and an interface.

**Request for Removing Tunnel Segment:** It is a message that a parent controller sends a child controller to request the child controller for removing the tunnel segment created by the child controller.

**Reply for Removing Tunnel Segment:** It is a message that a child controller sends a parent controller to reply the parent controller for a request message for removing tunnel segment after receiving the request message from the parent controller for removing tunnel segment and removing the tunnel segment as requested, which normally contains a status of removing tunnel segment.

#### 6.2.1. Contents of Messages

This section describes the contents in each of the messages and gives the format of each of messages in individual encoding, which is the same as in group encoding. Some of the objects in the messages are defined in the following sections.

##### 6.2.1.1. Message for Controller Relation Discovery

A message for controller relation discovery is exchanged between a parent controller and a child controller for discovering their parent-child relation.

A message for controller relation discovery (CRDis message for short) sent from a local controller to a remote controller comprises:

- o Local controller attributes
- o Remote controller attributes after the local controller receives the remote controller attributes from a remote end and determines that the relation between the local controller and the remote controller can be formed.

The format of the CRDis message is as follows:

```
<CRDis Message> ::= <Common Header>
                        <CRP>
                        <Local-Controller>
                        [<Remote-Controller>]
```

where CRP (Controller Request Parameters) object is defined in section Objects and TLVs.

#### 6.2.1.2. Message for Connections and Accesses Advertisement

After a child controller discovers its parent controller, it sends its parent controller a message for connections and accesses advertisement.

A message for connections and accesses advertisement (CAAdv message for short) from a child controller comprises:

- o Inter-domain links from the domain the child controller controls to its adjacent domains.
- o The addresses in the domain to be accessible to the outside of the domain.
- o Attributes of each of the boundary nodes of the domain.

The format of the CAAdv message is as follows:

```
<CAAdv Message> ::= <Common Header>
                    <CRP>
                    <Inter-Domain-Link-List>
                    [<Access-Address-List>]
where:
<Inter-Domain-Link-List> ::= <Inter-Domain-Link>
                             [<Inter-Domain-Link-List>]
<Access-Address-List> ::= <Access-Address>
                          [<Access-Address-List>]
```

#### 6.2.1.3. Request for Computing Path Segments

After receiving a request for creating an end to end tunnel from source A to destination Z for a given set of constraints, a parent controller allocates a global tunnel identifier (GTID) for the end to end tunnel crossing domains and a path identifier (PID) for an end to end path to be computed for the tunnel. The parent controller sends a request message to each of its related child controllers for computing a set of path segments in the domain the child controller controls in a special order. The parent controller builds a shortest path tree (SPT) using these path segments and obtains a shortest path from source A to destination Z that satisfies the constraints.

Note: The details of the path segments are hidden from the parent, which sees each of the segments as a link from one (boundary) node to another (boundary) node with a cost. The end to end path does not have any details from the parent's point of view, which may be considered as a domain path.

A request message for computing path segments (PSReq message for short) from a parent controller to a child controller comprises:

- o The address or identifier of the start-node (saying X) in the domain controlled by the child controller. From this node, a number of path segments are to be computed.
- o The global tunnel identifier (GTID) and the path identifier (PID). For the path of the tunnel, a number of path segments are to be computed.
- o An exception list containing the nodes that are on the SPT and in the domain controlled by the child controller or its adjacent domains.
- o The constraints for the path such as bandwidth constraints and color constraints.
- o A destination node Z. If Z is in the domain controlled by the child controller, the child controller computes a shortest path segment satisfying the constraints from node X to node Z within the domain.
- o Options for computing path segments:

E: E set to 1 indicating computing a shortest path segment satisfying the constraints from node X to each of the edge nodes of the domain controlled by the child controller except for the nodes in the exception list. E is set to 1 if there is not any previous hop of node X in the domain.

After receiving the request message, the child controller computes a shortest path segment satisfying the constraints from node X to each of the edge nodes of the domain controlled by the child controller except for the nodes in the exception list if E is 1. In addition, it computes a shortest path segment satisfying the constraints from node X to each of the edge nodes of the adjacent domains except for the edge nodes in the exception list just using the inter-domain links attached to node X if node X is an edge node of the domain and an end point of an inter-domain link.

The format of the PSReq message is as follows:



```

<PSReq Message> ::= <Common Header>
                    [<svec-list>]
                    <path-segment-request-list>
where:
  <svec-list> ::= <SVEC> [<svec-list>]
  <path-segment-request-list> ::=
    <path-segment-request>
    [<path-segment-request-list>]

  <path-segment-request> ::=
    <CRP>
    <Start-Node> <Tunnel-ID> <Path-ID>
    [<Destination>]
    [<OF>] [<LSPA>] [<BANDWIDTH>]
    [<metric-list>] [<RRO> [<BANDWIDTH>]] [<IRO>]
    [<LOAD-BALANCING>]
    <exception-list>

```

#### 6.2.1.4. Reply for Computing Path Segments

After receiving a request message from a parent controller for computing path segments, a child controller computes the path segments as requested in the message and sends the parent controller a reply message to reply the request message, which contains the path segments computed. The details of the path segments are hidden from the parent, which sees each of the path segments as a link with a cost.

A reply message for computing path segments (PSRep message for short) comprises:

- o The global tunnel identifier (GTID) and the path identifier (PID). For the path of the tunnel, the path segments are computed.
- o The address or identifier of the start-node (saying X) in the domain controlled by the child controller. From this node, the path segments are computed.
- o For each shortest path segment from node X to node Y computed, the address or identifier of node Y and the cost of the shortest path segment from node X to node Y.

The child controller stores the details about every shortest path segment computed under the global tunnel identifier (GTID) and the path identifier (PID) when it sends the reply message containing the path segments to the parent controller.

The child controller may delete the path segments computed for the global tunnel identifier (GTID) and the path identifier (PID) if it does not receive any request for keeping them from the parent controller for a given period of time.

The format of the PSRep message is as follows:

```

<PSRep Message> ::= <Common Header>
                        <path-segment-reply-list>
where:
  <path-segment-reply-list> ::=
    <path-segment-reply>
    [<path-segment-reply-list>]

  <path-segment-reply> ::=
    <CRP>
    <Tunnel-ID> <Path-ID>
    <Start-Node>
    [ <NO-PATH> | <segment-end-List> ]
    [<metric-list>]

```

#### 6.2.1.5. Request for Removing Path Segments

After a shortest path satisfying a set of constraints from source A to destination Z is computed, a parent controller may delete the path segments computed and stored in the related child controllers, which are not any part of the shortest path. A parent controller may send a child controller a request message for removing the path segments computed by the child controller and stored in the child controller.

1). A request message for removing path segments (RPSReq message for short) comprises:

- o The global tunnel identifier (GTID).

All the path segments stored under GTID in the child controller are to be removed.

2). A request message for removing path segments comprises:

- o The global tunnel identifier (GTID) and the path identifier (PID).

All the path segments stored under GTID and PID in the child controller are to be removed.

3). A request message for removing path segments comprises:

- o The global tunnel identifier (GTID) and the path identifier (PID)
- o A list of start point (or node) addresses or identifiers.

All the path segments stored in the child controller under GTID and PID and with a start point or node from the list of start point (or node) addresses or identifiers are to be removed.

4). A request message for removing path segments comprises:

- o The global tunnel identifier (GTID) and the path identifier (PID)
- o A list of start point (or node) addresses or identifiers
- o A list of pairs (start point, a list of end points), which identifies the path segments from start point of each pair to each of the end points in the list of the pairs.

In addition to the path segments as described in the previous message, the path segments stored in the child controller under GTID and PID and identified by the list of pairs are to be removed.

The format of the RPSReq message is as follows:

```

<RPSReq Message> ::= <Common Header>
                        <remove-path-segment-request-list>
where:
  <remove-path-segment-request-list> ::=
    <remove-path-segment-request>
    [<remove-path-segment-request-list>]

  <remove-path-segment-request> ::=
    <CRP>
    <Tunnel-ID> [<Path-ID>]
    [<start-node-list>]
    [<branch-List>]

  <start-node-list> ::= <Start-Node> [<start-node-list>]

  <branch-list> ::= <Branch> [<branch-list>]
  <Branch> ::= <Start-Node> <branch-end-list>

  <branch-end-list> ::= <Branch-End> [<branch-end-list>]

```

#### 6.2.1.6. Reply for Removing Path Segments

After removing the path segments as requested by a request message for removing path segments from a parent controller, a child controller sends the parent controller a reply message for removing path segments.

A reply message for removing path segments (RPSRep message for short) comprises:

- o The global tunnel identifier (GTID) and the path identifier (PID)
- o Status of the path segments removal:

Success: The path segments requested for removal are removed successfully.

Fail: The path segments requested for removal can not be removed.

- o Error code and reasons for failure if the status is Fail.

The format of the RPSRep message is as follows:

```
<RPSRep Message> ::= <Common Header>
                        <remove-path-segment-reply-list>
where:
  <remove-path-segment-reply-list> ::=
    <remove-path-segment-reply>
    [<remove-path-segment-reply-list>]

  <remove-path-segment-reply> ::=
    <CRP>
    <Tunnel-ID> [<Path-ID>]
    <Status>
    [<Reasons>]
```

#### 6.2.1.7. Request for Keeping Path Segments

After a shortest path satisfying a set of constraints from source A to destination Z is computed, a parent controller may send a request message for keeping path segments to each of the related child controllers to keep the path segments on the shortest path.

A request message for keeping path segments (KPSReq message for short) comprises:

- o The global tunnel identifier (GTID) and the path identifier (PID).
- o A list of pairs (start point, end point), each of which identifies the path segment from the start point of the pair to the end point of the pair.

The child controller will keep the path segments given by the list of pairs (start point, end point) stored under GTID and PID. It will remove all the other path segments stored under GTID and PID.

The format of the KPSReq message is as follows:

```

<KPSReq Message> ::= <Common Header>
                        <keep-path-segment-request-list>
where:
  <keep-path-segment-request-list> ::=
    <keep-path-segment-request>
    [<keep-path-segment-request-list>]

  <keep-path-segment-request> ::=
    <CRP>
    <Tunnel-ID> <Path-ID>
    <segment-list>

  <segment-list> ::= <Segment> [<segment-list>]
  <Segment> ::= <Segment-Start> <Segment-End>

```

#### 6.2.1.8. Reply for Keeping Path Segments

After keeping path segments as requested by a request message for keeping path segments from a parent controller, a child controller sends the parent controller a reply message for keeping path segments.

A reply message for keeping path segments (KPSRep message for short) comprises:

- o The global tunnel identifier (GTID) and the path identifier (PID).
- o Status of the path segment retention:

Success: The path segments requested for retention are retained successfully.

Fail: The path segments requested for retention can not be retained.

- o Error code and reasons for failure if the status is Fail.

The format of the KPSRep message is as follows:

```
<KPSRep Message> ::= <Common Header>
                        <keep-path-segment-reply-list>
where:
  <keep-path-segment-reply-list> ::=
    <keep-path-segment-reply>
    [<keep-path-segment-reply-list>]

  <keep-path-segment-reply> ::=
    <CRP>
    <Tunnel-ID> <Path-ID>
    <Status>
    [<Reasons>]
```

#### 6.2.1.9. Request for Creating Tunnel Segment

After obtaining the end to end shortest point to point (P2P) path, a parent controller creates a tunnel along the path crossing multiple domains through sending a request message for creating tunnel segment to each of the child controllers along the path in a reverse direction to create a tunnel segment.

A request message for creating tunnel segment (CTSReq message for short) comprises:

- o The global tunnel identifier (GTID) and the path identifier (PID).
- o A path segment from a start point to an end point for parent without domain topology or a path segment details/ERO for parent with domain topology.
- o A label and an interface if the domain controlled by the child control is not a destination domain.

For parent without domain topology, the child controller allocates and reserves link bandwidth along the path segment identified by the start point and end point, assigns labels along the path segment, and writes cross connects on each of the nodes along the path segment.

For parent with domain topology, the child controller assigns labels along the path segment ERO and writes cross connects on each of the

nodes along the path segment. The link bandwidth along the path segment is allocated and reserved by the parent controller.

For the non destination domain, the child controller writes the cross connect on the edge node to the downstream domain using the label and the interface from the downstream domain in the message.

For the non source domain, the child controller will include a label and an interface in a message to be sent to the parent controller. The interface connects the edge node of the upstream domain along the path. The label is allocated for the interface on the node that is the next hop of the edge node.

The format of the CTSReq message is as follows:

```

<CTSReq Message> ::= <Common Header>
                        <create-tunnel-segment-request-list>
where:
  <create-tunnel-segment-request-list> ::=
    <create-tunnel-segment-request>
    [<create-tunnel-segment-request-list>]

  <create-tunnel-segment-request> ::=
    <CRP>
    <Tunnel-ID> <Path-ID>
    <Path-Segment>
    [<Label> <Interface>]

  <Path-Segment> ::= [<Segment-Start> <Segment-End> | <ERO> ]

```

#### 6.2.1.10. Reply for Creating Tunnel Segment

After creating tunnel segment as requested by a request message for creating tunnel segment from a parent controller, a child controller sends the parent controller a reply message for creating tunnel segment.

A reply message for creating tunnel segment (CTSRep message for short) comprises:

- o The global tunnel identifier (GTID) and the path identifier (PID).
- o Status of the tunnel segment creation:

Success: The tunnel segment requested is created successfully.

Fail: The tunnel segments requested can not be created.

- o A label and an interface if the domain controlled by the child controller is not source domain and the status is Success.
- o Error code and reasons for failure if the status is Fail.

For the non source domain controlled by the child controller, the interface in the message connects the edge node of the upstream domain along the path, the label is allocated for the interface on the node that is the next hop of the edge node.

The format of the CTSRep message is as follows:

```

<CTSRep Message> ::= <Common Header>
                        <create-tunnel-segment-reply-list>
where:
  <create-tunnel-segment-reply-list> ::=
    <create-tunnel-segment-reply>
    [<create-tunnel-segment-reply-list>]

  <create-tunnel-segment-reply> ::=
    <CRP>
    <Tunnel-ID> <Path-ID>
    <Status> [<Label> <Interface>]
    [<Reasons>]

```

#### 6.2.1.11. Request for Removing Tunnel Segment

When a parent controller receives a request for deleting a tunnel from a user or an application, or receives a reply message for creating tunnel segment with status of Fail from a child controller, the parent controller will delete the tunnel through sending a request message for removing tunnel segment to each of the related child controllers.

A request message for removing tunnel segment (RTSReq message for short) comprises:

- o The global tunnel identifier (GTID) and the path identifier (PID).

The child controller releases the labels assigned along the path segments under GTID and PID, and removes the cross connects on each of the nodes along the path segments. If the child controller reserved the link bandwidth along the path segments under GTID and



PID, it releases the link bandwidth reserved.

The format of the RTSReq message is as follows:

```
<RTSReq Message> ::= <Common Header>
                        <remove-tunnel-segment-request-list>
where:
  <remove-tunnel-segment-request-list> ::=
    <remove-tunnel-segment-request>
    [<remove-tunnel-segment-request-list>]

  <remove-tunnel-segment-request> ::=
    <CRP>
    <Tunnel-ID> [<Path-ID>]
```

#### 6.2.1.12. Reply for Removing Tunnel Segment

After removing the tunnel segment as requested by a request message for removing tunnel segment from a parent controller, a child controller sends the parent controller a reply message for removing tunnel segment.

A reply message for removing tunnel segment (RTSRep message for short) comprises:

- o The global tunnel identifier (GTID) and the path identifier (PID).
- o Status of the tunnel segment removal:
  - Success: The tunnel segment requested is removed successfully.
  - Fail: The tunnel segment requested can not be removed.
- o Error code and reasons for failure if the status is Fail.

The format of the RTSRep message is as follows:

```

<RTSRep Message> ::= <Common Header>
                        <remove-tunnel-segment-reply-list>
where:
  <reply-tunnel-segment-reply-list> ::=
    <remove-tunnel-segment-reply>
    [<remove-tunnel-segment-reply-list>]

  <remove-tunnel-segment-reply> ::=
    <CRP>
    <Tunnel-ID> [<Path-ID>]
    <Status>
    [<Reasons>]

```

### 6.2.2. Individual Encoding of Messages

The format of PCEP Message Common Header is as follows:

```

      0               1               2               3
    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  | Message-Type |           Message-Length           |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

Message-Type (8 bits): The following message types are currently defined (refer to RFC 5440):

Message-Type	Meaning
1	Open
2	Keepalive
3	Path Computation Request
4	Path Computation Reply
5	Notification
6	Error
7	Close

The new message types are defined as follows:

Message-Type	Meaning
mTBD1	Controller Relation Discovery
mTBD2	Connections and Accesses Advertisement
mTBD3	Path Segment Computation Request
mTBD4	Path Segment Computation Reply
mTBD5	Remove Path Segment Request
mTBD6	Remove Path Segment Reply
mTBD7	Keep Path Segment Request
mTBD8	Keep Path Segment Reply
mTBD9	Create Tunnel Segment Request
mTBD10	Create Tunnel Segment Reply
mTBD11	Remove Tunnel Segment Request
mTBD12	Remove Tunnel Segment Reply

Ver, Flags and Message-Length are defined as RFC 5440.

#### 6.2.3. Group Encoding of Messages

We can encode the tunnel related messages into two groups: one group comprises the request messages related to tunnel and the other comprises the reply messages related to tunnel. Thus we can have four new message types, which are defined in PCEP Message Common Header as follows:

Message-Type	Meaning
mTBD1	Controller Relation Discovery
mTBD2	Connections and Accesses Advertisement
mTBD3	Tunnel Segment Operation Request
mTBD4	Tunnel Segment Operation Reply

Ver, Flags, other message types and Message-Length in PCEP Message Common Header are defined as RFC 5440.

The Tunnel Segment Operation can be one of the followings:

Create Tunnel Segment: Create a segment of an end to end tunnel.

Remove Tunnel Segment: Remove a segment of an end to end tunnel.

Compute Path Segments: Compute some path segments to find an end to end path for an end to end tunnel.

Remove Path Segments: Remove some path segments.

Keep Path Segment: Keep path segments on an end to end path for an end to end tunnel.

Each of these operations can be indicated by a value of options field of an object such as CRP object following PCEP Message Common Header in a message.

#### 6.2.4. Embedded Encoding of Messages

Each of the request messages can be encoded as a Path Computation Request message with a value of options/operations in an existing object. Each of the reply messages can be encoded as a Path Computation Reply message with a value of options/operations in an existing object.

A new options/operations field of 3 bits may be defined in the existing RP object. Thus each of the five request messages for supporting HSCS can be represented by a Path Computation Request message with a corresponding Options value in the RP object listed below. Each of the five reply messages for supporting HSCS can be represented by a Path Computation Reply message with a corresponding Options value in the RP object listed below.

Options Value	Meaning
oTBD1	Path Segment Computation Request/Reply
oTBD2	Remove Path Segment Request/Reply
oTBD3	Keep Path Segment Request/Reply
oTBD4	Create Tunnel Segment Request/Reply
oTBD5	Remove Tunnel Segment Request/Reply

Each request/reply message contains the contents for the message described in the previous section.

The Controller Relation Discovery message may be encoded as a Open message with a flag or a value of options/operations in an existing object. The Open message as a Controller Relation Discovery message contains the contents for the Discovery message described in the previous section.

The Connections and Accesses Advertisement message may be encoded as a Report message with a flag or a value of options/operations in an existing object such as SRP object. The Report message as a Connections and Accesses Advertisement message contains the contents of the Connections and Accesses Advertisement message described in the previous section.

#### 6.2.5. Mixed Encoding of Messages

Some of the above encodings can be combined to form a mixed encoding of the messages for supporting HSCS. For example, one mixed encoding of the messages is as follows:

- o Using Individual Encoding for Connections and Accesses Advertisement message and
- o Using Embedded Encoding for Controller Relation Discovery, all the request and reply messages for supporting HSCS.

Another mixed encoding of messages is below:

- o Using Embedded Encoding for Controller Relation Discovery;
- o Using Individual Encoding for Connections and Accesses Advertisement message and
- o Using Group Encoding for all the request and reply messages for supporting HSCS.

#### 6.3. Controller Relation Discovery

This section presents two approaches for discovering controller relation. One uses the Open Message with some simple extensions. The other uses a new message for Controller Relation Discovery, called a discovery message.

##### 6.3.1. Using Open Message

For a parent controller P and a child controller C connected by a PCE session and having a normal PCE peer adjacency, their parent-child relation is discovered through Open Messages exchanged between the parent controller and the child controller. The following is a sequence of events related to a controller relation discovery.

Controller P sends controller C an Open Message containing a capability TLV with parent flag PC set to 1 after controller C is configured as a child controller over the PCE session between P and C.

```

                P                                C
    Configure C as                                Configure P as
    Child Controller                              Parent Controller

                Open Message (PC=1)
    -----> Remote P is Parent and
                is same as configured
                Form Child-Parent relation

                Open Message (CC=1)
    <-----
    Remote C is Child and
    is same as configured
    Form Parent-Child relation
  
```

When C receives the Open Message from P and determines that PC=1 in the message is consistent with the parent controller configured locally, it forms Child-Parent relation between C and P. It sends controller P an Open Message containing a capability TLV with child controller flag CC set to 1 after controller P is configured as a parent controller over the PCE session between C and P.

When P receives the Open Message from C and determines that CC=1 in the message is consistent with the Child controller configured locally, it forms Parent-Child relation between P and C.

After the Parent-Child relation between P and C is formed, this relation is broken if the configuration "C as Child Controller" on parent controller P is deleted or "P as Parent Controller" on child controller C is removed.

When the configuration "C as Child Controller" is deleted from parent controller P, P breaks/removes the Parent-Child relation between P and C and sends C an Open Message with PC = 0. When child controller C receives the Open Message with PC = 0 from P, it determines that the remote end P is no longer its parent controller as configured locally and breaks/removes the Child-Parent relation between C and P.

When the configuration "P as Parent Controller" is deleted from child controller C, C breaks/removes the Child-Parent relation between C and P and sends P an Open Message with CC = 0. When parent controller P receives the Open Message with CC = 0 from C, it determines that the remote end C is no longer its child controller as configured locally and breaks/removes the Parent-Child relation between P and C.

### 6.3.2. Using Discovery Message

For a parent controller P and a child controller C connected by a PCE session and having a normal PCE peer adjacency, their parent-child relation is discovered through messages for controller relation discovery exchanged between the parent controller and the child controller. The following is a sequence of events related to a controller relation discovery.

Controller P sends controller C a message containing a local controller (LC=) P with a parent flag set to 1 after controller C is configured as a child controller over a PCE session between P and C.

<p>P</p> <p>Configure C as child</p> <p>message (LC=P)</p> <p>-----&gt;</p> <p>message (LC=C, RC=P)</p> <p>&lt;-----</p> <p>Remote see me and same as configured</p> <p>Form Parent-Child relation</p> <p>Add C as remote controller</p> <p>message (LC=P, RC=C)</p> <p>-----&gt;</p>	<p>C</p> <p>Configure P as parent</p> <p>LC in Msg same as configured</p> <p>Add P as remote controller</p> <p>Remote see me</p> <p>Form Child-Parent relation</p>
---	--

When C receives the message from P and determines that the local controller (LC=) P in the message is the same as the parent controller configured locally, it sends controller P a message containing local controller (LC=) C and remote controller (RC=) P.

When P receives the message from C and determines that the local controller (LC=) C in the message is the same as the child controller configured locally and the remote controller C sees me controller P (RC=P in the message), it forms a parent-child relation between P and C and sends controller C another message containing local controller (LC=) P and remote controller (RC=) C.

When C receives the message from P and determines that the local controller (LC=) P in the message is the same as the parent controller configured locally and the remote controller P sees me controller C (RC=C in the message), it forms a child-parent relation between C and P.

#### 6.4. Connections and Accesses Advertisement

A child controller sends its parent controller a message for connections and accesses, which contains the connections (i.e., inter-domain links) connecting the domain that the child controller controls to other adjacent domains, and the addresses/prefixes (i.e., the access points) in the domain to be accessible from outside of the domain.

When there is a change on the connections and the accesses of the domain, the child controller sends its parent controller a updated message for the connections and accesses, which contains the latest connections and accesses of the domain.

A parent controller stores the connections and accesses for each of its child controllers according to the messages for connections and accesses received from the child controllers. For a updated message, it updates the connections and accesses accordingly.

When a child controller is down, its parent controller may remove the connections and accesses of the domain controlled by the child controller.

After connections and accesses advertisement, a parent controller has the exterior information about all the domains controlled by its child controllers. In other words, a parent controller has the connections among the domains (i.e., the inter-domain links connecting the domains) controlled by its child controllers and the addresses/prefixes (i.e., access points) in the domains to be accessible.

A connection comprises: the attributes for a link connecting domains and the attributes for the end points of the link. The attributes for an end point of a link comprises the type of the end point node such as ABR or ASBR, and the domain of the end point such AS number and area number.

An access point comprises an address or a prefix of a domain to be accessible outside of the domain.

#### 6.5. Tunnel Creation

This section describes a couple of procedures for computing a shortest end to end path for a tunnel, and then a procedure for creating the tunnel along the path. One procedure for computing a end to end path takes two rounds of computations. The first round obtains an end to end path without any details on any of the path segments along the path. This path can be considered as a domain



path. In the second round, the details on each of the path segments along the domain path are computed. The other procedure is to get an end to end path in one round.

#### 6.5.1. Computing Path in Two Rounds

After a parent controller receives a request for creating an end to end tunnel from source A to destination Z for a given set of constraints, it computes an end to end path in two rounds as follows:

Round 1: Obtain a domain path

Roughly speaking, obtaining a domain path consists of the following three steps:

Step 1: The parent controller sends a request message to each of its related child controllers for computing a set of path segments in the domain the child controller controls in a special order.

Step 2: After a child controller receives the request message, it computes the path segments as requested and sends the parent controller a reply message with the path segments computed as links. It does not store any details about the path segments it computes. The details of the path segments are hidden from the parent controller, which sees each of the segments as a link from one (boundary) node to another (boundary) node with a cost.

Step 3: The parent controller builds a shortest path tree (SPT) using these path segments and obtains a shortest path from source A to destination Z that satisfies the constraints.

Details for obtaining a domain path are described below:

Step 1: The parent controller selects the node just added to the SPT (Initially, it selects the source).

Step 2: After selecting the node just added into the SPT, the parent controller chooses the child controller controlling the domain containing the node, and determines whether the node is destination.

For destination node, the parent controller stops computing path since the end to end (domain) path from source to destination is in the SPT, which is from the root of the SPT to the node (destination node) in the SPT.

For non-destination node X, the parent controller sends the child controller a request message for computing path segments in the domain controlled by the child controller.

- o After receiving the request message, the child controller computes the path segments as requested and sends the parent controller a reply message with the path segments computed as links. It does not store any details about the path segments it computes. The details of the path segments are hidden from the parent controller, which sees each of the segments as a link from one (boundary) node to another (boundary) node with a cost.

Step 3: After receiving the reply message from the child controller, the parent controller updates the candidate list with the links, picks up a node in the candidate list with the minimum cost and adds it into the SPT. Repeat step 1.

Round 2: Obtain the path details

After obtaining a domain path, the parent controller may initiate a BRPC procedure along the domain path to get the end to end path. Each of the child controllers controlling the domains along the domain path may store the details of the path segment it computes using a path key.

#### 6.5.2. Computing Path in One Round

For a top level parent without domain topology, the parent controller computes a shortest point to point (P2P) path for a tunnel from a source to a destination satisfying a set of constraints given to the tunnel through building a shortest path tree (SPT). The SPT is built from the source as the root of the SPT with an empty candidate list in the following steps.

Step 1: The parent controller selects the node just added to the SPT (Initially, it selects the source).

Step 2: After selecting the node just added into the SPT, the parent controller chooses the child controller controlling the domain containing the node, and determines whether the node is destination.

For destination node, the parent controller stops computing path since the end to end path from source to destination is in the SPT, which is from the root of the SPT to the node (destination node) in the SPT.

For non-destination node X, the parent controller sends the child controller a request message for computing path segments related to the domain controlled by the child controller. The request contains the exception list for the domain and flag E.

- o After receiving the request message, the child controller computes a shortest path segment from node X to each of the edge nodes of the domain not in the exception list if E is 1.
- o In addition, it computes a shortest path segment from node X to each of the edge nodes of the adjacent domains not in the exception list just using the inter-domain links attached to node X if node X is an edge node and there is an inter-domain link attached to it.
- o If node X is in the destination domain, it computes a shortest path segment from node X to the destination.
- o It sends the parent controller a reply message with the path segments computed as links and stores the details of the path segments temporarily.

Step 3: After receiving the reply message from the child controller, the parent controller updates the candidate list with the links, picks up a node in the candidate list with the minimum cost and adds it into the SPT. Repeat step 1.

For a parent without domain topology, if the parent controller is also a child controller of another upper level parent controller, after receiving a request for computing path segments from the upper level parent controller, the parent controller computes each of the path segments as requested in the same way as described above. It records and maintains the path segments computed under the GTID and PID in the request message received from the upper level parent controller.

In addition, for each path segment to be computed, it allocates a new GTID and PID for the path segment and computes the path segment through sending a request message for computing path segments to each of its related child controllers using the new GTID and PID.

When the parent as a child controller receives a request message for removing path segments from the upper level parent controller, it removes the path segments computed by each of its related child controllers through sending a request message for removing path segments to each of the related child controllers, and then it removes the path segments crossing multiple domains controlled by its

child controllers.

#### 6.5.3. Creating Tunnel along Path

After obtaining the end to end shortest point to point (P2P) path, the parent controller creates a tunnel along the path crossing multiple domains through requesting the child controllers along the path in a reverse direction.

For a parent without domain topology, the following is the procedure for creating the tunnel along the path, which is initiated by the parent controller starting from domain X = destination domain.

Step 1: The parent controller sends the child controller controlling domain X a request message for creating tunnel segment in domain X.

- o After receiving the request message from the parent controller, the child controller creates the tunnel segment in domain X it controls through reserving the resources such as link bandwidth, allocating labels along the path segment and writing a cross connect on every node in the domain along the path.
- o If the child controller is not destination controller, the request message contains an label and interface for the next hop of the edge node of domain X. The label is allocated by the controller that controls the downstream domain of domain X. The child controller uses this label and an incoming label allocated for the incoming interface on the edge node to write a cross connect on the edge node.
- o The child controller sends the parent controller a reply message with the status of the tunnel segment creation. The reply message contains an incoming label and interface for the next hop of the edge node of the upstream domain of domain X if domain X is not source domain.

Step 2: The parent controller receives the reply message from child controller C. If the status in the message is Fail, then it removes the tunnel segments created for the tunnel and return with failure for creating the tunnel.

Step 3: If child controller C is the source controller, then the end to end tunnel is created, and the parent controller and the child controllers along the tunnel maintain the information of the tunnel with the GTID and PID. The parent controller returns with success for creating the tunnel.

Step 4: Child controller C is not source controller. The reply message contains the label and interface, the parent controller repeats step 1 with domain X = the upstream domain of domain X. (In other words, it sends a request message to the child controller that controls the domain which is the upstream domain of the domain in which a tunnel segment is just created. The request contains the label and interface.)

For a parent with domain topology, the procedure for creating the tunnel along the path initiated by the parent controller is similar to the one described above, but has a few of changes to it, which are listed as follows:

- o The request message for creating tunnel segment sent to a child controller from the parent controller contains the detailed information about the path segment (such as ERO comprising every hop of the path segment) along which the tunnel segment to be created.
- o The child controller does not check or reserve resources such as link bandwidth along the path segment if the parent controller is responsible for allocating and reserving the resources along the path for the tunnel.
- o The child controller does not assign any labels along the path segment if the parent controller is responsible for assigning labels along the path for the tunnel. In this case, the request message for creating tunnel segment contains an label for every hop of the path segment. The reply message from the child controller to the parent controller does not contain any label or interface.

When the parent as a child controller receives a request message for creating tunnel segment along a path segment from the upper level parent controller, it gets the path segments for its related child controllers from the path segment in the message.

For the parent with domain topology, it obtains the detailed hop to hop information crossing multiple domains about the path segment stored by the parent controller using the GTID, PID and start point and end point of the path segment in the message received. The parent controller creates the tunnel segments in the multiple domains through sending a request message for creating tunnel to each of its related child controllers along the path in a reverse direction.

For the parent without domain topology, it obtains the detailed information about the path segment stored by the parent controller using the GTID, PID and start point and end point of the path segment

in the message received. The detailed information includes multiple path segments, each of which crosses a domain controlled by one of its related child controllers. These multiple path segments constitute the path segment in the message, which crosses multiple domains. The parent controller creates the tunnel segments in the multiple domains through sending a request message for creating tunnel to each of its related child controllers along the path in a reverse direction. For each of the path segments crossing a domain, the parent controller creates a tunnel segment along the path segment through sending a request message for creating tunnel to its child controller controlling the domain.

## 6.6. Objects and TLVs

### 6.6.1. CRP Objects

A Controller Request Parameters (CRP) object carried within each of the new messages for supporting HSCS is used to specify various parameters of a tunnel related operation request. The CRP object has Object-Class octBD1 and CRP Object-Type = 1. The format of the CRP body is as follows

Object-Class = ocTBD1 (CRP)										Object-Type = 1																					
0					1					2					3																
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										E																					
Request-ID																															
~										Optional TLVs										~											

The following flags are currently defined:

- o E (Edges of Domain): E set to 1 indicating computing a shortest path segment satisfying a given set of constraints from a start node to each of the edge nodes of the domain controlled by a child controller except for the nodes in a given exception list.

For Group Encoding of messages, a new Options field of 3 bits is defined in the flags field of the CRP object to tell the receiver of a message that the request/reply is for one of the five request/reply messages for supporting HSCS as follows:

Options	Meaning
1	Path Segment Computation Request/Reply
2	Remove Path Segment Request/Reply
3	Keep Path Segment Request/Reply
4	Create Tunnel Segment Request/Reply
5	Remove Tunnel Segment Request/Reply

### 6.6.2. LOCAL-CONTROLLER Object

A LOCAL-CONTROLLER (LC) Object is carried within a Controller Relation Discovery message. Two LC objects are defined: one for IPv4 and the other for IPv6. These two objects have the same Object-Class ocTBD2 but have different Object-Types.

#### 6.6.2.1. LOCAL-CONTROLLER Object for IPv4

The LOCAL-CONTROLLER Object for IPv4 (LC-IPv4 for short) has Object-Class ocTBD2 and Object-Type otTBD21. The format of the LC-IPv4 body is as follows:

```

      Object-Class = ocTBD2      Object-Type = otTBD21
      0              1              2              3
      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                                     |P| Level |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     Controller IPv4 Address                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+
~                                     Optional TLVs                                     ~
+-----+-----+-----+-----+-----+-----+-----+-----+

```

The LC-IPv4 object body has a 32-bit Flags field and a 32-bit Controller IPv4 Address. It may contain additional TLVs. No TLVs are currently defined.

The following flags are currently defined:

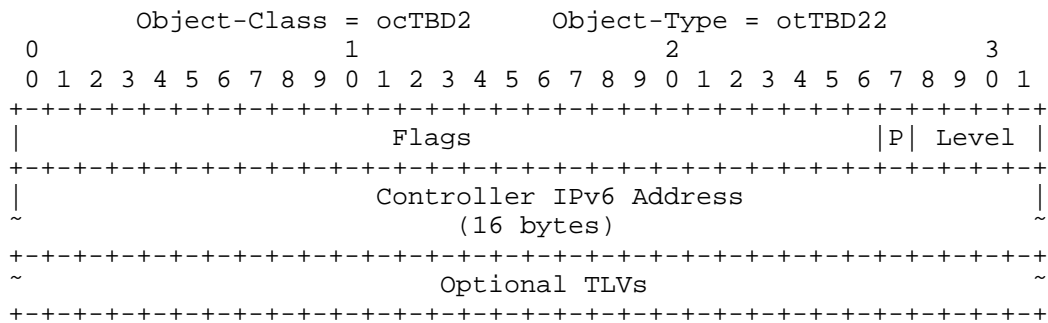
- o P (Parent Controller): P set to 1 indicating that the local controller is a Parent controller.
- o Level (Level as Parent): Level indicates the level of a controller as a parent controller. Level 0 means the highest (i.e., top) level as a parent controller. Level  $i$  ( $i > 0$ ) for a parent controller C means that C as a child controller has a parent controller of level  $(i - 1)$ .

Unassigned bits in the Flags field are considered reserved. They MUST be set to zero on transmission and MUST be ignored on receipt.

The Controller IPv4 Address indicates an IPv4 address of the local controller.

#### 6.6.2.2. LOCAL-CONTROLLER Object for IPv6

The LOCAL-CONTROLLER Object for IPv6 (LC-IPv6 for short) has Object-Class octBD2 and Object-Type otTBD22. The format of the LC-IPv6 body is as follows:



The LC-IPv6 object body has a 32-bit Flags field and a 128-bit Controller IPv6 Address. It may contain additional TLVs. No TLVs are currently defined.

The flag P (1 bit) and Level (4 bits) in the 32-bit Flags are the same as those defined in the LOCAL-CONTROLLER Object for IPv4.

The Controller IPv6 Address indicates an IPv6 address of the local controller.

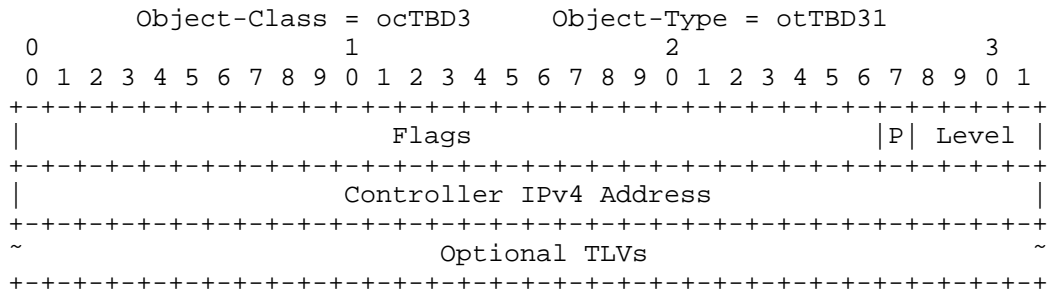
#### 6.6.3. REMOTE-CONTROLLER Object

When a local controller receives a Controller Relation Discovery message from a remote controller, the local controller MUST include a REMOTE-CONTROLLER (RC) Object with the remote controller in a Controller Relation Discovery message to be sent to the remote controller. Two RC objects are defined: one for IPv4 and the other for IPv6. These two objects have the same Object-Class octBD3 but have different Object-Types.



#### 6.6.3.1. REMOTE-CONTROLLER Object for IPv4

The REMOTE-CONTROLLER Object for IPv4 (RC-IPv4 for short) has Object-Class ocTBD3 and Object-Type otTBD31. The format of the RC-IPv4 body is as follows:



The RC-IPv4 object body has a 32-bit Flags field and a 32-bit Controller IPv4 Address. It may contain additional TLVs. No TLVs are currently defined.

The following flags are currently defined:

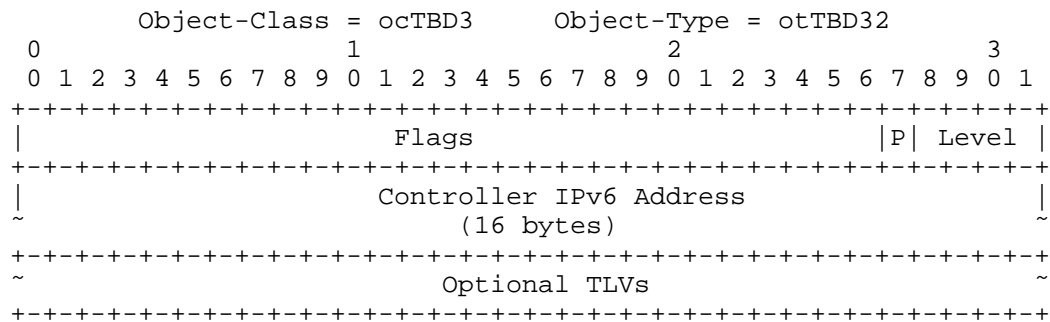
- o P (Parent Controller): P set to 1 indicating that the remote controller is a Parent controller.
- o Level (Level as Parent): Level indicates the level of a controller as a parent controller. Level 0 means the highest (i.e., top) level as a parent controller. Level  $i$  ( $i > 0$ ) for a parent controller C means that C as a child controller has a parent controller of level  $(i - 1)$ .

Unassigned bits in the Flags field are considered reserved. They MUST be set to zero on transmission and MUST be ignored on receipt.

The Controller IPv4 Address indicates an IPv4 address of the remote controller.

#### 6.6.3.2. REMOTE-CONTROLLER Object for IPv6

The REMOTE-CONTROLLER Object for IPv6 (RC-IPv6 for short) has Object-Class ocTBD3 and Object-Type otTBD32. The format of the RC-IPv6 body is as follows:



The LC-IPv6 object body has a 32-bit Flags field and a 128-bit Controller IPv6 Address. It may contain additional TLVs. No TLVs are currently defined.

The flag P (1 bit) and Level (4 bits) in the 32-bit Flags are the same as those defined in the REMOTE-CONTROLLER Object for IPv4.

The Controller IPv6 Address indicates an IPv6 address of the remote controller.

#### 6.6.4. CONNECTION and ACCESS Object

The CONNECTION and ACCESS Object (CA for short) has Object-Class ocTBD4. Three Object-Types are defined under CA object:

- o CA Inter-Domain Link: CA Object-Type is 1.
- o CA Access IPv4 Prefix: CA Object-Type is 2.
- o CA Access IPv6 Prefix: CA Object-Type is 3.

The format of each of these object bodies is as follows:

```

    Object-Class = ocTBD4 (Connection and Access)
    Object-Type = 1 (CA Inter-Domain Link)
      0          1          2          3
      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
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     AS Number                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     Area-ID TLV                                 |
~-----~-----~-----~-----~-----~-----~-----~-----~-----~
|                                     IGP Router-ID TLV                           |
~-----~-----~-----~-----~-----~-----~-----~-----~-----~
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     Inter-Domain Link TLVs                       |
~-----~-----~-----~-----~-----~-----~-----~-----~-----~
+-----+-----+-----+-----+-----+-----+-----+-----+-----+

```

Each of the Inter-Domain Link TLVs describes an inter-domain link and comprises a number of inter-domain link Sub-TLVs.

```

    Object-Class = ocTBD4 (Connection and Access)
    Object-Type = 2 (CA Access IPv4 Prefix)
      0          1          2          3
      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
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     AS Number                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     Area-ID TLV                                 |
~-----~-----~-----~-----~-----~-----~-----~-----~-----~
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     Access IPv4 Prefix TLVs                       |
~-----~-----~-----~-----~-----~-----~-----~-----~-----~
+-----+-----+-----+-----+-----+-----+-----+-----+-----+

```

```

Object-Class = ocTBD4 (Connection and Access)
Object-Type = 3 (CA Access IPv6 Prefix)
 0               1               2               3
 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
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     AS Number                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     Area-ID TLV                                     |
~                                                                                   ~
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     Access IPv6 Prefix TLVs                       |
~                                                                                   ~
+-----+-----+-----+-----+-----+-----+-----+-----+

```

The format of the Area-ID TLV is shown below:

```

 0               1               2               3
 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 (tTBD1)          |          Length (4)          |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     Area Number                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

The format of the OSPF Router-ID TLV is shown below:

```

 0               1               2               3
 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 (tTBD2)          |          Length (4)          |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     OSPF Router ID                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

The format of the ISIS Router-ID TLV is shown below:

```

 0               1               2               3
 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 (tTBD3)          |          Length (6)          |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     ISO Node-ID                                     |
~                                                                                   ~
+-----+-----+-----+-----+-----+-----+-----+-----+

```

The format of the Access IPv4 Prefix TLV is shown as follows:

```

      0                   1                   2                   3
      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 (tTBD4)                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+
| Prefix Length | IPv4 Prefix (variable) |~
+-----+-----+-----+-----+-----+-----+-----+-----+

```

The format of the Access IPv6 Prefix TLV is illustrated below:

```

      0                   1                   2                   3
      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 (tTBD5)                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+
| Prefix Length | IPv6 Prefix (variable) |~
+-----+-----+-----+-----+-----+-----+-----+-----+

```

The format of the Inter-Domain link TLV is illustrated below:

```

      0                   1                   2                   3
      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 (tTBD6)                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     Inter-Domain Link Sub-TLVs                                     |~
+-----+-----+-----+-----+-----+-----+-----+-----+

```

The format of the Inter-Domain Link Type Sub-TLV is illustrated below:

```

      0                   1                   2                   3
      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 (1)                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     Inter-Domain Link Type                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

The Inter-Domain Link Type sub-TLV defines the type of the inter-domain link:

- 1 - Point-to-point
- 2 - Multi-access

The Inter-Domain Link Type sub-TLV is TLV type 1, and is one octet in length.

The format of the Remote AS Number ID Sub-TLV is illustrated below:

```

      0               1               2               3
      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 (2)             |               Length (4)         |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     |                                     |
|               Remote AS Number     |                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

The Remote AS Number field has 4 octets. When only two octets are used for the AS number, as in current deployments, the left (high-order) two octets MUST be set to zero.

The format of the Remote Area-ID Sub-TLV is shown below:

```

      0               1               2               3
      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 (3)             |               Length (4)         |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     |                                     |
|               Area Number          |                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

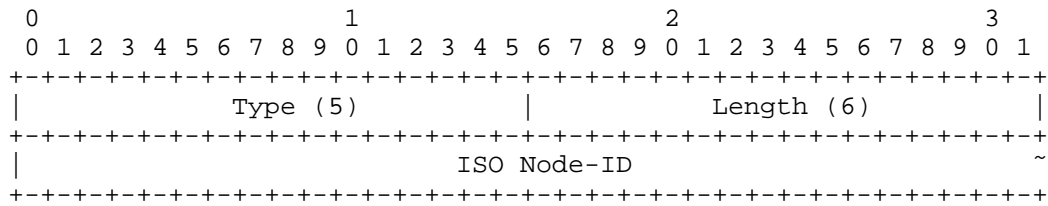
The format of the Remote OSPF Router-ID Sub-TLV is shown below:

```

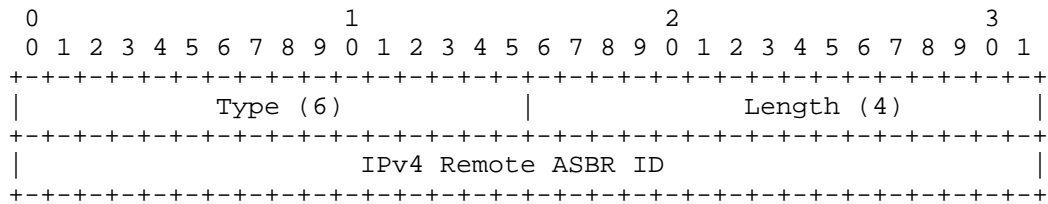
      0               1               2               3
      0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     |                                     |
|               Type (4)             |               Length (4)         |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     |                                     |
|               OSPF Router ID       |                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

The format of the Remote ISIS Router-ID Sub-TLV is shown below:

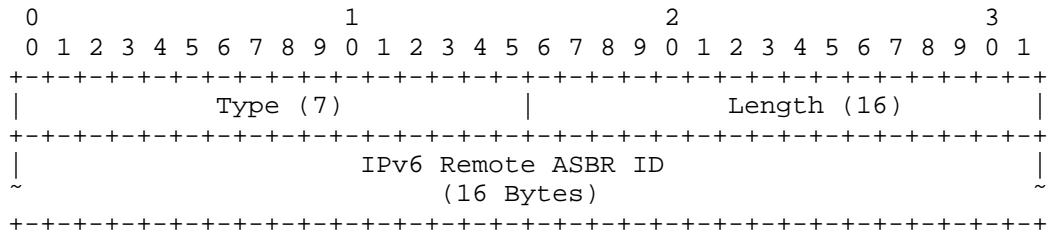


The format of the IPv4 Remote ASBR ID Sub-TLV is illustrated below:



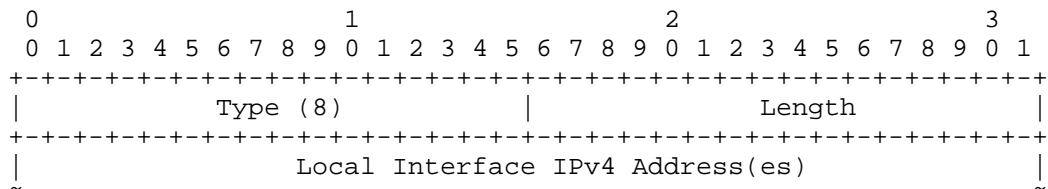
The IPv4 Remote ASBR ID sub-TLV MUST be included if the neighboring ASBR has an IPv4 address.

The format of the IPv6 Remote ASBR ID Sub-TLV is illustrated below:



The IPv6 Remote ASBR ID sub-TLV MUST be included if the neighboring ASBR has an IPv6 address.

The format of the Local Interface IPv4 Address Sub-TLV is shown below:



```

+-----+

```

The Local Interface IPv4 Address sub-TLV specifies the IPv4 address(es) of the interface corresponding to the inter-domain link. If there are multiple local addresses on the link, they are all listed in this sub-TLV.

The Local Interface IPv4 Address sub-TLV is TLV type 8, and is 4N octets in length, where N is the number of local IPv4 addresses.

The format of the Local Interface IPv6 Address Sub-TLV is illustrated below:

```

      0               1               2               3
      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 (9)                   |          Length                   |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     |                                     |
|          Local Interface IPv6 Address(es)                               |
|                                     |                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

The Local Interface IPv6 Address sub-TLV specifies the IPv6 address(es) of the interface corresponding to the inter-domain link. If there are multiple local addresses on the link, they are all listed in this sub-TLV.

The Local Interface IPv6 Address sub-TLV is TLV type 9, and is 16N octets in length, where N is the number of local IPv6 addresses.

The format of the Remote Interface IPv4 Address Sub-TLV is illustrated below:

```

      0               1               2               3
      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 (10)                  |          Length                   |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     |                                     |
|          Neighbor Interface IPv4 Address(es)                           |
|                                     |                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

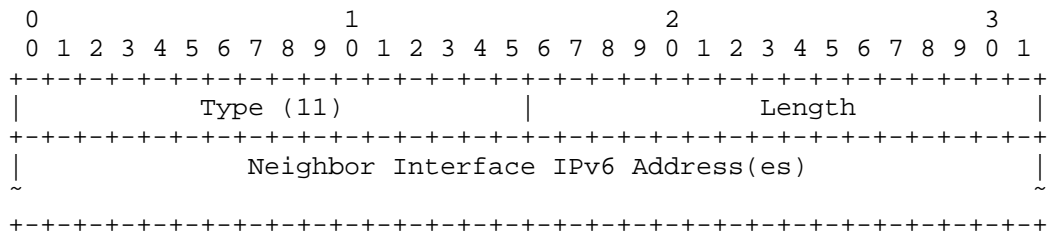
The Remote Interface IPv4 Address sub-TLV specifies the IPv4 address(es) of the neighbor's interface corresponding to the inter-domain link. This and the local address are used to discern multiple



parallel links between systems. If there are multiple remote addresses on the link, they are all listed in this sub-TLV.

The Remote Interface IPv4 Address sub-TLV is TLV type 10, and is  $4N$  octets in length, where  $N$  is the number of neighbor IPv4 addresses.

The format of the Remote Interface IPv6 Address Sub-TLV is illustrated below:



The Remote Interface IPv6 Address sub-TLV specifies the IPv6 address(es) of the neighbor's interface corresponding to the inter-domain link. If there are multiple neighbor addresses on the link, they are all listed in this sub-TLV.

The Remote Interface IPv6 Address sub-TLV is TLV type 11, and is  $16N$  octets in length, where  $N$  is the number of neighbor IPv6 addresses.

#### 6.6.5. NODE Object

The NODE Object has Object-Class octBD5. A nuber of Object-Types are defined under NODE object below:

1. IPv4 START-NODE: NODE Object-Type is 1.
2. IPv6 START-NODE: NODE Object-Type is 2.
3. IPv4 DESTINATION-NODE-LIST: NODE Object-Type is 3.
4. IPv6 DESTINATION-NODE-LIST: NODE Object-Type is 4.
5. IPv4 SEGMENT-END-NODE-LIST: NODE Object-Type is 5.
6. IPv6 SEGMENT-END-NODE-LIST: NODE Object-Type is 6.
7. IPv4 EXCEPTION-NODE-LIST: NODE Object-Type is 7.

8. IPv6 EXCEPTION-NODE-LIST: NODE Object-Type is 8.
9. NODE-IGP-METRIC-LIST: NODE Object-Type is 9.
10. NODE-TE-METRIC-LIST: NODE Object-Type is 10.
11. NODE-HOP-COUNT-LIST: NODE Object-Type is 11.

The format of NODE object body for IPv4 START-NODE is as follows:

```

Object-Class = octBD5 (NODE)
Object-Type = 1 (IPv4 START-NODE)
0               1               2               3
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
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                               Start Node IPv4 Address                               |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
```

The Start Node IPv4 Address is the IPv4 address of a start node.

The format of NODE object body for IPv6 START-NODE is as follows:

```

Object-Class = octBD5 (NODE)
Object-Type = 2 (IPv6 START-NODE)
0               1               2               3
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
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                               Start Node IPv6 Address                               |
|                               (16 bytes)                                           |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
```

The Start Node IPv6 Address is the IPv6 address of a start node.

The format of NODE object body for IPv4 DESTINATION-NODE-LIST is as follows:

```

    Object-Class = ocTBD5 (NODE)
    Object-Type = 3 (IPv4 DESTINATION-NODE-LIST)
      0               1               2               3
      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
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     Destination Node 1 IPv4 Address                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     . . . . .                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     Destination Node n IPv4 Address                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

The IPv4 DESTINATION-NODE-LIST contains n destination node IPv4 addresses. An IPv4 DESTINATION-NODE-LIST is also called an IPv4 DESTINATION-NODES.

The format of NODE object body for IPv6 DESTINATION-NODE-LIST is as follows:

```

    Object-Class = ocTBD5 (NODE)
    Object-Type = 4 (IPv6 DESTINATION-NODE-LIST)
      0               1               2               3
      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
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     Destination Node 1 IPv6 Address                                     |
|                                     (16 bytes)                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     . . . . .                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     Destination Node n IPv6 Address                                     |
|                                     (16 bytes)                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

The IPv6 DESTINATION-NODE-LIST contains n destination node IPv6 addresses. An IPv6 DESTINATION-NODE-LIST is also called an IPv6 DESTINATION-NODES.

The format of NODE object body for IPv4 SEGMENT-END-NODE-LIST is as follows:

```

Object-Class = ocTBD5 (NODE)
Object-Type = 5 (IPv4 SEGMENT-END-NODE-LIST)
0           1           2           3
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
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     Segment End Node 1 IPv4 Address                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     . . . . .                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     Segment End Node n IPv4 Address                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

The IPv4 SEGMENT-END-NODE-LIST contains n segment node IPv4 addresses. An IPv4 SEGMENT-END-NODE-LIST is also called an IPv4 SEGMENT-END-NODES.

The format of NODE object body for IPv6 SEGMENT-END-NODE-LIST is as follows:

```

Object-Class = ocTBD5 (NODE)
Object-Type = 6 (IPv6 SEGMENT-END-NODE-LIST)
0           1           2           3
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
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     Segment End Node 1 IPv6 Address                                     |
|                                     (16 bytes)                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     . . . . .                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     Segment End Node n IPv6 Address                                     |
|                                     (16 bytes)                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

The IPv6 SEGMENT-END-NODE-LIST contains n segment end node IPv6 addresses. An IPv6 SEGMENT-END-NODE-LIST is also called an IPv6 SEGMENT-END-NODES.

The format of NODE object body for IPv4 EXCEPTION-NODE-LIST is as follows:

```

    Object-Class = octBD5 (NODE)
    Object-Type = 7 (IPv4 EXCEPTION-NODE-LIST)
      0          1          2          3
      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
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     Exception Node 1 IPv4 Address                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     . . . . .                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     Exception Node n IPv4 Address                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

The IPv4 SEGMENT-END-NODE-LIST contains n node IPv4 addresses in an exception list. An IPv4 EXCEPTION-NODE-LIST is also called an IPv4 EXCEPTION-LIST.

The format of NODE object body for IPv6 EXCEPTION-NODE-LIST is as follows:

```

    Object-Class = octBD5 (NODE)
    Object-Type = 8 (IPv6 EXCEPTION-NODE-LIST)
      0          1          2          3
      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
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     Exception Node 1 IPv6 Address                                     |
|                                     (16 bytes)                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     . . . . .                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     Exception Node n IPv6 Address                                     |
|                                     (16 bytes)                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

The IPv6 EXCEPTION-NODE-LIST contains n node IPv6 addresses in an exception list. An IPv6 EXCEPTION-NODE-LIST is also called an IPv6 EXCEPTION-LIST.

The format of NODE object body for NODE-IGP-METRIC-LIST is as follows:

```

Object-Class = ocTBD5 (NODE)
Object-Type = 9 (NODE-IGP-METRIC-LIST)
0              1              2              3
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
+-----+-----+-----+-----+-----+-----+-----+-----+
|               Segment End Node 1 IGP Metric Value               |
+-----+-----+-----+-----+-----+-----+-----+-----+
|               . . . . .                                         |
+-----+-----+-----+-----+-----+-----+-----+-----+
|               Segment End Node n IGP Metric Value               |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

The NODE-IGP-METRIC-LIST contains n IGP metrics for n segment end nodes.

The format of NODE object body for NODE-TE-METRIC-LIST is as follows:

```

Object-Class = ocTBD5 (NODE)
Object-Type = 10 (NODE-TE-METRIC-LIST)
0              1              2              3
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
+-----+-----+-----+-----+-----+-----+-----+-----+
|               Segment End Node 1 TE Metric Value               |
+-----+-----+-----+-----+-----+-----+-----+-----+
|               . . . . .                                         |
+-----+-----+-----+-----+-----+-----+-----+-----+
|               Segment End Node n TE Metric Value               |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

The NODE-TE-METRIC-LIST contains n TE metrics for n segment end nodes.

The format of NODE object body for NODE-HOP-COUNT-LIST is as follows:

```

Object-Class = octBD5 (NODE)
Object-Type = 11 (NODE-HOP-COUNT-LIST)
0               1               2               3
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
+-----+-----+-----+-----+-----+-----+-----+-----+
|               Segment End Node 1 Hop Counts Value               |
+-----+-----+-----+-----+-----+-----+-----+-----+
|               . . . . .               |
+-----+-----+-----+-----+-----+-----+-----+-----+
|               Segment End Node n Hop Counts Value               |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

The NODE-HOP-COUNT-LIST contains n hop counts values for n segment end nodes.

#### 6.6.6. TUNNEL Object

The TUNNEL Object has Object-Class octBD6. Two Object-Types are defined under TUNNEL object:

1. TUNNEL-ID: TUNNEL Object-Type is 1.
2. TUNNEL-PATH-ID: TUNNEL Object-Type is 2.

The format of TUNNEL object body for TUNNEL-ID is as follows:

```

Object-Class = octBD6 (TUNNEL)
Object-Type = 1 (TUNNEL-ID)
0               1               2               3
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
+-----+-----+-----+-----+-----+-----+-----+-----+
|               Tunnel ID               |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

The Tunnel ID in the body is a 32-bit unique number for identifying a tunnel globally.

The format of TUNNEL object body for TUNNEL-PATH-ID is as follows:

```

Object-Class = octBD6 (TUNNEL)   Object-Type = 2 (PATH-ID)
0               1               2               3
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
+-----+-----+-----+-----+-----+-----+-----+-----+
|               Path ID               |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

The Path ID in the body is a 16-bit number for uniquely identifying a path under a tunnel.

#### 6.6.7. STATUS Object

The STATUS Object has Object-Class octBD7. The format of STATUS object body has following format:

```

    Object-Class = octBD7 (STATUS)
    Object-Type = 1
      0               1               2               3
    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
+-----+-----+-----+-----+-----+-----+-----+-----+
| Status Code | Reason | Reserved |
+-----+-----+-----+-----+-----+-----+-----+-----+
~                               Optional TLVs                               ~
+-----+-----+-----+-----+-----+-----+-----+-----+

```

The status code (or status for short) in a STATUS may be one of the followings:

- 1 (SUCCESS): Indicating a request is successfully finished.
- 2 (FAIL): Indicating a request can not be finished.

When the status is FAIL, the Reason gives a reason for the failure and the Optional TLVs give some more details about failure.

#### 6.6.8. LABEL Object

The LABEL Object has Object-Class octBD8. The format of LABEL object body has following format:

```

    Object-Class = octBD8 (LABEL)
    Object-Type = 1
      0               1               2               3
    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
+-----+-----+-----+-----+-----+-----+-----+-----+
|                               (top label)                               |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

The contents of a LABEL is a single label, encoded in 4 octets.



### 6.6.9. INTERFACE Object

The INTERFACE Object has Object-Class octBD9. Three Object-Types are defined under INTERFACE object:

1. Index: Object-Type is 1.
2. IPv4 Address: Object-Type is 2.
3. IPv6 Address: Object-Type is 3.

The format of INTERFACE object body for interface index has following format:

```

Object-Class = octBD9 (INTERFACE)
Object-Type = 1 (Index)
0               1               2               3
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
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     Interface Index                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

The Interface Index is a single interface index, encoded in 4 octets.

The format of INTERFACE object body for interface IPv4 address has following format:

```

Object-Class = octBD9 (INTERFACE)
Object-Type = 2 (IPv4 Address)
0               1               2               3
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
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     Interface IPv4 Address                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

The Interface IPv4 Address is a single interface IPv4 address, encoded in 4 octets.

The format of INTERFACE object body for interface IPv6 address has following format:

```

    Object-Class = ocTBD9 (INTERFACE)
    Object-Type = 3 (IPv6 Address)
      0               1               2               3
      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
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     Interface IPv6 Address                                     |
|~                                     (16 bytes)                                     ~|
+-----+-----+-----+-----+-----+-----+-----+-----+

```

The Interface IPv6 Address is a single interface IPv6 address, encoded in 16 octets.

## 7. Security Considerations

The mechanism described in this document does not raise any new security issues for the PCEP protocols.

## 8. IANA Considerations

This section specifies requests for IANA allocation.

## 9. Acknowledgement

The authors would like to thank people for their valuable comments on this draft.

## 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, <<http://www.rfc-editor.org/info/rfc2119>>.
- [RFC4655] Farrel, A., Vasseur, J., and J. Ash, "A Path Computation Element (PCE)-Based Architecture", RFC 4655, DOI 10.17487/RFC4655, August 2006, <<http://www.rfc-editor.org/info/rfc4655>>.
- [RFC5440] Vasseur, JP., Ed. and JL. Le Roux, Ed., "Path Computation Element (PCE) Communication Protocol (PCEP)", RFC 5440, DOI 10.17487/RFC5440, March 2009, <<http://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, <<http://www.rfc-editor.org/info/rfc5441>>.

[RFC5392] Chen, M., Zhang, R., and X. Duan, "OSPF Extensions in Support of Inter-Autonomous System (AS) MPLS and GMPLS Traffic Engineering", RFC 5392, DOI 10.17487/RFC5392, January 2009, <<http://www.rfc-editor.org/info/rfc5392>>.

[RFC5316] Chen, M., Zhang, R., and X. Duan, "ISIS Extensions in Support of Inter-Autonomous System (AS) MPLS and GMPLS Traffic Engineering", RFC 5316, DOI 10.17487/RFC5316, December 2008, <<http://www.rfc-editor.org/info/rfc5316>>.

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

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

## 10.2. Informative References

[RFC1136] Hares, S. and D. Katz, "Administrative Domains and Routing Domains: A model for routing in the Internet", RFC 1136, DOI 10.17487/RFC1136, December 1989, <<http://www.rfc-editor.org/info/rfc1136>>.

[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, <<http://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, <<http://www.rfc-editor.org/info/rfc4216>>.

[RFC6006] Zhao, Q., Ed., King, D., Ed., Verhaeghe, F., Takeda, T., Ali, Z., and J. Meuric, "Extensions to the Path Computation Element Communication Protocol (PCEP) for Point-to-Multipoint Traffic Engineering Label Switched Paths", RFC 6006, DOI 10.17487/RFC6006, September 2010, <<http://www.rfc-editor.org/info/rfc6006>>.

[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, <<http://www.rfc-editor.org/info/rfc6805>>.

#### Appendix A. Details on Embedded Encoding of Messages

A new options field of 3 bits is defined in the flags field of the RP object to tell the receiver of the message that the request/reply is for one of the five request/reply messages for supporting HSCS as follows:

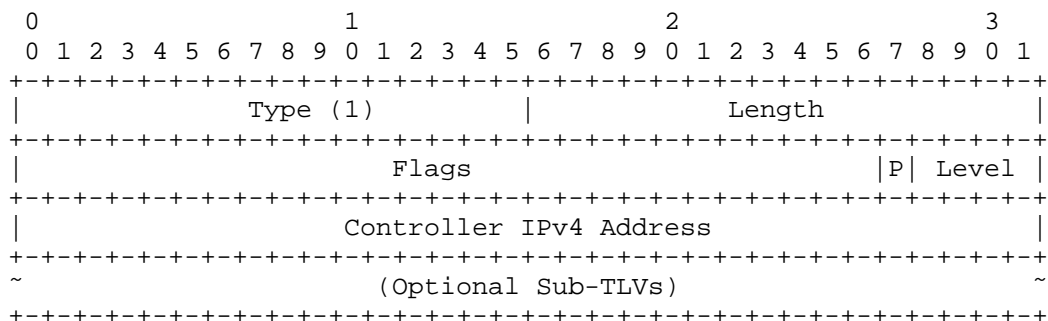
Options	Meaning
1	Path Segment Computation Request/Reply
2	Remove Path Segment Request/Reply
3	Keep Path Segment Request/Reply
4	Create Tunnel Segment Request/Reply
5	Remove Tunnel Segment Request/Reply

A new flag E of 1 bit is defined in the flags field of the RP object. Flag E set to 1 indicating computing a shortest path segment satisfying a given set of constraints from a start node to each of the edge nodes of the domain controlled by a child controller except for the nodes in a given exception list.

##### A.1. Message for Controller Relation Discovery

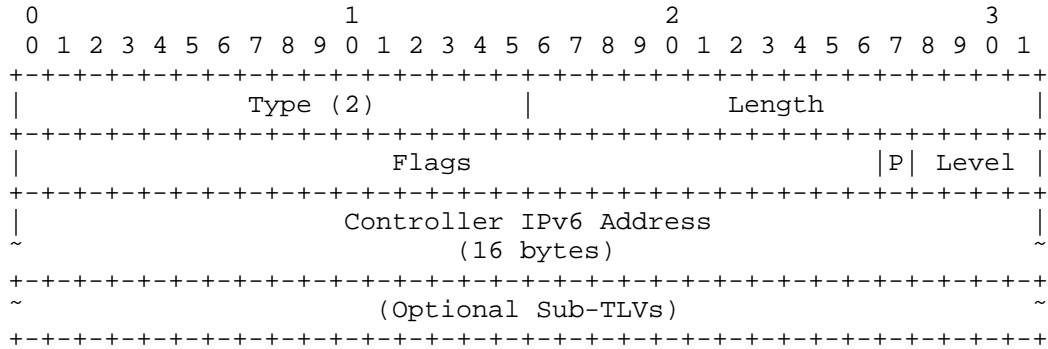
The new TLV defined in the Open Object in section Capability Discovery is extended to contain Sub-TLVs for local controller and remote controller. Thus Open Message with the Open Object containing the new TLV can be used as Message for Controller Relation Discovery. Four optional Sub-TLVs are defined as follows:

###### 1. Local Controller IPv4 Sub-TLV



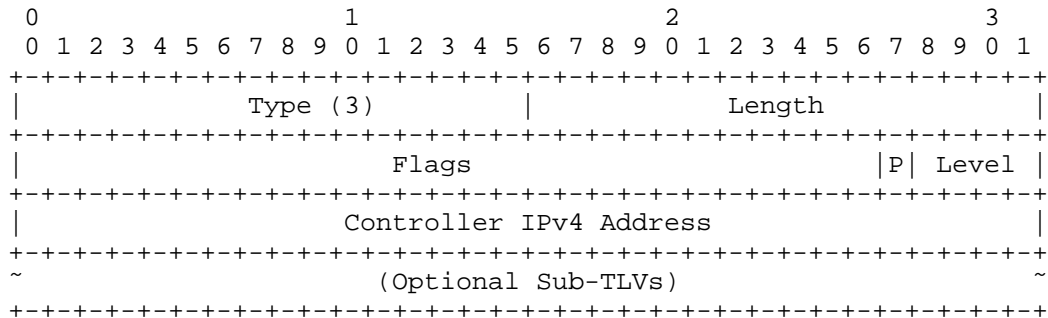
The meanings of each field in the Sub-TLV is the same as described in section LOCAL-CONTROLLER Object for IPv4.

## 2. Local Controller IPv6 Sub-TLV



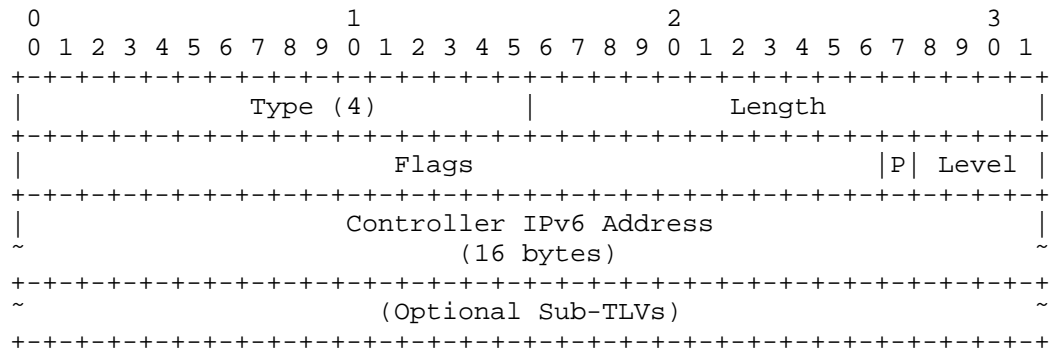
The meanings of each field in the Sub-TLV is the same as described in section LOCAL-CONTROLLER Object for IPv6.

## 3. Remote Controller IPv4 Sub-TLV



The meanings of each field in the Sub-TLV is the same as described in section REMOTE-CONTROLLER Object for IPv4.

## 4. Remote Controller IPv6 Sub-TLV



The meanings of each field in the Sub-TLV is the same as described in section REMOTE-CONTROLLER Object for IPv6.

#### A.2. Message for Connections and Accesses Advertisement

The format of the CAAdv message is as follows:

```

<CAAdv Message> ::= <Common Header>
                    <SRP>
                    <Inter-Domain-Link-List>
                    [<Access-Address-List>]
where:
<Inter-Domain-Link-List> ::= <Inter-Domain-Link>
                             [<Inter-Domain-Link-List>]
<Access-Address-List> ::= <Access-Address>
                          [<Access-Address-List>]

```

#### A.3. Request for Computing Path Segments

The format of the PSReq message is as follows:

```

<PSReq Message> ::= <Common Header>
                    [<svec-list>]
                    <path-segment-request-list>
where:
  <svec-list> ::= <SVEC> [<svec-list>]
  <path-segment-request-list> ::=
    <path-segment-request>
    [<path-segment-request-list>]

  <path-segment-request> ::=
    <RP> <END-POINTS> [<OF>] [<LSPA>] [<BANDWIDTH>]
    <Tunnel-ID> <Path-ID>
    [<metric-list>] [<RRO> [<BANDWIDTH>]] [<IRO>]
    [<LOAD-BALANCING>]
    <exception-list>

```

#### A.4. Reply for Computing Path Segments

The format of the PSRep message is as follows:

```

<PSRep Message> ::= <Common Header>
                    <path-segment-reply-list>
where:
  <path-segment-reply-list> ::=
    <path-segment-reply>
    [<path-segment-reply-list>]

  <path-segment-reply> ::=
    <RP> [<NO-PATH>] [<attribute-list>]
    <Tunnel-ID> <Path-ID>
    <Start-Node>
    [ <NO-PATH> | <segment-end-List> ]
    [<attribute-list>]

```

#### A.5. Request for Removing Path Segments

The format of the RPSReq message is as follows:

```

<RPSReq Message> ::= <Common Header>
                        <remove-path-segment-request-list>
where:
  <remove-path-segment-request-list> ::= =
                        <remove-path-segment-request>
                        [<remove-path-segment-request-list>]

  <remove-path-segment-request> ::=
                        <RP>
                        <Tunnel-ID> [<Path-ID>]
                        [<start-node-list>]
                        [<branch-List>]

  <start-node-list> ::= <Start-Node> [<start-node-list>]

  <branch-list> ::= <Branch> [<branch-list>]
  <Branch> ::= <Start-Node> <branch-end-list>

  <branch-end-list> ::= <Branch-End> [<branch-end-list>]

```

#### A.6. Reply for Removing Path Segments

The format of the RPSRep message is as follows:

```

<RPSRep Message> ::= <Common Header>
                        <remove-path-segment-reply-list>
where:
  <remove-path-segment-reply-list> ::=
                        <remove-path-segment-reply>
                        [<remove-path-segment-reply-list>]

  <remove-path-segment-reply> ::=
                        <RP>
                        <Tunnel-ID> [<Path-ID>]
                        <Status>
                        [<Reasons>]

```

#### A.7. Request for Keeping Path Segments

The format of the KPSReq message is as follows:



```

<KPSReq Message> ::= <Common Header>
                        <keep-path-segment-request-list>
where:
  <keep-path-segment-request-list> ::= =
                        <keep-path-segment-request>
                        [<keep-path-segment-request-list>]

  <keep-path-segment-request> ::=
                        <RP>
                        <Tunnel-ID> <Path-ID>
                        <segment-list>

  <segment-list> ::= <Segment> [<segment-list>]
  <Segment> ::= <Segment-Start> <Segment-End>

```

#### A.8. Reply for Keeping Path Segments

The format of the KPSRep message is as follows:

```

<KPSRep Message> ::= <Common Header>
                        <keep-path-segment-reply-list>
where:
  <keep-path-segment-reply-list> ::=
                        <keep-path-segment-reply>
                        [<keep-path-segment-reply-list>]

  <keep-path-segment-reply> ::=
                        <RP>
                        <Tunnel-ID> <Path-ID>
                        <Status>
                        [<Reasons>]

```

#### A.9. Request for Creating Tunnel Segment

The format of the CTSReq message is as follows:

```

<CTSReq Message> ::= <Common Header>
                        <create-tunnel-segment-request-list>
where:
  <create-tunnel-segment-request-list> ::=
    <create-tunnel-segment-request>
    [<create-tunnel-segment-request-list>]

  <create-tunnel-segment-request> ::=
    <RP>
    <Tunnel-ID> <Path-ID>
    <Path-Segment>
    [<Label> <Interface>]

  <Path-Segment> ::= [<Segment-Start> <Segment-End> | <ERO> ]

```

#### A.10. Reply for Creating Tunnel Segment

The format of the CTSRep message is as follows:

```

<CTSRep Message> ::= <Common Header>
                        <create-tunnel-segment-reply-list>
where:
  <create-tunnel-segment-reply-list> ::=
    <create-tunnel-segment-reply>
    [<create-tunnel-segment-reply-list>]

  <create-tunnel-segment-reply> ::=
    <RP>
    <Tunnel-ID> <Path-ID>
    <Status> [<Label> <Interface>]
    [<Reasons>]

```

#### A.11. Request for Removing Tunnel Segment

The format of the RTSReq message is as follows:

```

<RTSReq Message> ::= <Common Header>
                        <remove-tunnel-segment-request-list>
where:
  <remove-tunnel-segment-request-list> ::=
    <remove-tunnel-segment-request>
    [<remove-tunnel-segment-request-list>]

  <remove-tunnel-segment-request> ::
    <RP>
    <Tunnel-ID> [<Path-ID>]

```

## A.12. Reply for Removing Tunnel Segment

The format of the RTSRep message is as follows:

```
<RTSRep Message> ::= <Common Header>
                        <remove-tunnel-segment-reply-list>
where:
  <reply-tunnel-segment-reply-list> ::=
    <remove-tunnel-segment-reply>
    [<remove-tunnel-segment-reply-list>]

  <remove-tunnel-segment-reply> ::=
    <RP>
    <Tunnel-ID> [<Path-ID>]
    <Status>
    [<Reasons>]
```

## Authors' Addresses

Huaimo Chen  
Huawei Technologies  
Boston, MA,  
USA

EMail: [Huaimo.chen@huawei.com](mailto:Huaimo.chen@huawei.com)

Mehmet Toy  
Comcast  
1800 Bishops Gate Blvd.  
Mount Laurel, NJ 08054  
USA

EMail: [mehmet\\_toy@cable.comcast.com](mailto:mehmet_toy@cable.comcast.com)

Lei Liu  
Fujitsu  
USA

EMail: [lliu@us.fujitsu.com](mailto:lliu@us.fujitsu.com)

Vic Liu  
China Mobile  
No.32 Xuanwumen West Street, Xicheng District  
Beijing, 100053  
China

EMail: liuzhiheng@chinamobile.com



PCE Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: May 3, 2018

H. Chen  
Huawei Technologies  
M. Toy  
Verizon  
L. Liu  
Fujitsu  
V. Liu  
China Mobile  
October 30, 2017

PCE Hierarchical SDNs  
draft-chen-pce-h-sdns-03

Abstract

This document presents extensions to the Path Computation Element Communication Protocol (PCEP) for supporting a hierarchical SDN control system, which comprises multiple SDN controllers controlling a network with a number of domains.

Status of This Memo

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

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

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

This Internet-Draft will expire on May 3, 2018.

Copyright Notice

Copyright (c) 2017 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 . . . . .	4
2. Terminology . . . . .	4
3. Conventions Used in This Document . . . . .	6
4. Requirements . . . . .	6
5. Overview of Hierarchical SDN Control System . . . . .	6
6. Extensions to PCEP . . . . .	9
6.1. Capability Discovery . . . . .	9
6.2. New Messages for Hierarchical SDN Control System . . . . .	10
6.2.1. Contents of Messages . . . . .	12
6.2.2. Individual Encoding of Messages . . . . .	24
6.2.3. Group Encoding of Messages . . . . .	25
6.2.4. Embedded Encoding of Messages . . . . .	26
6.2.5. Mixed Encoding of Messages . . . . .	27
6.3. Controller Relation Discovery . . . . .	27
6.3.1. Using Open Message . . . . .	27
6.3.2. Using Discovery Message . . . . .	29
6.4. Connections and Accesses Advertisement . . . . .	30
6.5. Tunnel Creation . . . . .	30
6.5.1. Computing Path in Two Rounds . . . . .	31
6.5.2. Computing Path in One Round . . . . .	32
6.5.3. Creating Tunnel along Path . . . . .	34
6.6. Objects and TLVs . . . . .	36
6.6.1. CRP Objects . . . . .	36
6.6.2. LOCAL-CONTROLLER Object . . . . .	37
6.6.3. REMOTE-CONTROLLER Object . . . . .	38
6.6.4. CONNECTION and ACCESS Object . . . . .	40
6.6.5. NODE Object . . . . .	47
6.6.6. TUNNEL Object . . . . .	53
6.6.7. STATUS Object . . . . .	54
6.6.8. LABEL Object . . . . .	54
6.6.9. INTERFACE Object . . . . .	55
7. Security Considerations . . . . .	56
8. IANA Considerations . . . . .	56
9. Acknowledgement . . . . .	56
10. References . . . . .	56
10.1. Normative References . . . . .	56
10.2. Informative References . . . . .	57
Appendix A. Details on Embedded Encoding of Messages . . . . .	58
A.1. Message for Controller Relation Discovery . . . . .	58
A.2. Message for Connections and Accesses Advertisement . . . . .	60
A.3. Request for Computing Path Segments . . . . .	60

A.4.	Reply for Computing Path Segments . . . . .	61
A.5.	Request for Removing Path Segments . . . . .	61
A.6.	Reply for Removing Path Segments . . . . .	62
A.7.	Request for Keeping Path Segments . . . . .	62
A.8.	Reply for Keeping Path Segments . . . . .	63
A.9.	Request for Creating Tunnel Segment . . . . .	63
A.10.	Reply for Creating Tunnel Segment . . . . .	64
A.11.	Request for Removing Tunnel Segment . . . . .	64
A.12.	Reply for Removing Tunnel Segment . . . . .	65



## 1. Introduction

A domain is a collection of network elements within a common sphere of address management or routing procedure which are operated by a single organization or administrative authority. Examples of such domains include IGP (OSPF or IS-IS) areas and Autonomous Systems.

For scalability, security, interoperability and manageability, a big network is organized as a number of domains. For example, a big network running OSPF as routing protocol is organized as a number of OSPF areas. A network running BGP is organized as multiple Autonomous Systems, each of which has a number of IGP areas.

The concepts of Software Defined Networks (SDN) have been shown to reduce the overall network CapEx and OpEx, whilst facilitating the deployment of services and enabling new features. The core principles of SDN include: centralized control to allow optimized usage of network resources and provisioning of network elements across domains.

For a network with a number of domains, it is natural to have multiple SDN controllers, each of which controls a domain in the network. To achieve a centralized control on the network, a hierarchical architecture of controllers is a good fit. At top level of the hierarchy, it is a parent controller that is not a child controller. The parent controller controls a number of child controllers. Some of these child controllers are not parent controllers. Each of them controls a domain. Some other child controllers are also parent controllers, each of which controls multiple child controllers, and so on.

This document presents extensions to the Path Computation Element Communication Protocol (PCEP) for supporting a hierarchical SDN control system, which comprises multiple SDN controllers controlling a network with a number of domains.

## 2. Terminology

The following terminology is used in this document.

ABR: Area Border Router. Router used to connect two IGP areas (Areas in OSPF or levels in IS-IS).

ASBR: Autonomous System Border Router. Router used to connect together ASes of the same or different service providers via one or more inter-AS links.

BN: Boundary Node. A boundary node is either an ABR in the context of inter-area Traffic Engineering or an ASBR in the context of inter-AS Traffic Engineering. A Boundary Node is also called an Edge Node.

Entry BN of domain(n): a BN connecting domain(n-1) to domain(n) along the path found from the source node to the BN, where domain(n-1) is the previous hop (or upstream) domain of domain(n). An Entry BN is also called an in-BN or in-edge node.

Exit BN of domain(n): a BN connecting domain(n) to domain(n+1) along the path found from the source node to the BN, where domain(n+1) is the next hop (or downstream) domain of domain(n). An Exit BN is also called a out-BN or out-edge node.

Source Domain: For a tunnel from a source to a destination, the domain containing the source is the source domain for the tunnel.

Destination Domain: For a tunnel from a source to a destination, the domain containing the destination is the destination domain for the tunnel.

Source Controller: A controller controlling the source domain.

Destination Controller: A controller controlling the destination domain.

Parent Controller: A parent controller is a controller that communicates with a number of child controllers and controls a network with multiple domains through the child controllers. A PCE can be enhanced to be a parent controller.

Child Controller: A child controller is a controller that communicates with one parent controller and controls a domain in a network. A PCE can be enhanced to be a child controller.

Exception list: An exception list for a domain contains the nodes in the domain and its adjacent domains that are on the shortest path tree (SPT) that the parent controller is building.

GTID: Global Tunnel Identifier. It is used to identify a tunnel in a network.

PID: Path Identifier. It is used to identify a path for a tunnel in a network.

Inter-area TE LSP: a TE LSP that crosses an IGP area boundary.

Inter-AS TE LSP: a TE LSP that crosses an AS boundary.

LSP: Label Switched Path

LSR: Label Switching Router

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.

PCE(i): a PCE with the scope of domain(i).

TED: Traffic Engineering Database.

This document uses terminology defined in [RFC5440].

### 3. Conventions Used in This Document

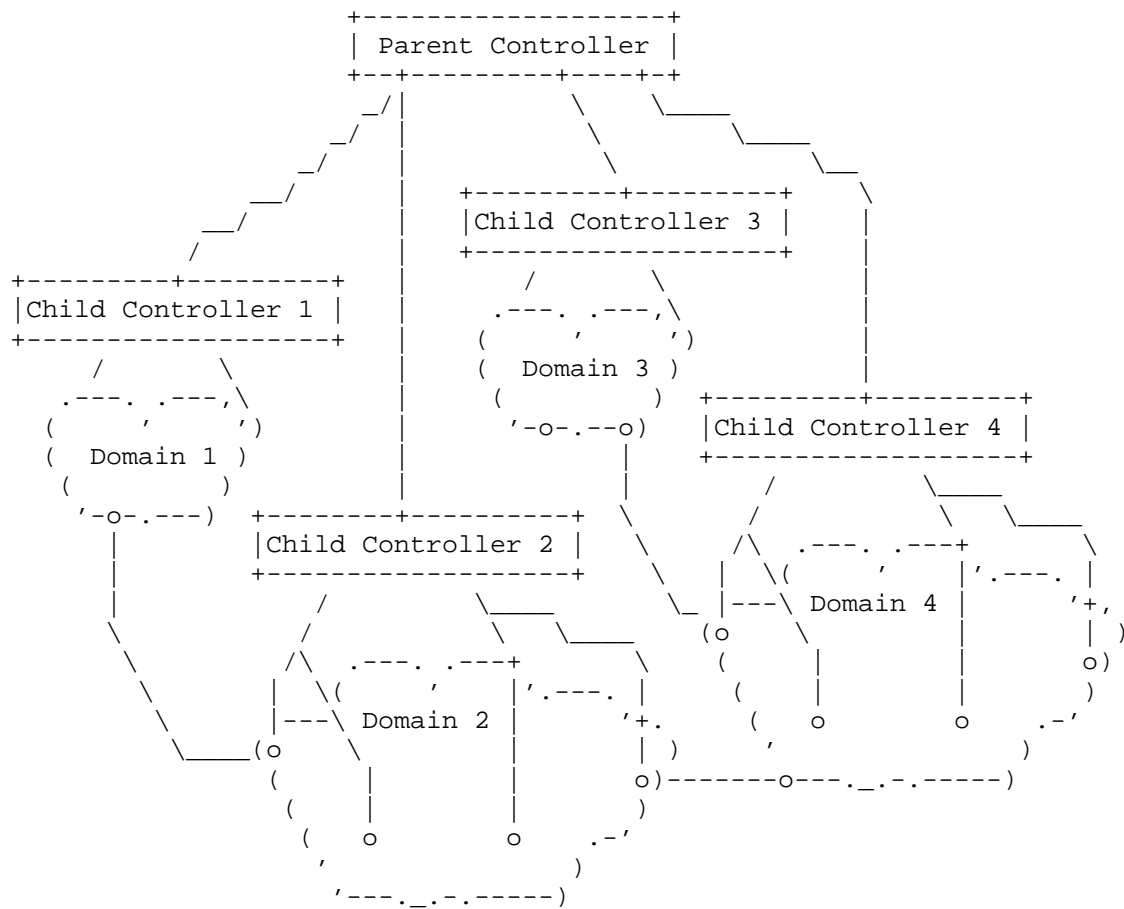
The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

### 4. Requirements

This section summarizes the requirements for Hierarchical SDN Control System (need more text here).

### 5. Overview of Hierarchical SDN Control System

The Figure below illustrates a hierarchical SDN control system. There is one Parent Controller and four Child Controllers: Child Controller 1, Child Controller 2, Child Controller 3 and Child Controller 4.



The parent controller communicates with these four child controllers and controls them, each of which controls (or is responsible for) a domain. Child controller 1 controls domain 1, Child controller 2 controls domain 2, Child controller 3 controls domain 3, and Child controller 4 controls domain 4.

One level of hierarchy of controllers is illustrated in the figure above. There is one parent controller at top level, which is not a child controller. Under the parent controller, there are four child controllers, which are not parent controllers.

In a general case, at top level there is one parent controller that is not a child controller, there are some controllers that are both parent controllers and child controllers, and there are a number of child controllers that are not parent controllers. This is a system

of multiple levels of hierarchies, in which one parent controller controls or communicates with a first number of child controllers, some of which are also parent controllers, each of which controls or communicates with a second number of child controllers, and so on.

Considering one parent controller and its child controllers, each of the child controllers controls a domain and has the topology information on the domain, the parent controller does not have the topology information on any domain controlled by a child controller normally. This is called parent without domain topology.

In some special cases, the parent controller has the topology information on a region consisting of the domains controlled by its child controllers. In other words, the parent controller has the topology information on the domains controlled by its child controllers and the topology/inter-connections among these domains. This is called parent with domain topology.

The parent controller receives requests for creating end to end tunnels from users or applications. For each request, the parent controller is responsible for obtaining a path for the tunnel and creating the tunnel along the path.

For parent without domain topology, the parent controller asks each of its related child controllers to compute path segments from an entry boundary node to exit boundary nodes in the domain it controls or path segments from an exit boundary node in its domain to entry boundary nodes of other adjacent domains just using the inter-domain links attached to the exit boundary node. The details of the segments are hidden from the parent, which sees each of the segments as a link from a boundary node to another boundary node with a cost. The parent controller builds a shortest path tree (SPT) using the path segments computed as links to get the end to end path and then creates the tunnel along the path by asking its related child controllers.

The end to end path does not have any details from the parent's point of view. It can be considered as a sequence of domains containing the shortest path. Along this sequence of domains, the details of the end to end path can be obtained. And then the tunnel along the path with details can be created.

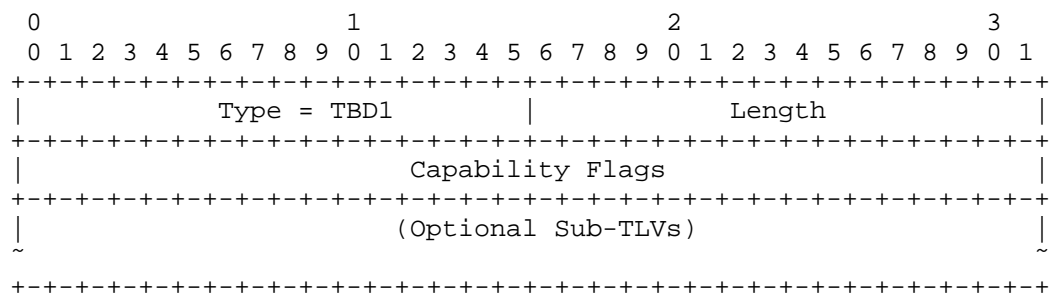
For parent with domain topology, the parent controller computes a path for the tunnel using the topology information on the domains controlled by its child controllers. And then it creates the tunnel along the path computed through asking its related child controllers.

## 6. Extensions to PCEP

This section describes the extensions to PCEP for a Hierarchical SDN Control System (HSCS). The extensions include the definition of a new flag in the RP object, a global tunnel identifier (GTID), a path identifier (PID), a list of path segments and an exception list in the PCReq and PCRep message.

### 6.1. Capability Discovery

During a PCEP session establishment between two PCEP speakers (PCE or PCC), each of them advertises its capabilities for HSCS through the Open Message with the Open Object containing a new TLV to indicate its capabilities for HSCS. This new TLV is called HSCS capability TLV. It has the following format.



The type of the TLV is TBD1. It has a length of 4 octets plus the size of optional Sub-TLVs. The value of the TLV comprises a capability flags field of 32 bits, which are numbered from the most significant as bit zero. Each bit represents a capability.

- o PC (Parent Controller - 1 bit): Bit 0 is used as PC flag. It is set to 1 indicating a parent controller.
- o CC (Child Controller - 1 bit): Bit 1 is used as PC flag. It is set to 1 indicating a child controller.
- o PS (Path Segments - 1 bit): Bit 2 is used as PS flag. It is set to 1 indicating support for computing path segments for HSCS
- o TS (Tunnel Segment - 1 bit): Bit 3 is used as TS flag. It is set to 1 indicating support for creating tunnel segment for HSCS

- o ET (End to end Tunnel - 1 bit): Bit 4 is used as ET flag. It is set to 1 indicating support for creation and maintenance of end to end LSP tunnels

## 6.2. New Messages for Hierarchical SDN Control System

This section describes the contents and semantics of the new messages, and presents a few of different encodings for the messages.

There are a number of new messages for supporting HSCS. These new messages can be encoded in a few of ways as follows:

- o To use a new type at top level for each of the new messages. This is called individual encoding.
- o To use a new type at top level for each group of the new messages and a option/operation/sub-type value for every message in the group. This is called group encoding.
- o To use/re-use existing messages and a value of options/operations for each new message in an existing message. This is called embedded encoding.
- o To combine the ways above. This is called mixed encoding.

Various types of messages for supporting HSCS are listed below. Note that many new messages may not be needed for some procedures/options. For example, four messages Request and Reply for Removing Path Segments and Request and Reply for Keeping Path Segments are not needed if path segments computed are not stored/remembered by a child controller. But in this case, the path segment in each domain along the end to end path computed needs to be re-computed when a tunnel along the path is set up.

Message for Controller Relation Discovery: It is a message exchanged between a parent controller and a child controller for discovering their parent-child relation.

Message for Connections and Accesses Advertisement: It is a message that a child controller sends its parent controller to describe the connections from the domain it controls to its adjacent domains and the access points in the domain to be accessible outside of the domain.

Request for Computing Path Segments: It is a message that a parent controller sends a child controller to request the child controller for computing path segments in the domain the child controller controls.

Reply for Computing Path Segments: It is a message that a child controller sends a parent controller to reply the parent controller for a request message for computing path segments after receiving the request message from the parent controller for computing path segments and computing path segments as requested, which normally contains the path segments computed.

Request for Removing Path Segments: It is a message that a parent controller sends a child controller to request the child controller for removing the path segments computed by the child controller and stored in the child controller.

Reply for Removing Path Segments: It is a message that a child controller sends a parent controller to reply the parent controller for a request message for removing a set of path segments after receiving the request message from the parent controller for removing path segments and removing the path segments as requested, which normally contains a status of removing path segments.

Request for Keeping Path Segments: It is a message that a parent controller sends a child controller to request the child controller for keeping a set of path segments computed by the child controller and stored in the child controller.

Reply for Keeping Path Segments: It is a message that a child controller sends a parent controller to reply the parent controller for a request message for keeping path segments after receiving the request message from the parent controller for keeping path segments and keeping the path segments as requested, which normally contains a status of keeping path segments.

Request for Creating Tunnel Segment: It is a message that a parent controller sends a child controller to request the child controller for creating tunnel segments related to the domain the child controller controls.

Reply for Creating Tunnel Segment: It is a message that a child controller sends a parent controller to reply the parent controller for a request message for creating tunnel segment after receiving the request message from the parent controller for creating tunnel segment and creating tunnel segment as requested, which normally contains a status of creating tunnel segment and a label and an interface.



**Request for Removing Tunnel Segment:** It is a message that a parent controller sends a child controller to request the child controller for removing the tunnel segment created by the child controller.

**Reply for Removing Tunnel Segment:** It is a message that a child controller sends a parent controller to reply the parent controller for a request message for removing tunnel segment after receiving the request message from the parent controller for removing tunnel segment and removing the tunnel segment as requested, which normally contains a status of removing tunnel segment.

#### 6.2.1. Contents of Messages

This section describes the contents in each of the messages and gives the format of each of messages in individual encoding, which is the same as in group encoding. Some of the objects in the messages are defined in the following sections.

##### 6.2.1.1. Message for Controller Relation Discovery

A message for controller relation discovery is exchanged between a parent controller and a child controller for discovering their parent-child relation.

A message for controller relation discovery (CRDis message for short) sent from a local controller to a remote controller comprises:

- o Local controller attributes
- o Remote controller attributes after the local controller receives the remote controller attributes from a remote end and determines that the relation between the local controller and the remote controller can be formed.

The format of the CRDis message is as follows:

```
<CRDis Message> ::= <Common Header>
                      <CRP>
                      <Local-Controller>
                      [<Remote-Controller>]
```

where CRP (Controller Request Parameters) object is defined in section Objects and TLVs.

#### 6.2.1.2. Message for Connections and Accesses Advertisement

After a child controller discovers its parent controller, it sends its parent controller a message for connections and accesses advertisement.

A message for connections and accesses advertisement (CAAdv message for short) from a child controller comprises:

- o Inter-domain links from the domain the child controller controls to its adjacent domains.
- o The addresses in the domain to be accessible to the outside of the domain.
- o Attributes of each of the boundary nodes of the domain.

The format of the CAAdv message is as follows:

```
<CAAdv Message> ::= <Common Header>
                    <CRP>
                    <Inter-Domain-Link-List>
                    [<Access-Address-List>]
where:
<Inter-Domain-Link-List> ::= <Inter-Domain-Link>
                             [<Inter-Domain-Link-List>]
<Access-Address-List> ::= <Access-Address>
                          [<Access-Address-List>]
```

#### 6.2.1.3. Request for Computing Path Segments

After receiving a request for creating an end to end tunnel from source A to destination Z for a given set of constraints, a parent controller allocates a global tunnel identifier (GTID) for the end to end tunnel crossing domains and a path identifier (PID) for an end to end path to be computed for the tunnel. The parent controller sends a request message to each of its related child controllers for computing a set of path segments in the domain the child controller controls in a special order. The parent controller builds a shortest path tree (SPT) using these path segments and obtains a shortest path from source A to destination Z that satisfies the constraints.

Note: The details of the path segments are hidden from the parent, which sees each of the segments as a link from one (boundary) node to another (boundary) node with a cost. The end to end path does not have any details from the parent's point of view, which may be considered as a domain path.

A request message for computing path segments (PSReq message for short) from a parent controller to a child controller comprises:

- o The address or identifier of the start-node (saying X) in the domain controlled by the child controller. From this node, a number of path segments are to be computed.
- o The global tunnel identifier (GTID) and the path identifier (PID). For the path of the tunnel, a number of path segments are to be computed.
- o An exception list containing the nodes that are on the SPT and in the domain controlled by the child controller or its adjacent domains.
- o The constraints for the path such as bandwidth constraints and color constraints.
- o A destination node Z. If Z is in the domain controlled by the child controller, the child controller computes a shortest path segment satisfying the constraints from node X to node Z within the domain.
- o Options for computing path segments:

E: E set to 1 indicating computing a shortest path segment satisfying the constraints from node X to each of the edge nodes of the domain controlled by the child controller except for the nodes in the exception list. E is set to 1 if there is not any previous hop of node X in the domain.

After receiving the request message, the child controller computes a shortest path segment satisfying the constraints from node X to each of the edge nodes of the domain controlled by the child controller except for the nodes in the exception list if E is 1. In addition, it computes a shortest path segment satisfying the constraints from node X to each of the edge nodes of the adjacent domains except for the edge nodes in the exception list just using the inter-domain links attached to node X if node X is an edge node of the domain and an end point of an inter-domain link.

The format of the PSReq message is as follows:

```

<PSReq Message> ::= <Common Header>
                    [<svec-list>]
                    <path-segment-request-list>
where:
  <svec-list> ::= <SVEC> [<svec-list>]
  <path-segment-request-list> ::=
    <path-segment-request>
    [<path-segment-request-list>]

  <path-segment-request> ::=
    <CRP>
    <Start-Node> <Tunnel-ID> <Path-ID>
    [<Destination>]
    [<OF>] [<LSPA>] [<BANDWIDTH>]
    [<metric-list>] [<RRO> [<BANDWIDTH>]] [<IRO>]
    [<LOAD-BALANCING>]
    <exception-list>

```

#### 6.2.1.4. Reply for Computing Path Segments

After receiving a request message from a parent controller for computing path segments, a child controller computes the path segments as requested in the message and sends the parent controller a reply message to reply the request message, which contains the path segments computed. The details of the path segments are hidden from the parent, which sees each of the path segments as a link with a cost.

A reply message for computing path segments (PSRep message for short) comprises:

- o The global tunnel identifier (GTID) and the path identifier (PID). For the path of the tunnel, the path segments are computed.
- o The address or identifier of the start-node (saying X) in the domain controlled by the child controller. From this node, the path segments are computed.
- o For each shortest path segment from node X to node Y computed, the address or identifier of node Y and the cost of the shortest path segment from node X to node Y.

The child controller stores the details about every shortest path segment computed under the global tunnel identifier (GTID) and the path identifier (PID) when it sends the reply message containing the path segments to the parent controller.

The child controller may delete the path segments computed for the global tunnel identifier (GTID) and the path identifier (PID) if it does not receive any request for keeping them from the parent controller for a given period of time.

The format of the PSRep message is as follows:

```

<PSRep Message> ::= <Common Header>
                        <path-segment-reply-list>
where:
  <path-segment-reply-list> ::=
    <path-segment-reply>
    [<path-segment-reply-list>]

  <path-segment-reply> ::=
    <CRP>
    <Tunnel-ID> <Path-ID>
    <Start-Node>
    [ <NO-PATH> | <segment-end-List> ]
    [<metric-list>]

```

#### 6.2.1.5. Request for Removing Path Segments

After a shortest path satisfying a set of constraints from source A to destination Z is computed, a parent controller may delete the path segments computed and stored in the related child controllers, which are not any part of the shortest path. A parent controller may send a child controller a request message for removing the path segments computed by the child controller and stored in the child controller.

1). A request message for removing path segments (RPSReq message for short) comprises:

- o The global tunnel identifier (GTID).

All the path segments stored under GTID in the child controller are to be removed.

2). A request message for removing path segments comprises:

- o The global tunnel identifier (GTID) and the path identifier (PID).

All the path segments stored under GTID and PID in the child controller are to be removed.

3). A request message for removing path segments comprises:

- o The global tunnel identifier (GTID) and the path identifier (PID)
- o A list of start point (or node) addresses or identifiers.

All the path segments stored in the child controller under GTID and PID and with a start point or node from the list of start point (or node) addresses or identifiers are to be removed.

4). A request message for removing path segments comprises:

- o The global tunnel identifier (GTID) and the path identifier (PID)
- o A list of start point (or node) addresses or identifiers
- o A list of pairs (start point, a list of end points), which identifies the path segments from start point of each pair to each of the end points in the list of the pairs.

In addition to the path segments as described in the previous message, the path segments stored in the child controller under GTID and PID and identified by the list of pairs are to be removed.

The format of the RPSReq message is as follows:

```

<RPSReq Message> ::= <Common Header>
                        <remove-path-segment-request-list>
where:
  <remove-path-segment-request-list> ::=
    <remove-path-segment-request>
    [<remove-path-segment-request-list>]

  <remove-path-segment-request> ::=
    <CRP>
    <Tunnel-ID> [<Path-ID>]
    [<start-node-list>]
    [<branch-List>]

  <start-node-list> ::= <Start-Node> [<start-node-list>]

  <branch-list> ::= <Branch> [<branch-list>]
  <Branch> ::= <Start-Node> <branch-end-list>

  <branch-end-list> ::= <Branch-End> [<branch-end-list>]

```

#### 6.2.1.6. Reply for Removing Path Segments

After removing the path segments as requested by a request message for removing path segments from a parent controller, a child controller sends the parent controller a reply message for removing path segments.

A reply message for removing path segments (RPSRep message for short) comprises:

- o The global tunnel identifier (GTID) and the path identifier (PID)
- o Status of the path segments removal:

Success: The path segments requested for removal are removed successfully.

Fail: The path segments requested for removal can not be removed.

- o Error code and reasons for failure if the status is Fail.

The format of the RPSRep message is as follows:

```
<RPSRep Message> ::= <Common Header>
                        <remove-path-segment-reply-list>
where:
  <remove-path-segment-reply-list> ::=
    <remove-path-segment-reply>
    [<remove-path-segment-reply-list>]

  <remove-path-segment-reply> ::=
    <CRP>
    <Tunnel-ID> [<Path-ID>]
    <Status>
    [<Reasons>]
```

#### 6.2.1.7. Request for Keeping Path Segments

After a shortest path satisfying a set of constraints from source A to destination Z is computed, a parent controller may send a request message for keeping path segments to each of the related child controllers to keep the path segments on the shortest path.

A request message for keeping path segments (KPSReq message for short) comprises:

- o The global tunnel identifier (GTID) and the path identifier (PID).
- o A list of pairs (start point, end point), each of which identifies the path segment from the start point of the pair to the end point of the pair.

The child controller will keep the path segments given by the list of pairs (start point, end point) stored under GTID and PID. It will remove all the other path segments stored under GTID and PID.

The format of the KPSReq message is as follows:

```
<KPSReq Message> ::= <Common Header>
                        <keep-path-segment-request-list>
where:
  <keep-path-segment-request-list> ::=
    <keep-path-segment-request>
    [<keep-path-segment-request-list>]

  <keep-path-segment-request> ::=
    <CRP>
    <Tunnel-ID> <Path-ID>
    <segment-list>

  <segment-list> ::= <Segment> [<segment-list>]
  <Segment> ::= <Segment-Start> <Segment-End>
```

#### 6.2.1.8. Reply for Keeping Path Segments

After keeping path segments as requested by a request message for keeping path segments from a parent controller, a child controller sends the parent controller a reply message for keeping path segments.

A reply message for keeping path segments (KPSRep message for short) comprises:

- o The global tunnel identifier (GTID) and the path identifier (PID).
- o Status of the path segment retention:

Success: The path segments requested for retention are retained successfully.



Fail: The path segments requested for retention can not be retained.

- o Error code and reasons for failure if the status is Fail.

The format of the KPSRep message is as follows:

```
<KPSRep Message> ::= <Common Header>
                        <keep-path-segment-reply-list>
where:
  <keep-path-segment-reply-list> ::=
    <keep-path-segment-reply>
    [<keep-path-segment-reply-list>]

  <keep-path-segment-reply> ::=
    <CRP>
    <Tunnel-ID> <Path-ID>
    <Status>
    [<Reasons>]
```

#### 6.2.1.9. Request for Creating Tunnel Segment

After obtaining the end to end shortest point to point (P2P) path, a parent controller creates a tunnel along the path crossing multiple domains through sending a request message for creating tunnel segment to each of the child controllers along the path in a reverse direction to create a tunnel segment.

A request message for creating tunnel segment (CTSReq message for short) comprises:

- o The global tunnel identifier (GTID) and the path identifier (PID).
- o A path segment from a start point to an end point for parent without domain topology or a path segment details/ERO for parent with domain topology.
- o A label and an interface if the domain controlled by the child control is not a destination domain.

For parent without domain topology, the child controller allocates and reserves link bandwidth along the path segment identified by the start point and end point, assigns labels along the path segment, and writes cross connects on each of the nodes along the path segment.

For parent with domain topology, the child controller assigns labels along the path segment ERO and writes cross connects on each of the

nodes along the path segment. The link bandwidth along the path segment is allocated and reserved by the parent controller.

For the non destination domain, the child controller writes the cross connect on the edge node to the downstream domain using the label and the interface from the downstream domain in the message.

For the non source domain, the child controller will include a label and an interface in a message to be sent to the parent controller. The interface connects the edge node of the upstream domain along the path. The label is allocated for the interface on the node that is the next hop of the edge node.

The format of the CTSReq message is as follows:

```

<CTSReq Message> ::= <Common Header>
                        <create-tunnel-segment-request-list>
where:
  <create-tunnel-segment-request-list> ::=
    <create-tunnel-segment-request>
    [<create-tunnel-segment-request-list>]

  <create-tunnel-segment-request> ::=
    <CRP>
    <Tunnel-ID> <Path-ID>
    <Path-Segment>
    [<Label> <Interface>]

  <Path-Segment> ::= [<Segment-Start> <Segment-End> | <ERO> ]

```

#### 6.2.1.10. Reply for Creating Tunnel Segment

After creating tunnel segment as requested by a request message for creating tunnel segment from a parent controller, a child controller sends the parent controller a reply message for creating tunnel segment.

A reply message for creating tunnel segment (CTSRep message for short) comprises:

- o The global tunnel identifier (GTID) and the path identifier (PID).
- o Status of the tunnel segment creation:

Success: The tunnel segment requested is created successfully.

Fail: The tunnel segments requested can not be created.

- o A label and an interface if the domain controlled by the child controller is not source domain and the status is Success.
- o Error code and reasons for failure if the status is Fail.

For the non source domain controlled by the child controller, the interface in the message connects the edge node of the upstream domain along the path, the label is allocated for the interface on the node that is the next hop of the edge node.

The format of the CTSRep message is as follows:

```

<CTSRep Message> ::= <Common Header>
                        <create-tunnel-segment-reply-list>
where:
  <create-tunnel-segment-reply-list> ::=
    <create-tunnel-segment-reply>
    [<create-tunnel-segment-reply-list>]

  <create-tunnel-segment-reply> ::=
    <CRP>
    <Tunnel-ID> <Path-ID>
    <Status> [<Label> <Interface>]
    [<Reasons>]

```

#### 6.2.1.11. Request for Removing Tunnel Segment

When a parent controller receives a request for deleting a tunnel from a user or an application, or receives a reply message for creating tunnel segment with status of Fail from a child controller, the parent controller will delete the tunnel through sending a request message for removing tunnel segment to each of the related child controllers.

A request message for removing tunnel segment (RTSReq message for short) comprises:

- o The global tunnel identifier (GTID) and the path identifier (PID).

The child controller releases the labels assigned along the path segments under GTID and PID, and removes the cross connects on each of the nodes along the path segments. If the child controller reserved the link bandwidth along the path segments under GTID and

PID, it releases the link bandwidth reserved.

The format of the RTSReq message is as follows:

```
<RTSReq Message> ::= <Common Header>
                        <remove-tunnel-segment-request-list>
where:
  <remove-tunnel-segment-request-list> ::=
    <remove-tunnel-segment-request>
    [<remove-tunnel-segment-request-list>]

  <remove-tunnel-segment-request> ::=
    <CRP>
    <Tunnel-ID> [<Path-ID>]
```

#### 6.2.1.12. Reply for Removing Tunnel Segment

After removing the tunnel segment as requested by a request message for removing tunnel segment from a parent controller, a child controller sends the parent controller a reply message for removing tunnel segment.

A reply message for removing tunnel segment (RTSRep message for short) comprises:

- o The global tunnel identifier (GTID) and the path identifier (PID).
- o Status of the tunnel segment removal:
  - Success: The tunnel segment requested is removed successfully.
  - Fail: The tunnel segment requested can not be removed.
- o Error code and reasons for failure if the status is Fail.

The format of the RTSRep message is as follows:

```

<RTSRep Message> ::= <Common Header>
                        <remove-tunnel-segment-reply-list>
where:
  <reply-tunnel-segment-reply-list> ::=
    <remove-tunnel-segment-reply>
    [<remove-tunnel-segment-reply-list>]

  <remove-tunnel-segment-reply> ::=
    <CRP>
    <Tunnel-ID> [<Path-ID>]
    <Status>
    [<Reasons>]

```

### 6.2.2. Individual Encoding of Messages

The format of PCEP Message Common Header is as follows:

```

      0               1               2               3
    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  | Message-Type |           Message-Length           |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

Message-Type (8 bits): The following message types are currently defined (refer to RFC 5440):

Message-Type	Meaning
1	Open
2	Keepalive
3	Path Computation Request
4	Path Computation Reply
5	Notification
6	Error
7	Close

The new message types are defined as follows:

Message-Type	Meaning
mTBD1	Controller Relation Discovery
mTBD2	Connections and Accesses Advertisement
mTBD3	Path Segment Computation Request
mTBD4	Path Segment Computation Reply
mTBD5	Remove Path Segment Request
mTBD6	Remove Path Segment Reply
mTBD7	Keep Path Segment Request
mTBD8	Keep Path Segment Reply
mTBD9	Create Tunnel Segment Request
mTBD10	Create Tunnel Segment Reply
mTBD11	Remove Tunnel Segment Request
mTBD12	Remove Tunnel Segment Reply

Ver, Flags and Message-Length are defined as RFC 5440.

#### 6.2.3. Group Encoding of Messages

We can encode the tunnel related messages into two groups: one group comprises the request messages related to tunnel and the other comprises the reply messages related to tunnel. Thus we can have four new message types, which are defined in PCEP Message Common Header as follows:

Message-Type	Meaning
mTBD1	Controller Relation Discovery
mTBD2	Connections and Accesses Advertisement
mTBD3	Tunnel Segment Operation Request
mTBD4	Tunnel Segment Operation Reply

Ver, Flags, other message types and Message-Length in PCEP Message Common Header are defined as RFC 5440.

The Tunnel Segment Operation can be one of the followings:

Create Tunnel Segment: Create a segment of an end to end tunnel.

Remove Tunnel Segment: Remove a segment of an end to end tunnel.

Compute Path Segments: Compute some path segments to find an end to end path for an end to end tunnel.

Remove Path Segments: Remove some path segments.

Keep Path Segment: Keep path segments on an end to end path for an end to end tunnel.

Each of these operations can be indicated by a value of options field of an object such as CRP object following PCEP Message Common Header in a message.

#### 6.2.4. Embedded Encoding of Messages

Each of the request messages can be encoded as a Path Computation Request message with a value of options/operations in an existing object. Each of the reply messages can be encoded as a Path Computation Reply message with a value of options/operations in an existing object.

A new options/operations field of 3 bits may be defined in the existing RP object. Thus each of the five request messages for supporting HSCS can be represented by a Path Computation Request message with a corresponding Options value in the RP object listed below. Each of the five reply messages for supporting HSCS can be represented by a Path Computation Reply message with a corresponding Options value in the RP object listed below.

Options Value	Meaning
oTBD1	Path Segment Computation Request/Reply
oTBD2	Remove Path Segment Request/Reply
oTBD3	Keep Path Segment Request/Reply
oTBD4	Create Tunnel Segment Request/Reply
oTBD5	Remove Tunnel Segment Request/Reply

Each request/reply message contains the contents for the message described in the previous section.

The Controller Relation Discovery message may be encoded as a Open message with a flag or a value of options/operations in an existing object. The Open message as a Controller Relation Discovery message contains the contents for the Discovery message described in the previous section.

The Connections and Accesses Advertisement message may be encoded as a Report message with a flag or a value of options/operations in an existing object such as SRP object. The Report message as a Connections and Accesses Advertisement message contains the contents of the Connections and Accesses Advertisement message described in the previous section.

#### 6.2.5. Mixed Encoding of Messages

Some of the above encodings can be combined to form a mixed encoding of the messages for supporting HSCS. For example, one mixed encoding of the messages is as follows:

- o Using Individual Encoding for Connections and Accesses Advertisement message and
- o Using Embedded Encoding for Controller Relation Discovery, all the request and reply messages for supporting HSCS.

Another mixed encoding of messages is below:

- o Using Embedded Encoding for Controller Relation Discovery;
- o Using Individual Encoding for Connections and Accesses Advertisement message and
- o Using Group Encoding for all the request and reply messages for supporting HSCS.

#### 6.3. Controller Relation Discovery

This section presents two approaches for discovering controller relation. One uses the Open Message with some simple extensions. The other uses a new message for Controller Relation Discovery, called a discovery message.

##### 6.3.1. Using Open Message

For a parent controller P and a child controller C connected by a PCE session and having a normal PCE peer adjacency, their parent-child relation is discovered through Open Messages exchanged between the parent controller and the child controller. The following is a sequence of events related to a controller relation discovery.

Controller P sends controller C an Open Message containing a capability TLV with parent flag PC set to 1 after controller C is configured as a child controller over the PCE session between P and C.



```

      P                                     C
Configure C as                             Configure P as
Child Controller                           Parent Controller

      Open Message (PC=1)
      -----> Remote P is Parent and
                is same as configured
                Form Child-Parent relation

      Open Message (CC=1)
      <-----
Remote C is Child and
is same as configured
Form Parent-Child relation

```

When C receives the Open Message from P and determines that PC=1 in the message is consistent with the parent controller configured locally, it forms Child-Parent relation between C and P. It sends controller P an Open Message containing a capability TLV with child controller flag CC set to 1 after controller P is configured as a parent controller over the PCE session between C and P.

When P receives the Open Message from C and determines that CC=1 in the message is consistent with the Child controller configured locally, it forms Parent-Child relation between P and C.

After the Parent-Child relation between P and C is formed, this relation is broken if the configuration "C as Child Controller" on parent controller P is deleted or "P as Parent Controller" on child controller C is removed.

When the configuration "C as Child Controller" is deleted from parent controller P, P breaks/removes the Parent-Child relation between P and C and sends C an Open Message with PC = 0. When child controller C receives the Open Message with PC = 0 from P, it determines that the remote end P is no longer its parent controller as configured locally and breaks/removes the Child-Parent relation between C and P.

When the configuration "P as Parent Controller" is deleted from child controller C, C breaks/removes the Child-Parent relation between C and P and sends P an Open Message with CC = 0. When parent controller P receives the Open Message with CC = 0 from C, it determines that the remote end C is no longer its child controller as configured locally and breaks/removes the Parent-Child relation between P and C.

### 6.3.2. Using Discovery Message

For a parent controller P and a child controller C connected by a PCE session and having a normal PCE peer adjacency, their parent-child relation is discovered through messages for controller relation discovery exchanged between the parent controller and the child controller. The following is a sequence of events related to a controller relation discovery.

Controller P sends controller C a message containing a local controller (LC=) P with a parent flag set to 1 after controller C is configured as a child controller over a PCE session between P and C.

<p>P</p> <p>Configure C as child</p> <p>message (LC=P)</p> <p>-----&gt;</p> <p>message (LC=C, RC=P)</p> <p>&lt;-----</p> <p>Remote see me and same as configured</p> <p>Form Parent-Child relation</p> <p>Add C as remote controller</p> <p>message (LC=P, RC=C)</p> <p>-----&gt;</p>	<p>C</p> <p>Configure P as parent</p> <p>LC in Msg same as configured</p> <p>Add P as remote controller</p> <p>Remote see me</p> <p>Form Child-Parent relation</p>
---	--

When C receives the message from P and determines that the local controller (LC=) P in the message is the same as the parent controller configured locally, it sends controller P a message containing local controller (LC=) C and remote controller (RC=) P.

When P receives the message from C and determines that the local controller (LC=) C in the message is the same as the child controller configured locally and the remote controller C sees me controller P (RC=P in the message), it forms a parent-child relation between P and C and sends controller C another message containing local controller (LC=) P and remote controller (RC=) C.

When C receives the message from P and determines that the local controller (LC=) P in the message is the same as the parent controller configured locally and the remote controller P sees me controller C (RC=C in the message), it forms a child-parent relation between C and P.

#### 6.4. Connections and Accesses Advertisement

A child controller sends its parent controller a message for connections and accesses, which contains the connections (i.e., inter-domain links) connecting the domain that the child controller controls to other adjacent domains, and the addresses/prefixes (i.e., the access points) in the domain to be accessible from outside of the domain.

When there is a change on the connections and the accesses of the domain, the child controller sends its parent controller a updated message for the connections and accesses, which contains the latest connections and accesses of the domain.

A parent controller stores the connections and accesses for each of its child controllers according to the messages for connections and accesses received from the child controllers. For a updated message, it updates the connections and accesses accordingly.

When a child controller is down, its parent controller may remove the connections and accesses of the domain controlled by the child controller.

After connections and accesses advertisement, a parent controller has the exterior information about all the domains controlled by its child controllers. In other words, a parent controller has the connections among the domains (i.e., the inter-domain links connecting the domains) controlled by its child controllers and the addresses/prefixes (i.e., access points) in the domains to be accessible.

A connection comprises: the attributes for a link connecting domains and the attributes for the end points of the link. The attributes for an end point of a link comprises the type of the end point node such as ABR or ASBR, and the domain of the end point such AS number and area number.

An access point comprises an address or a prefix of a domain to be accessible outside of the domain.

#### 6.5. Tunnel Creation

This section describes a couple of procedures for computing a shortest end to end path for a tunnel, and then a procedure for creating the tunnel along the path. One procedure for computing a end to end path takes two rounds of computations. The first round obtains an end to end path without any details on any of the path segments along the path. This path can be considered as a domain

path. In the second round, the details on each of the path segments along the domain path are computed. The other procedure is to get an end to end path in one round.

#### 6.5.1. Computing Path in Two Rounds

After a parent controller receives a request for creating an end to end tunnel from source A to destination Z for a given set of constraints, it computes an end to end path in two rounds as follows:

Round 1: Obtain a domain path

Roughly speaking, obtaining a domain path consists of the following three steps:

Step 1: The parent controller sends a request message to each of its related child controllers for computing a set of path segments in the domain the child controller controls in a special order.

Step 2: After a child controller receives the request message, it computes the path segments as requested and sends the parent controller a reply message with the path segments computed as links. It does not store any details about the path segments it computes. The details of the path segments are hidden from the parent controller, which sees each of the segments as a link from one (boundary) node to another (boundary) node with a cost.

Step 3: The parent controller builds a shortest path tree (SPT) using these path segments and obtains a shortest path from source A to destination Z that satisfies the constraints.

Details for obtaining a domain path are described below:

Step 1: The parent controller selects the node just added to the SPT (Initially, it selects the source).

Step 2: After selecting the node just added into the SPT, the parent controller chooses the child controller controlling the domain containing the node, and determines whether the node is destination.

For destination node, the parent controller stops computing path since the end to end (domain) path from source to destination is in the SPT, which is from the root of the SPT to the node (destination node) in the SPT.

For non-destination node X, the parent controller sends the child controller a request message for computing path segments in the domain controlled by the child controller.

- o After receiving the request message, the child controller computes the path segments as requested and sends the parent controller a reply message with the path segments computed as links. It does not store any details about the path segments it computes. The details of the path segments are hidden from the parent controller, which sees each of the segments as a link from one (boundary) node to another (boundary) node with a cost.

Step 3: After receiving the reply message from the child controller, the parent controller updates the candidate list with the links, picks up a node in the candidate list with the minimum cost and adds it into the SPT. Repeat step 1.

Round 2: Obtain the path details

After obtaining a domain path, the parent controller may initiate a BRPC procedure along the domain path to get the end to end path. Each of the child controllers controlling the domains along the domain path may store the details of the path segment it computes using a path key.

#### 6.5.2. Computing Path in One Round

For a top level parent without domain topology, the parent controller computes a shortest point to point (P2P) path for a tunnel from a source to a destination satisfying a set of constraints given to the tunnel through building a shortest path tree (SPT). The SPT is built from the source as the root of the SPT with an empty candidate list in the following steps.

Step 1: The parent controller selects the node just added to the SPT (Initially, it selects the source).

Step 2: After selecting the node just added into the SPT, the parent controller chooses the child controller controlling the domain containing the node, and determines whether the node is destination.

For destination node, the parent controller stops computing path since the end to end path from source to destination is in the SPT, which is from the root of the SPT to the node (destination node) in the SPT.

For non-destination node X, the parent controller sends the child controller a request message for computing path segments related to the domain controlled by the child controller. The request contains the exception list for the domain and flag E.

- o After receiving the request message, the child controller computes a shortest path segment from node X to each of the edge nodes of the domain not in the exception list if E is 1.
- o In addition, it computes a shortest path segment from node X to each of the edge nodes of the adjacent domains not in the exception list just using the inter-domain links attached to node X if node X is an edge node and there is an inter-domain link attached to it.
- o If node X is in the destination domain, it computes a shortest path segment from node X to the destination.
- o It sends the parent controller a reply message with the path segments computed as links and stores the details of the path segments temporarily.

Step 3: After receiving the reply message from the child controller, the parent controller updates the candidate list with the links, picks up a node in the candidate list with the minimum cost and adds it into the SPT. Repeat step 1.

For a parent without domain topology, if the parent controller is also a child controller of another upper level parent controller, after receiving a request for computing path segments from the upper level parent controller, the parent controller computes each of the path segments as requested in the same way as described above. It records and maintains the path segments computed under the GTID and PID in the request message received from the upper level parent controller.

In addition, for each path segment to be computed, it allocates a new GTID and PID for the path segment and computes the path segment through sending a request message for computing path segments to each of its related child controllers using the new GTID and PID.

When the parent as a child controller receives a request message for removing path segments from the upper level parent controller, it removes the path segments computed by each of its related child controllers through sending a request message for removing path segments to each of the related child controllers, and then it removes the path segments crossing multiple domains controlled by its

child controllers.

#### 6.5.3. Creating Tunnel along Path

After obtaining the end to end shortest point to point (P2P) path, the parent controller creates a tunnel along the path crossing multiple domains through requesting the child controllers along the path in a reverse direction.

For a parent without domain topology, the following is the procedure for creating the tunnel along the path, which is initiated by the parent controller starting from domain X = destination domain.

Step 1: The parent controller sends the child controller controlling domain X a request message for creating tunnel segment in domain X.

- o After receiving the request message from the parent controller, the child controller creates the tunnel segment in domain X it controls through reserving the resources such as link bandwidth, allocating labels along the path segment and writing a cross connect on every node in the domain along the path.
- o If the child controller is not destination controller, the request message contains an label and interface for the next hop of the edge node of domain X. The label is allocated by the controller that controls the downstream domain of domain X. The child controller uses this label and an incoming label allocated for the incoming interface on the edge node to write a cross connect on the edge node.
- o The child controller sends the parent controller a reply message with the status of the tunnel segment creation. The reply message contains an incoming label and interface for the next hop of the edge node of the upstream domain of domain X if domain X is not source domain.

Step 2: The parent controller receives the reply message from child controller C. If the status in the message is Fail, then it removes the tunnel segments created for the tunnel and return with failure for creating the tunnel.

Step 3: If child controller C is the source controller, then the end to end tunnel is created, and the parent controller and the child controllers along the tunnel maintain the information of the tunnel with the GTID and PID. The parent controller returns with success for creating the tunnel.

Step 4: Child controller C is not source controller. The reply message contains the label and interface, the parent controller repeats step 1 with domain X = the upstream domain of domain X. (In other words, it sends a request message to the child controller that controls the domain which is the upstream domain of the domain in which a tunnel segment is just created. The request contains the label and interface.)

For a parent with domain topology, the procedure for creating the tunnel along the path initiated by the parent controller is similar to the one described above, but has a few of changes to it, which are listed as follows:

- o The request message for creating tunnel segment sent to a child controller from the parent controller contains the detailed information about the path segment (such as ERO comprising every hop of the path segment) along which the tunnel segment to be created.
- o The child controller does not check or reserve resources such as link bandwidth along the path segment if the parent controller is responsible for allocating and reserving the resources along the path for the tunnel.
- o The child controller does not assign any labels along the path segment if the parent controller is responsible for assigning labels along the path for the tunnel. In this case, the request message for creating tunnel segment contains an label for every hop of the path segment. The reply message from the child controller to the parent controller does not contain any label or interface.

When the parent as a child controller receives a request message for creating tunnel segment along a path segment from the upper level parent controller, it gets the path segments for its related child controllers from the path segment in the message.

For the parent with domain topology, it obtains the detailed hop to hop information crossing multiple domains about the path segment stored by the parent controller using the GTID, PID and start point and end point of the path segment in the message received. The parent controller creates the tunnel segments in the multiple domains through sending a request message for creating tunnel to each of its related child controllers along the path in a reverse direction.

For the parent without domain topology, it obtains the detailed information about the path segment stored by the parent controller using the GTID, PID and start point and end point of the path segment



in the message received. The detailed information includes multiple path segments, each of which crosses a domain controlled by one of its related child controllers. These multiple path segments constitute the path segment in the message, which crosses multiple domains. The parent controller creates the tunnel segments in the multiple domains through sending a request message for creating tunnel to each of its related child controllers along the path in a reverse direction. For each of the path segments crossing a domain, the parent controller creates a tunnel segment along the path segment through sending a request message for creating tunnel to its child controller controlling the domain.

## 6.6. Objects and TLVs

### 6.6.1. CRP Objects

A Controller Request Parameters (CRP) object carried within each of the new messages for supporting HSCS is used to specify various parameters of a tunnel related operation request. The CRP object has Object-Class octBD1 and CRP Object-Type = 1. The format of the CRP body is as follows

[illegible]

The following flags are currently defined:

- o E (Edges of Domain): E set to 1 indicating computing a shortest path segment satisfying a given set of constraints from a start node to each of the edge nodes of the domain controlled by a child controller except for the nodes in a given exception list.

For Group Encoding of messages, a new Options field of 3 bits is defined in the flags field of the CRP object to tell the receiver of a message that the request/reply is for one of the five request/reply messages for supporting HSCS as follows:

Options	Meaning
1	Path Segment Computation Request/Reply
2	Remove Path Segment Request/Reply
3	Keep Path Segment Request/Reply
4	Create Tunnel Segment Request/Reply
5	Remove Tunnel Segment Request/Reply

### 6.6.2. LOCAL-CONTROLLER Object

A LOCAL-CONTROLLER (LC) Object is carried within a Controller Relation Discovery message. Two LC objects are defined: one for IPv4 and the other for IPv6. These two objects have the same Object-Class ocTBD2 but have different Object-Types.

#### 6.6.2.1. LOCAL-CONTROLLER Object for IPv4

The LOCAL-CONTROLLER Object for IPv4 (LC-IPv4 for short) has Object-Class ocTBD2 and Object-Type otTBD21. The format of the LC-IPv4 body is as follows:

```

      Object-Class = ocTBD2      Object-Type = otTBD21
      0              1              2              3
      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                                     |P| Level |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     Controller IPv4 Address                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+
~                                     Optional TLVs                                     ~
+-----+-----+-----+-----+-----+-----+-----+-----+

```

The LC-IPv4 object body has a 32-bit Flags field and a 32-bit Controller IPv4 Address. It may contain additional TLVs. No TLVs are currently defined.

The following flags are currently defined:

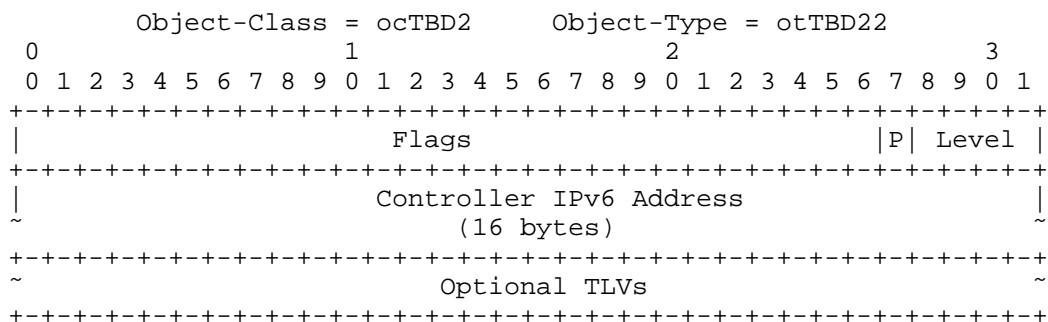
- o P (Parent Controller): P set to 1 indicating that the local controller is a Parent controller.
- o Level (Level as Parent): Level indicates the level of a controller as a parent controller. Level 0 means the highest (i.e., top) level as a parent controller. Level  $i$  ( $i > 0$ ) for a parent controller C means that C as a child controller has a parent controller of level  $(i - 1)$ .

Unassigned bits in the Flags field are considered reserved. They MUST be set to zero on transmission and MUST be ignored on receipt.

The Controller IPv4 Address indicates an IPv4 address of the local controller.

#### 6.6.2.2. LOCAL-CONTROLLER Object for IPv6

The LOCAL-CONTROLLER Object for IPv6 (LC-IPv6 for short) has Object-Class octBD2 and Object-Type otTBD22. The format of the LC-IPv6 body is as follows:



The LC-IPv6 object body has a 32-bit Flags field and a 128-bit Controller IPv6 Address. It may contain additional TLVs. No TLVs are currently defined.

The flag P (1 bit) and Level (4 bits) in the 32-bit Flags are the same as those defined in the LOCAL-CONTROLLER Object for IPv4.

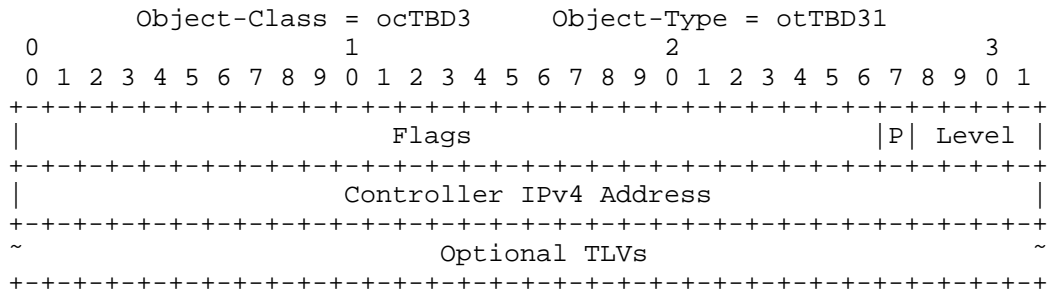
The Controller IPv6 Address indicates an IPv6 address of the local controller.

#### 6.6.3. REMOTE-CONTROLLER Object

When a local controller receives a Controller Relation Discovery message from a remote controller, the local controller MUST include a REMOTE-CONTROLLER (RC) Object with the remote controller in a Controller Relation Discovery message to be sent to the remote controller. Two RC objects are defined: one for IPv4 and the other for IPv6. These two objects have the same Object-Class octBD3 but have different Object-Types.

#### 6.6.3.1. REMOTE-CONTROLLER Object for IPv4

The REMOTE-CONTROLLER Object for IPv4 (RC-IPv4 for short) has Object-Class ocTBD3 and Object-Type otTBD31. The format of the RC-IPv4 body is as follows:



The RC-IPv4 object body has a 32-bit Flags field and a 32-bit Controller IPv4 Address. It may contain additional TLVs. No TLVs are currently defined.

The following flags are currently defined:

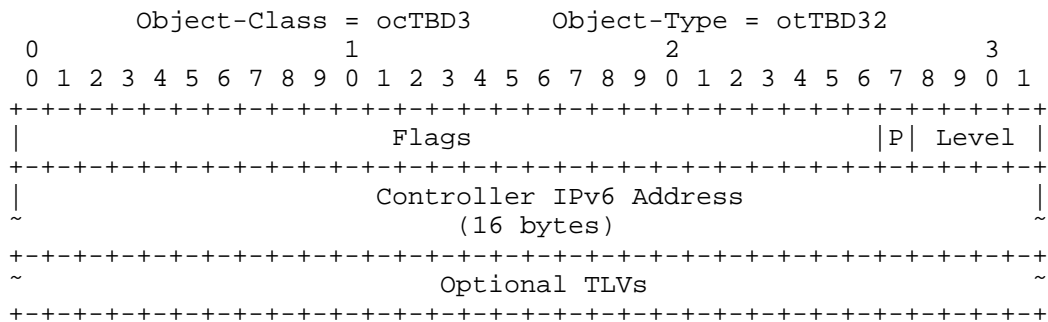
- o P (Parent Controller): P set to 1 indicating that the remote controller is a Parent controller.
- o Level (Level as Parent): Level indicates the level of a controller as a parent controller. Level 0 means the highest (i.e., top) level as a parent controller. Level  $i$  ( $i > 0$ ) for a parent controller C means that C as a child controller has a parent controller of level  $(i - 1)$ .

Unassigned bits in the Flags field are considered reserved. They MUST be set to zero on transmission and MUST be ignored on receipt.

The Controller IPv4 Address indicates an IPv4 address of the remote controller.

#### 6.6.3.2. REMOTE-CONTROLLER Object for IPv6

The REMOTE-CONTROLLER Object for IPv6 (RC-IPv6 for short) has Object-Class ocTBD3 and Object-Type otTBD32. The format of the RC-IPv6 body is as follows:



The LC-IPv6 object body has a 32-bit Flags field and a 128-bit Controller IPv6 Address. It may contain additional TLVs. No TLVs are currently defined.

The flag P (1 bit) and Level (4 bits) in the 32-bit Flags are the same as those defined in the REMOTE-CONTROLLER Object for IPv4.

The Controller IPv6 Address indicates an IPv6 address of the remote controller.

#### 6.6.4. CONNECTION and ACCESS Object

The CONNECTION and ACCESS Object (CA for short) has Object-Class ocTBD4. Three Object-Types are defined under CA object:

- o CA Inter-Domain Link: CA Object-Type is 1.
- o CA Access IPv4 Prefix: CA Object-Type is 2.
- o CA Access IPv6 Prefix: CA Object-Type is 3.

The format of each of these object bodies is as follows:

```

    Object-Class = ocTBD4 (Connection and Access)
    Object-Type = 1 (CA Inter-Domain Link)
      0          1          2          3
      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
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     AS Number                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     Area-ID TLV                                |
~-----~-----~-----~-----~-----~-----~-----~-----~-----~
|                                     IGP Router-ID TLV                          |
~-----~-----~-----~-----~-----~-----~-----~-----~-----~
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     Inter-Domain Link TLVs                      |
~-----~-----~-----~-----~-----~-----~-----~-----~-----~
+-----+-----+-----+-----+-----+-----+-----+-----+-----+

```

Each of the Inter-Domain Link TLVs describes an inter-domain link and comprises a number of inter-domain link Sub-TLVs.

```

    Object-Class = ocTBD4 (Connection and Access)
    Object-Type = 2 (CA Access IPv4 Prefix)
      0          1          2          3
      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
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     AS Number                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     Area-ID TLV                                |
~-----~-----~-----~-----~-----~-----~-----~-----~-----~
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     Access IPv4 Prefix TLVs                      |
~-----~-----~-----~-----~-----~-----~-----~-----~-----~
+-----+-----+-----+-----+-----+-----+-----+-----+-----+

```

```

Object-Class = ocTBD4 (Connection and Access)
Object-Type = 3 (CA Access IPv6 Prefix)
 0               1               2               3
 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
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     AS Number                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     Area-ID TLV                                 |
~                                                                                   ~
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     Access IPv6 Prefix TLVs                     |
~                                                                                   ~
+-----+-----+-----+-----+-----+-----+-----+-----+

```

The format of the Area-ID TLV is shown below:

```

 0               1               2               3
 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 (tTBD1)          |          Length (4)          |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     Area Number                             |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

The format of the OSPF Router-ID TLV is shown below:

```

 0               1               2               3
 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 (tTBD2)          |          Length (4)          |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     OSPF Router ID                         |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

The format of the ISIS Router-ID TLV is shown below:

```

 0               1               2               3
 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 (tTBD3)          |          Length (6)          |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     ISO Node-ID                             |
~                                                                                   ~
+-----+-----+-----+-----+-----+-----+-----+-----+

```

The format of the Access IPv4 Prefix TLV is shown as follows:

```

      0                   1                   2                   3
      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 (tTBD4)                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+
| Prefix Length | IPv4 Prefix (variable) |~
+-----+-----+-----+-----+-----+-----+-----+-----+

```

The format of the Access IPv6 Prefix TLV is illustrated below:

```

      0                   1                   2                   3
      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 (tTBD5)                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+
| Prefix Length | IPv6 Prefix (variable) |~
+-----+-----+-----+-----+-----+-----+-----+-----+

```

The format of the Inter-Domain link TLV is illustrated below:

```

      0                   1                   2                   3
      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 (tTBD6)                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     Inter-Domain Link Sub-TLVs                                     |~
+-----+-----+-----+-----+-----+-----+-----+-----+

```

The format of the Inter-Domain Link Type Sub-TLV is illustrated below:

```

      0                   1                   2                   3
      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 (1)                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     Inter-Domain Link Type                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

The Inter-Domain Link Type sub-TLV defines the type of the inter-domain link:



- 1 - Point-to-point
- 2 - Multi-access

The Inter-Domain Link Type sub-TLV is TLV type 1, and is one octet in length.

The format of the Remote AS Number ID Sub-TLV is illustrated below:

```

      0               1               2               3
      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 (2)             |               Length (4)           |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     |                                     |
|               Remote AS Number     |                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

The Remote AS Number field has 4 octets. When only two octets are used for the AS number, as in current deployments, the left (high-order) two octets MUST be set to zero.

The format of the Remote Area-ID Sub-TLV is shown below:

```

      0               1               2               3
      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 (3)             |               Length (4)           |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     |                                     |
|               Area Number          |                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

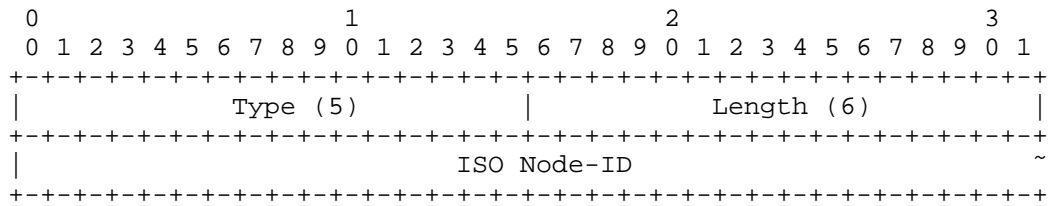
The format of the Remote OSPF Router-ID Sub-TLV is shown below:

```

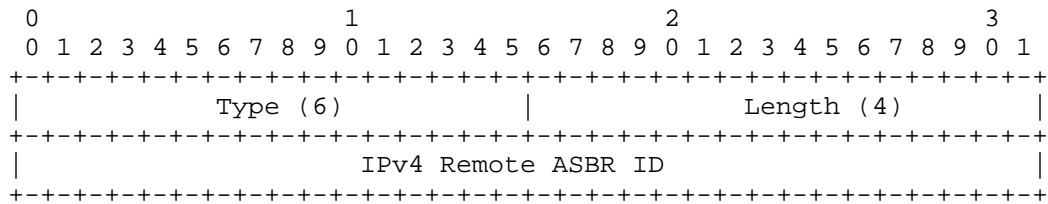
      0               1               2               3
      0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     |                                     |
|               Type (4)             |               Length (4)           |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     |                                     |
|               OSPF Router ID       |                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

The format of the Remote ISIS Router-ID Sub-TLV is shown below:

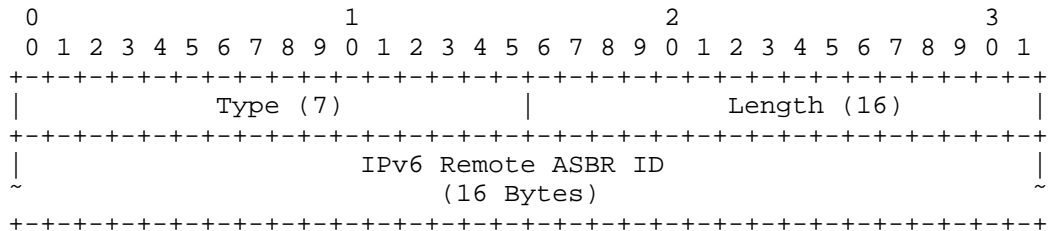


The format of the IPv4 Remote ASBR ID Sub-TLV is illustrated below:



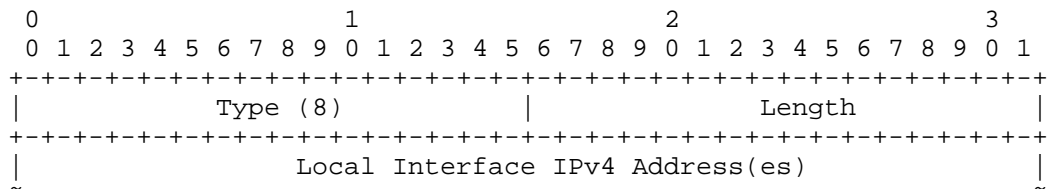
The IPv4 Remote ASBR ID sub-TLV MUST be included if the neighboring ASBR has an IPv4 address.

The format of the IPv6 Remote ASBR ID Sub-TLV is illustrated below:



The IPv6 Remote ASBR ID sub-TLV MUST be included if the neighboring ASBR has an IPv6 address.

The format of the Local Interface IPv4 Address Sub-TLV is shown below:



```

+-----+

```

The Local Interface IPv4 Address sub-TLV specifies the IPv4 address(es) of the interface corresponding to the inter-domain link. If there are multiple local addresses on the link, they are all listed in this sub-TLV.

The Local Interface IPv4 Address sub-TLV is TLV type 8, and is 4N octets in length, where N is the number of local IPv4 addresses.

The format of the Local Interface IPv6 Address Sub-TLV is illustrated below:

```

      0               1               2               3
      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 (9)             |               Length             |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     |                                     |
|               Local Interface IPv6 Address(es)                       |
|                                     |                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

The Local Interface IPv6 Address sub-TLV specifies the IPv6 address(es) of the interface corresponding to the inter-domain link. If there are multiple local addresses on the link, they are all listed in this sub-TLV.

The Local Interface IPv6 Address sub-TLV is TLV type 9, and is 16N octets in length, where N is the number of local IPv6 addresses.

The format of the Remote Interface IPv4 Address Sub-TLV is illustrated below:

```

      0               1               2               3
      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 (10)            |               Length             |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     |                                     |
|               Neighbor Interface IPv4 Address(es)                     |
|                                     |                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+

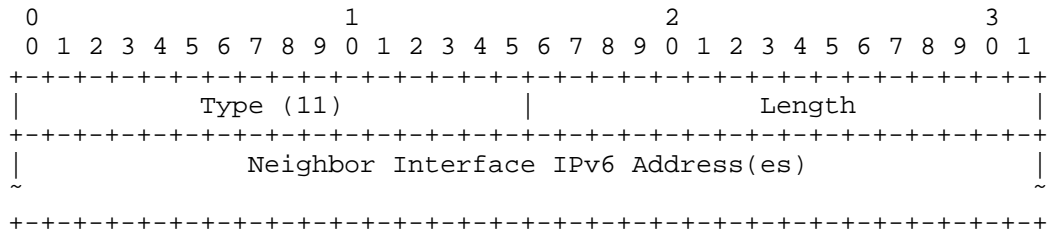
```

The Remote Interface IPv4 Address sub-TLV specifies the IPv4 address(es) of the neighbor's interface corresponding to the inter-domain link. This and the local address are used to discern multiple

parallel links between systems. If there are multiple remote addresses on the link, they are all listed in this sub-TLV.

The Remote Interface IPv4 Address sub-TLV is TLV type 10, and is  $4N$  octets in length, where  $N$  is the number of neighbor IPv4 addresses.

The format of the Remote Interface IPv6 Address Sub-TLV is illustrated below:



The Remote Interface IPv6 Address sub-TLV specifies the IPv6 address(es) of the neighbor's interface corresponding to the inter-domain link. If there are multiple neighbor addresses on the link, they are all listed in this sub-TLV.

The Remote Interface IPv6 Address sub-TLV is TLV type 11, and is  $16N$  octets in length, where  $N$  is the number of neighbor IPv6 addresses.

#### 6.6.5. NODE Object

The NODE Object has Object-Class octBD5. A number of Object-Types are defined under NODE object below:

1. IPv4 START-NODE: NODE Object-Type is 1.
2. IPv6 START-NODE: NODE Object-Type is 2.
3. IPv4 DESTINATION-NODE-LIST: NODE Object-Type is 3.
4. IPv6 DESTINATION-NODE-LIST: NODE Object-Type is 4.
5. IPv4 SEGMENT-END-NODE-LIST: NODE Object-Type is 5.
6. IPv6 SEGMENT-END-NODE-LIST: NODE Object-Type is 6.
7. IPv4 EXCEPTION-NODE-LIST: NODE Object-Type is 7.

- 8. IPv6 EXCEPTION-NODE-LIST: NODE Object-Type is 8.
- 9. NODE-IGP-METRIC-LIST: NODE Object-Type is 9.
- 10. NODE-TE-METRIC-LIST: NODE Object-Type is 10.
- 11. NODE-HOP-COUNT-LIST: NODE Object-Type is 11.

The format of NODE object body for IPv4 START-NODE is as follows:

```

Object-Class = octBD5 (NODE)
Object-Type = 1 (IPv4 START-NODE)
0               1               2               3
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
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                               Start Node IPv4 Address                               |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
```

The Start Node IPv4 Address is the IPv4 address of a start node.

The format of NODE object body for IPv6 START-NODE is as follows:

```

Object-Class = octBD5 (NODE)
Object-Type = 2 (IPv6 START-NODE)
0               1               2               3
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
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                               Start Node IPv6 Address                               |
|                               (16 bytes)                                           |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
```

The Start Node IPv6 Address is the IPv6 address of a start node.

The format of NODE object body for IPv4 DESTINATION-NODE-LIST is as follows:

```

    Object-Class = ocTBD5 (NODE)
    Object-Type = 3 (IPv4 DESTINATION-NODE-LIST)
      0               1               2               3
      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
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     Destination Node 1 IPv4 Address                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     . . . . .                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     Destination Node n IPv4 Address                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

The IPv4 DESTINATION-NODE-LIST contains n destination node IPv4 addresses. An IPv4 DESTINATION-NODE-LIST is also called an IPv4 DESTINATION-NODES.

The format of NODE object body for IPv6 DESTINATION-NODE-LIST is as follows:

```

    Object-Class = ocTBD5 (NODE)
    Object-Type = 4 (IPv6 DESTINATION-NODE-LIST)
      0               1               2               3
      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
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     Destination Node 1 IPv6 Address                                     |
|                                     (16 bytes)                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     . . . . .                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     Destination Node n IPv6 Address                                     |
|                                     (16 bytes)                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

The IPv6 DESTINATION-NODE-LIST contains n destination node IPv6 addresses. An IPv6 DESTINATION-NODE-LIST is also called an IPv6 DESTINATION-NODES.

The format of NODE object body for IPv4 SEGMENT-END-NODE-LIST is as follows:

```

    Object-Class = ocTBD5 (NODE)
    Object-Type = 5 (IPv4 SEGMENT-END-NODE-LIST)
      0               1               2               3
      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
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     Segment End Node 1 IPv4 Address                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     . . . . .                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     Segment End Node n IPv4 Address                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

The IPv4 SEGMENT-END-NODE-LIST contains n segment node IPv4 addresses. An IPv4 SEGMENT-END-NODE-LIST is also called an IPv4 SEGMENT-END-NODES.

The format of NODE object body for IPv6 SEGMENT-END-NODE-LIST is as follows:

```

    Object-Class = ocTBD5 (NODE)
    Object-Type = 6 (IPv6 SEGMENT-END-NODE-LIST)
      0               1               2               3
      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
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     Segment End Node 1 IPv6 Address                                     |
|                                     (16 bytes)                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     . . . . .                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     Segment End Node n IPv6 Address                                     |
|                                     (16 bytes)                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

The IPv6 SEGMENT-END-NODE-LIST contains n segment end node IPv6 addresses. An IPv6 SEGMENT-END-NODE-LIST is also called an IPv6 SEGMENT-END-NODES.

The format of NODE object body for IPv4 EXCEPTION-NODE-LIST is as follows:

```

    Object-Class = octBD5 (NODE)
    Object-Type = 7 (IPv4 EXCEPTION-NODE-LIST)
      0          1          2          3
      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
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     Exception Node 1 IPv4 Address                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     . . . . .                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     Exception Node n IPv4 Address                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

The IPv4 SEGMENT-END-NODE-LIST contains n node IPv4 addresses in an exception list. An IPv4 EXCEPTION-NODE-LIST is also called an IPv4 EXCEPTION-LIST.

The format of NODE object body for IPv6 EXCEPTION-NODE-LIST is as follows:

```

    Object-Class = octBD5 (NODE)
    Object-Type = 8 (IPv6 EXCEPTION-NODE-LIST)
      0          1          2          3
      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
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     Exception Node 1 IPv6 Address                                     |
|                                     (16 bytes)                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     . . . . .                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     Exception Node n IPv6 Address                                     |
|                                     (16 bytes)                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

The IPv6 EXCEPTION-NODE-LIST contains n node IPv6 addresses in an exception list. An IPv6 EXCEPTION-NODE-LIST is also called an IPv6 EXCEPTION-LIST.

The format of NODE object body for NODE-IGP-METRIC-LIST is as follows:



```

Object-Class = ocTBD5 (NODE)
Object-Type = 9 (NODE-IGP-METRIC-LIST)
0              1              2              3
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
+-----+-----+-----+-----+-----+-----+-----+-----+
|               Segment End Node 1 IGP Metric Value               |
+-----+-----+-----+-----+-----+-----+-----+-----+
|               . . . . .                                         |
+-----+-----+-----+-----+-----+-----+-----+-----+
|               Segment End Node n IGP Metric Value               |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

The NODE-IGP-METRIC-LIST contains n IGP metrics for n segment end nodes.

The format of NODE object body for NODE-TE-METRIC-LIST is as follows:

```

Object-Class = ocTBD5 (NODE)
Object-Type = 10 (NODE-TE-METRIC-LIST)
0              1              2              3
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
+-----+-----+-----+-----+-----+-----+-----+-----+
|               Segment End Node 1 TE Metric Value               |
+-----+-----+-----+-----+-----+-----+-----+-----+
|               . . . . .                                         |
+-----+-----+-----+-----+-----+-----+-----+-----+
|               Segment End Node n TE Metric Value               |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

The NODE-TE-METRIC-LIST contains n TE metrics for n segment end nodes.

The format of NODE object body for NODE-HOP-COUNT-LIST is as follows:

```

Object-Class = octBD5 (NODE)
Object-Type = 11 (NODE-HOP-COUNT-LIST)
0           1           2           3
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
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     Segment End Node 1 Hop Counts Value                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     . . . . .                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     Segment End Node n Hop Counts Value                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

The NODE-HOP-COUNT-LIST contains n hop counts values for n segment end nodes.

#### 6.6.6. TUNNEL Object

The TUNNEL Object has Object-Class octBD6. Two Object-Types are defined under TUNNEL object:

1. TUNNEL-ID: TUNNEL Object-Type is 1.
2. TUNNEL-PATH-ID: TUNNEL Object-Type is 2.

The format of TUNNEL object body for TUNNEL-ID is as follows:

```

Object-Class = octBD6 (TUNNEL)
Object-Type = 1 (TUNNEL-ID)
0           1           2           3
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
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     Tunnel ID                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

The Tunnel ID in the body is a 32-bit unique number for identifying a tunnel globally.

The format of TUNNEL object body for TUNNEL-PATH-ID is as follows:

```

Object-Class = octBD6 (TUNNEL)   Object-Type = 2 (PATH-ID)
0           1           2           3
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
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     Path ID                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

The Path ID in the body is a 16-bit number for uniquely identifying a path under a tunnel.

#### 6.6.7. STATUS Object

The STATUS Object has Object-Class octBD7. The format of STATUS object body has following format:

```

    Object-Class = octBD7 (STATUS)
    Object-Type = 1
      0               1               2               3
    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
+-----+-----+-----+-----+-----+-----+-----+-----+
| Status Code | Reason | Reserved |
+-----+-----+-----+-----+-----+-----+-----+-----+
~                               Optional TLVs                               ~
+-----+-----+-----+-----+-----+-----+-----+-----+

```

The status code (or status for short) in a STATUS may be one of the followings:

- 1 (SUCCESS): Indicating a request is successfully finished.
- 2 (FAIL): Indicating a request can not be finished.

When the status is FAIL, the Reason gives a reason for the failure and the Optional TLVs give some more details about failure.

#### 6.6.8. LABEL Object

The LABEL Object has Object-Class octBD8. The format of LABEL object body has following format:

```

    Object-Class = octBD8 (LABEL)
    Object-Type = 1
      0               1               2               3
    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
+-----+-----+-----+-----+-----+-----+-----+-----+
|                               (top label)                               |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

The contents of a LABEL is a single label, encoded in 4 octets.

### 6.6.9. INTERFACE Object

The INTERFACE Object has Object-Class octBD9. Three Object-Types are defined under INTERFACE object:

1. Index: Object-Type is 1.
2. IPv4 Address: Object-Type is 2.
3. IPv6 Address: Object-Type is 3.

The format of INTERFACE object body for interface index has following format:

```

Object-Class = octBD9 (INTERFACE)
Object-Type = 1 (Index)
0              1              2              3
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
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     Interface Index                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

The Interface Index is a single interface index, encoded in 4 octets.

The format of INTERFACE object body for interface IPv4 address has following format:

```

Object-Class = octBD9 (INTERFACE)
Object-Type = 2 (IPv4 Address)
0              1              2              3
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
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     Interface IPv4 Address                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

The Interface IPv4 Address is a single interface IPv4 address, encoded in 4 octets.

The format of INTERFACE object body for interface IPv6 address has following format:

```

    Object-Class = ocTBD9 (INTERFACE)
    Object-Type = 3 (IPv6 Address)
      0               1               2               3
      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
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     Interface IPv6 Address                                     |
|                                     (16 bytes)                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

The Interface IPv6 Address is a single interface IPv6 address, encoded in 16 octets.

## 7. Security Considerations

The mechanism described in this document does not raise any new security issues for the PCEP protocols.

## 8. IANA Considerations

This section specifies requests for IANA allocation.

## 9. Acknowledgement

The authors would like to thank people for their valuable comments on this draft.

## 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>>.
- [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>>.
- [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>>.

[RFC5392] Chen, M., Zhang, R., and X. Duan, "OSPF Extensions in Support of Inter-Autonomous System (AS) MPLS and GMPLS Traffic Engineering", RFC 5392, DOI 10.17487/RFC5392, January 2009, <<https://www.rfc-editor.org/info/rfc5392>>.

[RFC5316] Chen, M., Zhang, R., and X. Duan, "ISIS Extensions in Support of Inter-Autonomous System (AS) MPLS and GMPLS Traffic Engineering", RFC 5316, DOI 10.17487/RFC5316, December 2008, <<https://www.rfc-editor.org/info/rfc5316>>.

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

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

## 10.2. Informative References

[RFC1136] Hares, S. and D. Katz, "Administrative Domains and Routing Domains: A model for routing in the Internet", RFC 1136, DOI 10.17487/RFC1136, December 1989, <<https://www.rfc-editor.org/info/rfc1136>>.

[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>>.

[RFC6006] Zhao, Q., Ed., King, D., Ed., Verhaeghe, F., Takeda, T., Ali, Z., and J. Meuric, "Extensions to the Path Computation Element Communication Protocol (PCEP) for Point-to-Multipoint Traffic Engineering Label Switched Paths", RFC 6006, DOI 10.17487/RFC6006, September 2010, <<https://www.rfc-editor.org/info/rfc6006>>.

[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>>.

#### Appendix A. Details on Embedded Encoding of Messages

A new options field of 3 bits is defined in the flags field of the RP object to tell the receiver of the message that the request/reply is for one of the five request/reply messages for supporting HSCS as follows:

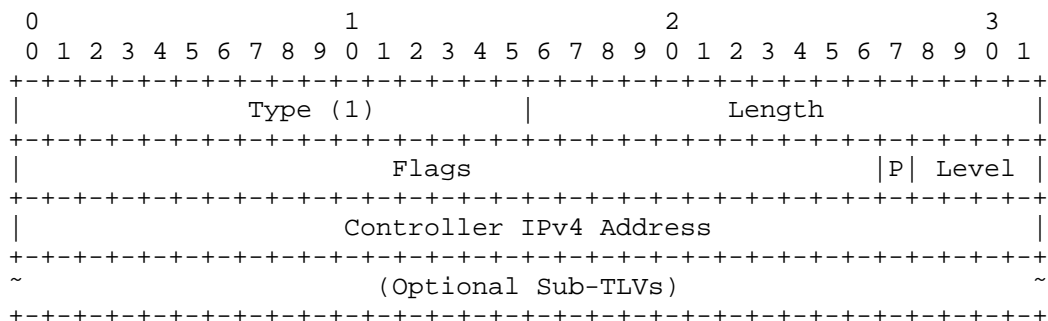
Options	Meaning
1	Path Segment Computation Request/Reply
2	Remove Path Segment Request/Reply
3	Keep Path Segment Request/Reply
4	Create Tunnel Segment Request/Reply
5	Remove Tunnel Segment Request/Reply

A new flag E of 1 bit is defined in the flags field of the RP object. Flag E set to 1 indicating computing a shortest path segment satisfying a given set of constraints from a start node to each of the edge nodes of the domain controlled by a child controller except for the nodes in a given exception list.

##### A.1. Message for Controller Relation Discovery

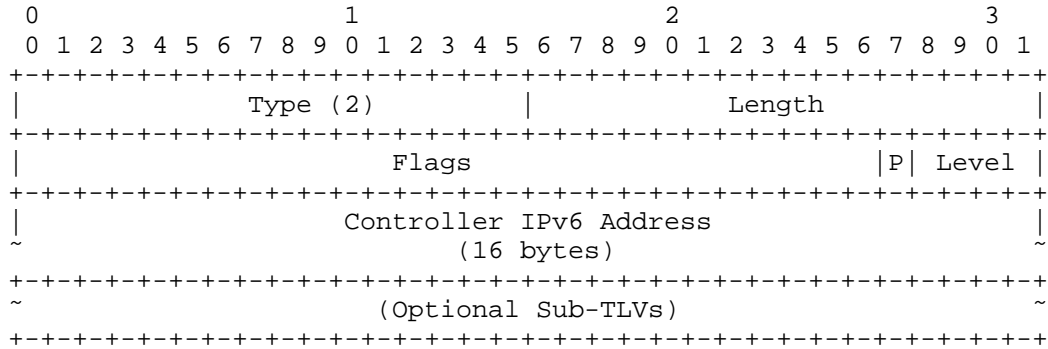
The new TLV defined in the Open Object in section Capability Discovery is extended to contain Sub-TLVs for local controller and remote controller. Thus Open Message with the Open Object containing the new TLV can be used as Message for Controller Relation Discovery. Four optional Sub-TLVs are defined as follows:

###### 1. Local Controller IPv4 Sub-TLV



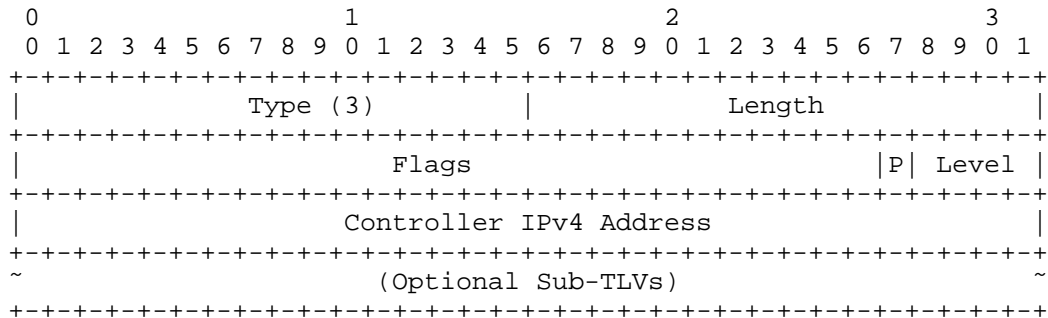
The meanings of each field in the Sub-TLV is the same as described in section LOCAL-CONTROLLER Object for IPv4.

## 2. Local Controller IPv6 Sub-TLV



The meanings of each field in the Sub-TLV is the same as described in section LOCAL-CONTROLLER Object for IPv6.

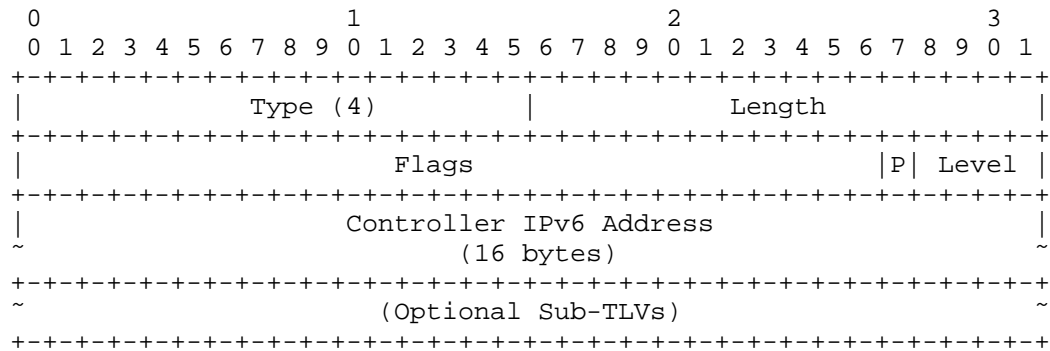
## 3. Remote Controller IPv4 Sub-TLV



The meanings of each field in the Sub-TLV is the same as described in section REMOTE-CONTROLLER Object for IPv4.

## 4. Remote Controller IPv6 Sub-TLV





The meanings of each field in the Sub-TLV is the same as described in section REMOTE-CONTROLLER Object for IPv6.

#### A.2. Message for Connections and Accesses Advertisement

The format of the CAAdv message is as follows:

```

<CAAdv Message> ::= <Common Header>
                    <SRP>
                    <Inter-Domain-Link-List>
                    [<Access-Address-List>]
where:
<Inter-Domain-Link-List> ::= <Inter-Domain-Link>
                             [<Inter-Domain-Link-List>]
<Access-Address-List> ::= <Access-Address>
                          [<Access-Address-List>]

```

#### A.3. Request for Computing Path Segments

The format of the PSReq message is as follows:

```

<PSReq Message> ::= <Common Header>
                    [<svec-list>]
                    <path-segment-request-list>
where:
  <svec-list> ::= <SVEC> [<svec-list>]
  <path-segment-request-list> ::=
    <path-segment-request>
    [<path-segment-request-list>]

  <path-segment-request> ::=
    <RP> <END-POINTS> [<OF>] [<LSPA>] [<BANDWIDTH>]
    <Tunnel-ID> <Path-ID>
    [<metric-list>] [<RRO> [<BANDWIDTH>]] [<IRO>]
    [<LOAD-BALANCING>]
    <exception-list>

```

#### A.4. Reply for Computing Path Segments

The format of the PSRep message is as follows:

```

<PSRep Message> ::= <Common Header>
                    <path-segment-reply-list>
where:
  <path-segment-reply-list> ::=
    <path-segment-reply>
    [<path-segment-reply-list>]

  <path-segment-reply> ::=
    <RP> [<NO-PATH>] [<attribute-list>]
    <Tunnel-ID> <Path-ID>
    <Start-Node>
    [ <NO-PATH> | <segment-end-List> ]
    [<attribute-list>]

```

#### A.5. Request for Removing Path Segments

The format of the RPSReq message is as follows:

```

<RPSReq Message> ::= <Common Header>
                        <remove-path-segment-request-list>
where:
  <remove-path-segment-request-list> ::= =
                        <remove-path-segment-request>
                        [<remove-path-segment-request-list>]

  <remove-path-segment-request> ::=
                        <RP>
                        <Tunnel-ID> [<Path-ID>]
                        [<start-node-list>]
                        [<branch-List>]

  <start-node-list> ::= <Start-Node> [<start-node-list>]

  <branch-list> ::= <Branch> [<branch-list>]
  <Branch> ::= <Start-Node> <branch-end-list>

  <branch-end-list> ::= <Branch-End> [<branch-end-list>]

```

#### A.6. Reply for Removing Path Segments

The format of the RPSRep message is as follows:

```

<RPSRep Message> ::= <Common Header>
                        <remove-path-segment-reply-list>
where:
  <remove-path-segment-reply-list> ::=
                        <remove-path-segment-reply>
                        [<remove-path-segment-reply-list>]

  <remove-path-segment-reply> ::=
                        <RP>
                        <Tunnel-ID> [<Path-ID>]
                        <Status>
                        [<Reasons>]

```

#### A.7. Request for Keeping Path Segments

The format of the KPSReq message is as follows:

```

<KPSReq Message> ::= <Common Header>
                        <keep-path-segment-request-list>
where:
  <keep-path-segment-request-list> ::= =
                        <keep-path-segment-request>
                        [<keep-path-segment-request-list>]

  <keep-path-segment-request> ::=
                        <RP>
                        <Tunnel-ID> <Path-ID>
                        <segment-list>

  <segment-list> ::= <Segment> [<segment-list>]
  <Segment> ::= <Segment-Start> <Segment-End>

```

#### A.8. Reply for Keeping Path Segments

The format of the KPSRep message is as follows:

```

<KPSRep Message> ::= <Common Header>
                        <keep-path-segment-reply-list>
where:
  <keep-path-segment-reply-list> ::=
                        <keep-path-segment-reply>
                        [<keep-path-segment-reply-list>]

  <keep-path-segment-reply> ::=
                        <RP>
                        <Tunnel-ID> <Path-ID>
                        <Status>
                        [<Reasons>]

```

#### A.9. Request for Creating Tunnel Segment

The format of the CTSReq message is as follows:

```

<CTSReq Message> ::= <Common Header>
                        <create-tunnel-segment-request-list>
where:
  <create-tunnel-segment-request-list> ::=
    <create-tunnel-segment-request>
    [<create-tunnel-segment-request-list>]

  <create-tunnel-segment-request> ::=
    <RP>
    <Tunnel-ID> <Path-ID>
    <Path-Segment>
    [<Label> <Interface>]

  <Path-Segment> ::= [<Segment-Start> <Segment-End> | <ERO> ]

```

#### A.10. Reply for Creating Tunnel Segment

The format of the CTSRep message is as follows:

```

<CTSRep Message> ::= <Common Header>
                        <create-tunnel-segment-reply-list>
where:
  <create-tunnel-segment-reply-list> ::=
    <create-tunnel-segment-reply>
    [<create-tunnel-segment-reply-list>]

  <create-tunnel-segment-reply> ::=
    <RP>
    <Tunnel-ID> <Path-ID>
    <Status> [<Label> <Interface>]
    [<Reasons>]

```

#### A.11. Request for Removing Tunnel Segment

The format of the RTSReq message is as follows:

```

<RTSReq Message> ::= <Common Header>
                        <remove-tunnel-segment-request-list>
where:
  <remove-tunnel-segment-request-list> ::=
    <remove-tunnel-segment-request>
    [<remove-tunnel-segment-request-list>]

  <remove-tunnel-segment-request> ::
    <RP>
    <Tunnel-ID> [<Path-ID>]

```

## A.12. Reply for Removing Tunnel Segment

The format of the RTSRep message is as follows:

```
<RTSRep Message> ::= <Common Header>
                        <remove-tunnel-segment-reply-list>
where:
  <reply-tunnel-segment-reply-list> ::=
    <remove-tunnel-segment-reply>
    [<remove-tunnel-segment-reply-list>]

  <remove-tunnel-segment-reply> ::=
    <RP>
    <Tunnel-ID> [<Path-ID>]
    <Status>
    [<Reasons>]
```

## Authors' Addresses

Huaimo Chen  
Huawei Technologies  
Boston, MA,  
USA

EMail: [Huaimo.chen@huawei.com](mailto:Huaimo.chen@huawei.com)

Mehmet Toy  
Verizon  
USA

EMail: [mehmet.toy@verizon.com](mailto:mehmet.toy@verizon.com)

Lei Liu  
Fujitsu  
USA

EMail: [lliu@us.fujitsu.com](mailto:lliu@us.fujitsu.com)

Vic Liu  
China Mobile  
No.32 Xuanwumen West Street, Xicheng District  
Beijing, 100053  
China

EMail: liu.cmri@gmail.com





PCE Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: August 16, 2016

D. Dhody  
Y. Lee  
Huawei Technologies  
D. Ceccarelli  
Ericsson  
J. Shin  
SK Telecom  
D. King  
Lancaster University

February 16, 2016

Hierarchical Stateful Path Computation Element (PCE).  
draft-dhodylee-pce-stateful-hpce-00

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).

The Hierarchical Path Computation Element (H-PCE) architecture, provides an architecture to allow the optimum sequence of inter-connected domains to be selected, and network policy to be applied if applicable, via the use of a hierarchical relationship between PCEs.

Combining the capabilities of Stateful PCE and the Hierarchical PCE would be advantageous. This document describes general considerations and use cases for the deployment of Stateful PCE(s) using the Hierarchical PCE architecture.

Status of This Memo

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

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

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

This Internet-Draft will expire on August 16, 2016.

## Copyright Notice

Copyright (c) 2016 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
1.1. Requirements Language . . . . .	3
2. Terminology . . . . .	3
3. Hierarchical Stateful PCE . . . . .	4
3.1. Passive Operations . . . . .	4
3.2. Active Operations . . . . .	7
3.3. PCE Initiation Operation . . . . .	8
3.3.1. Per Domain Stitched LSP . . . . .	8
4. Other Considerations . . . . .	10
4.1. Applicability to Inter-Layer . . . . .	10
4.2. Applicability to ACTN . . . . .	11
5. Security Considerations . . . . .	12
6. Manageability Considerations . . . . .	12
6.1. Control of Function and Policy . . . . .	12
6.2. Information and Data Models . . . . .	12
6.3. Liveness Detection and Monitoring . . . . .	12
6.4. Verify Correct Operations . . . . .	12
6.5. Requirements On Other Protocols . . . . .	12
6.6. Impact On Network Operations . . . . .	12
7. IANA Considerations . . . . .	12
8. Acknowledgments . . . . .	12
9. References . . . . .	12
9.1. Normative References . . . . .	12
9.2. Informative References . . . . .	13
Appendix A. Contributor Addresses . . . . .	14
Authors' Addresses . . . . .	14

## 1. Introduction

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.

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 (LSPDB).

[I-D.ietf-pce-stateful-pce-app] 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.

[I-D.ietf-pce-stateful-pce] 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. [I-D.ietf-pce-pce-initiated-lsp] describes the setup, maintenance and teardown of PCE-initiated LSPs under the stateful PCE model.

[I-D.ietf-pce-stateful-pce] also describes the active stateful PCE. The active PCE functionality allows a PCE to reroute an existing LSP or make changes to the attributes of an existing LSP, or delegate control of specific LSPs to a new PCE.

The ability to compute shortest constrained TE LSPs in Multiprotocol Label Switching (MPLS) and Generalized MPLS (GMPLS) networks across multiple domains has been identified as a key motivation for PCE development. [RFC6805] describes a Hierarchical PCE (H-PCE) architecture which can be used for computing end-to-end paths for inter-domain MPLS Traffic Engineering (TE) and GMPLS Label Switched Paths (LSPs). Within the Hierarchical PCE (H-PCE) architecture [RFC6805], the Parent PCE (P-PCE) is used to compute a multi-domain path based on the domain connectivity information. A Child PCE (C-PCE) may be responsible for a single domain or multiple domains, it is used to compute the intra-domain path based on its domain topology information.

This document presents general considerations for stateful PCE(s) in hierarchical PCE architecture. In particular, the behavior changes and additions to the existing stateful PCE mechanisms (including PCE-initiated LSP setup and active PCE usage) in the context of networks using the H-PCE architecture.

### 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].

## 2. Terminology

The terminology is as per [RFC4655], [RFC5440], [RFC6805], and [I-D.ietf-pce-stateful-pce].



### 3. Hierarchical Stateful PCE

As described in [RFC6805], in the hierarchical PCE architecture, a P-PCE maintains a domain topology map that contains the child domains (seen as vertices in the topology) and their interconnections (links in the topology). The P-PCE has no information about the content of the child domains. Each child domain has at least one PCE capable of computing paths across the domain. These PCEs are known as C-PCEs and have a direct relationship with the P-PCE. The P-PCE builds the domain topology map either via direct configuration (allowing network policy to also be applied) or from learned information received from each C-PCE.

[I-D.ietf-pce-stateful-pce] specifies new functions to support a stateful PCE. It also specifies that a function can be initiated either from a PCC towards a PCE (C-E) or from a PCE towards a PCC (E-C).

This document extends these functions to support H-PCE Architecture from a C-PCE towards a P-PCE (CE-PE) or from a P-PCE towards a C-PCE (PE-CE). All PCE types herein (i.e., PE or CE) are assumed to be 'stateful PCE'.

A number of interactions are expected in the Hierarchical Stateful PCE architecture, these include:

LSP State Report (CE-PE): a child stateful PCE sends an LSP state report to a Parent Stateful PCE whenever the state of a LSP changes.

LSP State Synchronization (CE-PE): after the session between the Child and Parent stateful PCEs is initialized, the P-PCE must learn the state of C-PCE's TE LSPs.

LSP Control Delegation (CE-PE,PE-CE): a C-PCE grants to the P-PCE the right to update LSP attributes on one or more LSPs; the C-PCE may withdraw the delegation or the P-PCE may give up the delegation at any time.

LSP Update Request (PE-CE): a stateful P-PCE requests modification of attributes on a C-PCE's TE LSP.

PCE LSP Initiation Request (PE-CE): a stateful P-PCE requests C-PCE to initiate a TE LSP.

#### 3.1. Passive Operations

Procedures as described in [RFC6805] are applied, where the ingress C-PCE sends a request to the P-PCE. The P-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 C-PCEs responsible for each of the domains on the

candidate domain paths. Each C-PCE computes a set of candidate path segments across its domain and sends the results to the P-PCE. The P-PCE uses this information to select path segments and concatenate them to derive the optimal end-to-end inter-domain path. The end-to-end path is then sent to the C-PCE that received the initial path request, and this C-PCE passes the path on to the PCC that issued the original request.

As per [I-D.ietf-pce-stateful-pce], PCC sends an LSP State Report carried on a PCRpt message to the C-PCE, indicating the LSP's status. The C-PCE MAY further propagate the State Report to the P-PCE. A local policy at C-PCE MAY dictate which LSPs to be reported to the P-PCE. The PCRpt message is sent from C-PCE to P-PCE.

State synchronization mechanism as described in [I-D.ietf-pce-stateful-pce] and [I-D.ietf-pce-stateful-sync-optimizations] are applicable to PCEP session between C-PCE and P-PCE as well.

Taking the sample hierarchical domain topology example from [RFC6805] as the reference topology for the entirety of this document.

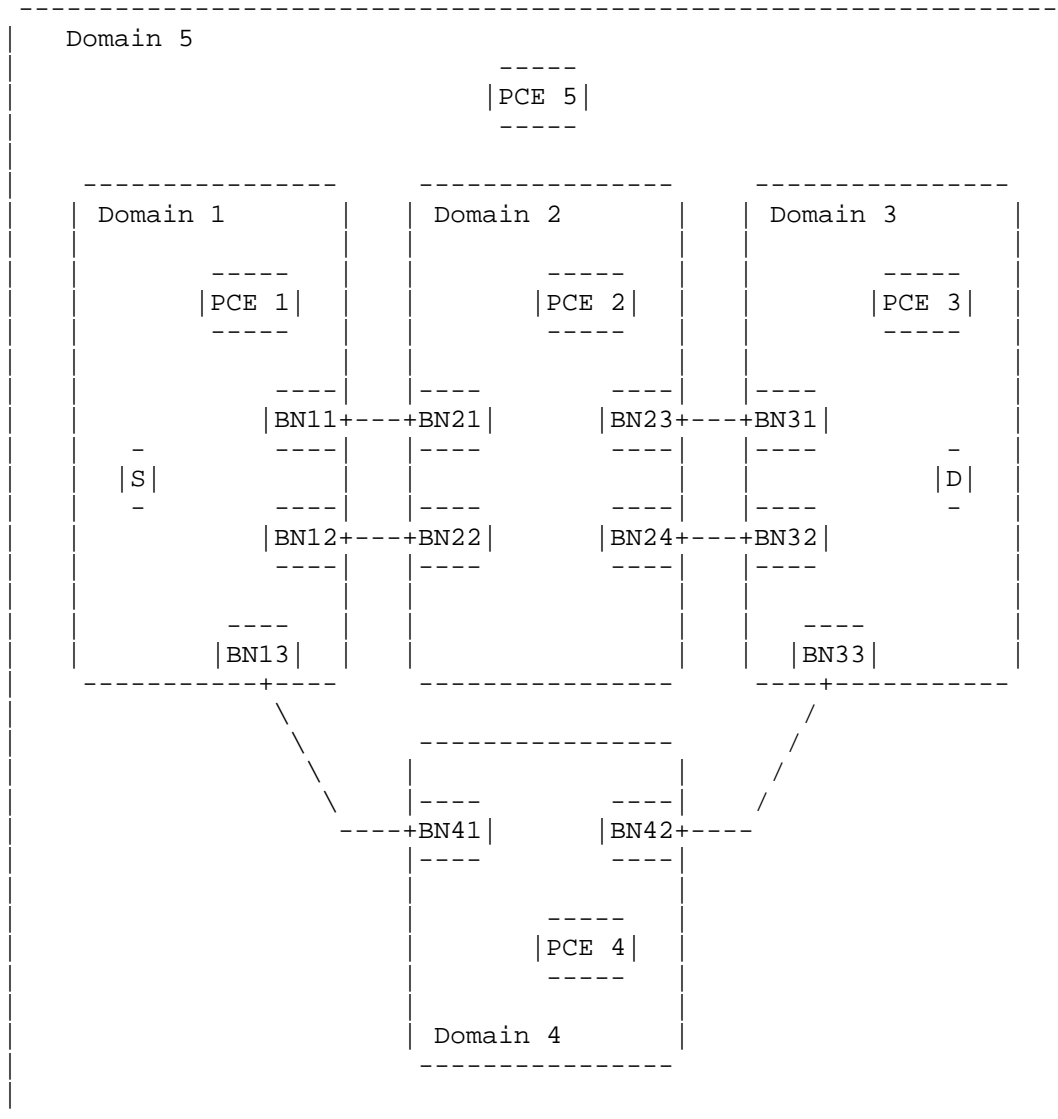


Figure 1: Sample Hierarchical Domain Topology

Steps 1 to 11 are exactly as described in section 4.6.2 (Hierarchical PCE End-to-End Path Computation Procedure) of [RFC6805], the following additional steps are added for stateful PCE:

- (1) The Ingress LSR initiates the setup of the LSP as per the path and reports to the PCE1 the LSP status ("GOING-UP").
- (2) The PCE1 further reports the status of the LSP to the P-PCE (PCE5).

- (3) The Ingress LSR notifies the LSP state to PCE1 when the state is "UP".
- (4) The PCE1 further reports the status of the LSP to the P-PCE (PCE5).

### 3.2. Active Operations

[I-D.ietf-pce-stateful-pce] describes the case of active stateful PCE. The active PCE functionality uses two specific PCEP messages:

- o Update Request (PCUpd)
- o State Report (PCRpt)

The first is sent by the PCE to a Path Computation Client (PCC) for modifying LSP attributes. The PCC sends back a PCRpt to acknowledge the requested operation. PCRpt has the same structure of PCNtf message.

As per [I-D.ietf-pce-stateful-pce], Delegation is an operation to grant a PCE, temporary rights to modify a subset of LSP parameters on one or more PCC's LSPs. The C-PCE may further choose to delegate to P-PCE based on a local policy. The PCRpt message with "D" (delegate) flag is sent from C-PCE to P-PCE.

To update an LSP, a PCE send to the PCC, an LSP Update Request using a PCUpd message. For LSP delegated to the P-PCE via the child PCE, the P-PCE can use the same PCUpd message to request change to the C-PCE (the Ingress domain PCE), the PCE further propagates the update request to the PCC.

The P-PCE uses the same mechanism described in Section 3.1 to compute the end to end path using PCReq and PCRep messages.

The following additional steps are also initially performed, for active operations, again using the reference architecture described in Figure 1 (Sample Hierarchical Domain Topology).

- (1) The Ingress LSR delegates the LSP to the PCE1 via PCRpt message with D flag set.
- (2) The PCE1 further delegates the LSP to the P-PCE (PCE5).

Steps 4 to 10 of section 4.6.2 of [RFC6805] are executed to determine the end to end path.

- (3) The P-PCE (PCE5) sends the update request to the C-PCE (PCE1) via PCUpd message.
- (4) The PCE1 further updates the LSP to the Ingress LSR (PCC).
- (5) The Ingress LSR initiates the setup of the LSP as per the path and reports to the PCE1 the LSP status ("GOING-UP").
- (6) The PCE1 further reports the status of the LSP to the P-PCE (PCE5).
- (7) The Ingress LSR notifies the LSP state to PCE1 when the state is "UP".





- (8) The PCE1 further reports the status of the LSP to the P-PCE (PCE5).

### 3.3. PCE Initiation Operation

[I-D.ietf-pce-pce-initiated-lsp] describes the setup, maintenance and teardown of PCE-initiated LSPs under the stateful PCE model, without the need for local configuration on the PCC, thus allowing for a dynamic network that is centrally controlled and deployed. To instantiate or delete an LSP, the PCE sends the Path Computation LSP Initiate Request (PCInitiate) message to the PCC. In case of inter-domain LSP in Hierarchical PCE architecture, the initiation operations can be carried out at the P-PCE. In which case after P-PCE finishes the E2E path computation, it can send the PCInitiate message to the C-PCE (the Ingress domain PCE), the PCE further propagates the initiate request to the PCC.

The following additional steps are also initially performed, for PCE initiated operations, again using the reference architecture described in Figure 1 (Sample Hierarchical Domain Topology):

- (1) The P-PCE (PCE5) is requested to initiate a LSP.

Steps 4 to 10 of section 4.6.2 of [RFC6805] are executed to determine the end to end path.

- (2) The P-PCE (PCE5) sends the initiate request to the child PCE (PCE1) via PCInitiate message.
- (3) The PCE1 further propagates the initiate message to the Ingress LSR (PCC).
- (4) The Ingress LSR initiates the setup of the LSP as per the path and reports to the PCE1 the LSP status ("GOING-UP").
- (5) The PCE1 further reports the status of the LSP to the P-PCE (PCE5).
- (6) The Ingress LSR notifies the LSP state to PCE1 when the state is "UP".
- (7) The PCE1 further reports the status of the LSP to the P-PCE (PCE5).

#### 3.3.1. Per Domain Stitched LSP

The hierarchical PCE architecture as per [RFC6805] is primarily used for E2E LSP. With PCE-Initiated capability, another mode of operation is possible, where multiple intra-domain LSP are initiated

in each domain which are further stitched to form an E2E LSP. The P-PCE sends PCInitiate message to each C-PCE separately to initiate individual LSP segments along the domain path. These individual per domain LSP are stitched together by some mechanism, which is out of scope of this document.

The following additional steps are also initially performed, for the Per Domain stitched LSP operation, again using the reference architecture described in Figure 1 (Sample Hierarchical Domain Topology):

- (1) The P-PCE (PCE5) is requested to initiate a LSP.

Steps 4 to 10 of section 4.6.2 of [RFC6805] are executed to determine the end to end path, which are broken into per-domain LSPs say -

- o S-BN41
- o BN41-BN33
- o BN33-D

It should be noted that the P-PCE MAY use other mechanisms to determine the suitable per-domain LSPs (apart from [RFC6805]).

For LSP (BN33-D)

- (2) The P-PCE (PCE5) sends the initiate request to the child PCE (PCE3) via PCInitiate message for LSP (BN33-D).
- (3) The PCE3 further propagates the initiate message to BN33.
- (4) BN33 initiates the setup of the LSP as per the path and reports to the PCE3 the LSP status ("GOING-UP").
- (5) The PCE3 further reports the status of the LSP to the P-PCE (PCE5).
- (6) The node BN33 notifies the LSP state to PCE3 when the state is "UP".
- (7) The PCE3 further reports the status of the LSP to the P-PCE (PCE5).

For LSP (BN41-BN33)

- (8) The P-PCE (PCE5) sends the initiate request to the child PCE (PCE4) via PCInitiate message for LSP (BN41-BN33).
- (9) The PCE4 further propagates the initiate message to BN41.

- (10) BN41 initiates the setup of the LSP as per the path and reports to the PCE4 the LSP status ("GOING-UP").
- (11) The PCE4 further reports the status of the LSP to the P-PCE (PCE5).
- (12) The node BN41 notifies the LSP state to PCE4 when the state is "UP".
- (13) The PCE4 further reports the status of the LSP to the P-PCE (PCE5).

For LSP (S-BN41)

- (14) The P-PCE (PCE5) sends the initiate request to the child PCE (PCE1) via PCInitiate message for LSP (S-BN41).
- (15) The PCE1 further propagates the initiate message to node S.
- (16) S initiates the setup of the LSP as per the path and reports to the PCE1 the LSP status ("GOING-UP").
- (17) The PCE1 further reports the status of the LSP to the P-PCE (PCE5).
- (18) The node S notifies the LSP state to PCE1 when the state is "UP".
- (19) The PCE1 further reports the status of the LSP to the P-PCE (PCE5).

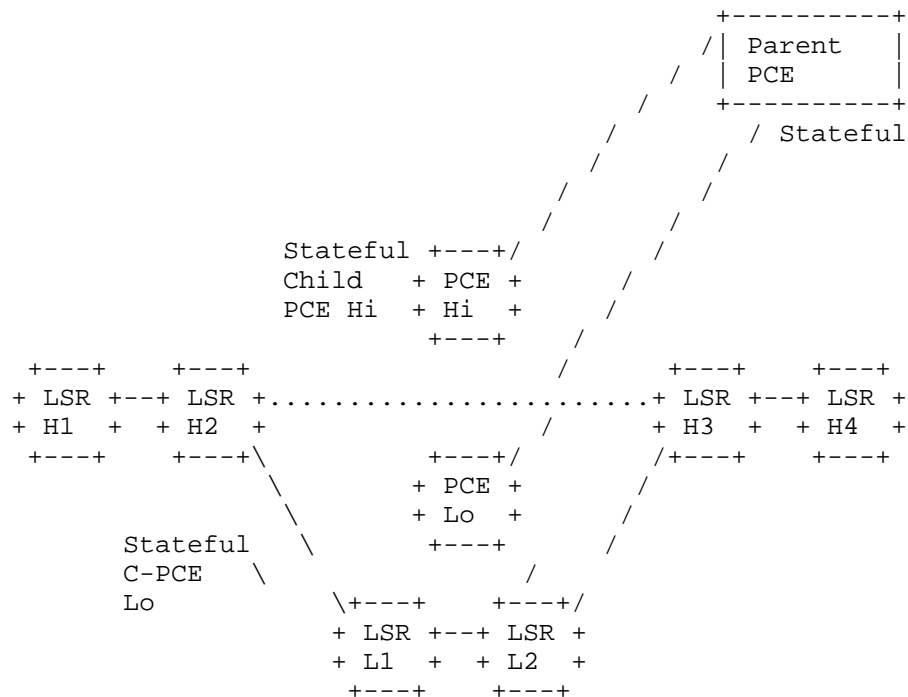
Additionally:

- (20) once P-PCE receives report of each per-domain LSP, it should use some stitching mechanism, which is out of scope of this document.

#### 4. Other Considerations

##### 4.1. Applicability to Inter-Layer

[RFC5623] describes a framework for applying the PCE-based architecture to inter-layer (G)MPLS traffic engineering. The H-PCE Stateful architecture with stateful P-PCE coordinating with the stateful C-PCEs of higher and lower layer is shown in the figure below.



All procedures described in Section 3 are applicable to inter-layer path setup as well.

#### 4.2. Applicability to ACTN

[I-D.ceccarelli-teas-actn-framework] describes framework for Abstraction and Control of TE Networks (ACTN), where each Physical Network Controller (PNC) is equivalent to C-PCE and P-PCE is the Multi-Domain Service Coordinator (MDSC). The Per domain stitched LSP as per the Hierarchical PCE architecture described in Section 3.3.1 and Section 4.1 is well suited for ACTN.

In ACTN framework, Customer Network Controller (CNC) can request the MDSC to check if there is a possibility to meet Virtual Network (VN) requirements (before requesting for VN provision). The H-PCE architecture as described in [RFC6805] can supports via the use of PCReq and PCRep messages between the P-PCE and C-PCEs.

## 5. Security Considerations

## 6. Manageability Considerations

### 6.1. Control of Function and Policy

TBD.

### 6.2. Information and Data Models

TBD.

### 6.3. Liveness Detection and Monitoring

TBD.

### 6.4. Verify Correct Operations

TBD.

### 6.5. Requirements On Other Protocols

TBD.

### 6.6. Impact On Network Operations

TBD.

## 7. IANA Considerations

## 8. Acknowledgments

## 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, <<http://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, <<http://www.rfc-editor.org/info/rfc5440>>.

- [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, <<http://www.rfc-editor.org/info/rfc6805>>.
- [I-D.ietf-pce-stateful-pce]  
Crabbe, E., Minei, I., Medved, J., and R. Varga, "PCEP Extensions for Stateful PCE", draft-ietf-pce-stateful-pce-13 (work in progress), December 2015.
- [I-D.ietf-pce-pce-initiated-lsp]  
Crabbe, E., Minei, I., Sivabalan, S., and R. Varga, "PCEP Extensions for PCE-initiated LSP Setup in a Stateful PCE Model", draft-ietf-pce-pce-initiated-lsp-05 (work in progress), October 2015.

## 9.2. 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, <<http://www.rfc-editor.org/info/rfc4655>>.
- [RFC5623] Oki, E., Takeda, T., Le Roux, JL., and A. Farrel, "Framework for PCE-Based Inter-Layer MPLS and GMPLS Traffic Engineering", RFC 5623, DOI 10.17487/RFC5623, September 2009, <<http://www.rfc-editor.org/info/rfc5623>>.
- [I-D.ietf-pce-stateful-pce-app]  
Zhang, X. and I. Minei, "Applicability of a Stateful Path Computation Element (PCE)", draft-ietf-pce-stateful-pce-app-05 (work in progress), October 2015.
- [I-D.ietf-pce-stateful-sync-optimizations]  
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", draft-ietf-pce-stateful-sync-optimizations-04 (work in progress), November 2015.
- [I-D.ceccarelli-teas-actn-framework]  
Ceccarelli, D. and Y. Lee, "Framework for Abstraction and Control of Transport Networks", draft-ceccarelli-teas-actn-framework-00 (work in progress), June 2015.

## Appendix A. Contributor Addresses

Avantika  
Huawei Technologies  
Leela Palace  
Bangalore, Karnataka 560008  
India

EMail: avantika.sushilkumar@huawei.com

Xian Zhang  
Huawei Technologies  
Bantian, Longgang District  
Shenzhen, Guangdong 518129  
P.R.China

EMail: zhang.xian@huawei.com

## Authors' Addresses

Dhruv Dhody  
Huawei Technologies  
Divyashree Techno Park, Whitefield  
Bangalore, Karnataka 560037  
India

EMail: dhruv.ietf@gmail.com

Young Lee  
Huawei Technologies  
5340 Legacy Drive, Building 3  
Plano, TX 75023  
USA

EMail: leeyoung@huawei.com

Daniele Ceccarelli  
Ericsson  
Torshamnsgatan, 48  
Stockholm  
Sweden

EMail: daniele.ceccarelli@ericsson.com



Jongyoon Shin  
SK Telecom  
6 Hwangsaoul-ro, 258 beon-gil, Bundang-gu, Seongnam-si,  
Gyeonggi-do 463-784  
Republic of Korea

EMail: jongyoon.shin@sk.com

Dan King

EMail: d.king@lancaster.ac.uk

PCE Working Group  
Internet-Draft  
Intended status: Informational  
Expires: September 14, 2017

D. Dhody  
Y. Lee  
Huawei Technologies  
D. Ceccarelli  
Ericsson  
J. Shin  
SK Telecom  
D. King  
Lancaster University  
O. Gonzalez de Dios  
Telefonica I+D  
March 13, 2017

Hierarchical Stateful Path Computation Element (PCE).  
draft-dhodylee-pce-stateful-hpce-03

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).

The Hierarchical Path Computation Element (H-PCE) architecture, provides an architecture to allow the optimum sequence of inter-connected domains to be selected, and network policy to be applied if applicable, via the use of a hierarchical relationship between PCEs.

Combining the capabilities of Stateful PCE and the Hierarchical PCE would be advantageous. This document describes general considerations and use cases for the deployment of Stateful PCE(s) using the Hierarchical PCE architecture.

Status of This Memo

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

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

Internet-Drafts are draft documents valid for a maximum of six months

and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

#### Copyright Notice

Copyright (c) 2016 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
1.1. Requirements Language . . . . .	3
2. Terminology . . . . .	3
3. Hierarchical Stateful PCE . . . . .	4
3.1. Passive Operations . . . . .	4
3.2. Active Operations . . . . .	7
3.3. PCE Initiation Operation . . . . .	8
3.3.1. Per Domain Stitched LSP . . . . .	8
4. Other Considerations . . . . .	10
4.1. Applicability to Inter-Layer . . . . .	10
4.2. Applicability to ACTN . . . . .	11
5. Security Considerations . . . . .	12
6. Manageability Considerations . . . . .	12
6.1. Control of Function and Policy . . . . .	12
6.2. Information and Data Models . . . . .	12
6.3. Liveness Detection and Monitoring . . . . .	12
6.4. Verify Correct Operations . . . . .	12
6.5. Requirements On Other Protocols . . . . .	12
6.6. Impact On Network Operations . . . . .	12
7. IANA Considerations . . . . .	12
8. Acknowledgments . . . . .	12
9. References . . . . .	12
9.1. Normative References . . . . .	12
9.2. Informative References . . . . .	13
Appendix A. Contributor Addresses . . . . .	14
Authors' Addresses . . . . .	14

## 1. Introduction

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.

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 (LSPDB)).

[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.

[I-D.ietf-pce-stateful-pce] 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. [I-D.ietf-pce-pce-initiated-lsp] describes the setup, maintenance and teardown of PCE-initiated LSPs under the stateful PCE model.

[I-D.ietf-pce-stateful-pce] also describes the active stateful PCE. The active PCE functionality allows a PCE to reroute an existing LSP or make changes to the attributes of an existing LSP, or delegate control of specific LSPs to a new PCE.

The ability to compute shortest constrained TE LSPs in Multiprotocol Label Switching (MPLS) and Generalized MPLS (GMPLS) networks across multiple domains has been identified as a key motivation for PCE development. [RFC6805] describes a Hierarchical PCE (H-PCE) architecture which can be used for computing end-to-end paths for inter-domain MPLS Traffic Engineering (TE) and GMPLS Label Switched Paths (LSPs). Within the Hierarchical PCE (H-PCE) architecture [RFC6805], the Parent PCE (P-PCE) is used to compute a multi-domain path based on the domain connectivity information. A Child PCE (C-PCE) may be responsible for a single domain or multiple domains, it is used to compute the intra-domain path based on its domain topology information.

This document presents general considerations for stateful PCE(s) in hierarchical PCE architecture. In particular, the behavior changes

and additions to the existing stateful PCE mechanisms (including PCE-initiated LSP setup and active PCE usage) in the context of networks using the H-PCE architecture.

### 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].

### 2. Terminology

The terminology is as per [RFC4655], [RFC5440], [RFC6805], and [I-D.ietf-pce-stateful-pce].

### 3. Hierarchical Stateful PCE

As described in [RFC6805], in the hierarchical PCE architecture, a P-PCE maintains a domain topology map that contains the child domains (seen as vertices in the topology) and their interconnections (links in the topology). The P-PCE has no information about the content of the child domains. Each child domain has at least one PCE capable of computing paths across the domain. These PCEs are known as C-PCEs and have a direct relationship with the P-PCE. The P-PCE builds the domain topology map either via direct configuration (allowing network policy to also be applied) or from learned information received from each C-PCE.

[I-D.ietf-pce-stateful-pce] specifies new functions to support a stateful PCE. It also specifies that a function can be initiated either from a PCC towards a PCE (C-E) or from a PCE towards a PCC (E-C).

This document extends these functions to support H-PCE Architecture from a C-PCE towards a P-PCE (CE-PE) or from a P-PCE towards a C-PCE (PE-CE). All PCE types herein (i.e., PE or CE) are assumed to be 'stateful PCE'.

A number of interactions are expected in the Hierarchical Stateful PCE architecture, these include:

LSP State Report (CE-PE): a child stateful PCE sends an LSP state report to a Parent Stateful PCE whenever the state of a LSP changes.

LSP State Synchronization (CE-PE): after the session between the Child and Parent stateful PCEs is initialized, the P-PCE must learn the state of C-PCE's TE LSPs.

LSP Control Delegation (CE-PE,PE-CE): a C-PCE grants to the P-PCE the right to update LSP attributes on one or more LSPs; the C-PCE may withdraw the delegation or the P-PCE may give up the delegation at any time.

LSP Update Request (PE-CE): a stateful P-PCE requests modification of attributes on a C-PCE's TE LSP.

PCE LSP Initiation Request (PE-CE): a stateful P-PCE requests C-PCE to initiate a TE LSP.

Note that this hierarchy is recursive and thus a LSR could delegate the control to a PCE, which may delegate to its parent, which may further delegate it to its parent (if it exist or needed). Similarly update operations could also be applied recursively.

[I-D.ietf-pce-hierarchy-extensions] defines the H-PCE capability TLV that should be used in the OPEN message to advertise the H-PCE capability. [I-D.ietf-pce-stateful-pce] defines the stateful PCE capability TLV. The presence of both TLVs represent the support for stateful H-PCE operations as described in this document.

[I-D.litkowski-pce-state-sync] describes the procedures to allow a stateful communication between PCEs for various use-cases. The procedures and extensions as described in Section 3 of [I-D.litkowski-pce-state-sync] are also applicable to Child and Parent PCE communication.

### 3.1. Passive Operations

Procedures as described in [RFC6805] are applied, where the ingress C-PCE sends a request to the P-PCE. The P-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 C-PCEs responsible for each of the domains on the candidate domain paths. Each C-PCE computes a set of candidate path segments across its domain and sends the results to the P-PCE. The P-PCE uses this information to select path segments and concatenate them to derive the optimal end-to-end inter-domain path. The end-to-end path is then sent to the C-PCE that received the initial path request, and this C-PCE passes the path on to the PCC that issued the original request.

As per [I-D.ietf-pce-stateful-pce], PCC sends an LSP State Report carried on a PCRpt message to the C-PCE, indicating the LSP's status. The C-PCE MAY further propagate the State Report to the P-PCE. A local policy at C-PCE MAY dictate which LSPs to be reported to the P-PCE. The PCRpt message is sent from C-PCE to P-PCE.

State synchronization mechanism as described in [I-D.ietf-pce-stateful-pce] and [I-D.ietf-pce-stateful-sync-optimizations] are applicable to PCEP session between C-PCE and P-PCE as well.

Taking the sample hierarchical domain topology example from [RFC6805] as the reference topology for the entirety of this document.

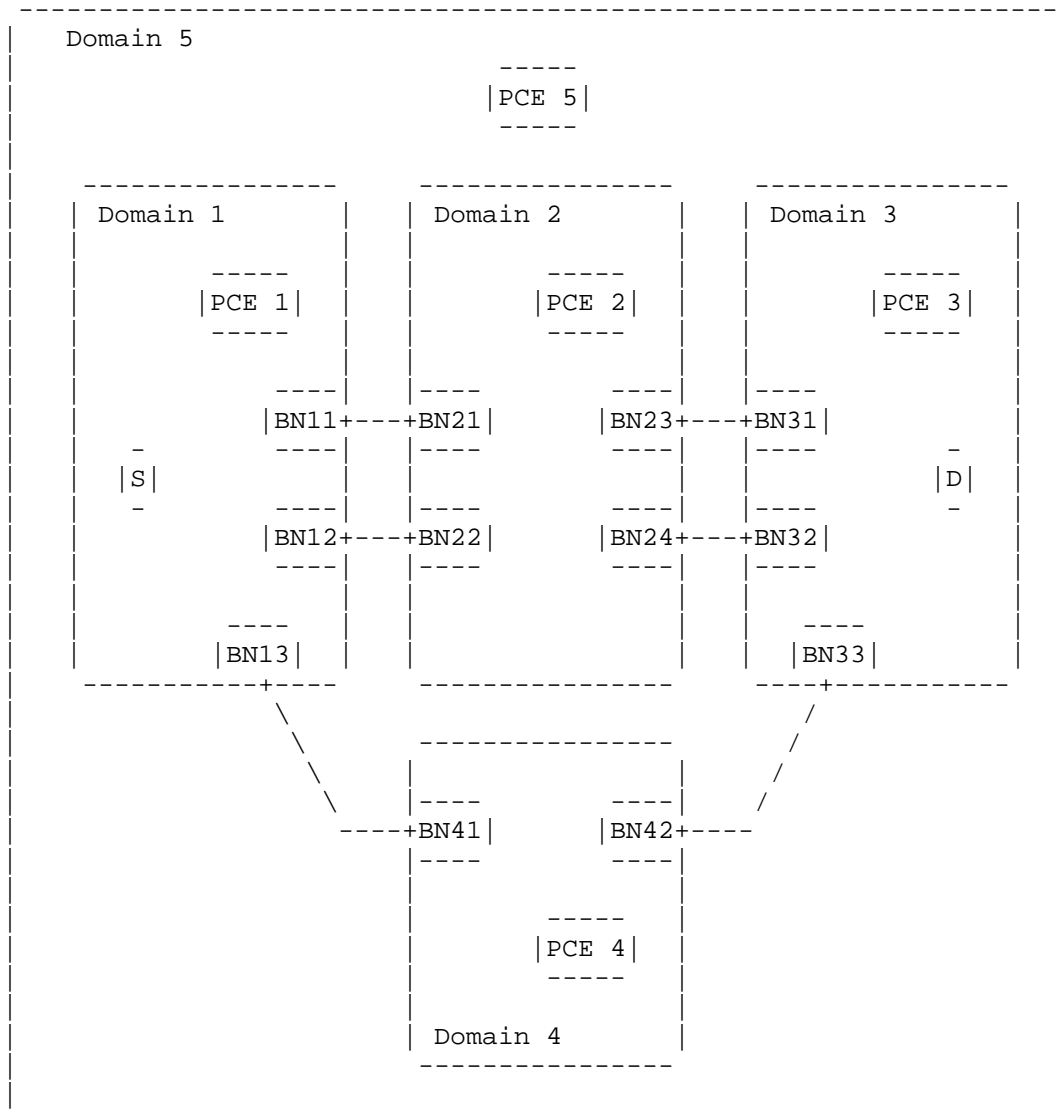


Figure 1: Sample Hierarchical Domain Topology

Steps 1 to 11 are exactly as described in section 4.6.2 (Hierarchical PCE End-to-End Path Computation Procedure) of [RFC6805], the following additional steps are added for stateful PCE:

- (1) The Ingress LSR initiates the setup of the LSP as per the path and reports to the PCE1 the LSP status ("GOING-UP").



- (2) The PCE1 further reports the status of the LSP to the P-PCE (PCE5).
- (3) The Ingress LSR notifies the LSP state to PCE1 when the state is "UP".
- (4) The PCE1 further reports the status of the LSP to the P-PCE (PCE5).

### 3.2. Active Operations

[I-D.ietf-pce-stateful-pce] describes the case of active stateful PCE. The active PCE functionality uses two specific PCEP messages:

- o Update Request (PCUpd)
- o State Report (PCRpt)

The first is sent by the PCE to a Path Computation Client (PCC) for modifying LSP attributes. The PCC sends back a PCRpt to acknowledge the requested operation or report any change in LSP's state.

As per [RFC8051], Delegation is an operation to grant a PCE, temporary rights to modify a subset of LSP parameters on one or more PCC's LSPs. The C-PCE may further choose to delegate to P-PCE based on a local policy. The PCRpt message with "D" (delegate) flag is sent from C-PCE to P-PCE.

To update an LSP, a PCE send to the PCC, an LSP Update Request using a PCUpd message. For LSP delegated to the P-PCE via the child PCE, the P-PCE can use the same PCUpd message to request change to the C-PCE (the Ingress domain PCE), the PCE further propagates the update request to the PCC.

The P-PCE uses the same mechanism described in Section 3.1 to compute the end to end path using PCReq and PCRep messages.

The following additional steps are also initially performed, for active operations, again using the reference architecture described in Figure 1 (Sample Hierarchical Domain Topology).

- (1) The Ingress LSR delegates the LSP to the PCE1 via PCRpt message with D flag set.
- (2) The PCE1 further delegates the LSP to the P-PCE (PCE5).

Steps 4 to 10 of section 4.6.2 of [RFC6805] are executed to determine the end to end path.

- (3) The P-PCE (PCE5) sends the update request to the C-PCE (PCE1) via PCUpd message.
- (4) The PCE1 further updates the LSP to the Ingress LSR (PCC).
- (5) The Ingress LSR initiates the setup of the LSP as per the path and reports to the PCE1 the LSP status ("GOING-UP").
- (6) The PCE1 further reports the status of the LSP to the P-PCE (PCE5).
- (7) The Ingress LSR notifies the LSP state to PCE1 when the state is "UP".
- (8) The PCE1 further reports the status of the LSP to the P-PCE (PCE5).

### 3.3. PCE Initiation Operation

[I-D.ietf-pce-pce-initiated-lsp] describes the setup, maintenance and teardown of PCE-initiated LSPs under the stateful PCE model, without the need for local configuration on the PCC, thus allowing for a dynamic network that is centrally controlled and deployed. To instantiate or delete an LSP, the PCE sends the Path Computation LSP Initiate Request (PCInitiate) message to the PCC. In case of inter-domain LSP in Hierarchical PCE architecture, the initiation operations can be carried out at the P-PCE. In which case after P-PCE finishes the E2E path computation, it can send the PCInitiate message to the C-PCE (the Ingress domain PCE), the PCE further propagates the initiate request to the PCC.

The following additional steps are also initially performed, for PCE initiated operations, again using the reference architecture described in Figure 1 (Sample Hierarchical Domain Topology):

- (1) The P-PCE (PCE5) is requested to initiate a LSP.

Steps 4 to 10 of section 4.6.2 of [RFC6805] are executed to determine the end to end path.

- (2) The P-PCE (PCE5) sends the initiate request to the child PCE (PCE1) via PCInitiate message.
- (3) The PCE1 further propagates the initiate message to the Ingress LSR (PCC).
- (4) The Ingress LSR initiates the setup of the LSP as per the path

and reports to the PCE1 the LSP status ("GOING-UP").

- (5) The PCE1 further reports the status of the LSP to the P-PCE (PCE5).
- (6) The Ingress LSR notifies the LSP state to PCE1 when the state is "UP".
- (7) The PCE1 further reports the status of the LSP to the P-PCE (PCE5).

### 3.3.1. Per Domain Stitched LSP

The hierarchical PCE architecture as per [RFC6805] is primarily used for E2E LSP. With PCE-Initiated capability, another mode of operation is possible, where multiple intra-domain LSPs are initiated in each domain which are further stitched to form an E2E LSP. The P-PCE sends PCInitiate message to each C-PCE separately to initiate individual LSP segments along the domain path. These individual per domain LSP are stitched together by some mechanism, which is out of scope of this document. The P-PCE may also send the PCInitiate message to the ingress C-PCE to initiate the E2E LSP separately.

The following additional steps are also initially performed, for the Per Domain stitched LSP operation, again using the reference architecture described in Figure 1 (Sample Hierarchical Domain Topology):

- (1) The P-PCE (PCE5) is requested to initiate a LSP.

Steps 4 to 10 of section 4.6.2 of [RFC6805] are executed to determine the end to end path, which are broken into per-domain LSPs say -

- o S-BN41
- o BN41-BN33
- o BN33-D

It should be noted that the P-PCE MAY use other mechanisms to determine the suitable per-domain LSPs (apart from [RFC6805]).

For LSP (BN33-D)

- (2) The P-PCE (PCE5) sends the initiate request to the child PCE (PCE3) via PCInitiate message for LSP (BN33-D).

- (3) The PCE3 further propagates the initiate message to BN33.
- (4) BN33 initiates the setup of the LSP as per the path and reports to the PCE3 the LSP status ("GOING-UP").
- (5) The PCE3 further reports the status of the LSP to the P-PCE (PCE5).
- (6) The node BN33 notifies the LSP state to PCE3 when the state is "UP".
- (7) The PCE3 further reports the status of the LSP to the P-PCE (PCE5).

For LSP (BN41-BN33)

- (8) The P-PCE (PCE5) sends the initiate request to the child PCE (PCE4) via PCInitiate message for LSP (BN41-BN33).
- (9) The PCE4 further propagates the initiate message to BN41.
- (10) BN41 initiates the setup of the LSP as per the path and reports to the PCE4 the LSP status ("GOING-UP").
- (11) The PCE4 further reports the status of the LSP to the P-PCE (PCE5).
- (12) The node BN41 notifies the LSP state to PCE4 when the state is "UP".
- (13) The PCE4 further reports the status of the LSP to the P-PCE (PCE5).

For LSP (S-BN41)

- (14) The P-PCE (PCE5) sends the initiate request to the child PCE (PCE1) via PCInitiate message for LSP (S-BN41).
- (15) The PCE1 further propagates the initiate message to node S.
- (16) S initiates the setup of the LSP as per the path and reports to the PCE1 the LSP status ("GOING-UP").
- (17) The PCE1 further reports the status of the LSP to the P-PCE (PCE5).
- (18) The node S notifies the LSP state to PCE1 when the state is "UP".

- (19) The PCE1 further reports the status of the LSP to the P-PCE (PCE5).

Additionally:

- (20) Once P-PCE receives report of each per-domain LSP, it should use some stitching mechanism, which is out of scope of this document. In this step, P-PCE (PCE5) could also initiate an E2E LSP (S-D) by sending the PCInitiate message to Ingress C-PCE (PCE1).

#### 4. Other Considerations

##### 4.1. Applicability to Inter-Layer

[RFC5623] describes a framework for applying the PCE-based architecture to inter-layer (G)MPLS traffic engineering. The H-PCE Stateful architecture with stateful P-PCE coordinating with the stateful C-PCEs of higher and lower layer is shown in the figure below.

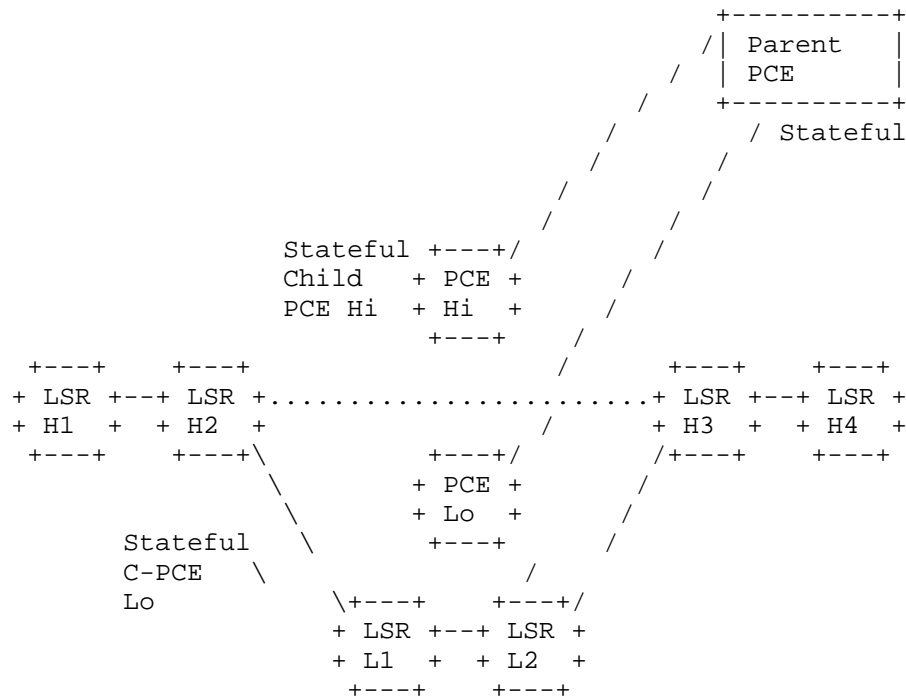


Figure 2: Sample Inter-Layer Topology

All procedures described in Section 3 are applicable to inter-layer path setup as well.

#### 4.2. Applicability to ACTN

[I-D.ietf-teas-actn-framework] describes framework for Abstraction and Control of TE Networks (ACTN), where each Physical Network Controller (PNC) is equivalent to C-PCE and P-PCE is the Multi-Domain Service Coordinator (MDSC). The Per domain stitched LSP as per the Hierarchical PCE architecture described in Section 3.3.1 and Section 4.1 is well suited for ACTN.

[I-D.dhody-pce-applicability-actn] examines the applicability of PCE to the ACTN framework. To support the function of multi domain coordination via hierarchy, the stateful hierarchy of PCEs plays a crucial role.

In ACTN framework, Customer Network Controller (CNC) can request the MDSC to check if there is a possibility to meet Virtual Network (VN) requirements (before requesting for VN provision). The H-PCE architecture as described in [RFC6805] can supports via the use of PCReq and PCRep messages between the P-PCE and C-PCEs.

#### 5. Scalability Considerations

It should be noted that if all the C-PCEs would report all the LSPs in their domain, it could lead to scalability issues for the P-PCE. Thus it is recommended to only report the LSPs which are involved in H-PCE, i.e. the LSPs which are either delegated to the P-PCE or initiated by the P-PCE. Scalability considerations for PCEP as per [I-D.ietf-pce-stateful-pce] continue to apply for the PCEP session between child and parent PCE.

#### 6. Security Considerations

The security considerations listed in [I-D.ietf-pce-stateful-pce],[RFC6805] and [RFC5440] apply to this document as well. As per [RFC6805], it is expected that the parent PCE will require all child PCEs to use full security when communicating with the parent.

Any multi-domain operation necessarily involves the exchange of information across domain boundaries. This is bound to represent a significant security and confidentiality risk especially when the child domains are controlled by different commercial concerns. PCEP allows individual PCEs to maintain confidentiality of their domain path information using path-keys [RFC5520], and the hierarchical PCE architecture is specifically designed to enable as much isolation of

domain topology and capabilities information as is possible. The LSP state in the PCRpt message SHOULD continue to use this.

The security consideration for PCE-Initiated LSP as per [I-D.ietf-pce-pce-initiated-lsp] is also applicable from P-PCE to C-PCE.

Thus securing the PCEP session (between the P-PCE and the C-PCE) using mechanism like TCP Authentication Option (TCP-AO) [RFC5925] or Transport Layer Security (TLS) [I-D.ietf-pce-pceps] is RECOMMENDED.

## 7. Manageability Considerations

All manageability requirements and considerations listed in [RFC5440], [RFC6805], [I-D.ietf-pce-stateful-pce], and [I-D.ietf-pce-pce-initiated-lsp] apply to Stateful H-PCE defined in this document. In addition, requirements and considerations listed in this section apply.

### 7.1. Control of Function and Policy

Support of the hierarchical procedure will be controlled by the management organization responsible for each child PCE. The parent PCE must only accept path computation requests from authorized child PCEs. If a parent PCE receives report from an unauthorized child PCE, the report should be dropped. All mechanism as described in [I-D.ietf-pce-stateful-pce] and [I-D.ietf-pce-pce-initiated-lsp] continue to apply.

### 7.2. Information and Data Models

An implementation SHOULD allow the operator to view the stateful and H-PCE capabilities advertised by each peer. The PCEP YANG module [I-D.ietf-pce-pcep-yang] can be extended to include details stateful H-PCE.

### 7.3. 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].

### 7.4. 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 [I-D.ietf-pce-stateful-pce].

## 7.5. Requirements On Other Protocols

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

## 7.6. Impact On Network Operations

Mechanisms defined in [RFC5440] and [I-D.ietf-pce-stateful-pce] also apply to PCEP extensions defined in this document.

The stateful H-PCE technique brings the applicability of stateful PCE as described in [RFC8051], for the LSP traversing multiple domains.

## 8. IANA Considerations

There are no IANA considerations.

## 9. Acknowledgments

Thanks to Manuela Scarella, Haomian Zheng, Sergio Marmo, Stefano Parodi, Giacomo Agostini, Jeff Tantsura and Rajan Rao for 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, <<http://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, <<http://www.rfc-editor.org/info/rfc5440>>.
- [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, <<http://www.rfc-editor.org/info/rfc6805>>.
- [I-D.ietf-pce-stateful-pce] Crabbe, E., Minei, I., Medved, J., and R. Varga, "PCEP Extensions for Stateful PCE", draft-ietf-pce-stateful-pce-18 (work in progress), December 2016.



[I-D.ietf-pce-pce-initiated-lsp]  
Crabbe, E., Minei, I., Sivabalan, S., and R. Varga, "PCEP Extensions for PCE-initiated LSP Setup in a Stateful PCE Model", draft-ietf-pce-pce-initiated-lsp-09 (work in progress), March 2017.

## 10.2. 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, <<http://www.rfc-editor.org/info/rfc4655>>.
- [RFC5520] Bradford, R., Ed., Vasseur, JP., and A. Farrel, "Preserving Topology Confidentiality in Inter-Domain Path Computation Using a Path-Key-Based Mechanism", RFC 5520, DOI 10.17487/RFC5520, April 2009, <<http://www.rfc-editor.org/info/rfc5520>>.
- [RFC5623] Oki, E., Takeda, T., Le Roux, JL., and A. Farrel, "Framework for PCE-Based Inter-Layer MPLS and GMPLS Traffic Engineering", RFC 5623, DOI 10.17487/RFC5623, September 2009, <<http://www.rfc-editor.org/info/rfc5623>>.
- [RFC5925] Touch, J., Mankin, A., and R. Bonica, "The TCP Authentication Option", RFC 5925, DOI 10.17487/RFC5925, June 2010, <<http://www.rfc-editor.org/info/rfc5925>>.
- [RFC8051] Zhang, X., Ed. and I. Minei, Ed., "Applicability of a Stateful Path Computation Element (PCE)", RFC 8051, DOI 10.17487/RFC8051, January 2017, <<http://www.rfc-editor.org/info/rfc8051>>.
- [I-D.ietf-pce-stateful-sync-optimizations]  
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", draft-ietf-pce-stateful-sync-optimizations-09 (work in progress), February 2017.
- [I-D.ietf-teas-actn-framework]  
Ceccarelli D. and Y. Lee, "Framework for Abstraction and Control of Transport Networks", draft-ietf-teas-actn-framework-04 (work in progress), February 2017.
- [I-D.dhody-pce-applicability-actn]  
Dhody, D., Lee, Y., and D. Ceccarelli, "Applicability of Path Computation Element (PCE) for Abstraction and

Control of TE Networks (ACTN)", draft-dhody-pce-applicability-actn-01 (work in progress), October 2016.

[I-D.litkowski-pce-state-sync]

Litkowski, S., Sivabalan, S., and D. Dhody, "Inter Stateful Path Computation Element communication procedures", draft-litkowski-pce-state-sync-01 (work in progress), February 2017.

[I-D.ietf-pce-hierarchy-extensions]

Zhang, F., Zhao, Q., Dios, O., Casellas, R., and D. King, "Extensions to Path Computation Element Communication Protocol (PCEP) for Hierarchical Path Computation Elements (PCE)", draft-ietf-pce-hierarchy-extensions-03 (work in progress), July 2016.

[I-D.ietf-pce-pceps]

Lopez, D., Dios, O., Wu, W., and D. Dhody, "Secure Transport for PCEP", draft-ietf-pce-pceps-11 (work in progress), January 2017.

## Appendix A. Contributor Addresses

Avantika  
Huawei Technologies  
Divyashree Techno Park, Whitefield  
Bangalore, Karnataka 560066  
India

EMail: avantika.sushilkumar@huawei.com

Xian Zhang  
Huawei Technologies  
Bantian, Longgang District  
Shenzhen, Guangdong 518129  
P.R.China

EMail: zhang.xian@huawei.com

Udayasree Palle  
Huawei Technologies  
Divyashree Techno Park, Whitefield  
Bangalore, Karnataka 560066  
India

EMail: udayasree.palle@huawei.com

## Authors' Addresses

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

EMail: dhruv.ietf@gmail.com

Young Lee  
Huawei Technologies  
5340 Legacy Drive, Building 3  
Plano, TX 75023  
USA

EMail: leeyoung@huawei.com

Daniele Ceccarelli  
Ericsson

Torshamnsgatan, 48  
Stockholm  
Sweden

EMail: daniele.ceccarelli@ericsson.com

Jongyoon Shin  
SK Telecom  
6 Hwangsaoul-ro, 258 beon-gil, Bundang-gu, Seongnam-si,  
Gyeonggi-do 463-784  
Republic of Korea

EMail: jongyoon.shin@sk.com

Dan King  
Lancaster University  
UK

EMail: d.king@lancaster.ac.uk

Oscar Gonzalez de Dios  
Telefonica I+D  
Don Ramon de la Cruz 82-84  
Madrid, 28045  
Spain

Phone: +34913128832  
Email: ogondio@tid.es

PCE Working Group

Internet-Draft

Intended Status: Standards track

Expires: August 2016

Y. Lee  
D. Dhody  
Huawei Technologies  
D. Ceccarelli  
Ericsson

February 25, 2016

PCEP Extensions for Establishing Relationships Between Sets of LSPs  
and Virtual Networks

draft-leedhody-pce-vn-association-00.txt

Abstract

This document describes how to extend PCE association mechanism introduced by [PCE-Association] to further associate sets of LSPs with a higher-level structure such as a virtual network requested by clients or applications. This extended association mechanism can be used to facilitate virtual network control using PCE architecture.

Status of this Memo

This Internet-Draft is submitted to IETF 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 July 25, 2016.

#### Copyright Notice

Copyright (c) 2016 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.....	2
1.1. Requirements Language.....	3
2. Terminology.....	4
3. Operation Overview.....	4
4. Extensions to PCEP.....	4
5. Applicability to H-PCE architecture.....	6
6. Security Considerations.....	7
7. IANA Considerations.....	7
7.1. Association Object Type Indicator.....	7
7.2. PCEP TLV Type Indicator.....	8
7.3. PCEP Error.....	8
8. References.....	8
8.1. Normative References.....	8
8.2. Informative References.....	9
Author's Addresses.....	9

#### 1. Introduction

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-stateful-pce-app] 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. [I-D.ietf-pce-stateful-pce] 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.

[I-D.ietf-pce-pce-initiated-lsp] describes the setup, maintenance and teardown of PCE-initiated LSPs under the stateful PCE model. Within the hierarchical PCE architecture, a PCE is used to initiate or delete LSPs to a PCC.

[I-D.ietf-pce-association-group] introduces a generic mechanism to create a grouping of LSPs. This grouping can then be used to define association between sets of LSPs or between a set of LSPs and a set of attributes.

[ACTN-REQ] describes various Virtual Network (VN) operations initiated by a customer/application. In this context, there is a need for associating a set of LSPs with a VN "construct" to facilitate VN operations in PCE architecture. This association allows the PCEs to identify which LSPs belong to a certain VN.

This document specifies a PCEP extension to associate a set of LSPs based on Virtual Network or customer.

### 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].

## 2. Terminology

The terminology is as per [RFC4655], [RFC5440], [RFC6805], and [I-D.ietf-pce-stateful-pce].

## 3. Operation Overview

As per [I-D.ietf-pce-association-group], LSPs are associated with other LSPs with which they interact by adding them to a common association group. In this draft, this grouping is used to define associations between a set of LSPs and a virtual network.

One new optional Association Object-type is defined based on the generic Association object -

- o VN Association Group (VNAG)

Thus this document defines one new association type called "VN Association Type" of value TBD1. The scope and handling of VNAG identifier is similar to the generic association identifier defined in [I-D.ietf-pce-association-group].

## 4. Extensions to PCEP

[I-D.ietf-pce-association-group] introduces the ASSOCIATION object, the format of VNAG is as follows:



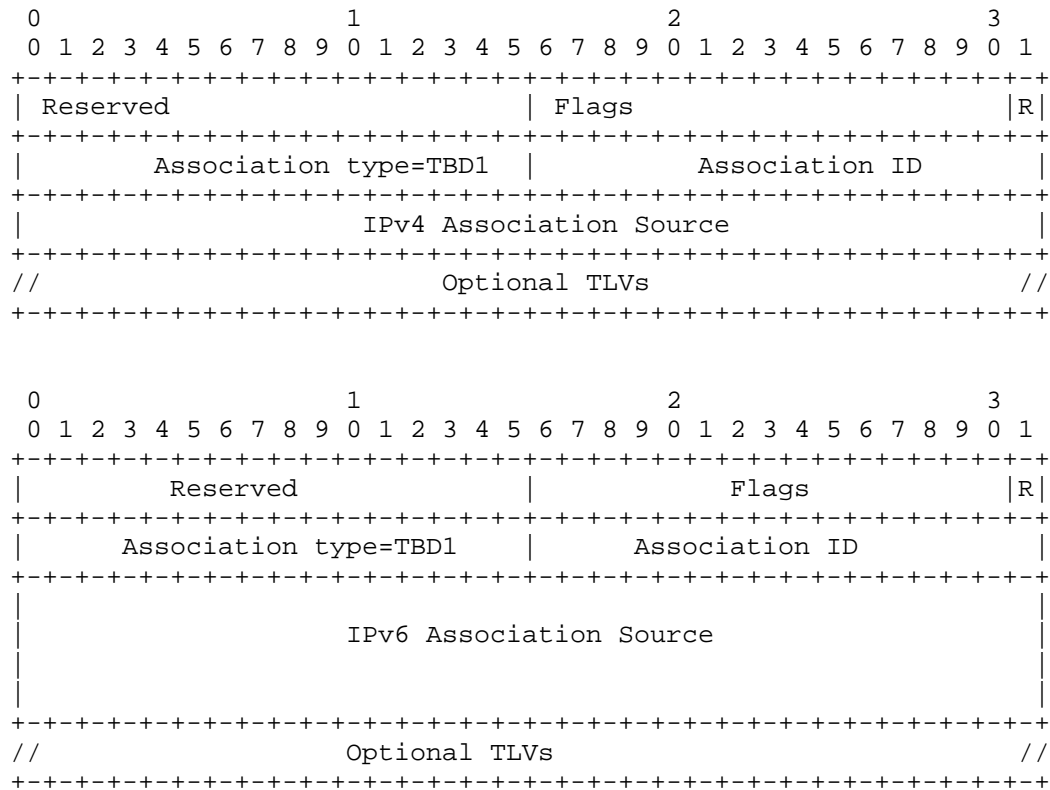


Figure 1: The VNAG Object formats

Please refer to [I-D.ietf-pce-association-group] for the definition of each field in Figure 1. This document defines one mandatory TLV.

o VIRTUAL-NETWORK-TLV: Used to communicate the VN Identifier.

The format of VIRTUAL-NETWORK-TLV is as follows.

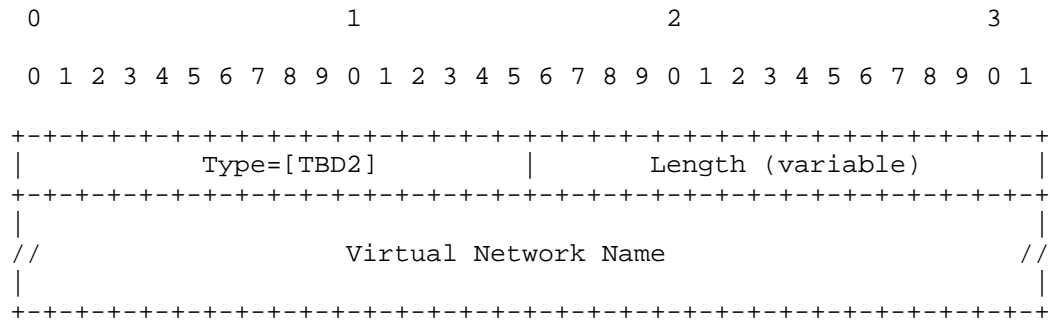


Figure 2: The VIRTUAL-NETWORK-TLV formats

Type: TBD2 (to be allocated by IANA)

Length: Variable Length

Virtual Network Name(variable): symbolic name for the VN.

The VIRTUAL-NETWORK-TLV MUST be included in VNAG object. If a PCEP speaker receives the VNAG object without the VIRTUAL-NETWORK-TLV, it MUST send a PCErr message with Error-Type= 6 (mandatory object missing) and Error-Value=TBD3 (VIRTUAL-NETWORK-TLV missing) and close the session.

## 5. Applicability to H-PCE architecture

The ability to compute shortest constrained TE LSPs in Multiprotocol Label Switching (MPLS) and Generalized MPLS (GMPLS) networks across multiple domains has been identified as a key motivation for PCE development. [RFC6805] describes a Hierarchical PCE (H-PCE) architecture which can be used for computing end-to-end paths for inter-domain MPLS Traffic Engineering (TE) and GMPLS Label Switched Paths (LSPs). Within the hierarchical PCE architecture, the parent PCE is used to compute a multi-domain path based on the domain connectivity information. A child PCE may be responsible for a single domain or multiple domains, it is used to compute the intra-domain path based on its domain topology information.

[I-D.ietf-dhodylee-stateful-HPCE] introduces general considerations for stateful PCE(s) in hierarchical PCE architecture. In

particular, the behavior changes and additions to the existing stateful PCE mechanisms in the context of a H-PCE architecture.

In Stateful H-PCE architecture, the Parent PCE receives a virtual network creation request by its client over its Northbound API. This VN is uniquely identified by an Association ID in VNAG as well as the VIRTUAL-NETWORK name. This VN may comprise multiple LSPs in the network in a single domain or across multiple domains.

As the Parent PCE computes the optimum E2E paths for each tunnel in VN, it MUST associate each LSP with the VN to which it belongs. Parent PCE sends a PCInitiate Message with this association information in the VNAG Object (See Section 4 for details). This in effect binds an LSP that is to be instantiated at the child PCE with the VN.

Whenever changes occur with the instantiated LSP in a domain network, the domain child PCE reports the changes using a PCRpt Message in which the VNAG Object indicates the relationship between the LSP and the VN.

Whenever an update occurs with VNs in the Parent PCE (via the client's request), the parent PCE sends an PCUpd Message to inform each affected child PCE of this change.

## 6. Security Considerations

TDB

## 7. IANA Considerations

### 7.1. Association Object Type Indicator

This document defines the following new association type originally defined in [I-D.ietf-pce-association-group].

Value	Name	Reference
TBD1	VN Association Type	[This I.D.]

## 7.2. PCEP TLV Type Indicator

This document defines the following new PCEP TLV; IANA is requested to make the following allocations from this registry at <http://www.iana.org/assignments/pcep/pcep.xhtml>; see PCEP TLV Type Indicators.

Value	Name	Reference
TBD2	VIRTUAL-NETWORK-TLV	[This I.D.]

## 7.3. PCEP Error

IANA is requested to make the following allocations from this registry at <http://www.iana.org/assignments/pcep/pcep.xhtml>; see PCEP-ERROR Object Error Types and Values.

This document defines new Error-Type and Error-Value for the following new error conditions:

Error-Type	Meaning
6	Mandatory Object missing  Error-value=TBD3: VIRTUAL-NETWORK TLV missing

## 8. References

### 8.1. Normative References

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

[I-D.ietf-pce-pce-initiated-lsp] E. Crabbe, et. al., "PCEP Extensions for PCE-initiated LSP Setup in a Stateful PCE Model", draft-ietf-pce-pce-initiated-lsp, work in progress.

[I-D.ietf-pce-association-group] I. Minei, Ed., "PCEP Extensions for Establishing Relationships Between Sets of LSPs", draft-ietf-pce-association-group, work in progress.

[I-D.ietf-dhodylee-stateful-HPCE] Dhody, D. and Lee, Y.,  
"Hierarchical Stateful Path Computation Element (PCE)",  
draft-dhodylee-pce-stateful-hpce, work in progress.

## 8.2. Informative References

- [RFC4655] Farrel, A., Vasseur, J.-P., and J. Ash, "A Path Computation Element (PCE)-Based Architecture", RFC 4655, August 2006.
- [RFC5440] Vasseur, JP., Ed. and JL. Le Roux, Ed., "Path Computation Element (PCE) Communication Protocol (PCEP)", RFC 5440, March 2009.
- [RFC4655] Farrel, A., Vasseur, J., and J. Ash, "A Path Computation Element (PCE)-Based Architecture", RFC 4655, August 2006.
- [RFC6805] A. Farrel and D. King, "The Application of the Path Computation Element Architecture to the Determination of a Sequence of Domains in MPLS and GMPLS", RFC 6805, November 2012.
- [I-D.ietf-pce-stateful-pce-app] Zhang, X., ED, and Minei, I., ED,  
"Applicability of a Stateful Path Computation Element (PCE)", draft-ietf-pce-stateful-pce-app, work-in-progress.
- [ACTN-REQ] Y. Lee, D. Dhody, S. Belotti, K. Pithewan, and D. Ceccarelli, "Requirements for Abstraction and Control of TE Networks", draft-ietf-teas-actn-requirements, work in progress.

## Author's Addresses

Young Lee (Editor)  
Huawei Technologies  
5340 Legacy Drive, Building 3  
Plano, TX 75023, USA

Email: leeyoung@huawei.com

Dhruv Dhody (Editor)  
Huawei Technologies  
Divyashree Technopark, Whitefield  
Bangalore, Karnataka 560037  
India

EMail: dhruv.ietf@gmail.com

Daniele Ceccarelli  
Ericsson  
Torshamnsgatan, 48  
Stockholm, Sweden

EMail: daniele.ceccarelli@ericsson.com

Xian Zhang  
Huawei Technologies

Email: zhang.xian@huawei.com



PCE Working Group  
Internet-Draft  
Intended Status: Standards track  
Expires: December 21, 2019

Y. Lee  
Futurewei  
X. Zhang  
Huawei Technologies  
D. Ceccarelli  
Ericsson

June 19, 2019

Path Computation Element communication Protocol (PCEP) extensions  
for Establishing Relationships between sets of LSPs and Virtual  
Networks  
draft-leedhody-pce-vn-association-08

## Abstract

This document describes how to extend Path Computation Element (PCE) Communication Protocol (PCEP) association mechanism introduced by the PCEP Association Group specification, to further associate sets of LSPs with a higher-level structure such as a virtual network (VN) requested by clients or applications. This extended association mechanism can be used to facilitate virtual network control using PCE architecture.

## Status of this Memo

This Internet-Draft is submitted to IETF 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  
<https://www.ietf.org/ietf/lid-abstracts.txt>

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

This Internet-Draft will expire on December 21, 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 (<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
1.1. Requirements Language . . . . .	4
2. Terminology . . . . .	5
3. Operation Overview . . . . .	5
4. Extensions to PCEP . . . . .	7
5. Applicability to H-PCE architecture . . . . .	8
6. Implementation Status . . . . .	9
6.1. Huawei's Proof of Concept based on ONOS . . . . .	9
7. Security Considerations . . . . .	10
8. IANA Considerations . . . . .	10
8.1. Association Object Type Indicator . . . . .	10
8.2. PCEP TLV Type Indicator . . . . .	10
8.3. PCEP Error . . . . .	11
9. Manageability Considerations . . . . .	11
9.1. Control of Function and Policy . . . . .	11
9.2. Information and Data Models . . . . .	11
9.3. Liveness Detection and Monitoring . . . . .	11
9.4. Verify Correct Operations . . . . .	11
9.5. Requirements On Other Protocols . . . . .	11
9.6. Impact On Network Operations . . . . .	12
10. References . . . . .	12
10.1. Normative References . . . . .	12

10.2. Informative References . . . . .	12
--	----

## 1. Introduction

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.

[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 setup, maintenance and teardown of PCE-initiated LSPs under the stateful PCE model.

[I-D.ietf-pce-association-group] introduces a generic mechanism to create a grouping of LSPs. This grouping can then be used to define association between sets of LSPs or between a set of LSPs and a set of attributes.

[RFC8453] describes various Virtual Network (VN) operations initiated by a customer/application. In this context, there is a need for associating a set of LSPs with a VN "construct" to facilitate VN operations in PCE architecture. This association allows the PCEs to identify which LSPs belong to a certain VN. The PCE could then use this association to optimize all LSPs belonging to the VN together. The PCE could further take VN specific actions on the LSPs such as relaxation of constraints, policy actions, setting default behavior etc.

[I-D.ietf-pce-applicability-actn] examines the PCE and ACTN architecture and describes how the PCE architecture is applicable to ACTN. [RFC6805] and [I-D.ietf-pce-stateful-hpce] describes a hierarchy of stateful PCEs with Parent PCE coordinating multi-domain path computation function between Child PCE(s) and thus making it the base for PCE applicability for ACTN. In this text child PCE would be same as Provisioning Network Controller (PNC), and the parent PCE as

Multi-domain Service Coordinator (MDSC) [RFC8453].

This document specifies a PCEP extension to associate a set of LSPs based on Virtual Network (VN) (or customer). A Virtual Network (VN) is a customer view of the TE network. Depending on the agreement between client and provider various VN operations and VN views are possible as described in [RFC8453].

### 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. Terminology

The terminology is as per [RFC4655], [RFC5440], [RFC6805], [RFC8231] and [RFC8453].

## 3. Operation Overview

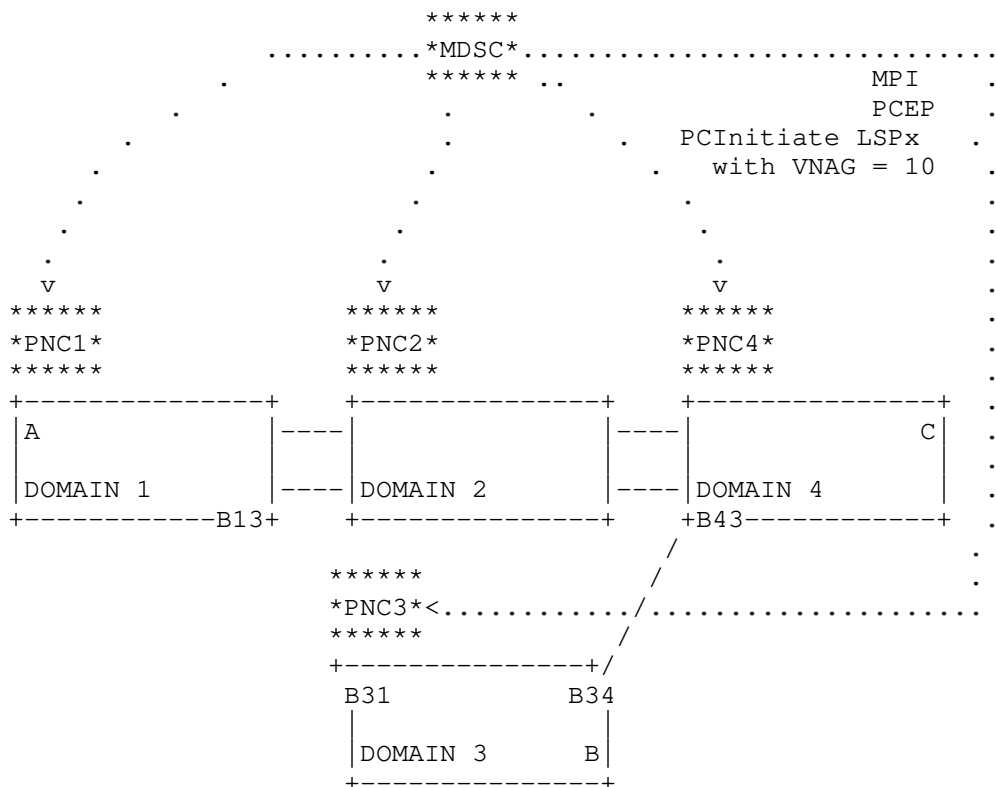
As per [I-D.ietf-pce-association-group], LSPs are associated with other LSPs with which they interact by adding them to a common association group.

An association group based on VN is useful for various optimizations that should be applied by considering all the LSPs in the association. This includes, but not limited to -

- o Path Computation: When computing path for a LSP, the impact of this LSP, on the other LSPs belonging to the same VN is useful to analyze. The aim would be optimize overall VN and all LSPs, rather than a single LSP. Also, the optimization criteria such as minimize the load of the most loaded link (MLL) [RFC5541] and other could be applied for all the LSP belonging to the same VN, identified by the VN association.

- o Path Re-Optimization: The child PCE or the parent PCE would like to use advanced path computation algorithm and optimization technique that consider all the LSPs belonging to a VN/customer and optimize them all together during the re-optimization.

This association is useful in PCEP session between parent PCE (MDSC) and child PCE (PNC). The figure describes a typical VN operations using PCEP for illustration purpose.



```
MDSC -> Parent PCE
PNC  -> Child  PCE
MPI  -> PCEP
```

In this draft, this grouping is used to define associations between a set of LSPs and a virtual network, a new association group is defined below -

- o VN Association Group (VNAG)

One new Association type is defined as described in the Association object -

- o Association type = TBD1 ("VN Association") for VNAG

The scope and handling of VNAG identifier is similar to the generic association identifier defined in [I-D.ietf-pce-association-group].

In this document VNAG object refers to an Association Object with the Association type set to "VNAG".

Local policies on the PCE MAY define the computational and optimization behavior for the LSPs in the VN. An LSP MUST NOT belong to more than one VNAG. If an implementation encounters more than one VNAG, it MUST consider the first occurrence and ignore the others.

[I-D.ietf-pce-association-group] specify the mechanism for the capability advertisement of the association types supported by a PCEP speaker by defining a ASSOC-Type-List TLV to be carried within an OPEN object. This capability exchange for the association type described in this document (i.e. VN Association Type) MUST be done before using the policy association. Thus the PCEP speaker MUST include the VN Association Type (TBD1) in the ASSOC-Type-List TLV before using the VNAG in the PCEP messages.

This Association-Type is dynamic in nature and created by the Parent PCE (MDSC) for the LSPs belonging to the same VN or customer. These associations are conveyed via PCEP messages to the PCEP peer. Operator-configured Association Range MUST NOT be set for this association-type and MUST be ignored.

#### 4. Extensions to PCEP

The format of VNAG is as per the ASSOCIATION object [I-D.ietf-pce-association-group].

This document defines one mandatory TLV "VIRTUAL-NETWORK-TLV" and one new optional TLV "VENDOR-INFORMATION-TLV"; apart from this TLV, VENDOR-INFORMATION-TLV can be used to carry arbitrary vendor specific information.

- o VIRTUAL-NETWORK-TLV: Used to communicate the VN Identifier.
- o VENDOR-INFORMATION-TLV: Used to communicate arbitrary vendor specific behavioral information, described in [RFC7470].

The format of VIRTUAL-NETWORK-TLV is as follows.

```

      0                               1                               2                               3
      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
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                                     | Length (variable)                |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                                     |                                     |
//                                     Virtual Network Name                //
|                                     |                                     |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

Figure 1: The VIRTUAL-NETWORK-TLV formats

Type: TBD2 (to be allocated by IANA)

Length: Variable Length

Virtual Network Name (variable): an unique symbolic name for the VN. It SHOULD be a string of printable ASCII characters, without a NULL terminator. The VN name is a human-readable string that identifies a VN. The VN name MUST remain constant throughout an LSP's lifetime, which may span across multiple consecutive PCEP sessions and/or PCC restarts. The VN name MAY be specified by an operator or auto-generated by the PCEP speaker.

The VIRTUAL-NETWORK-TLV MUST be included in VNAG object. If a PCEP speaker receives the VNAG object without the VIRTUAL-NETWORK-TLV, it MUST send a PCErr message with Error-Type=6 (mandatory object missing) and Error-Value=TBD3 (VIRTUAL-NETWORK-TLV missing) and close the session.

The format of VENDOR-INFORMATION-TLV is defined in [RFC7470].

## 5. Applicability to H-PCE architecture

The ability to compute shortest constrained TE LSPs in Multiprotocol Label Switching (MPLS) and Generalized MPLS (GMPLS) networks across multiple domains has been identified as a key motivation for PCE development. [RFC6805] describes a Hierarchical PCE (H-PCE) architecture which can be used for computing end-to-end paths for inter-domain MPLS Traffic Engineering (TE) and GMPLS Label Switched Paths (LSPs). Within the hierarchical PCE architecture, the parent PCE is used to compute a multi-domain path based on the domain connectivity information. A child PCE may be responsible for a single domain or multiple domains, it is used to compute the intra-domain path based on its domain topology information.

[I-D.ietf-pce-stateful-hpce] introduces general considerations for stateful PCE(s) in hierarchical PCE architecture. In particular, the behavior changes and additions to the existing stateful PCE mechanisms in the context of a H-PCE architecture.

In Stateful H-PCE architecture, the Parent PCE receives a virtual network creation request by its client over its Northbound API. This VN is uniquely identified by an Association ID in VNAG as well as the VIRTUAL-NETWORK name. This VN may comprise multiple LSPs in the network in a single domain or across multiple domains.

As the Parent PCE computes the optimum E2E paths for each tunnel in

VN, it MUST associate each LSP with the VN to which it belongs. Parent PCE sends a PCInitiate Message with this association information in the VNAG Object (See Section 4 for details). This in effect binds an LSP that is to be instantiated at the child PCE with the VN.

Whenever changes occur with the instantiated LSP in a domain network, the domain child PCE reports the changes using a PCRpt Message in which the VNAG Object indicates the relationship between the LSP and the VN.

Whenever an update occurs with VNs in the Parent PCE (via the client's request), the parent PCE sends an PCUpd Message to inform each affected child PCE of this change.

The Child PCE could then use this association to optimize all LSPs belonging to the same VN association together. The Child PCE could further take VN specific actions on the LSPs such as relaxation of constraints, policy actions, setting default behavior etc. The parent PCE could also maintain all E2E LSP or per-domain path segments under a single VN association.

## 6. Implementation Status

[Note to the RFC Editor - remove this section before publication, as well as remove the reference to RFC 7942.]

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 RFC 7942 [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 RFC 7942, "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".

### 6.1. Huawei's Proof of Concept based on ONOS



The PCE function was developed in the ONOS open source platform. This extension was implemented on a private version as a proof of concept to ACTN.

- o Organization: Huawei
- o Implementation: Huawei's PoC based on ONOS
- o Description: PCEP as a southbound plugin was added to ONOS. To support ACTN, this extension in PCEP is used. Refer <https://wiki.onosproject.org/display/ONOS/PCEP+Protocol>
- o Maturity Level: Prototype
- o Coverage: Full
- o Contact: satishk@huawei.com

## 7. Security Considerations

This document defines one new type for association, which do not add any new security concerns beyond those discussed in [RFC5440], [RFC8231] and [I-D.ietf-pce-association-group] in itself.

Some deployments may find the Virtual Network Name and the VN associations as extra sensitive; and thus should employ suitable PCEP security mechanisms like TCP-AO [RFC5925] or [RFC8253].

## 8. IANA Considerations

### 8.1. Association Object Type Indicator

This document defines a new association type, originally defined in [I-D.ietf-pce-association-group], for path protection. IANA is requested to make the assignment of a new value for the sub-registry "ASSOCIATION Type Field" (request to be created in [I-D.ietf-pce-association-group]), as follows:

Value	Name	Reference
TBD1	VN Association Type	[This I.D.]

### 8.2. PCEP TLV Type Indicator

This document defines a new TLV for carrying additional information of LSPs within a path protection association group. IANA is requested to make the assignment of a new value for the existing "PCEP TLV Type Indicators" registry as follows:

Value	Name	Reference
TBD2	VIRTUAL-NETWORK-TLV	[This I.D.]

### 8.3. PCEP Error

This document defines new Error-Type and Error-Value related to path protection association. IANA is requested to allocate new error values within the "PCEP-ERROR Object Error Types and Values" sub-registry of the PCEP Numbers registry, as follows:

Error-Type	Meaning
------------	---------

6	Mandatory Object missing Error-value=TBD3: VIRTUAL-NETWORK TLV missing [This I.D.]
---	---

## 9. Manageability Considerations

### 9.1. Control of Function and Policy

An operator MUST BE allowed to mark LSPs that belong to the same VN. This could also be done automatically based on the VN configuration.

### 9.2. Information and Data Models

The PCEP YANG module [I-D.ietf-pce-pcep-yang] should support the association between LSPs including VN association.

### 9.3. 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].

### 9.4. Verify Correct Operations

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

### 9.5. Requirements On Other Protocols

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

## 9.6. 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].

## 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.
- [RFC5440] Vasseur, JP., Ed. and JL. Le Roux, Ed., "Path Computation Element (PCE) Communication Protocol (PCEP)", RFC 5440, March 2009.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, May 2017.
- [RFC8231] E. Crabbe, I. Minei, J. Medved, and R. Varga, "PCEP Extensions for Stateful PCE", RFC 8231, September 2017.
- [RFC8281] E. Crabbe, et. al., "PCEP Extensions for PCE-initiated LSP Setup in a Stateful PCE Model", RFC 8281, December 2017.
- [I-D.ietf-pce-association-group] I, Minei, Ed., "PCEP Extensions for Establishing Relationships Between Sets of LSPs", draft-ietf-pce-association-group, work in progress.

### 10.2. Informative References

- [RFC4655] Farrel, A., Vasseur, J.-P., and J. Ash, "A Path Computation Element (PCE)-Based Architecture", RFC 4655, August 2006.
- [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>>.
- [RFC6805] A. Farrel and D. King, "The Application of the Path Computation Element Architecture to the Determination of a Sequence of Domains in MPLS and GMPLS", RFC 6805, November 2012.
- [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.

- [RFC8453] Ceccarelli, D., Ed. and Y. Lee, Ed., "Framework for Abstraction and Control of TE Networks (ACTN)", RFC 8453, DOI 10.17487/RFC8453, August 2018, <<https://www.rfc-editor.org/info/rfc8453>>.
- [I-D.ietf-pce-applicability-actn] Dhody D., Lee Y., and D. Ceccarelli, "Applicability of Path Computation Element (PCE) for Abstraction and Control of TE Networks (ACTN)", draft-ietf-pce-applicability-actn, work-in-progress.
- [I-D.ietf-pce-stateful-hpce] Dhody, D. and Lee, Y., "Hierarchical Stateful Path Computation Element (PCE)", draft-ietf-pce-stateful-hpce, work in progress.
- [RFC5541] Le Roux, JL., Vasseur, JP., and Y. Lee, "Encoding of Objective Functions in the Path Computation Element Communication Protocol (PCEP)", RFC 5541, DOI 10.17487/RFC5541, June 2009, <<http://www.rfc-editor.org/info/rfc5541>>.
- [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, <<http://www.rfc-editor.org/info/rfc7470>>.
- [RFC8051] Zhang, X., Ed. and I. Minei, Ed., "Applicability of a Stateful Path Computation Element (PCE)", RFC 8051, DOI 10.17487/RFC8051, January 2017, <<http://www.rfc-editor.org/info/rfc8051>>.
- [RFC8253] Lopez, D., Dios, O., Wu, W., and D. Dhody, "Secure Transport for PCEP", RFC 8253, October 2017 .
- [I-D.ietf-pce-pcep-yang] Dhody, D., Hardwick, J., Beeram, V., and j. jeffrant@gmail.com, "A YANG Data Model for Path Computation Element Communications Protocol (PCEP)", draft-ietf-pce-pcep-yang (work in progress).

Contributor's Addresses

Dhruv Dhody  
Huawei Technologies  
Divyashree Technopark, Whitefield  
Bangalore, Karnataka 560066  
India

Email: dhruv.ietf@gmail.com

Qin Wu  
Huawei Technologies  
China

Email: bill.wu@huawei.com

Author's Addresses

Young Lee  
Futurewei  
5340 Legacy Drive, Building 3  
Plano, TX 75023,  
USA

Email: younglee.tx@gmail.com

Xian Zhang  
Huawei Technologies  
China

Email: zhang.xian@huawei.com

Daniele Ceccarelli  
Ericsson  
Torshamnsgatan, 48  
Stockholm,  
Sweden

Email: daniele.ceccarelli@ericsson.com

Network Working Group  
Internet-Draft  
Intended status: Informational  
Expires: September 19, 2016

Z. Li  
X. Chen  
Huawei Technologies  
March 18, 2016

PCEP Extensions for Bidirectional Forwarding Detection  
draft-li-pce-bfd-00

## Abstract

This document describes the extensions to the PCEP to notify BFD parameters for LSPs from PCE to PCC for PCE-initiated LSP. The extensions include BFD protocol parameters and allow PCC to support BFD for PCE-Initiated LSP whose BFD session is a bi-directional co-routed channel.

## Requirements Language

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

## Status of This Memo

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

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

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

This Internet-Draft will expire on September 19, 2016.

## Copyright Notice

Copyright (c) 2016 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 . . . . .	2
2. Terminology . . . . .	3
3. Bootstrapping Bi-directional Co-routed BFD Session . . . . .	3
3.1. Bootstrapping BFD session without LSP Ping . . . . .	3
3.2. Bootstrapping BFD session with LSP Ping . . . . .	4
4. TLVs of PCEP Extensions for BFD . . . . .	4
4.1. BFD Reverse Path TLV . . . . .	4
4.2. BFD Generic TLV . . . . .	5
4.3. BFD Authentication TLV . . . . .	6
5. IANA Considerations . . . . .	7
6. Security Considerations . . . . .	7
7. Normative References . . . . .	7
Authors' Addresses . . . . .	8

## 1. Introduction

RFC 5884 [RFC5884] describes the applicability of BFD in relation to LSP Ping for detecting rapidly a Multiprotocol Label Switching (MPLS) Label Switched Path (LSP) data plane failure. It also describes procedures for using BFD in MPLS environment. The LSP BFD detecting can be bidirectional LSP or unidirectional LSP (so long as there is some return path). If the path from ingress LSR to egress LSR is not co-routed with the path from egress LSR to ingress LSR, the failure to deliver BFD control packets from egress LSR to ingress LSR can lead to false negatives, making ingress LSR deduces that the LSP has failed.

I-D.ietf-pce-pce-initiated-lsp [I-D.ietf-pce-pce-initiated-lsp] introduces the procedure of PCE-initiated LSPs under the stateful PCE model. PCC will automatically set up the MPLS RSVP-TE LSP according to the explicit path PCE provides. BFD session for the PCE-initiated LSP can also be created dynamically and the return path is implicitly the shortest path. Such BFD session whose forward and reverse paths are possibly not co-routed has the same problem as mentioned above.

This document describes the extensions to the PCEP to notify BFD parameters for LSPs from PCE to PCC for PCE-initiated LSP. The extensions allow PCC to set up BFD session for PCE-Initiated LSP.

The BFD control packets can be exchanged over a bi-directional co-routed channel.

The BFD protocol parameters such as detection time multiplier, desired Min TX Interval, required Min RX Interval for PCE-initiated LSP come from the public template or global configuration on PCC. The extensions of PCEP include generic BFD protocol parameters too. It can be used to notify PCC by PCE to adjust these parameters for special LSP.

## 2. Terminology

BFD: Bidirectional Forwarding Detection

LSP: Label Switching Path

This document uses the following terms defined in RFC 5440 [RFC5440]: PCC, PCE, PCEP.

The following term is defined in I-D.ietf-pce-pce-initiated-lsp [I-D.ietf-pce-pce-initiated-lsp]:

PCE-initiated LSP: LSP that is instantiated as a result of a request from the PCE.

## 3. Bootstrapping Bi-directional Co-routed BFD Session

PCE computes the path for one LSP from the ingress LSR to egress LSR and initiates the creation of this LSP on ingress LSR. The LSP is called as LSP1. PCE can initiate the creation of LSP on egress LSR according to the co-routed path from egress LSR to ingress LSR. The LSP is called as LSP2.

To make the BFD session for LSP1 over the co-routed path to avoid the false detection there are two solutions as below:

### 3.1. Bootstrapping BFD session without LSP Ping

BFD for MPLS LSP uses LSP Ping carrying local discriminator to bootstrapping BFD session in order to associate the FEC representing the LSP with the BFD session indicated by discriminators. But PCE knows the two co-routed LSPs and can allocate the pair of discriminators for the co-routed LSPs.

PCE notify ingress LSR and egress LSR to set up BFD session for LSP1 by the BFD extensions of PCEP the pair of discriminators and notify PCC not necessary to send LSP ping message and directly to set up BFD



session. My discriminator in BFD control packets along LSP1 is your discriminator in BFD control packets along LSP2 and vice versa.

By this method the same BFD session is set up not only for LSP1 but also LSP2.

How to guarantee the discriminators allocated by PCE and PCC are not the same is out of scope of this document.

### 3.2. Bootstrapping BFD session with LSP Ping

I-D.ietf-mpls-bfd-directed [I-D.ietf-mpls-bfd-directed] defines the BFD Reverse Path TLV as an extension to LSP Ping RFC 4379 [RFC4379] and proposes that it to be used to instruct the egress BFD peer to use specified path for its BFD control packets associated with the particular BFD session.

After PCE initiates PCC to set up the LSP PCC delegates the MPLS RSVP-LSP with LSP-IDENTIFIERS TLV including FEC information to PCE. Therefore after ingress LSR and egress LSR set up the LSP1 and LSP2 independently they will delegate the LSP1 and LSP2 FEC information to PCE independently.

PCE notify ingress LSR to set up BFD session for LSP1 carrying the FEC information about the reverse LSP, LSP2. The information received by ingress LSR via PCEP can be set to the BFD Reverse Path TLV in the LSP Ping message. Following the procedure defined in [draft-ietf-mpls-bfd-directed-02] the BFD session would be a bi-directional co-routed channel and no false detection would be notified.

PCE notify Egress LSR to set up BFD session for LSP2 following the same process above.

By this method two BFD sessions are set up for LSP1 and LSP2 independently.

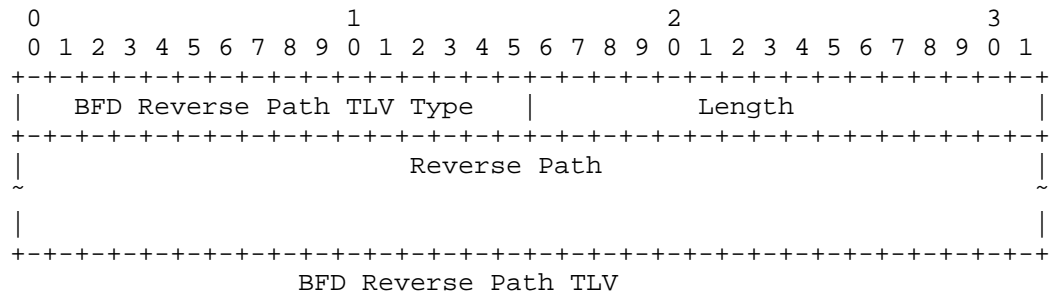
## 4. TLVs of PCEP Extensions for BFD

### 4.1. BFD Reverse Path TLV

The BFD Reverse Path TLV provides BFD parameters used to indicate the reverse path for BFD session.

This is an optional TLV defined for the LSPA Object. This TLV is included in the LSPA Object with PCUpd message.

The format of the BFD Reverse Path TLV is shown in the following figure:



BFD Reverse Path TLV Type is 2 octets in length and the value is to be assigned by IANA.

Length is 2 octets in length and defines the length in octets of the Reverse Path field.

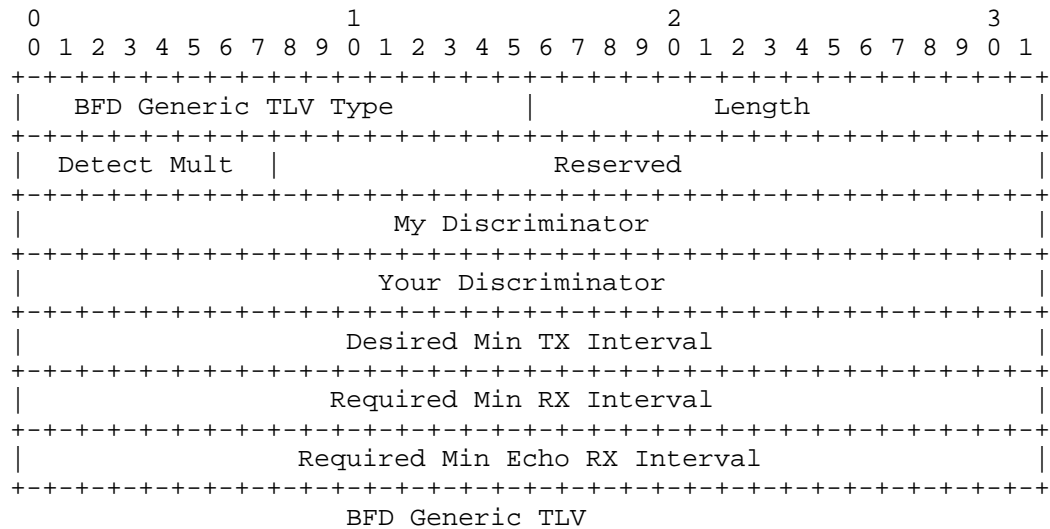
Reverse Path refers to Reverse Path defined in BFD Reverse Path TLV in I-D.ietf-mpls-bfd-directed [I-D.ietf-mpls-bfd-directed].

#### 4.2. BFD Generic TLV

The BFD Generic TLV provides BFD generic parameters of BFD session.

This is an optional TLV defined for the LSPA Object. This TLV is included in the LSPA Object with PCInitiate or PCUpd message.

The format of the BFD Generic TLV is shown in the following figure:



BFD Generic Path TLV Type is 2 octets in length and the value is to be assigned by IANA.

Length is 2 octets in length and defines the fixed length 20.

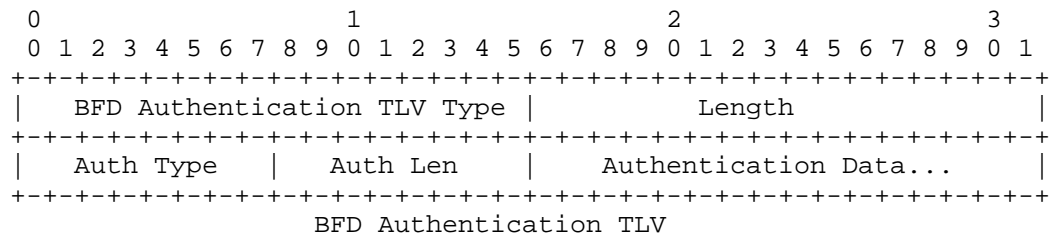
The value of TLV refers to Generic BFD Control Packet Format in RFC 5880 [RFC5880].

#### 4.3. BFD Authentication TLV

The BFD Authentication TLV provides BFD authentication parameters of BFD session.

This is an optional TLV defined for the LSPA Object. This TLV is included in the LSPA Object with PCInitiate or PCUpd message.

The format of the BFD Generic Path TLV is shown in the following figure:



BFD Authentication TLV Type is 2 octets in length and the value is to be assigned by IANA.

Length is 2 octets in length and defines the length in octets of the value of BFD Authentication TLV.

The value of TLV refers to the optional Authentication Section in RFC 5880 [RFC5880].

## 5. IANA Considerations

TBD.

## 6. Security Considerations

TBD.

## 7. Normative References

[I-D.ietf-mpls-bfd-directed]

Mirsky, G., Tantsura, J., Varlashkin, I., and M. Chen, "Bidirectional Forwarding Detection (BFD) Directed Return Path", draft-ietf-mpls-bfd-directed-02 (work in progress), March 2016.

[I-D.ietf-pce-pce-initiated-lsp]

Crabbe, E., Minei, I., Sivabalan, S., and R. Varga, "PCEP Extensions for PCE-initiated LSP Setup in a Stateful PCE Model", draft-ietf-pce-pce-initiated-lsp-05 (work in progress), October 2015.

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

[RFC4379] Kompella, K. and G. Swallow, "Detecting Multi-Protocol Label Switched (MPLS) Data Plane Failures", RFC 4379, DOI 10.17487/RFC4379, February 2006, <<http://www.rfc-editor.org/info/rfc4379>>.

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

- [RFC5880] Katz, D. and D. Ward, "Bidirectional Forwarding Detection (BFD)", RFC 5880, DOI 10.17487/RFC5880, June 2010, <<http://www.rfc-editor.org/info/rfc5880>>.
- [RFC5884] Aggarwal, R., Kompella, K., Nadeau, T., and G. Swallow, "Bidirectional Forwarding Detection (BFD) for MPLS Label Switched Paths (LSPs)", RFC 5884, DOI 10.17487/RFC5884, June 2010, <<http://www.rfc-editor.org/info/rfc5884>>.

## Authors' Addresses

Zhenbin Li  
Huawei Technologies  
Huawei Bld., No.156 Beiqing Rd.  
Beijing 100095  
China

Email: [lizhenbin@huawei.com](mailto:lizhenbin@huawei.com)

Xia Chen  
Huawei Technologies  
Huawei Bld., No.156 Beiqing Rd.  
Beijing 100095  
China

Email: [jescia.chenxia@huawei.com](mailto:jescia.chenxia@huawei.com)

Network Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: August 31, 2016

Z. Li  
X. Chen  
S. Zhuang  
Huawei Technologies  
February 28, 2016

PCEP Extension for Flow Specification  
draft-li-pce-pcep-flowspec-00

Abstract

Dissemination of the traffic flow specifications was first introduced in the BGP protocol via RFC 5575. In order to distribute the flow specifications from PCE controller to network device without BGP protocol it is desirable to extend PCEP with flow specification information.

This document specifies a set of extensions to PCEP to support dissemination of flow specifications. The extensions include the instantiation, updation and deletion of flowspecifications.

Requirements Language

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

Status of This Memo

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

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

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

This Internet-Draft will expire on August 31, 2016.

## Copyright Notice

Copyright (c) 2016 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 . . . . .	2
2. Terminology . . . . .	4
3. Procedures for Dissemination of FlowSpec . . . . .	4
3.1. Overview of Procedures . . . . .	4
3.2. Capability Advertisement . . . . .	5
3.3. Operations . . . . .	5
4. PCEP Messages . . . . .	6
4.1. PCEP FlowSpec Message . . . . .	6
5. Objects and TLVs . . . . .	7
5.1. OPEN Object . . . . .	8
5.1.1. PCE FlowSpec Capability TLV . . . . .	8
5.2. FLOW Object . . . . .	8
5.3. ACTION Object . . . . .	10
6. IANA Considerations . . . . .	12
7. Security Considerations . . . . .	12
8. Acknowledgements . . . . .	12
9. References . . . . .	12
9.1. Normative References . . . . .	12
9.2. Informative References . . . . .	12
Appendix A. Contributor Addresses . . . . .	13
Appendix B. Example Usage . . . . .	13
Authors' Addresses . . . . .	15

## 1. Introduction

Dissemination of the traffic flow specifications was first introduced in the BGP protocol [RFC5575]. The traffic flow specification is comprised of traffic filtering rules and actions. The routers which received the flow specification can take advantage of the ACL (Access Control List) or firewall capabilities in the router's forwarding path. The routers can classify the packets according to the traffic

filtering rules and shape, rate limit, filter, or redirect packets based on the actions. The flow specification carried by BGP can be used to automate inter-domain coordination of traffic filtering to mitigate (distributed) denial-of-service attacks and can also be used to provide traffic filtering in the context of a BGP/MPLS VPN service.

[RFC5575] also defines that a flow specification received from an external autonomous system will need to be validated against unicast routing before being accepted. [I-D.ietf-idr-bgp-flowspec-oid] describes a modification to the validation procedure defined in [I-D.ietf-idr-bgp-flowspec-oid] for the dissemination of BGP flow specifications. The modification proposed enables flow specifications to be originated from a centralized BGP route controller.

[I-D.ietf-ospf-flowspec-extensions] defines the extensions to OSPF to distribute flow specifications in the networks that only deploy an IGP (Interior Gateway Protocol) (e.g., OSPF). It also defines the validation procedures for imposing the filtering information on the routers.

[RFC5440] describes the Path Computation Element Protocol (PCEP). PCEP defines the communication between a Path Computation Client (PCC) and a Path Control Element (PCE), or between PCE and PCE, enabling computation of Multiprotocol Label Switching (MPLS) for Traffic Engineering Label Switched Path (TE LSP) characteristics.

Stateful pce [I-D.ietf-pce-stateful-pce] specifies a set of extensions to PCEP to enable stateful control of TE LSPs between and across PCEP sessions in compliance with [RFC4657]. It includes mechanisms to effect LSP state synchronization between PCCs and PCEs, delegation of control of LSPs to PCEs, and PCE control of timing and sequence of path computations within and across PCEP sessions and focuses on a model where LSPs are configured on the PCC and control over them is delegated to the PCE. [I-D.ietf-pce-pce-initiated-lsp] describes the setup, maintenance and teardown of PCE- initiated LSPs under the stateful PCE model, without the need for local configuration on the PCC, thus allowing for a dynamic network that is centrally controlled and deployed.

In case PCE is used to initiate tunnels via PCEP, it is desirable to use the same protocol to also distribute the flow specifications to describe what data flows on those tunnels. Thus, in order to distribute the flow specifications from PCE controller to network device, PCEP is extended with flow specification information in this document.



This document specifies a set of extensions to PCEP to support dissemination of flow specifications. The flow specifications can be disseminated between PCEP peers such as from PCE to PCC or between PCEs. The extensions include the creation, updation and withdrawal of flow specifications via PCEP.

The values of flow filtering rules and actions mainly refer to the BGP flow specification and IGP specification. This document extends new actions which are redirecting to LSP (referred by Symbolic Path Name, IPv4 LSP, or IPv6 LSP).

## 2. Terminology

This document uses the terms defined in [RFC5440] and [RFC5575].

This document uses the terms defined in [RFC5440]: PCC, PCE, PCEP Peer.

The following term is from [RFC5575]. It is used frequently throughout this document:

Flow Specification (FlowSpec): A flow specification is an n-tuple consisting of several matching criteria that can be applied to IP traffic, including filters and actions. Each FlowSpec consists of a set of filters and a set of actions.

## 3. Procedures for Dissemination of FlowSpec

### 3.1. Overview of Procedures

A PCC or PCE indicates its ability to support PCE FlowSpec during the PCEP Initialization Phase via "PCE FlowSpec Capability" TLV (see details in Section 5.1.1).

This section introduces the procedure to support PCE FlowSpec as follows:

Firstly both the PCE and PCC advertise the PCE FlowSpec Capability during the PCE session initiation phase.

On the PCEP session with PCE FlowSpec Capability PCE communicates with PCC to create, update and withdraw PCE FlowSpec.

[Editor's Note - The procedure about PCE FlowSpec synchronization, the session failure process, etc. will be specified in the future version.]

### 3.2. Capability Advertisement

During PCEP session establishment, both the PCC and the PCE must announce their support of PCEP extensions for FlowSpec defined in this document.

A PCEP Speaker (PCE or PCC) includes the "PCE FlowSpec Capability" TLV, described in Section 5.1.1, in the OPEN Object to advertise its support for PCEP extensions for PCE FlowSpec Capability.

The presence of the PCE FlowSpec Capability TLV in PCE's OPEN message indicates that the PCE can support distribute the FlowSpec to PCC.

The presence of such Capability TLV in PCC's OPEN Object indicates that the PCC can be in support of Flowspec functionality to instantiate the FlowSpec according to the PCE's indication and can apply the FlowSpec to the incoming packets.

If PCE has such capability TLV and PCC has no such capability TLV PCE MUST NOT send the PCE messages with FlowSpec information. And if PCC receives such messages it should send PCErr message to PCE.

[Editor's Note - PCE discovery via IGP should also be extended for this.]

### 3.3. Operations

To instantiate a FlowSpec which is comprised of a set of FlowSpec filter rules and actions, the PCE sends a new PCEP message (called FlowSpec message) to the PCC. The FlowSpec message MUST include the SRP object[I-D.ietf-pce-stateful-pce], a new FLOW object (see details in Section 5.2) and a new ACTION object (see details in Section 5.3). FLOW object carries a set of FlowSpec filter rules. A list of ACTION objects specify a set of FlowSpec actions.

To update the FlowSpec actions of a specified FlowSpec which has been created, the same PCEP message "FlowSpec" is used. The PCE sends a FlowSpec message to the PCC. The FlowSpec message MUST include the SRP object, FLOW object and ACTION object.

To delete the specified FlowSpec which has been created, the PCE sends a FlowSpec message to the PCC with a flag indicating the removal action. The FlowSpec message MUST include the SRP object (with R flag set) and FLOW object.

#### 4. PCEP Messages

As defined in [RFC5440], a PCEP message consists of a common header followed by a variable-length body made of a set of objects that can be either mandatory or optional. An object is said to be mandatory in a PCEP message when the object must be included for the message to be considered valid. For each PCEP message type, a set of rules is defined that specify the set of objects that the message can carry. An implementation **MUST** form the PCEP messages using the object ordering specified in this document.

To support the PCEP FlowSpec functionality one new PCEP messages is introduced.

##### 4.1. PCEP FlowSpec Message

A FlowSpec message which is also referred to as FlowSpec message is a PCEP message sent by a PCE to a PCC to trigger creation, modification or deletion of a FlowSpec.

The Message-Type field of the PCEP common header for the FlowSpec message is to be assigned by IANA. The FlowSpec message **MUST** include the SRP and the FLOW objects.

If FlowSpec message is used to create or update the FlowSpec, it **MUST** include the ACTION objects too.

If FlowSpec message is used to delete the FlowSpec the ACTION objects **SHOULD NOT** be carried and the SRP object is set with the R flag.

A FlowSpec is identified by a PCEP specific identifier FS-ID.

The format of a FlowSpec message for creation or deletion of FlowSpec is as follows:

```

<FlowSpec Message> ::= <Common Header>
                        <flowspec-list>

```

Where:

```

<flowspec-list> ::= <flowspec-request>[<flowspec-list>]

```

```

<flowspec-request> ::= (<flowspec-create-or-update> |
                        <flowspec-delete>)

```

```

<flowspec-create-or-update> ::= <SRP>
                                <FLOW>
                                <action-list>

```

```

<flowspec-delete> ::= <SRP>
                      <FLOW>

```

Where:

```

<action-list> ::= <ACTION>[<action-list>]

```

The SRP object defined in [I-D.ietf-pce-stateful-pce] can be used in this document to correlate FlowSpec requests sent by the PCE with the error reports sent by the PCC.

Every FlowSpec requests from the PCE sends a new SRP-ID-NUMBER as described in [I-D.ietf-pce-stateful-pce]. This number is unique per PCEP session and is incremented each time an FlowSpec operation (creation, update, deletion etc) is requested from the PCE. The value of the SRP-ID-NUMBER MAY be echoed back by the PCC in PCErr messages to allow for correlation between requests made by the PCE and errors generated by the PCC. Procedure of dissemination of FlowSpec from PCE share the same number space of the SRP-ID-NUMBER with procedure of stateful PCE.

The FLOW and ACTION objects are new objects introduced in this document.

## 5. Objects and TLVs

The PCEP objects defined in this document are compliant with the PCEP object format defined in [RFC5440].

New TLVs about FlowSpec filtering rules are defined. The value portion of the new TLVs can reuse the structure defined in [RFC5575] and [I-D.ietf-idr-flow-spec-v6]. New TLVs about FlowSpec actions are also defined. The value portion of the new TLVs can reuse the structure defined in [I-D.ietf-ospf-flowspec-extensions]. This document also defines two new actions: Redirect to IPv4 LSP and Redirect to IPv6 LSP.

## 5.1. OPEN Object

### 5.1.1. PCE FlowSpec Capability TLV

The PCE-FLOWSPEC-CAPABILITY TLV is an optional TLV associated with the OPEN Object [RFC5440] to exchange PCE FlowSpec capability of PCEP speakers.

Its format is shown in the following figure:

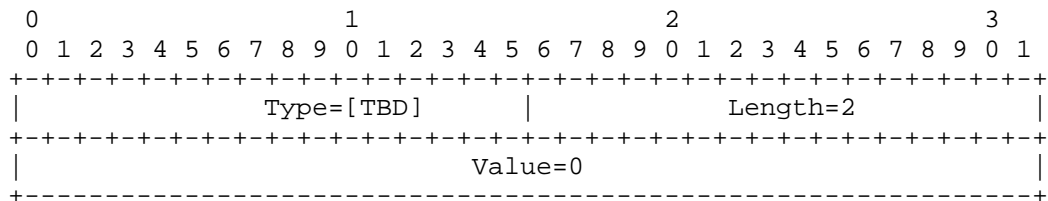


Figure 1: PCE-FLOWSPEC-CAPABILITY TLV format

The type of the TLV is to be assigned by IANA and it has a fixed length of 2 octets. The value field is set to default value 0.

The inclusion of this TLV in an OPEN object indicate that the sender can perform FlowSpec handling in PCEP.

## 5.2. FLOW Object

The FLOW object MUST be present within FlowSpec messages. The FLOW object carries a set of FlowSpec filter rules.

FLOW Object-Class is to be assigned by IANA.

Two FLOW Object-Type are defined so far:

- o IPv4 FLOW: FLOW Object-Type is 1.
- o IPv6 FLOW: FLOW Object-Type is 2.

The format of the FLOW object is as follows:

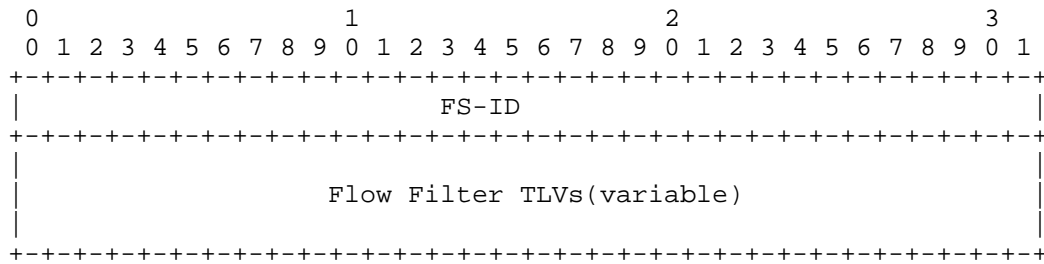


Figure 2: FLOW Object Body Format

FS-ID(32-bit): A PCEP-specific identifier for the FlowSpec information. A PCE creates an unique FS-ID for each FlowSpec that is constant for the lifetime of a PCEP session. All subsequent PCEP messages then address the FlowSpec by the FS-ID. The values of 0 and 0xFFFFFFFF are reserved.

Flow Filter TLVs(variable): The FLOW object body has a variable length and may contain one or more additional TLVs.

The following flow filter types are supported:

Type	Description	Ref TLV	Value defined in
TBD1	Destination IPv4 Prefix	1	RFC5575
TBD2	Source IPv4 Prefix	2	RFC5575
TBD3	IP Protocol	3	RFC5575
TBD4	Port	4	RFC5575
TBD5	Destination port	5	RFC5575
TBD6	Source port	6	RFC5575
TBD7	ICMP type	7	RFC5575
TBD8	ICMP code	8	RFC5575
TBD9	TCP flags	9	RFC5575
TBD10	Packet length	10	RFC5575
TBD11	DSCP	11	RFC5575
TBD12	Fragment	12	RFC5575
TBD13	Flow Label	13	I-D.ietf-idr-flow-spec-v6
TBD14	Destination IPv6 Prefix	1	I-D.ietf-idr-flow-spec-v6
TBD15	Source IPv6 Prefix	2	I-D.ietf-idr-flow-spec-v6
TBD16	Next Header	3	I-D.ietf-idr-flow-spec-v6

Table 2: Flow Filter Types

### 5.3. ACTION Object

The ACTION object MUST be present within FlowSpec messages when creating or updating the FlowSpec. The ACTION object carries a set of FlowSpec actions.

ACTION Object-Class is to be assigned by IANA.

ACTION Object-Type is 1.

The format of the ACTION object body is:

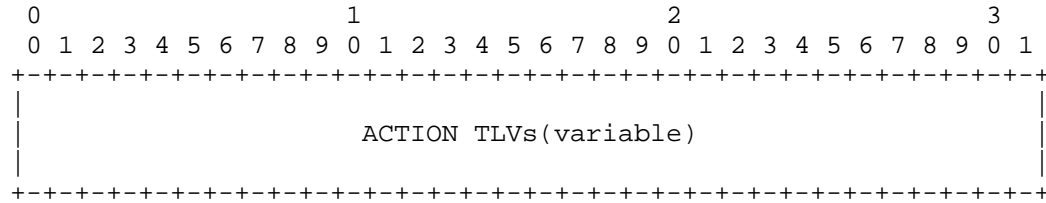


Figure 3: ACTION Object Body Format

The ACTION object body has a variable length and may contain one or more additional TLVs.

The following FlowSpec action types are supported:

Type	Description	Ref TLV	Value defined in
TBD17	traffic-rate	TBD	I-D.ietf-ospf-flowspec-extensions
TBD18	traffic-action	TBD	I-D.ietf-ospf-flowspec-extensions
TBD19	traffic-marking	TBD	I-D.ietf-ospf-flowspec-extensions
TBD20	redirect-to-IPv4	TBD	I-D.ietf-ospf-flowspec-extensions
TBD21	redirect-to-IPv6	TBD	I-D.ietf-ospf-flowspec-extensions
18(*)	IPV4-LSP-IDENTIFIERS	-	I-D.ietf-pce-stateful-pce
19(*)	IPV6-LSP-IDENTIFIERS	-	I-D.ietf-pce-stateful-pce
17(*)	Symbolic-Path-Name	-	I-D.ietf-pce-stateful-pce

Table 3: Flow Action Types

(\*) The type is defined in [I-D.ietf-pce-stateful-pce]



## 6. IANA Considerations

TBD.

## 7. Security Considerations

TBD.

## 8. Acknowledgements

TBD.

## 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, <<http://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, <<http://www.rfc-editor.org/info/rfc5440>>.
- [RFC5575] Marques, P., Sheth, N., Raszuk, R., Greene, B., Mauch, J., and D. McPherson, "Dissemination of Flow Specification Rules", RFC 5575, DOI 10.17487/RFC5575, August 2009, <<http://www.rfc-editor.org/info/rfc5575>>.
- [I-D.ietf-pce-stateful-pce] Crabbe, E., Minei, I., Medved, J., and R. Varga, "PCEP Extensions for Stateful PCE", draft-ietf-pce-stateful-pce-13 (work in progress), December 2015.
- [I-D.ietf-pce-pce-initiated-lsp] Crabbe, E., Minei, I., Sivabalan, S., and R. Varga, "PCEP Extensions for PCE-initiated LSP Setup in a Stateful PCE Model", draft-ietf-pce-pce-initiated-lsp-05 (work in progress), October 2015.

## 9.2. Informative References

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

[I-D.ietf-idr-bgp-flowspec-oid]

Uttaro, J., Filsfils, C., Smith, D., Alcaide, J., and P. Mohapatra, "Revised Validation Procedure for BGP Flow Specifications", draft-ietf-idr-bgp-flowspec-oid-02 (work in progress), January 2014.

[I-D.ietf-idr-flow-spec-v6]

Raszuk, R., Pithawala, B., McPherson, D., and A. Andy, "Dissemination of Flow Specification Rules for IPv6", draft-ietf-idr-flow-spec-v6-06 (work in progress), November 2014.

[I-D.ietf-ospf-flowspec-extensions]

Liang, Q., You, J., Wu, N., Fan, P., Patel, K., and A. Lindem, "OSPF Extensions for Flow Specification", draft-ietf-ospf-flowspec-extensions-00 (work in progress), June 2015.

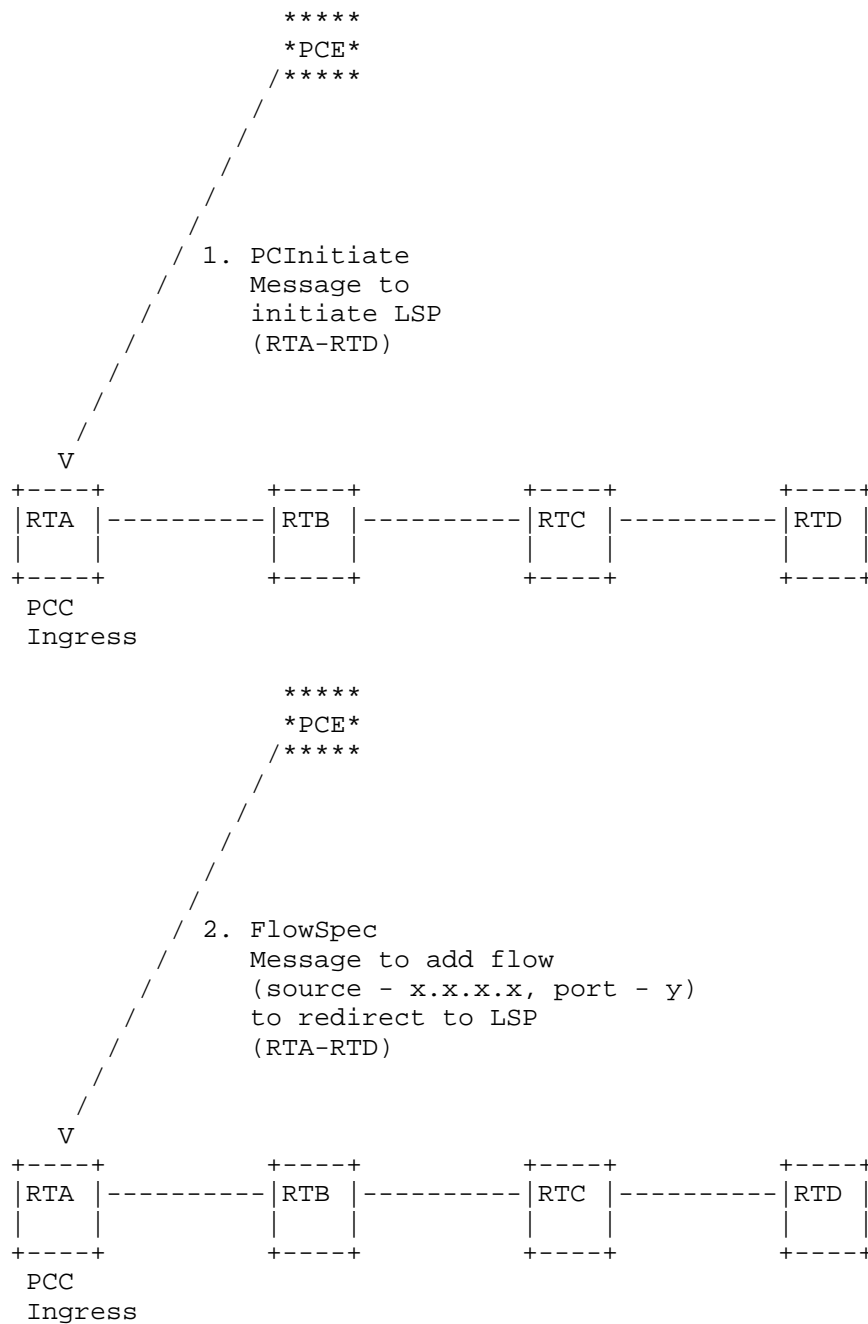
#### Appendix A. Contributor Addresses

Dhruv Dhody  
Huawei Technologies  
Divyashree Techno Park, Whitefield  
Bangalore, Karnataka 560066  
India  
Email: dhruv.ietf@gmail.com

Shankara  
Huawei Technologies  
Divyashree Techno Park, Whitefield  
Bangalore, Karnataka 560066  
India  
Email: shankara@huawei.com

#### Appendix B. Example Usage

Once PCE initiate tunnels, it needs to further decide what data needs to flow on the newly created tunnel, a flow specification can be created at the ingress to redirect the flow to the LSP as shown below.



Authors' Addresses

Zhenbin Li  
Huawei Technologies  
Huawei Bld., No.156 Beiqing Rd.  
Beijing 100095  
China

Email: lizhenbin@huawei.com

Xia Chen  
Huawei Technologies  
Huawei Bld., No.156 Beiqing Rd.  
Beijing 100095  
China

Email: jescia.chenxia@huawei.com

Shunwan Zhuang  
Huawei Technologies  
Huawei Bld., No.156 Beiqing Rd.  
Beijing 100095  
China

Email: zhuangshunwan@huawei.com

Network Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: July 2, 2018

D. Dhody, Ed.  
Huawei Technologies  
A. Farrel, Ed.  
Juniper Networks  
Z. Li  
Huawei Technologies  
December 29, 2017

PCEP Extension for Flow Specification  
draft-li-pce-pcep-flowspec-03

Abstract

The Path Computation Element (PCE) is a functional component capable of selecting the paths through a traffic engineered network. These paths may be supplied in response to requests for computation, or may be unsolicited directions issued by the PCE to network elements. Both approaches use the PCE Communication Protocol (PCEP) to convey the details of the computed path.

Traffic flows may be categorized and described using "Flow Specifications". RFC 5575 defines the Flow Specification and describes how it may be distributed in BGP to allow specific traffic flows to be associated with routes.

This document specifies a set of extensions to PCEP to support dissemination of Flow Specifications. This allows a PCE to indicate what traffic should be placed on each path that it is aware of.

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 July 2, 2018.

#### Copyright Notice

Copyright (c) 2017 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. Terminology . . . . .	4
3. Procedures for PCE Use of Flow Specifications . . . . .	4
3.1. Capability Advertisement . . . . .	5
3.1.1. PCEP OPEN Message . . . . .	5
3.1.2. IGP PCE Capabilities Advertisement . . . . .	5
3.2. Dissemination Procedures . . . . .	6
3.3. Flow Specification Synchronization . . . . .	7
4. PCE FlowSpec Capability TLV . . . . .	7
5. PCEP Flow Spec Object . . . . .	8
6. Flow Filter TLV . . . . .	9
7. Flow Specification TLVs . . . . .	9
8. Detailed Procedures . . . . .	12
8.1. Default Behavior and Backward Compatibility . . . . .	13
8.2. Composite Flow Specifications . . . . .	13
8.3. Modifying Flow Specifications . . . . .	13
8.4. Multiple Flow Specifications . . . . .	13
8.5. Adding and Removing Flow Specifications . . . . .	14
8.6. VPN Identifiers . . . . .	14
8.7. Priorities and Overlapping Flow Specifications . . . . .	14
9. PCEP Messages . . . . .	15
10. IANA Considerations . . . . .	18
10.1. PCEP Objects . . . . .	18
10.2. PCEP TLV Type Indicators . . . . .	18

10.3. Flow Specification TLV Type Indicators . . . . .	18
10.4. PCEP Error Codes . . . . .	19
10.5. PCE Capability Flag . . . . .	19
11. Security Considerations . . . . .	20
12. Manageability Considerations . . . . .	20
13. Acknowledgements . . . . .	21
14. References . . . . .	21
14.1. Normative References . . . . .	21
14.2. Informative References . . . . .	22
Appendix A. Contributors . . . . .	23
Authors' Addresses . . . . .	24

## 1. Introduction

[RFC4655] defines the Path Computation Element (PCE), a functional component capable of computing paths for use in traffic engineering networks. PCE was originally conceived for use in Multiprotocol Label Switching (MPLS) for Traffic Engineering (TE) networks to derive the routes of Label Switched Paths (LSPs). However, the scope of PCE was quickly extended to make it applicable to Generalized MPLS (GMPLS) networks, and more recent work has brought other traffic engineering technologies and planning applications into scope (for example, Segment Routing (SR) [I-D.ietf-pce-segment-routing]).

[RFC5440] describes the Path Computation Element Communication Protocol (PCEP). PCEP defines the communication between a Path Computation Client (PCC) and a PCE, or between PCE and PCE, enabling computation of path for MPLS-TE LSPs.

Stateful PCE [RFC8231] specifies a set of extensions to PCEP to enable control of TE-LSPs by a PCE that retains state about the the LSPs provisioned in the network (a stateful PCE). [RFC8281] describes the setup, maintenance, and teardown of LSPs initiated by a stateful PCE without the need for local configuration on the PCC, thus allowing for a dynamic network that is centrally controlled. [RFC8283] introduces the architecture for PCE as a central controller and describes how PCE can be viewed as a component that performs computation to place 'flows' within the network and decide how these flows are routed.

Dissemination of traffic flow specifications (Flow Specifications) was introduced for BGP in [RFC5575]. A Flow Specification is comprised of traffic filtering rules and actions. The routers that receive a Flow Specification can classify received packets according to the traffic filtering rules and can direct packets based on the actions.

When a PCE is used to initiate tunnels (such as TE-LSPs or SR paths) using PCEP, it is important that the head end of the tunnels understands what traffic to place on each tunnel. The data flows intended for a tunnel can be described using Flow Specifications, and when PCEP is in use for tunnel initiation it makes sense for that same protocol to be used to distribute the Flow Specifications that describe what data is to flow on those tunnels.

This document specifies a set of extensions to PCEP to support dissemination of Flow Specifications. The extensions include the creation, update, and withdrawal of Flow Specifications via PCEP and can be applied to tunnels initiated by the PCE or to tunnels where control is delegated to the PCE by the PCC. Furthermore, a PCC requesting a new path can include Flow Specifications in the request to indicate the purpose of the tunnel allowing the PCE to factor this in during the path computation.

Flow Specifications are carried in TLVs within a new Flow Spec Object defined in this document. The flow filtering rules indicated by the Flow Specifications are mainly defined by BGP Flow Specifications.

## 2. Terminology

This document uses the following terms defined in [RFC5440]: PCC, PCE, PCEP Peer.

The following term from [RFC5575] is used frequently throughout this document:

Flow Specification (FlowSpec): A Flow Specification is an n-tuple consisting of several matching criteria that can be applied to IP traffic, including filters and actions. Each FlowSpec consists of a set of filters and a set of actions.

This document uses the terms "stateful PCE" and "active PCE" as advocated in [RFC7399].

## 3. Procedures for PCE Use of Flow Specifications

There are three elements of procedure:

- o A PCE and a PCC must be able to indicate whether or not they support the use of Flow Specifications.
- o A PCE or PCC must be able to include Flow Specifications in PCEP messages with clear understanding of the applicability of those Flow Specifications in each case including whether the use of such



information is mandatory, constrained, or optional, and how overlapping Flow Specifications will be resolved..

- o Flow Specification information/state must be synchronized between PCEP peers so that, on recovery, the peers have the same understanding of which Flow Specifications apply.

The following subsections describe these points.

### 3.1. Capability Advertisement

#### 3.1.1. PCEP OPEN Message

During PCEP session establishment, a PCC or PCE that supports the procedures described in this document announces this fact by including the "PCE FlowSpec Capability" TLV (described in Section 4) in the OPEN Object carried in the PCEP Open message.

The presence of the PCE FlowSpec Capability TLV in the OPEN Object in a PCE's OPEN message indicates that the PCE can support distribute the FlowSpec to PCCs and can receive FlowSpecs in messages from the PCCs.

The presence of the PCE FlowSpec Capability TLV in the OPEN Object in a PCC's OPEN message indicates that the PCC supports the FlowSpec functionality described in this document.

If either one of a pair of PCEP peers does not indicate support of the functionality described in this document by not including the PCE FlowSpec Capability TLV in the OPEN Object in its OPEN message, then the other peer MUST NOT include a FlowSpec object in any PCEP message sent to the peer that does not support the procedures. If a FlowSpec object is received even though support has not been indicated, the receiver will respond with a PCerr message reporting the objects containing the FlowSpec as described in [RFC5440]: that is, it will use 'Unknown Object' if it does not support this specification, and 'Not supported object' if it supports this specification but has not chosen to support FlowSpec objects on this PCEP session.

#### 3.1.2. IGP PCE Capabilities Advertisement

The ability to advertise support for PCEP and PCE features in IGP advertisements is provided for OSPF in [RFC5088] and for IS-IS in [RFC5089]. The mechanism uses the PCE Discovery TLV which has a PCE-CAP-FLAGS sub-TLV containing bit-flags each of which indicates support for a different feature.

This document defines a new PCE-CAP-FLAGS sub-TLV bit, the FlowSpec Capable flag (bit number TBD1). Setting the bit indicates that an advertising PCE supports the procedures defined in this document.

Note that while PCE FlowSpec Capability may be advertised during discovery, PCEP speakers that wish to use Flow Specification in PCEP MUST negotiate PCE FlowSpec Capability during PCEP session setup, as specified in Section 3.1.1. A PCC MAY initiate PCE FlowSpec Capability negotiation at PCEP session setup even if it did not receive any IGP PCE capability advertisement.

### 3.2. Dissemination Procedures

This section describes the procedures to support Flow Specifications in PCEP messages.

The primary purpose of distributing Flow Specification information is to allow a PCE to indicate to a PCC what traffic it should place on a path (such as an LSP or an SR path). This means that the Flow Specification may be included in:

- o PCInitiate messages so that an active PCE can indicate the traffic to place on a path at the time that the PCE instantiates the path.
- o PCUpd messages so that an active PCE can indicate or change the traffic to place on a path that has already been set up.
- o PCRpt messages so that a PCC can report the traffic that the PCC plans to place on the path.
- o PCReq messages so that a PCC can indicate what traffic it plans to place on a path at the time it requests the PCE to perform a computation in case that information aids the PCE in its work.
- o PCRep messages so that a PCE that has been asked to compute a path can suggest which traffic could be placed on a path that a PCC may be about to set up.
- o PCErr messages so that issues related to paths and the traffic they carry can be reported to the PCE by the PCC, and so that problems with other PCEP messages that carry Flow Specifications can be reported.

To carry Flow Specifications in PCEP messages, this document defines a new PCEP object called the PCEP Flow Spec Object. The object is OPTIONAL in the messages described above and MAY appear more than once in each message.

The PCEP Flow Spec Object carries zero or one Flow Filter TLV which describes a traffic flow.

The inclusion of multiple PCEP Flow Spec Objects allows multiple traffic flows to be placed on a single path.

Once a PCE and PCC have established that they can both support the use of Flow Specifications in PCEP messages, such information may be exchanged at any time for new or existing paths.

The application and prioritization of Flow Specifications is described in Section 8.7.

### 3.3. Flow Specification Synchronization

The Flow Specifications are carried along with the LSP State information as per [RFC8231] making the Flow Specifications part of the LSP database (LSP-DB). Thus, the synchronization of the Flow Specification information is done as part of LSP-DB synchronization. This may be achieved using normal state synchronization procedures as described in [RFC8231] or enhanced state synchronization procedures as defined in [RFC8232].

The approach selected will be implementation and deployment specific and will depend on issues such as how the databases are constructed and what level of synchronization support is needed.

## 4. PCE FlowSpec Capability TLV

The PCE-FLOWSPEC-CAPABILITY TLV is an optional TLV that can be carried in the OPEN Object [RFC5440] to exchange PCE FlowSpec capabilities of PCEP speakers.

The format of the PCE-FLOWSPEC-CAPABILITY TLV follows the format of all PCEP TLVs as defined in [RFC5440] and is shown in Figure 1.

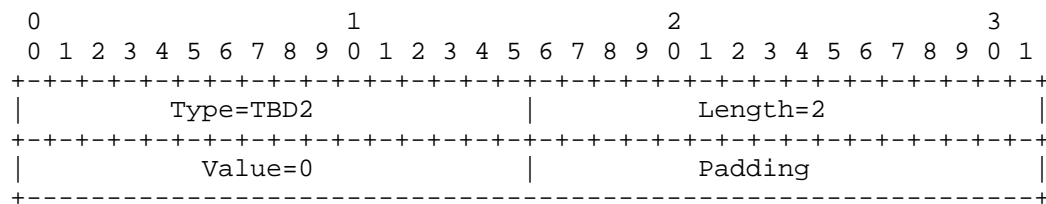


Figure 1: PCE-FLOWSPEC-CAPABILITY TLV format

The type of the PCE-FLOWSPEC-CAPABILITY TLV is TBD2 and it has a fixed length of 2 octets. The Value field is set to default value 0. The two bytes of padding MUST be set to zero and ignored on receipt.

The inclusion of this TLV in an OPEN object indicates that the sender can perform FlowSpec handling as defined in this document.

## 5. PCEP Flow Spec Object

The PCEP Flow Spec object defined in this document is compliant with the PCEP object format defined in [RFC5440]. It is OPTIONAL in the PCReq, PCRep, PCErr, PCInitiate, PCRpt, and PCUpd messages and MAY be present zero, one, or more times. Each instance of the object specifies a traffic flow.

The PCEP Flow Spec object carries a FlowSpec filter rule encoded in a TLV (as defined in Section 6).

The FLOW SPEC Object-Class is TBD3 (to be assigned by IANA).

The FLOW SPEC Object-Type is 1.

The format of the body of the PCEP Flow Spec object is shown in Figure 2

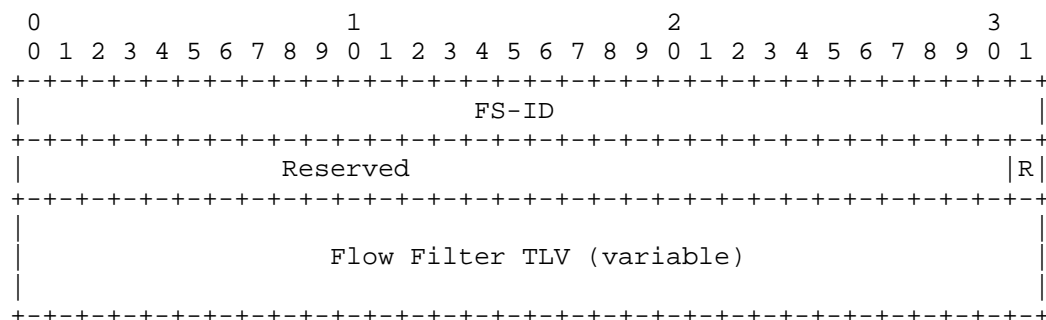


Figure 2: PCEP Flow Spec Object Body Format

FS-ID (32-bits): A PCEP-specific identifier for the FlowSpec information. A PCE creates an FS-ID for each FlowSpec, the value is unique within the scope of the PCE and is constant for the lifetime of a PCEP session. All subsequent PCEP messages can identify the FlowSpec using the FS-ID. The values 0 and 0xFFFFFFFF are reserved and MUST NOT be used.

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

R bit: The Remove bit is set when a PCEP Flow Spec Object is included in a PCEP message to indicate removal of the Flow Specification from the associated tunnel. If the bit is clear, the Flow Specification is being added or modified.

Flow Filter TLV (variable): One TLV MAY be included.

The Flow Filter TLV is OPTIONAL when the R bit is set. The TLV MUST be present when the R bit is clear. If the TLV is missing when the R bit is clear, the PCEP peer MUST respond with a PCErr message with error-type TBD8 (FlowSpec Error), error-value 2 (Malformed FlowSpec).

## 6. Flow Filter TLV

A new PCEP TLV is defined to convey Flow Specification filtering rules that specify what traffic is carried on a path. The TLV follows the format of all PCEP TLVs as defined in [RFC5440]. The Type field values come from the codepoint space for PCEP TLVs and has the value TBD4.

The Value field contains one or more sub-TLVs (the Flow Specification TLVs) as defined in Section 7. Only one Flow Filter TLV can be present and represents the complete definition of a Flow Specification for traffic to be placed on the tunnel indicated by the PCEP message in which the PCEP Flow Spec Object is carried. The set of Flow Specification TLVs in a single instance of a Flow Filter TLV are combined to indicate the specific Flow Specification.

Further Flow Specifications can be included in a PCEP message by including additional Flow Spec objects.

## 7. Flow Specification TLVs

Flow Filter TLV carries one or more Flow Specification sub-TLV. The Flow Specification TLV also follows the format of all PCEP TLVs as defined in [RFC5440], however, the Type values are selected from a separate IANA registry (see Section 10) rather than from the common PCEP TLV registry.

Type values are chosen so that there can be commonality with Flow Specifications defined for use with BGP. This is possible because the BGP Flow Spec encoding uses a single octet to encode the type where PCEP uses two octets. Thus the space of values for the Type field is partitioned as shown in Figure 3.

Range	
0	Reserved - must not be allocated.
1 .. 255	Per BGP registry defined by [RFC5575]. Not to be allocated in this registry.
256 .. 65535	New PCEP Flow Specs allocated according to the registry defined in this document.

Figure 3: Flow Specification TLV Type Ranges

The content of the Value field Flow in each TLV is specific to the type and describes the parameters of the Flow Specification. The definition of the format of many of these Value fields is inherited from BGP specifications as shown in Figure 4. Specifically, the inheritance is from [RFC5575] and [I-D.ietf-idr-flow-spec-v6], but may also be inherited from future BGP specifications.

When multiple Flow Specification TLVs are present in a single Flow Filter TLV they are combined to produce a more detailed description of a flow. For examples and rules about how this is achieved, see [RFC5575].

An implementation that receives a PCEP message carrying a Flow Specification TLV with a type value that it does not recognize or does not support MUST respond with a PCErr message with error-type TBD8 (FlowSpec Error), error-value 1 (Unsupported FlowSpec) and MUST NOT install the Flow Specification.

When used in other protocols (such as BGP) these Flow Specifications are also associated with actions to indicate how traffic matching the Flow Specification should be treated. In PCEP, however, the only action is to associate the traffic with a tunnel and to forward matching traffic on to that path, so no encoding of an action is needed.

Section 8.7 describes how overlapping Flow Specifications are prioritized and handled.

Type	Description	Value defined in
*	Destination IPv4 Prefix	[RFC5575]
*	Source IPv4 Prefix	[RFC5575]
*	IP Protocol	[RFC5575]
*	Port	[RFC5575]
*	Destination port	[RFC5575]
*	Source port	[RFC5575]
*	ICMP type	[RFC5575]
*	ICMP code	[RFC5575]
*	TCP flags	[RFC5575]
*	Packet length	[RFC5575]
*	DSCP	[RFC5575]
*	Fragment	[RFC5575]
*	Flow Label	[I-D.ietf-idr-flow-spec-v6]
*	Destination IPv6 Prefix	[I-D.ietf-idr-flow-spec-v6]
*	Source IPv6 Prefix	[I-D.ietf-idr-flow-spec-v6]
*	Next Header	[I-D.ietf-idr-flow-spec-v6]
TBD5	Route Distinguisher	[I-D.dhodylee-pce-pcep-ls]
TBD6	IPv4 Multicast Flow	[This.I-D]
TBD7	IPv6 Multicast Flow	[This.I-D]

\* Indicates that the TLV Type value comes from the value used in BGP.

Figure 4: Table of Flow Specification TLV Types

All Flow Specification TLVs with Types in the range 1 to 255 have Values defined for use in BGP (for example in [RFC5575] and [I-D.ietf-idr-flow-spec-v6]) and are set using the BGP encoding, but without the type or length octets (the relevant information is in the Type and Length fields of the TLV). The Value field is padded with trailing zeros to achieve 4-byte alignment if necessary.

[I-D.dhodylee-pce-pcep-ls] defines a way to convey identification of a VPN in PCEP via a Route Distinguisher (RD) [RFC4364] and encoded in ROUTE-DISTINGUISHER TLV. A Flow Specification TLV with Type TBD5 carries a Value field matching that in the ROUTE-DISTINGUISHER TLV and is used to identify that other flow filter information (for example, an IPv4 destination prefix) is associated with a specific VPN identified by the RD. See Section 8.6 for further discussion of VPN identification.

Although it may be possible to describe a multicast Flow Specification from the combination of other Flow Specification TLVs with specific values, it is more convenient to use a dedicated Flow Specification TLV. Flow Specification TLVs with Type values TBD6 and TBD7 are used to identify a multicast flow for IPv4 and IPv6 respectively. The Value field is encoded as shown in Figure 5.

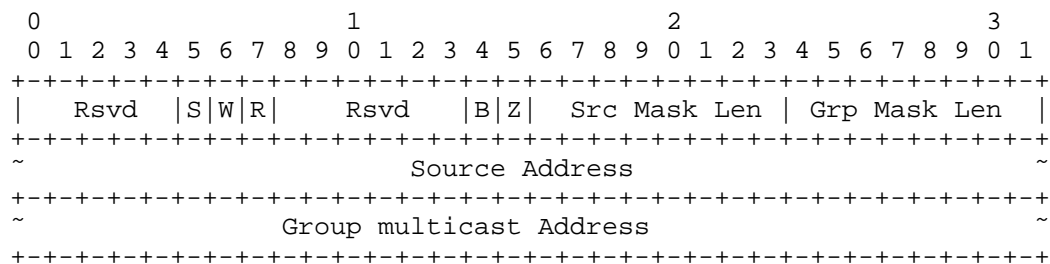


Figure 5: Multicast Flow Specification TLV Encoding

The fields of the two Multicast Flow Specification TLVs are as described in Section 4.9.1 of [RFC7761] noting that the two address fields are 32 bits for the IPv4 Multicast Flow and 128 bits for the IPv6 Multicast Flow. Reserved fields MUST be set to zero and ignored on receipt.

## 8. Detailed Procedures

This section outlines some specific detailed procedures for using the protocol extensions defined in this document.



### 8.1. Default Behavior and Backward Compatibility

The default behavior is that no Flow Specification is applied to a tunnel. That is, the default is that the Flow Spec object is not used as is the case in all systems before the implementation of this specification.

In this case it is a local matter (such as through configuration) how tunnel head ends are instructed what traffic to place on a tunnel.

[RFC5440]describes how receivers respond when they see unknown PCEP objects.

### 8.2. Composite Flow Specifications

Flow Specifications may be represented by a single Flow Specification TLV or may require a more complex description using multiple Flow Specification TLVs. For example, a flow indicated by a source-destination pair of IPv6 addresses would be described by the combination of Destination IPv6 Prefix and Source IPv6 Prefix Flow Specification TLVs.

### 8.3. Modifying Flow Specifications

A PCE may want to modify a Flow Specification associated with a tunnel, or a PCC may want to report a change to the Flow Specification it is using with a tunnel.

It is important that the specific Flow Specification is identified so that it is clear that this is a modification of an existing flow and not the addition of a new flow as described in Section 8.4. The FS-ID field of the PCEP Flow Spec Object is used to identify a specific Flow Specification.

When modifying a Flow Specification, all Flow Specification TLVs for the intended specification of the flow MUST be included in the PCEP Flow Spec Object and the FS-ID MUST be retained from the previous description of the flow.

### 8.4. Multiple Flow Specifications

It is possible that multiple flows will be place on a single tunnel. In some cases it is possible to to define these within a single PCEP Flow Spec Object: for example, two Destination IPv4 Prefix TLVs could be included to indicate that packets matching either prefix are acceptable. PCEP would consider this as a single Flow Specification identified by a single FS-ID.

In other scenarios the use of multiple Flow Specification TLVs would be confusing. For example, if flows from A to B and from C to D are to be included then using two Source IPv4 Prefix TLVs and two Destination IPv4 Prefix TLVs would be confusing (are flows from A to D included?). In these cases, each Flow Specification is carried in its own PCEP Flow Spec Object with multiple objects present on a single PCEP message. Use of separate objects also allows easier removal and modification of Flow Specifications.

#### 8.5. Adding and Removing Flow Specifications

The Remove bit in the the PCEP Flow Spec Object is left clear when a Flow Specification is being added or modified.

To remove a Flow Specification, a PCEP Flow Spec Object is included with the FS-ID matching the one being removed, and the R bit set to indicate removal. In this case it is not necessary to include any Flow Specification TLVs.

If the R bit is set and Flow Specification TLVs are present an implementation MAY ignore them. If the implementation checks the Flow Specification TLVs against those recorded for the FS-ID of the Flow Specification being removed and finds a mismatch, the Flow Specification MUST still be removed and the implementation SHOULD record a local exception or log.

#### 8.6. VPN Identifiers

VPN instances are identified in BGP using Route Distinguishers (RDs) [RFC4364]. These values are not normally considered to have any meaning outside of the network, and they are not encoded in data packets belonging to the VPNs. However, RDs provide a useful way of identifying VPN instances and are often manually or automatically assigned to VPNs as they are provisioned.

Thus the RD provides a useful way to indicate that traffic for a particular VPN should be placed on a given tunnel. The tunnel head end will need to interpret this Flow Specification not as a filter on the fields of data packets, but using the other mechanisms that it uses to identify VPN traffic. This could be based on the incoming port (for port-based VPNs) or may leverage knowledge of the VRF that is in use for the traffic.

#### 8.7. Priorities and Overlapping Flow Specifications

TBD

An implementation that receives a PCEP message carrying a Flow Specification that it cannot resolve against other Flow Specifications already installed MUST respond with a PCErr message with error-type TBD8 (FlowSpec Error), error-value 3 (Unresolvable conflict) and MUST NOT install the Flow Specification.

## 9. PCEP Messages

The figures below use the notation defined in [RFC5511].

The FLOW SPEC Object is OPTIONAL and MAY be carried in the PCEP messages.

The PCInitiate message is defined in [RFC8281] and updated as below:

```
<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>]
                                       [<flowspec-list>]
```

Where:

```
<flowspec-list> ::= <FLOWSPEC> [<flowspec-list>]
```

The PCUpd message is defined in [RFC8231] and updated as below:

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

Where:

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

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

Where:

$$\langle \text{path} \rangle ::= \langle \text{intended-path} \rangle \langle \text{intended-attribute-list} \rangle$$
$$\langle \text{flowspec-list} \rangle ::= \langle \text{FLOWSPEC} \rangle [\langle \text{flowspec-list} \rangle]$$

The PCRpt message is defined in [RFC8231] and updated as below:

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

Where:

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

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

Where:

```
<path> ::= <intended-path>
           [<actual-attribute-list><actual-path>]
           <intended-attribute-list>
```

$$\langle \text{flowspec-list} \rangle ::= \langle \text{FLOWSPEC} \rangle [\langle \text{flowspec-list} \rangle]$$

The PCReq message is defined in [RFC5440] and updated in [RFC8231], it is further updated below for flow specification:

```
<PCReq Message> ::= <Common Header>
                        [<svec-list>]
                        <request-list>
```

Where:

```
<svec-list> ::= <SVEC> [<svec-list>]

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

<request> ::= <RP>
               <END-POINTS>
               [<LSP>]
               [<LSPA>]
               [<BANDWIDTH>]
               [<metric-list>]
               [<RRO> [<BANDWIDTH>]]
               [<IRO>]
               [<LOAD-BALANCING>]
               [<flowspec-list>]
```

Where:

```
<flowspec-list> ::= <FLOWSPEC> [<flowspec-list>]
```

The PCRep message is defined in [RFC5440] and updated in [RFC8231], it is further updated below for flow specification:

```
<PCRep Message> ::= <Common Header>
                     <response-list>
```

Where:

```
<response-list> ::= <response> [<response-list>]

<response> ::= <RP>
               [<LSP>]
               [<NO-PATH>]
               [<attribute-list>]
               [<path-list>]
               [<flowspec-list>]
```

Where:

```
<flowspec-list> ::= <FLOWSPEC> [<flowspec-list>]
```

## 10. IANA Considerations

IANA maintains the "Path Computation Element Protocol (PCEP) Numbers" registry. This document requests IANA actions to allocate code points for the protocol elements defined in this document.

### 10.1. PCEP Objects

Each PCEP object has an Object-Class and an Object-Type. IANA maintains a subregistry called "PCEP Objects". IANA is requested to make an assignment from this subregistry as follows:

Object-Class	Value Name	Object-Type	Reference
TBD3	FLOW SPEC	0 (Reserved)	[This.I-D]
		1	[This.I-D]

### 10.2. PCEP TLV Type Indicators

IANA maintains a subregistry called "PCEP TLV Type Indicators". IANA is requested to make an assignment from this subregistry as follows:

Value	Meaning	Reference
TBD2	PCE-FLOWSPEC-CAPABILITY TLV	[This.I-D]
TBD4	FLOW FILTER TLV	[This.I-D]

### 10.3. Flow Specification TLV Type Indicators

IANA is requested to create a new subregistry call the PCEP Flow Specification TLV Type Indicators registry.

Allocations from this registry are to be made according to the following assignment policies [RFC8126]:

Range	Assignment policy
0	Reserved - must not be allocated.
1 .. 255	Reserved - must not be allocated. Usage mirrors the BGP FlowSpec registry [RFC5575].
258 .. 64506	Specification Required
64507 .. 65531	First Come First Served
65532 .. 65535	Experimental

IANA is requested to pre-populate this registry with values defined in this document as follows:

Value	Meaning
TBD5	Route Distinguisher
TBD6	IPv4 Multicast
TBD7	IPv6 Multicast

#### 10.4. PCEP Error Codes

IANA maintains a subregistry called "PCEP-ERROR Object Error Types and Values". Entries in this subregistry are described by Error-Type and Error-value. IANA is requested to make the following assignment from this subregistry:

Error-Type	Meaning	Error-value	Reference
TBD8	FlowSpec error	0: Unassigned	[This.I-D]
		1: Unsupported FlowSpec	[This.I-D]
		2: Malformed FlowSpec	[This.I-D]
		3: Unresolvable conflict	[This.I-D]
		4-255: Unassigned	[This.I-D]

#### 10.5. PCE Capability Flag

IANA maintains a subregistry called "Open Shortest Path First v2 (OSPFv2) Parameters" with a sub-registry called "Path Computation

Element (PCE) Capability Flags". IANA is requested to assign a new capability bit from this registry as follows:

Bit	Capability Description	Reference
TBD1	FlowSpec	[This.I-D]

## 11. Security Considerations

We may assume that a system that utilizes a remote PCE is subject to a number of vulnerabilities that could allow spurious LSPs or SR paths to be established or that could result in existing paths being modified or torn down. Such systems, therefore, apply security considerations as described in [RFC5440], [RFC6952], and [RFC8253].

The description of Flow Specifications associated with paths set up or controlled by a PCE add an further detail that could be attacked without tearing down LSPs or SR paths but causing traffic to be misrouted within the network. Therefore, the use of the security mechanisms for PCEP referenced above is important.

Visibility into the information carried in PCEP does not have direct privacy concerns for end-users' data, however, knowledge of how data is routed in a network may make that data more vulnerable. Of course, the ability to interfere with the way data s routed also makes the data more vulnerable. Furthermore, knowledge of the connected end-points (such as multicast receivers or VPN sites) is usually considered private customer information. Therefore, implementations or deployments concerned to protect privacy MUST apply the mechanisms described in the documents referenced above.

Experience with Flow Specifications in BGP systems indicates that they can become complex and that the overlap of Flow Specifications installed in different orders can lead to unexpected results. Although this is not directly a security issue per se, the confusion and unexpected forwarding behavior may be engineered or exploited by an attacker. Therefore, implementers and operators SHOULD pay careful attention to the Manageability Considerations described in Section 12.

## 12. Manageability Considerations

TBD



### 13. Acknowledgements

Thanks to Julian Lucek and Sudhir Cheruathur for useful discussions.

### 14. References

#### 14.1. Normative References

- [I-D.dhodylee-pce-pcep-ls]  
Dhody, D., Lee, Y., and D. Ceccarelli, "PCEP Extension for Distribution of Link-State and TE Information.", draft-dhodylee-pce-pcep-ls-08 (work in progress), June 2017.
- [I-D.ietf-idr-flow-spec-v6]  
McPherson, D., Raszuk, R., Pithawala, B., akarch@cisco.com, a., and S. Hares, "Dissemination of Flow Specification Rules for IPv6", draft-ietf-idr-flow-spec-v6-09 (work in progress), November 2017.
- [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>>.
- [RFC5511] Farrel, A., "Routing Backus-Naur Form (RBNF): A Syntax Used to Form Encoding Rules in Various Routing Protocol Specifications", RFC 5511, DOI 10.17487/RFC5511, April 2009, <<https://www.rfc-editor.org/info/rfc5511>>.
- [RFC5575] Marques, P., Sheth, N., Raszuk, R., Greene, B., Mauch, J., and D. McPherson, "Dissemination of Flow Specification Rules", RFC 5575, DOI 10.17487/RFC5575, August 2009, <<https://www.rfc-editor.org/info/rfc5575>>.
- [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>>.
- [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>>.

## 14.2. Informative References

- [I-D.ietf-pce-segment-routing]  
Sivabalan, S., Filsfils, C., Tantsura, J., Henderickx, W.,  
and J. Hardwick, "PCEP Extensions for Segment Routing",  
draft-ietf-pce-segment-routing-11 (work in progress),  
November 2017.
- [RFC4364] Rosen, E. and Y. Rekhter, "BGP/MPLS IP Virtual Private  
Networks (VPNs)", RFC 4364, DOI 10.17487/RFC4364, February  
2006, <<https://www.rfc-editor.org/info/rfc4364>>.
- [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>>.
- [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>>.
- [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>>.
- [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>>.
- [RFC7761] Fenner, B., Handley, M., Holbrook, H., Kouvelas, I.,  
Parekh, R., Zhang, Z., and L. Zheng, "Protocol Independent  
Multicast - Sparse Mode (PIM-SM): Protocol Specification  
(Revised)", STD 83, RFC 7761, DOI 10.17487/RFC7761, March  
2016, <<https://www.rfc-editor.org/info/rfc7761>>.
- [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>>.
- [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>>.
- [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>>.
- [RFC8283] Farrel, A., Ed., Zhao, Q., Ed., Li, Z., and C. Zhou, "An Architecture for Use of PCE and the PCE Communication Protocol (PCEP) in a Network with Central Control", RFC 8283, DOI 10.17487/RFC8283, December 2017, <<https://www.rfc-editor.org/info/rfc8283>>.

#### Appendix A. Contributors

Shankara  
Huawei Technologies  
Divyashree Techno Park,  
Whitefield Bangalore,  
Karnataka  
560066  
India

Email: [shankara@huawei.com](mailto:shankara@huawei.com)

Qiandeng Liang  
Huawei Technologies  
101 Software Avenue,  
Yuhuatai District  
Nanjing  
210012  
China

Email: [liangqiandeng@huawei.com](mailto:liangqiandeng@huawei.com)

Cyril Margaria

Juniper Networks  
200 Somerset Corporate Boulevard, Suite 4001  
Bridgewater, NJ  
08807  
USA

Email: cmargaria@juniper.net

Colby Barth  
Juniper Networks  
200 Somerset Corporate Boulevard, Suite 4001  
Bridgewater, NJ  
08807  
USA

Email: cbarth@juniper.net

Xia Chen  
Huawei Technologies  
Huawei Bld., No.156 Beiqing Rd.  
Beijing  
100095  
China

Email: jescia.chenxia@huawei.com

Shunwan Zhuang  
Huawei Technologies  
Huawei Bld., No.156 Beiqing Rd.  
Beijing  
100095  
China

Email: zhuangshunwan@huawei.com

#### Authors' Addresses

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

Email: dhruv.ietf@gmail.com

Adrian Farrel (editor)  
Juniper Networks

Email: [afarrel@juniper.net](mailto:afarrel@juniper.net)

Zhenbin Li  
Huawei Technologies  
Huawei Bld., No.156 Beiqing Rd.  
Beijing 100095  
China

Email: [lizhenbin@huawei.com](mailto:lizhenbin@huawei.com)



Network Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: September 15, 2016

Z. Li  
X. Chen  
Huawei Technologies  
March 14, 2016

PCEP Extensions for Tunnel Segment  
draft-li-pce-tunnel-segment-01

Abstract

[I-D.li-spring-tunnel-segment] introduces a new type of segment, Tunnel Segment, for the segment routing. Tunnel segment can be used to reduce SID stack depth of SR path, span the non-SR domain or provide differentiated services. A binding label can be assigned to tunnel segment. An upstream node can use such a binding label for steering traffic into the appropriate tunnel. This document specifies a set of extensions to PCEP to support that PCC reports binding label of tunnel to PCE and that PCE allocates label for tunnel and updates label binding of tunnel to PCC.

Requirements Language

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

Status of This Memo

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

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

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

This Internet-Draft will expire on September 15, 2016.

## Copyright Notice

Copyright (c) 2016 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 . . . . .	2
2. Terminology . . . . .	3
3. Procedures . . . . .	3
3.1. Procedure for PCC Reporting Label Binding . . . . .	3
3.2. Procedure for PCE Download Label Binding . . . . .	5
4. Objects . . . . .	6
4.1. TE object . . . . .	6
5. TLVs . . . . .	6
5.1. Tunnel Label Binding TLV . . . . .	6
5.2. PATH-SETUP-TYPE TLV . . . . .	7
5.3. Tunnel Related TLV . . . . .	7
6. IANA Considerations . . . . .	7
7. Security Considerations . . . . .	7
8. References . . . . .	7
8.1. Normative References . . . . .	7
8.2. Informative References . . . . .	8
Authors' Addresses . . . . .	9

## 1. Introduction

[I-D.li-spring-tunnel-segment] introduces a new type of segment, Tunnel Segment, for the segment routing. Tunnel segment can be used to reduce SID stack depth of SR path, span the non-SR domain or provide differentiated services. A binding label can be assigned to tunnel segment. An upstream node can use such a binding label for steering traffic into the appropriate tunnel. The tunnel segment can be allocated for RSVP-TE tunnel, SR-TE tunnel or IP tunnel.

[I-D.li-spring-tunnel-segment] defines the requirement of control plane to support use cases of tunnel segment. The PCE related requirements are as follows:



-- PCEP extensions SHOULD be introduced to advertise the binding relationship between a SID/label and the corresponding tunnel from a PCC to a PCE. Attributes of the tunnel MAY be carried optionally.

-- PCE SHOULD support to allocate SID/label for the corresponding tunnel dynamically.

-- PCEP extensions SHOULD be introduced to distribute the binding relationship between a SID/label and the corresponding tunnel from a PCE to a PCC. Attributes of the tunnel MAY be carried optionally.

This document specifies the protocol extensions to PCEP to support these requirements defined in [I-D.li-spring-tunnel-segment].

## 2. Terminology

SR: Segment Routing

SR-TE: Segment Routing Traffic Engineering

SR-TE Tunnel: Segment Routing Traffic Engineering Tunnel

This document uses the terms defined in [RFC5440]: PCC, PCE, PCEP Peer.

The following terms are defined in [I-D.ietf-pce-pce-initiated-lsp]:

PCE-initiated LSP: LSP that is instantiated as a result of a request from the PCE.

The following terms are defined in [I-D.chen-pce-pce-initiated-ip-tunnel]:

IP Tunnel: Tunnel that uses IP encapsulation.

PCE-initiated IP Tunnel: IP Tunnel that is instantiated as a result of a request from the PCE.

## 3. Procedures

### 3.1. Procedure for PCC Reporting Label Binding

In the procedure for PCC reporting the label binding PCC allocates the label and reports the label binding for the tunnel according to the local policy PCC. For report the label binding information, there are following options:

Option 1: [I-D.zhao-pce-pcep-extension-for-pce-controller] specifies the procedures and PCEP protocol extensions for using the PCE as the central controller where LSPs are calculated/setup/initiated and label forwarding entries are downloaded to PCC. It introduces the LABEL object to specify the label binding information in PCLabelUpd message. The label carried in LABEL object is mapped to specific LSP carried in LSP object or FEC carried in FEC object. The LABEL object defined in the document is to allocate label from the PCE to PCC and is not appropriate to represent the label binding for the tunnel which can be carried in other PCE objects.

Option 2: [I-D.sivabalan-pce-binding-label-sid] proposes an approach for reporting binding label/SID to Path Computation Element (PCE) for supporting PCE-based Traffic Engineering policies. It introduces the optional TLV called "TE-PATH-BINDING TLV" to carry binding label or SID for a TE path. This TLV is limited to traffic engineering and not appropriate for tunnel segment.

Option 3: PCEP-LS [I-D.dhodylee-pce-pcep-te-data-extn] extends the Path Computation Element Communication Protocol (PCEP) with TED population capability. A PCEP TE Report message (also referred to as TERpt message) is sent by a PCC to a PCE to report the TED state. The TE object is a mandatory object which carries TE information of a TE node or a TE link. [I-D.wu-pce-pcep-ls-sr-extension] introduces new extensions of PCEP-LS to export path segment information for Segment Routing.

This document adopts Option 3 and introduces a new type of TLV, TUNNEL-LABEL-BINDING TLV, which is a new optional TLV defined to report the label mapping for the tunnel in the case of Segment Routing. The tunnel can be PCE-initiated tunnel or created by PCC. [I-D.chen-pce-pce-initiated-ip-tunnel] defines the PCE-initiated IP tunnel and Tunnel object. Tunnel related TLVs defined in [I-D.chen-pce-pce-initiated-ip-tunnel] will be used when report label binding for the tunnel. In order to support Tunnel Segment for MPLS TE tunnel and SR-TE tunnel, this document introduces two new tunnel types for tunnel related TLVs: RSVP-TE tunnel and SR-TE tunnel.

In this document TE object will be extended to carry the label mapping information for the tunnel. A new Object-Type value is defined for the TE object to indicate Tunnel Segment. The TE object in TERpt message MUST carry both TUNNEL-LABEL-BINDING TLV and Tunnel Identifier TLV with the new Object-Type value. If a PCC wants to modify a previously reported label, it MUST send a TERpt message with the TUNNEL-LABEL-BINDING TLV containing the new label binding value. If the Tunnel Identifier TLV is missing, the PCE will generate an error with error-type 6 (mandatory object missing) and error-value which means Tunnel Identifier TLV missing and close the session. If

the TUNNEL-LABEL-BINDING TLV is missing, the PCE will generate an error with error-type 6 (mandatory object missing) and error-value which means TUNNEL-LABEL-BINDING TLV missing and close the session.

If a PCE does not recognize the TUNNEL-LABEL-BINDING TLV, it MUST ignore the TLV in accordance with [RFC5440]. If a PCE recognizes the TLV but does not support the TLV, it MUST send PCErr with Error-Type = 2 (Capability not supported). If there are more than one TUNNEL-LABEL-BINDING TLVs, only the first TLV MUST be processed and the rest MUST be silently ignored.

If a PCE does not recognize the Tunnel Identifier TLV, it MUST ignore the TLV in accordance with [RFC5440]. If a PCE recognizes the TLV but does not support the TLV, it MUST send PCErr with Error-Type = 2 (Capability not supported). If there are more than one Tunnel Identifier TLVs, only the first TLV MUST be processed and the rest MUST be silently ignored.

### 3.2. Procedure for PCE Download Label Binding

[I-D.zhao-pce-pcep-extension-for-pce-controller] has defined the Label Update Message (also referred to as PCLabelUpd) that is a PCEP message sent by a PCE to a PCC to download label or update the label mapping. The same message is also used to cleanup the label mapping. In the procedure for PCE downloading the label binding for Tunnel Segment PCE is responsible for allocating the label for PCE-initiated tunnel or the tunnel initiated by PCC and reported to PCE.

[I-D.chen-pce-pce-initiated-ip-tunnel] defines the PCE-initiated IP tunnel and the TUNNEL object. PCE uses the Label Update Message to download the label mapping for the tunnel in the case of Segment Routing. The PCLabelUpd Message is defined in [I-D.zhao-pce-pcep-extension-for-pce-controller] and the extension of the PCLabelUpd message for tunnel segment is defined as follows:

```
<pce-label-update> ::= (<pce-label-download>|<pce-label-map>
                        |<pce-label-tunnel-map>)
```

Where:

```
<pce-label-tunnel-map> ::= <SRP>
                           <LABEL>
                           <TUNNEL>
```

TUNNEL object refers to the definition of [I-D.chen-pce-pce-initiated-ip-tunnel] and this document extends the use of TUNNEL object in PCLabelUpd message. In order to support Tunnel Segment for MPLS TE tunnel and SR-TE tunnel, this document introduces two new tunnel types for TLVs used in TUNNEL object: RSVP-TE tunnel and SR-TE tunnel. The TUNNEL object is an optional object

and used in the tunnel segment mode in PCLabelUpd message. TUNNEL object in PCLabelUpd message MUST carry the TUNNEL-IDENTIFIER TLV with Tunnel ID set. If the TLV is missing, the PCC will generate a PCErr message with Error-Type=6 (mandatory object missing) and Error-Value which means Tunnel Identifier TLV missing and close the session.

To cleanup the label mapping for the tunnel the SRP object MUST set the R (remove) bit.

PCE downloads the label mapping to the ingress node of the tunnel and create the label forwarding entry for the tunnel segment. PCE can also download the label mapping to other nodes which will use the label mapping of the tunnel for SR path computation.

#### 4. Objects

##### 4.1. TE object

TE object is defined in [I-D.dhodylee-pce-pcep-te-data-extn]. This document defines a new Object-Type value for TE object:

- o Tunnel Segment: TE Object-Type is 3 (to be assigned by IANA).

#### 5. TLVs

##### 5.1. Tunnel Label Binding TLV

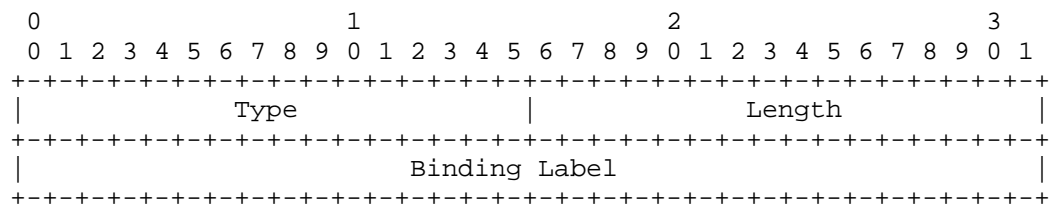


Figure 1: TUNNEL-LABEL-BINDING TLV

The type of the TLV is to be assigned by IANA and it has a fixed length of 4 octets.

The value contains the following fields:

Binding Label: contains the binding label which is generic.

## 5.2. PATH-SETUP-TYPE TLV

The PATH-SETUP-TYPE TLV is defined in [I-D.sivabalan-pce-lsp-setup-type]. This document defines following new PST value:

- o PST = 4(to be assigned by IANA): tunnel segment mode that means path setup will use the tunnel segment.

On a PCLabelUpd message, the PST=4 in PATH-SETUP-TYPE TLV in SRP object indicates that this LSP was setup using the tunnel segment.

## 5.3. Tunnel Related TLV

[I-D.chen-pce-pce-initiated-ip-tunnel] defines tunnel related TLVs including Tunnel Identifier TLV, Tunnel Name TLV, Tunnel Parameter TLV and Tunnel Attribute TLV. Tunnel Identifier TLV and Tunnel Parameter TLV contain the Tunnel Type field and only IP tunnel types are defined. This document defines following two new tunnel types to support RSVP-TE tunnel and SR-TE tunnel. The values are to be assigned by IANA and MUST NOT conflict with the registry for "BGP Tunnel Encapsulation Attribute Tunnel Types" [RFC5512] assigned by IANA.

Tunnel Type	Value
-----	-----
RSVP-TE	14
SR-TE	15

Tunnel Identifier TLV can be directly used for RSVP-TE tunnel and SR-TE tunnel. Tunnel Parameter TLV for RSVP-TE tunnel and SR-TE tunnel will be defined in the future version.

## 6. IANA Considerations

TBD.

## 7. Security Considerations

TBD.

## 8. References

### 8.1. Normative References

- [I-D.chen-pce-pce-initiated-ip-tunnel]  
Chen, X. and Z. Li, "PCE-initiated IP Tunnel", draft-chen-pce-pce-initiated-ip-tunnel-00 (work in progress), September 2015.
- [I-D.dhodylee-pce-pcep-te-data-extn]  
Dhody, D., Lee, Y., and D. Ceccarelli, "PCEP Extension for Transporting TE Data", draft-dhodylee-pce-pcep-te-data-extn-02 (work in progress), March 2015.
- [I-D.li-spring-tunnel-segment]  
Li, Z. and N. Wu, "Tunnel Segment in Segment Routing", draft-li-spring-tunnel-segment-00 (work in progress), September 2015.
- [I-D.wu-pce-pcep-ls-sr-extension]  
Wu, N. and Z. Li, "PCEP Link-State Extensions for Segment Routing", draft-wu-pce-pcep-ls-sr-extension-00 (work in progress), September 2015.
- [I-D.zhao-pce-pcep-extension-for-pce-controller]  
Zhao, Q., Zhao, K., Li, Z., Dhody, D., Palle, U., and T. Communications, "PCEP Procedures and Protocol Extensions for Using PCE as a Central Controller (PCECC) of LSPs", draft-zhao-pce-pcep-extension-for-pce-controller-02 (work in progress), October 2015.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<http://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, <<http://www.rfc-editor.org/info/rfc5440>>.
- [RFC5512] Mohapatra, P. and E. Rosen, "The BGP Encapsulation Subsequent Address Family Identifier (SAFI) and the BGP Tunnel Encapsulation Attribute", RFC 5512, DOI 10.17487/RFC5512, April 2009, <<http://www.rfc-editor.org/info/rfc5512>>.

## 8.2. Informative References

[I-D.ietf-pce-pce-initiated-lsp]

Crabbe, E., Minei, I., Sivabalan, S., and R. Varga, "PCEP Extensions for PCE-initiated LSP Setup in a Stateful PCE Model", draft-ietf-pce-pce-initiated-lsp-05 (work in progress), October 2015.

[I-D.sivabalan-pce-binding-label-sid]

Sivabalan, S., Filsfils, C., Previdi, S., Tantsura, J., Hardwick, J., and M. Nanduri, "Carrying Binding Label/Segment-ID in PCE-based Networks.", draft-sivabalan-pce-binding-label-sid-00 (work in progress), April 2015.

[I-D.sivabalan-pce-lsp-setup-type]

Sivabalan, S., Medved, J., Minei, I., Crabbe, E., and R. Varga, "Conveying path setup type in PCEP messages", draft-sivabalan-pce-lsp-setup-type-02 (work in progress), June 2014.

#### Authors' Addresses

Zhenbin Li  
Huawei Technologies  
Huawei Bld., No.156 Beiqing Rd.  
Beijing 100095  
China

Email: lizhenbin@huawei.com

Xia Chen  
Huawei Technologies  
Huawei Bld., No.156 Beiqing Rd.  
Beijing 100095  
China

Email: jescia.chenxia@huawei.com

Network Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: September 13, 2017

Z. Li  
X. Chen  
Huawei Technologies  
March 12, 2017

PCEP Extensions for Tunnel Segment  
draft-li-pce-tunnel-segment-02

Abstract

[I-D.li-spring-tunnel-segment] introduces a new type of segment, Tunnel Segment, for the segment routing. Tunnel segment can be used to reduce SID stack depth of SR path, span the non-SR domain or provide differentiated services. A binding label can be assigned to tunnel segment. An upstream node can use such a binding label for steering traffic into the appropriate tunnel. This document specifies a set of extensions to PCEP to support that PCC reports binding label of tunnel to PCE and that PCE allocates label for tunnel and updates label binding of tunnel to PCC.

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].

Status of This Memo

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

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

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

This Internet-Draft will expire on September 13, 2017.



## Copyright Notice

Copyright (c) 2017 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 . . . . .	2
2. Terminology . . . . .	3
3. Procedures . . . . .	3
3.1. Procedure for PCC Reporting Label Binding . . . . .	3
3.2. Procedure for PCE Downloading Label Binding . . . . .	5
4. Objects . . . . .	6
4.1. LS object . . . . .	6
5. TLVs . . . . .	6
5.1. Tunnel Label Binding TLV . . . . .	6
5.2. PATH-SETUP-TYPE TLV . . . . .	7
5.3. Tunnel Related TLV . . . . .	7
6. IANA Considerations . . . . .	7
7. Security Considerations . . . . .	7
8. References . . . . .	7
8.1. Normative References . . . . .	7
8.2. Informative References . . . . .	8
Authors' Addresses . . . . .	9

## 1. Introduction

[I-D.li-spring-tunnel-segment] introduces a new type of segment, Tunnel Segment, for the segment routing. Tunnel segment can be used to reduce SID stack depth of SR path, span the non-SR domain or provide differentiated services. A binding label can be assigned to tunnel segment. An upstream node can use such a binding label for steering traffic into the appropriate tunnel. The tunnel segment can be allocated for RSVP-TE tunnel, SR-TE tunnel or IP tunnel.

[I-D.li-spring-tunnel-segment] defines the requirement of control plane to support use cases of tunnel segment. The PCE related requirements are as follows:

-- PCEP extensions SHOULD be introduced to advertise the binding relationship between a SID/label and the corresponding tunnel from a PCC to a PCE. Attributes of the tunnel MAY be carried optionally.

-- PCE SHOULD support to allocate SID/label for the corresponding tunnel dynamically.

-- PCEP extensions SHOULD be introduced to distribute the binding relationship between a SID/label and the corresponding tunnel from a PCE to a PCC. Attributes of the tunnel MAY be carried optionally.

This document specifies the protocol extensions to PCEP to support these requirements defined in [I-D.li-spring-tunnel-segment].

## 2. Terminology

SR: Segment Routing

SR-TE: Segment Routing Traffic Engineering

SR-TE Tunnel: Segment Routing Traffic Engineering Tunnel

This document uses the terms defined in [RFC5440]: PCC, PCE, PCEP Peer.

The following terms are defined in [I-D.ietf-pce-pce-initiated-lsp]:

PCE-initiated LSP: LSP that is instantiated as a result of a request from the PCE.

The following terms are defined in [I-D.chen-pce-pce-initiated-ip-tunnel]:

IP Tunnel: Tunnel that uses IP encapsulation.

PCE-initiated IP Tunnel: IP Tunnel that is instantiated as a result of a request from the PCE.

## 3. Procedures

### 3.1. Procedure for PCC Reporting Label Binding

In the procedure for PCC reporting the label binding PCC allocates the label and reports the label binding for the tunnel according to the local policy PCC. For report the label binding information, there are following options:

Option 1: [I-D.zhao-pce-pcep-extension-for-pce-controller] specifies the procedures and PCEP protocol extensions for using the PCE as the central controller where LSPs are calculated/setup/initiated and label forwarding entries are downloaded to PCC. It introduces the LABEL object to specify the label binding information in PCLabelUpd message. The label carried in LABEL object is mapped to specific LSP carried in LSP object or FEC carried in FEC object. The LABEL object defined in the document is to allocate label from the PCE to PCC and is not appropriate to represent the label binding for the tunnel which can be carried in other PCE objects.

Option 2: [I-D.sivabalan-pce-binding-label-sid] proposes an approach for reporting binding label/SID to Path Computation Element (PCE) for supporting PCE-based Traffic Engineering policies. It introduces the optional TLV called "TE-PATH-BINDING TLV" to carry binding label or SID for a TE path. This TLV is limited to traffic engineering and not appropriate for tunnel segment.

Option 3: PCEP-LS [I-D.dhodylee-pce-pcep-ls] extends the Path Computation Element Communication Protocol (PCEP) with TED population capability. A PCEP LS Report message (also referred to as LSRpt message) is sent by a PCC to a PCE to report the TED state. The LS object is a mandatory object which carries TE information of a TE node or a TE link. [I-D.wu-pce-pcep-ls-sr-extension] introduces new extensions of PCEP-LS to export path segment information for Segment Routing.

This document adopts Option 3 and introduces a new type of TLV, TUNNEL-LABEL-BINDING TLV, which is a new optional TLV defined to report the label mapping for the tunnel in the case of Segment Routing. The tunnel can be PCE-initiated tunnel or created by PCC. [I-D.chen-pce-pce-initiated-ip-tunnel] defines the PCE-initiated IP tunnel and Tunnel object. Tunnel related TLVs defined in [I-D.chen-pce-pce-initiated-ip-tunnel] will be used when report label binding for the tunnel. In order to support Tunnel Segment for MPLS TE tunnel and SR-TE tunnel, this document introduces two new tunnel types for tunnel related TLVs: RSVP-TE tunnel and SR-TE tunnel.

In this document LS object will be extended to carry the label mapping information for the tunnel. A new Object-Type value is defined for the LS object to indicate Tunnel Segment. The LS object in LSRpt message MUST carry both TUNNEL-LABEL-BINDING TLV and Tunnel Identifier TLV with the new Object-Type value. If a PCC wants to modify a previously reported label, it MUST send a LSRpt message with the TUNNEL-LABEL-BINDING TLV containing the new label binding value. If the Tunnel Identifier TLV is missing, the PCE will generate an error with error-type 6 (mandatory object missing) and error-value which means Tunnel Identifier TLV missing and close the session. If

the TUNNEL-LABEL-BINDING TLV is missing, the PCE will generate an error with error-type 6 (mandatory object missing) and error-value which means TUNNEL-LABEL-BINDING TLV missing and close the session.

If a PCE does not recognize the TUNNEL-LABEL-BINDING TLV, it MUST ignore the TLV in accordance with [RFC5440]. If a PCE recognizes the TLV but does not support the TLV, it MUST send PCErr with Error-Type = 2 (Capability not supported). If there are more than one TUNNEL-LABEL-BINDING TLVs, only the first TLV MUST be processed and the rest MUST be silently ignored.

If a PCE does not recognize the Tunnel Identifier TLV, it MUST ignore the TLV in accordance with [RFC5440]. If a PCE recognizes the TLV but does not support the TLV, it MUST send PCErr with Error-Type = 2 (Capability not supported). If there are more than one Tunnel Identifier TLVs, only the first TLV MUST be processed and the rest MUST be silently ignored.

### 3.2. Procedure for PCE Downloading Label Binding

[I-D.zhao-pce-pcep-extension-for-pce-controller] has defined the Label Update Message (also referred to as PCLabelUpd) that is a PCEP message sent by a PCE to a PCC to download label or update the label mapping. The same message is also used to cleanup the label mapping. In the procedure for PCE downloading the label binding for Tunnel Segment PCE is responsible for allocating the label for PCE-initiated tunnel or the tunnel initiated by PCC and reported to PCE.

[I-D.chen-pce-pce-initiated-ip-tunnel] defines the PCE-initiated IP tunnel and the TUNNEL object. PCE uses the Label Update Message to download the label mapping for the tunnel in the case of Segment Routing. The PCLabelUpd Message is defined in [I-D.zhao-pce-pcep-extension-for-pce-controller] and the extension of the PCLabelUpd message for tunnel segment is defined as follows:

```
<pce-label-update> ::= (<pce-label-download>|<pce-label-map>
                        |<pce-label-tunnel-map>)
```

Where:

```
<pce-label-tunnel-map> ::= <SRP>
                           <LABEL>
                           <TUNNEL>
```

TUNNEL object refers to the definition of [I-D.chen-pce-pce-initiated-ip-tunnel] and this document extends the use of TUNNEL object in PCLabelUpd message. In order to support Tunnel Segment for MPLS TE tunnel and SR-TE tunnel, this document introduces two new tunnel types for TLVs used in TUNNEL object: RSVP-TE tunnel and SR-TE tunnel. The TUNNEL object is an optional object

and used in the tunnel segment mode in PCLabelUpd message. TUNNEL object in PCLabelUpd message MUST carry the TUNNEL-IDENTIFIER TLV with Tunnel ID set. If the TLV is missing, the PCC will generate a PCErr message with Error-Type=6 (mandatory object missing) and Error-Value which means Tunnel Identifier TLV missing and close the session.

To cleanup the label mapping for the tunnel the SRP object MUST set the R (remove) bit.

PCE downloads the label mapping to the ingress node of the tunnel and create the label forwarding entry for the tunnel segment. PCE can also download the label mapping to other nodes which will use the label mapping of the tunnel for SR path computation.

## 4. Objects

#### 4.1. LS object

LS object is defined in [I-D.draft-dhodylee-pce-pcep-ls]. This document defines a new Object-Type value for LS object:

- o Tunnel Segment: LS Object-Type is 5 (to be assigned by IANA).

## 5. TLVs

### 5.1. Tunnel Label Binding TLV

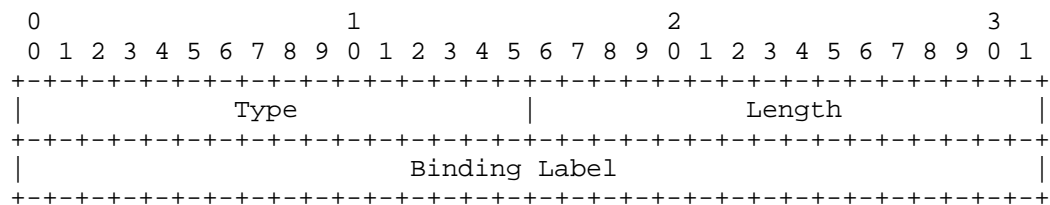


Figure 1: TUNNEL-LABEL-BINDING TLV

The type of the TLV is to be assigned by IANA and it has a fixed length of 4 octets.

The value contains the following fields:

Binding Label: contains the binding label which is generic.

## 5.2. PATH-SETUP-TYPE TLV

The PATH-SETUP-TYPE TLV is defined in [I-D.sivabalan-pce-lsp-setup-type]. This document defines following new PST value:

- o PST = 4(to be assigned by IANA): tunnel segment mode that means path setup will use the tunnel segment.

On a PCLabelUpd message, the PST=4 in PATH-SETUP-TYPE TLV in SRP object indicates that this LSP was setup using the tunnel segment.

## 5.3. Tunnel Related TLV

[I-D.chen-pce-pce-initiated-ip-tunnel] defines tunnel related TLVs including Tunnel Identifier TLV, Tunnel Name TLV, Tunnel Parameter TLV and Tunnel Attribute TLV. Tunnel Identifier TLV and Tunnel Parameter TLV contain the Tunnel Type field and only IP tunnel types are defined. This document defines following two new tunnel types to support RSVP-TE tunnel and SR-TE tunnel. The values are to be assigned by IANA and MUST NOT conflict with the registry for "BGP Tunnel Encapsulation Attribute Tunnel Types" [RFC5512] assigned by IANA.

Tunnel Type	Value
-----	-----
RSVP-TE	14
SR-TE	15

Tunnel Identifier TLV can be directly used for RSVP-TE tunnel and SR-TE tunnel. Tunnel Parameter TLV for RSVP-TE tunnel and SR-TE tunnel will be defined in the future version.

## 6. IANA Considerations

TBD.

## 7. Security Considerations

TBD.

## 8. References

### 8.1. Normative References

- [I-D.chen-pce-pce-initiated-ip-tunnel]  
Chen, X. and Z. Li, "PCE-initiated IP Tunnel", draft-chen-pce-pce-initiated-ip-tunnel-00 (work in progress), September 2015.
- [I-D.dhodylee-pce-pcep-ls]  
Dhody, D., Lee, Y., and D. Ceccarelli, "PCEP Extension for Distribution of Link-State and TE Information.", draft-dhodylee-pce-pcep-ls-07 (work in progress), March 2017.
- [I-D.li-spring-tunnel-segment]  
Li, Z. and N. Wu, "Tunnel Segment in Segment Routing", draft-li-spring-tunnel-segment-01 (work in progress), March 2016.
- [I-D.wu-pce-pcep-ls-sr-extension]  
Wu, N. and Z. Li, "PCEP Link-State Extensions for Segment Routing", draft-wu-pce-pcep-ls-sr-extension-00 (work in progress), September 2015.
- [I-D.zhao-pce-pcep-extension-for-pce-controller]  
Zhao, Q., Li, Z., Dhody, D., and C. Zhou, "PCEP Procedures and Protocol Extensions for Using PCE as a Central Controller (PCECC) of LSPs", draft-zhao-pce-pcep-extension-for-pce-controller-04 (work in progress), January 2017.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<http://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, <<http://www.rfc-editor.org/info/rfc5440>>.
- [RFC5512] Mohapatra, P. and E. Rosen, "The BGP Encapsulation Subsequent Address Family Identifier (SAFI) and the BGP Tunnel Encapsulation Attribute", RFC 5512, DOI 10.17487/RFC5512, April 2009, <<http://www.rfc-editor.org/info/rfc5512>>.

## 8.2. Informative References

[I-D.ietf-pce-pce-initiated-lsp]

Crabbe, E., Minei, I., Sivabalan, S., and R. Varga, "PCEP Extensions for PCE-initiated LSP Setup in a Stateful PCE Model", draft-ietf-pce-pce-initiated-lsp-09 (work in progress), March 2017.

[I-D.sivabalan-pce-binding-label-sid]

Sivabalan, S., Filsfils, C., Previdi, S., Tantsura, J., Hardwick, J., and M. Nanduri, "Carrying Binding Label/Segment-ID in PCE-based Networks.", draft-sivabalan-pce-binding-label-sid-02 (work in progress), October 2016.

[I-D.sivabalan-pce-lsp-setup-type]

Sivabalan, S., Medved, J., Minei, I., Crabbe, E., and R. Varga, "Conveying path setup type in PCEP messages", draft-sivabalan-pce-lsp-setup-type-02 (work in progress), June 2014.

#### Authors' Addresses

Zhenbin Li  
Huawei Technologies  
Huawei Bld., No.156 Beiqing Rd.  
Beijing 100095  
China

Email: lizhenbin@huawei.com

Xia Chen  
Huawei Technologies  
Huawei Bld., No.156 Beiqing Rd.  
Beijing 100095  
China

Email: jescia.chenxia@huawei.com



PCE  
Internet-Draft  
Intended status: Standards Track  
Expires: September 21, 2016

C. Margaria, Ed.  
C. Barth  
S. Cheruathur  
B. Tsai  
Juniper  
March 20, 2016

PCEP Procedures for Hierarchical Label Switched Paths  
draft-margaria-pce-pcep-hlsp-extension-00

Abstract

Label Switched Paths (LSPs) set up in Multiprotocol Label Switching (MPLS) or Generalized MPLS (GMPLS) networks can be used to form links to carry traffic in those networks or in other (client) networks. These LSPs can be referred to as Hierarchical LSPs (H-LSP). H-LSPs allow to improve the scalability of MPLS/GMPLS networks by creating hierarchies of TE-LSPs. Those hierarchies are an important state for optimal TE-Computation, therefore a stateful PCE should be able to discover and manage those H-LSPs. A PCE having a global view of the network, including Forwarding Adjacencies LSP (FA-LSP) and non FA-LSPs, can create more optimal hierarchies and (re-)compute the TE-LSPs path to make use of the H-LSPs. In particular a PCE can better leverage the Private H-LSP introduced by RFC6107 without influencing the IGP, allowing a less disruptive use of Hierarchies.

RFC6107 defined Protocol mechanisms to facilitate the establishment of such LSPs and to bundle traffic engineering (TE) links to reduce the load on routing protocols.

This document defines PCEP extensions to learn about and control those H-LSPs.

Status of This Memo

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

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

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

This Internet-Draft will expire on September 21, 2016.

#### Copyright Notice

Copyright (c) 2016 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
1.1. Requirements Language . . . . .	3
1.2. Solution overview . . . . .	3
2. H-LSP capability advertisement . . . . .	4
2.1. PCE Discovery Protocol . . . . .	4
2.2. OPEN Object extension HLSP-CAPABILITY TLV . . . . .	4
3. PCEP object and extensions . . . . .	5
3.1. The PCReq message . . . . .	5
3.2. The PCRep message . . . . .	6
3.3. The PCRpt message . . . . .	6
3.4. The PCUpd message . . . . .	6
3.5. The PCInitiate message . . . . .	7
3.6. LSP_TUNNEL_INTERFACE_ID Object . . . . .	7
3.6.1. Procedures . . . . .	7
4. Additional Error Type and Error Values Defined . . . . .	9
5. IANA Considerations . . . . .	10
5.1. PCEP Objects . . . . .	10
5.2. New PCEP TLVs . . . . .	11
5.3. RP Object Flag Field . . . . .	11
5.4. New PCEP Error Codes . . . . .	12
6. Security Considerations . . . . .	13
7. Acknowledgments . . . . .	14
8. References . . . . .	14
8.1. Normative References . . . . .	14
8.2. Informative References . . . . .	15
Authors' Addresses . . . . .	15

## 1. Introduction

Traffic Engineering (TE) links in a Multiprotocol Label Switching (MPLS) or a Generalized MPLS (GMPLS) network may be constructed from Label Switched Paths (LSPs) [RFC6107]. Such LSPs are defined as hierarchical LSPs (H-LSPs).

The mechanisms described in [RFC6107] enables the dynamically construction of provisioned hierarchical networks. The Path Computation Element Protocol (PCEP) defined in [RFC5440], [RFC5521], [RFC5541], [RFC5520], [I-D.ietf-pce-gmpls-pcep-extensions], [I-D.ietf-pce-stateful-pce] and [I-D.ietf-pce-pce-initiated-lsp] enable a PCE to compute paths for a range of switching technologies in a stateless and statefull manner, but does not allow a PCE to control the hierarchy of such LSPs. This document complements those RFCs to control H-LSPs.

This document provides the same level of control as [RFC6107], so that the PCE can provide the following information to the LSPs endpoints:

- Whether the LSP is an ordinary LSP or an H-LSP.

- In which IGP instances the LSP should be advertised as a link.

- How the client networks should make use of H-LSP and corresponding TE-links.

- Whether the TE-link should form part of a bundle (and if so, which bundle).

- How the link endpoints should be identified when advertised.

### 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 RFC 2119 [RFC2119].

### 1.2. Solution overview

The encoding and semantics associated with the control of H-LSPs is already considered and defined by [RFC6107]. This document reuses those definitisns and adapts them to PCEP. The following section describes the new PCEP new objects and associated procedures.

## 2. H-LSP capability advertisement

## 2.1. PCE Discovery Protocol

IGP-based PCE Discovery (PCED) is defined in [RFC5088] and [RFC5089] for the OSPF and IS-IS protocols. A new flag (bit TBA-1) is defined to advertise the H-LSP capability:

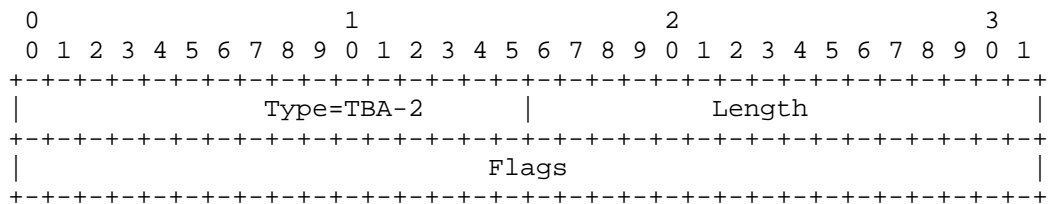
## Bit Capabilities

## TBA-1 : H-LSP Capability

## 2.2. OPEN Object extension HLSP-CAPABILITY TLV

In addition to the IGP advertisement, a PCEP speaker SHOULD be able to discover the other peer GMPLS capabilities during the Open message exchange. This capability is also useful to avoid misconfigurations. This document defines a new GMPLS-CAPABILITY TLV for use in the OPEN object to negotiate the H-LSP capability. The inclusion of this TLV in the OPEN message indicates that the PCC/PCE support the PCEP extensions defined in this document. A PCE that is able to support the extensions defined in this document MUST include the HLSP-CAPABILITY TLV in the OPEN message. If a PCEP Peer does not include the HLSP-CAPABILITY TLV in the OPEN message and the other PCEP peer does include the TLV, it is RECOMMENDED that each peer indicates a mismatch of capabilities. If any of the peers do not advertise the HLSP-CAPABILITY TLV, the extension defined in this document MUST NOT be used.

IANA has allocated value TBA-2 from the "PCEP TLV Type Indicators" sub-registry, as documented in Section 5.2 ("New PCEP TLVs"). The description is "HLSP-CAPABILITY". Its format is shown in the following figure.



No Flags are defined in this document, they are reserved for future use.

### 3. PCEP object and extensions

This section describes the required PCEP objects and extensions. The PCReq and PCRep messages are defined in [RFC5440]. The PCRpt and PCUpd messages are defined in [I-D.ietf-pce-stateful-pce] and the PCInitiate in [I-D.ietf-pce-pce-initiated-lsp]. The control of H-LSP by a PCE will reuse and adapt the information, encoding and procedure described in [RFC6107]. This document defines the LSP\_TUNNEL\_INTERFACE\_ID PCEP object for that purpose and it is carried in the following messages:

PCReq: The PCC indicates that it will form a H-LSP.

PCRep: If the object was present in the corresponding PCReq, the PCE may suggest IDs.

PCRpt: The PCC reports the state of the H-LSP.

PCUpd: The PCE requests the LSP to be used as H-LSP.

PCInitiate: The PCE requests the creation of a H-LSP.

#### 3.1. The PCReq message

The PCReq MAY include the LSP\_TUNNEL\_INTERFACE\_ID object. Multiple instances of the object MAY be included in the message, in case multiple IGP instances are the target, following [RFC6107], section 3.4. The presence of the object indicates that the PCC will setup the TE-LSP as a H-LSP. This MAY be used by the PCE as policy input. The PCC MAY set the IDs to 0, as described in Section 3.6.1.

The PCReq is modified as follows:

```
<request> ::= <RP>
               <END-POINTS>
               [<LSPA>]
               [<BANDWIDTH>]
               [<metric-list>]
               [<OF>]
               [<RRO> [<BANDWIDTH>]]
               [<IRO>]
               [<LOAD-BALANCING>]
               [<XRO>]
               [<LSP_TUNNEL_INTERFACE_ID>...]
```

### 3.2. The PCRep message

The PCE MAY include the LSP\_TUNNEL\_INTERFACE\_ID object from the corresponding PCReq. The PCE MUST NOT include the LSP\_TUNNEL\_INTERFACE\_ID if it was not present in the corresponding PCReq. If the IDs were set to 0 on request, the PCE SHOULD provide a recommended value, as described in Section 3.6.1.

The PCRep uses the <attribute-list> definition, which is extended as follows:

```
<attribute-list>::=[<LSPA>]
                    [<BANDWIDTH>]
                    [<metric-list>]
                    [<IRO>]
                    [<LSP_TUNNEL_INTERFACE_ID>...]
```

### 3.3. The PCRpt message

The PCRpt MAY include the LSP\_TUNNEL\_INTERFACE\_ID object. Multiple instances of the object MAY be included in the message, in the case where multiple IGP instances are the target, following [RFC6107], section 3.4 or to report the ingress and egress IDs. The presence of the object indicates that the PCC will setup the TE-LSP as a H-LSP. If the LSP object O(Operational) flag is DOWN, the PCC MAY set the IDs to 0, as described in Section 3.6.1. If the LSP object O flag is UP or ACTIVE the PCC SHOULD report at least 2 LSP\_TUNNEL\_INTERFACE\_IDS for a given target IGP instance, one for ingress and one for egress.

The PCRpt uses the <attribute-list> definition, which is extended as described in Section 3.2.

### 3.4. The PCUpd message

The PCUpd MAY include the LSP\_TUNNEL\_INTERFACE\_ID object. Multiple instances of the object MAY be included in the message, in the case where multiple IGP instances are the target, following [RFC6107], section 3.4 or to report the ingress and egress IDs. The presence of the object indicates that the PCC SHOULD setup the TE-LSP as a H-LSP. The PCE MUST NOT include any object type for the egress node. If the PCE includes the object type for the egress node the PCC MUST send a PCErr with error type TBA-5(PCC Hierarchy Issue) and error value 1(Egress LSP\_TUNNEL\_INTERFACE\_ID Object type in PCUp, PCRep or PCInitiate message). The PCE MAY set the IDs in accordance to Section 3.6.1.

The PCUpd use the <attribute-list> definition, which is extended as described in Section 3.2

Upon receipt of a PCUpd message with LSP\_TUNNEL\_INTERFACE\_ID, the PCC SHOULD try to setup the TE-LSP as a H-LSP based on its policies. If the PCC ignores the LSP\_TUNNEL\_INTERFACE\_ID, it MUST set the I bit. If the PCE requires the LSP to be an H-LSP, it MUST set the P(Processing) Flag in the object header.

If the PCE is tearing down the LSP, the client LSPs may be impacted. It is RECOMMENDED that the PCC uses the Gracefull link shutdown procedures described in [RFC4203], [RFC5307] and [RFC5817]. It can be desirable for a PCE to know in advance if the LSP carries traffic before initiating the teardown as it would result in a smoother transition in the case where the gracefull teardown procedures are not used. This indication is not a H-LSP specific operation and MAY be used in a more general context, therefore it is out of the scope of this document.

### 3.5. The PCInitiate message

The procedure for PCInitiate are the same as for PCUpd, described in Section 3.4.

### 3.6. LSP\_TUNNEL\_INTERFACE\_ID Object

IANA has allocated value TBA-3 from the "PCEP Objects" sub-registry, as documented in Section 5.1 ("New PCEP Object"). The description is "LSP\_TUNNEL\_INTERFACE\_ID". The following object-type are defined by this document:

#### Object-Type Description

- |   |   |
|---|---|
| 1 | Ingress Unnumbered Links with Action Identification.    |
| 2 | Ingress IPv4 Numbered Links with Action Identification. |
| 3 | Ingress IPv6 Numbered Links with Action Identification. |
| 4 | Egress Unnumbered Links with Action Identification.     |
| 5 | Egress IPv4 Numbered Links with Action Identification.  |
| 6 | Egress IPv6 Numbered Links with Action Identification.  |

The content and TLVs are those defined in [RFC6107]. The TLVs are not PCEP TLVs.

#### 3.6.1. Procedures

In [RFC6107] the interface IDs are allocated by the endpoints, this principle remains unchanged. In the context of PCEP the PCE does not manage the PCC ids. It may suggest IDs (numbered or unnumbered), but

the PCC remains in control of these allocations. The PCE can indicate to the PCC that the ID SHOULD be allocated by the PCC by setting the ID to the value of 0. This applies to the following fields:

Interface ID

LSR's Router ID

IPv4 Interface Address

IPv6 Interface Address

Component Link Identifier

IPv4 Numbered Component Link Identification

IPv6 Numbered Component Link Identification

The PCE MAY only set the Object-type (OT) in the range of 1 to 3, while the OT range of 4 to 6 are reserved for reporting the reverse Ids assigned by the egress node.

The ID MAY be 0 for OT 1 to 3 in the following cases:

PCReq: the PCC is indicating that the PCE SHOULD provide a value

PCRep: the PCE is indicating the PCC SHOULD do the allocation

PCRpt: when the LSP is DOWN or GOING-UP

PCUpd: the PCE is indicating the PCC SHOULD do the allocation

PCInitiate: the PCE is indicating the PCC SHOULD do the allocation

In case where the PCC is not able to allocate an address suitable for the H-LSP, it MUST reply with a PCErr with type TBA-5 (PCC Hierarchy Issue), value 9 (PCC Cannot allocate a IPv4 Interface Address), value 10 (PCC Cannot allocate a IPv6 Interface Address) or value 11 (PCC Cannot allocate an Unnumbered Interface Address).

The ID MAY be set by the PCE for OT in range of 1 to 3 in the following cases:

PCRep: the PCE is suggesting and ID to be used

PCUpd: the PCE is suggesting and ID to be used



PCInitiate: the PCE is suggesting an ID to be used

The PCC MAY use the suggested value. If the value is not used, the PCC SHOULD send a PCErr with type TBA-5 (PCC Hierarchy Issue) and a value 2 (Interface ID provided is invalid), 3 (LSR's Router ID provided is invalid), 4 (IPv4 Interface Address provided is invalid), 5 (IPv6 Interface Address provided is invalid), 6 (Component Link Identifier provided is invalid), 7 (IPv4 Numbered Component Link Identification provided is invalid) or 8 (IPv6 Numbered Component Link Identification provided is invalid).

The ID MUST NOT be 0 for OT 1 to 3 in the following cases:

PCRpt when the LSP is UP, ACTIVE or GOING-DOWN

#### 4. Additional Error Type and Error Values Defined

A PCEP-ERROR object is used to report a PCEP error and is characterized by an Error-Type that specifies the type of error while Error-value that provides additional information about the error. Additional error types and error values are defined to represent some of the errors related to the newly identified objects. For each PCEP error, an Error-Type and an Error-value are defined. Error-Type 1 to 10 are already defined in [RFC5440]. Two new Error-Type are introduced (value TBA-4 and TBA-5).

## Error-Type Error-value

Type=TBA-4	LSP Hierarchy Issue
Value=1:	Link advertisement not supported.
Value=2:	Link advertisement not allowed by policy.
Value=3:	TE link creation not supported.
Value=4:	TE link creation not allowed by policy.
Value=5:	Routing adjacency creation not supported.
Value=6:	Routing adjacency creation not allowed by policy.
Value=7:	Bundle creation not supported.
Value=8:	Bundle creation not allowed by policy.
Value=9:	Hierarchical LSP not supported.
Value=10:	LSP stitching not supported.
Value=11:	Link address type or family not supported.
Value=12:	IGP instance unknown.
Value=13:	IGP instance advertisement not allowed by policy.
Value=14:	Component link identifier not valid.
Value=15:	Unsupported component link identifier address family.
Type=TBA-5	PCC Hierarchy Issue
Value=1:	Egress LSP_TUNNEL_INTERFACE_ID Object type in PCUp, PCRep or PCInitiate message.
Value=2:	Interface ID provided is invalid.
Value=3:	LSR's Router ID provided is invalid.
Value=4:	IPv4 Interface Address provided is invalid.
Value=5:	IPv6 Interface Address provided is invalid.
Value=6:	Component Link Identifier provided is invalid.
Value=7:	IPv4 Numbered Component Link Identification provided is invalid.
Value=8:	IPv6 Numbered Component Link Identification provided is invalid.
Value=9:	PCC Cannot allocate a IPv4 Interface Address.
Value=10:	PCC Cannot allocate a IPv6 Interface Address.
Value=11:	PCC Cannot allocate an Unnumbered Interface Address.

## 5. IANA Considerations

## 5.1. PCEP Objects

IANA is requested to make the following Object-Type allocations from the "PCEP Objects" sub-registry.

Object Class Value	Name	Object-Type	Reference
TBA-3	LSP_TUNNEL_INTERFACE_ID	1: Ingress Unnumbered Links with Action Identification.	This document
		2: Ingress IPv4 Numbered Links with Action Identification.	This document
		3: Ingress IPv6 Numbered Links with Action Identification.	This document
		4: Egress Unnumbered Links with Action Identification.	This document
		5: Egress IPv4 Numbered Links with Action Identification.	This document
		6: Egress IPv6 Numbered Links with Action Identification.	This document
		7-15: Unassigned	This document

## 5.2. New PCEP TLVs

IANA manages the PCEP TLV code point registry (see [RFC5440]). This is maintained as the "PCEP TLV Type Indicators" sub-registry of the "Path Computation Element Protocol (PCEP) Numbers" registry. This document defines new PCEP TLVs, to be carried in the END-POINTS object with Generalized Endpoint object Type. IANA is requested to do the following allocation. The values here are suggested for use by IANA.

Value	Meaning	Reference
TBA-2	HLSP-CAPABILITY TLV	This document (section Section 2.2)

## 5.3. RP Object Flag Field

As described in new flag are defined in the RP Object Flag IANA is requested to make the following allocations from the OSPF registry, "PCE Capability Flags" sub-registry. The values here are suggested for use by IANA.

Bit	Description	Reference
TBA-1	H-LSP Capability	This document, Section 2.1

#### 5.4. New PCEP Error Codes

As described in Section 4, new PCEP Error-Type and Error Values are defined. IANA is requested to make the following allocation in the "PCEP-ERROR Object Error Types and Values" registry. The values here are suggested for use by IANA.

Error	name	Reference
Type=TBA-4	LSP Hierarchy Issue	This Document
Value=1:	Link advertisement not supported.	This Document
Value=2:	Link advertisement not allowed by policy.	This Document
Value=3:	TE link creation not supported.	This Document
Value=4:	TE link creation not allowed by policy.	This Document
Value=5:	Routing adjacency creation not supported.	This Document
Value=6:	Routing adjacency creation not allowed by policy.	This Document
Value=7:	Bundle creation not supported.	This Document
Value=8:	Bundle creation not allowed by policy.	This Document
Value=9:	Hierarchical LSP not supported.	This Document
Value=10:	LSP stitching not supported.	This Document
Value=11:	Link address type or family not supported.	This Document
Value=12:	IGP instance unknown.	This Document
Value=13:	IGP instance advertisement not allowed by policy.	This Document
Value=14:	Component link identifier not valid.	This Document
Value=15:	Unsupported component link identifier address family.	This Document
Type=TBA-5	PCC Hierarchy Issue	This Document
Value=1:	Egress LSP_TUNNEL_INTERFACE_ID Object type in PCUp, PCRep or PCInitiate message.	This Document
Value=2:	Interface ID provided is invalid.	This Document
Value=3:	LSR's Router ID provided is invalid.	This Document
Value=4:	IPv4 Interface Address provided is invalid.	This Document
Value=5:	IPv6 Interface Address provided is invalid.	This Document
Value=6:	Component Link Identifier provided is invalid.	This Document
Value=7:	IPv4 Numbered Component Link Identification provided is invalid.	This Document
Value=8:	IPv6 Numbered Component Link Identification provided is invalid.	This Document
Value=9:	PCC Cannot allocate a IPv4 Interface Address.	This Document
Value=10:	PCC Cannot allocate a IPv6 Interface Address.	This Document
Value=11:	PCC Cannot allocate an Unnumbered Interface Address.	This Document
Value=:	.	This Document

## 6. Security Considerations

## 7. Acknowledgments

## 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, <<http://www.rfc-editor.org/info/rfc2119>>.
- [RFC4203] Kompella, K., Ed. and Y. Rekhter, Ed., "OSPF Extensions in Support of Generalized Multi-Protocol Label Switching (GMPLS)", RFC 4203, DOI 10.17487/RFC4203, October 2005, <<http://www.rfc-editor.org/info/rfc4203>>.
- [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, <<http://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, <<http://www.rfc-editor.org/info/rfc5089>>.
- [RFC5307] Kompella, K., Ed. and Y. Rekhter, Ed., "IS-IS Extensions in Support of Generalized Multi-Protocol Label Switching (GMPLS)", RFC 5307, DOI 10.17487/RFC5307, October 2008, <<http://www.rfc-editor.org/info/rfc5307>>.
- [RFC5440] Vasseur, JP., Ed. and JL. Le Roux, Ed., "Path Computation Element (PCE) Communication Protocol (PCEP)", RFC 5440, DOI 10.17487/RFC5440, March 2009, <<http://www.rfc-editor.org/info/rfc5440>>.
- [RFC5520] Bradford, R., Ed., Vasseur, JP., and A. Farrel, "Preserving Topology Confidentiality in Inter-Domain Path Computation Using a Path-Key-Based Mechanism", RFC 5520, DOI 10.17487/RFC5520, April 2009, <<http://www.rfc-editor.org/info/rfc5520>>.
- [RFC5521] Oki, E., Takeda, T., and A. Farrel, "Extensions to the Path Computation Element Communication Protocol (PCEP) for Route Exclusions", RFC 5521, DOI 10.17487/RFC5521, April 2009, <<http://www.rfc-editor.org/info/rfc5521>>.

- [RFC5541] Le Roux, J.L., Vasseur, J.P., and Y. Lee, "Encoding of Objective Functions in the Path Computation Element Communication Protocol (PCEP)", RFC 5541, DOI 10.17487/RFC5541, June 2009, <<http://www.rfc-editor.org/info/rfc5541>>.
- [RFC6107] Shiimoto, K., Ed. and A. Farrel, Ed., "Procedures for Dynamically Signaled Hierarchical Label Switched Paths", RFC 6107, DOI 10.17487/RFC6107, February 2011, <<http://www.rfc-editor.org/info/rfc6107>>.

## 8.2. Informative References

- [I-D.ietf-pce-gmpls-pcep-extensions]  
Margaria, C., Dios, O., and F. Zhang, "PCEP extensions for GMPLS", draft-ietf-pce-gmpls-pcep-extensions-11 (work in progress), October 2015.
- [I-D.ietf-pce-pce-initiated-lsp]  
Crabbe, E., Minei, I., Sivabalan, S., and R. Varga, "PCEP Extensions for PCE-initiated LSP Setup in a Stateful PCE Model", draft-ietf-pce-pce-initiated-lsp-05 (work in progress), October 2015.
- [I-D.ietf-pce-stateful-pce]  
Crabbe, E., Minei, I., Medved, J., and R. Varga, "PCEP Extensions for Stateful PCE", draft-ietf-pce-stateful-pce-14 (work in progress), March 2016.
- [RFC5817] Ali, Z., Vasseur, J.P., Zamfir, A., and J. Newton, "Graceful Shutdown in MPLS and Generalized MPLS Traffic Engineering Networks", RFC 5817, DOI 10.17487/RFC5817, April 2010, <<http://www.rfc-editor.org/info/rfc5817>>.

## Authors' Addresses

Cyril Margaria (editor)  
Juniper  
200 Somerset Corporate Boulevard, , Suite 4001  
Bridgewater, NJ 08807  
USA

Email: [cmargaria@juniper.net](mailto:cmargaria@juniper.net)

Colby Barth  
Juniper

Email: cbarth@juniper.net

Sudhir Cheruathur  
Juniper

Email: scheruathur@juniper.net

Ben J.C. Tsai  
Juniper

Email: jtsai@juniper.net



PCE Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: July 13, 2016

U. Palle  
D. Dhody  
Huawei Technologies  
Y. Tanaka  
NTT Communications  
Z. Ali  
Cisco Systems  
V. Beeram  
Juniper Networks  
January 10, 2016

PCEP Extensions for PCE-initiated Point-to-Multipoint LSP Setup in a  
Stateful PCE Model  
draft-palle-pce-stateful-pce-initiated-p2mp-lsp-07

Abstract

The Path Computation Element (PCE) has been identified as an appropriate technology for the determination of the paths of point-to-multipoint (P2MP) TE LSPs. This document provides extensions required for Path Computation Element communication Protocol (PCEP) so as to enable the usage of a stateful PCE initiation capability in recommending P2MP TE LSP instantiation.

Status of This Memo

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

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

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

This Internet-Draft will expire on July 13, 2016.

Copyright Notice

Copyright (c) 2016 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
1.1. Requirements Language . . . . .	4
2. Terminology . . . . .	4
3. Architectural Overview . . . . .	4
3.1. Motivation . . . . .	4
3.2. Operation Overview . . . . .	4
4. Support of PCE Initiated P2MP TE LSPs . . . . .	5
5. IGP Extensions for PCE-Initiation for P2MP Capabilities Advertisement . . . . .	5
6. PCE-initiated P2MP TE LSP Operations . . . . .	6
6.1. The PCInitiate message . . . . .	6
6.2. P2MP TE LSP Instantiation . . . . .	8
6.3. P2MP TE LSP Deletion . . . . .	8
6.4. Adding and Pruning Leaves for the P2MP TE LSP . . . . .	8
6.5. P2MP TE LSP Delegation and Cleanup . . . . .	8
7. PCInitiate Message Fragmentation . . . . .	9
7.1. PCInitiate Fragmentation Procedure . . . . .	9
8. Non-Support of P2MP TE LSP Instantiation for Stateful PCE . . . . .	9
9. Security Considerations . . . . .	10
10. Manageability Considerations . . . . .	10
10.1. Control of Function and Policy . . . . .	10
10.2. Information and Data Models . . . . .	10
10.3. Liveness Detection and Monitoring . . . . .	10
10.4. Verify Correct Operations . . . . .	10
10.5. Requirements On Other Protocols . . . . .	10
10.6. Impact On Network Operations . . . . .	11
11. IANA Considerations . . . . .	11
11.1. PCE Capabilities in IGP Advertisements . . . . .	11
11.2. STATEFUL-PCE-CAPABILITY TLV . . . . .	11
11.3. Extension of PCEP-Error Object . . . . .	11
12. Security Considerations . . . . .	12
13. Acknowledgments . . . . .	12
14. References . . . . .	12
14.1. Normative References . . . . .	12
14.2. Informative References . . . . .	13
Appendix A. Contributor Addresses . . . . .	15

Authors' Addresses	15
--------------------	----

## 1. Introduction

As per [RFC4655], the Path Computation Element (PCE) is an entity that is capable of computing a network path or route based on a network graph, and applying computational constraints. A Path Computation Client (PCC) may make requests to a PCE for paths to be computed.

[RFC4857] describes how to set up point-to-multipoint (P2MP) Traffic Engineering Label Switched Paths (TE LSPs) for use in Multiprotocol Label Switching (MPLS) and Generalized MPLS (GMPLS) networks. The PCE has been identified as a suitable application for the computation of paths for P2MP TE LSPs ([RFC5671]).

The PCEP is designed as a communication protocol between PCCs and PCEs for point-to-point (P2P) path computations and is defined in [RFC5440]. The extensions of PCEP to request path computation for P2MP TE LSPs are described in [RFC6006].

Stateful PCEs are shown to be helpful in many application scenarios, in both MPLS and GMPLS networks, as illustrated in [I-D.ietf-pce-stateful-pce-app]. These scenarios apply equally to P2P and P2MP TE LSPs. [I-D.ietf-pce-stateful-pce] provides the fundamental extensions needed for stateful PCE to support general functionality for P2P TE LSP. Further [I-D.palle-pce-stateful-pce-p2mp] focuses on the extensions that are necessary in order for the deployment of stateful PCEs to support P2MP TE LSPs. It includes mechanisms to effect P2MP LSP state synchronization between PCCs and PCEs, delegation of control of P2MP LSPs to PCEs, and PCE control of timing and sequence of P2MP path computations within and across PCEP sessions and focuses on a model where P2MP LSPs are configured on the PCC and control over them is delegated to the PCE.

[I-D.ietf-pce-pce-initiated-lsp] provides the fundamental extensions needed for stateful PCE-initiated P2P TE LSP recommended instantiation.

This document describes the setup, maintenance and teardown of PCE-initiated P2MP LSPs under the stateful PCE model, without the need for local configuration on the PCC, thus allowing for a dynamic network that is centrally controlled and deployed.

### 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].

## 2. Terminology

Terminology used in this document is same as terminology used in [I-D.ietf-pce-stateful-pce], [I-D.ietf-pce-pce-initiated-lsp] and [RFC6006].

## 3. Architectural Overview

### 3.1. Motivation

[I-D.palle-pce-stateful-pce-p2mp] provides stateful control over P2MP TE LSPs that are locally configured on the PCC. This model relies on the Ingress taking an active role in delegating locally configured P2MP TE LSPs to the PCE, and is well suited in environments where the P2MP TE LSP placement is fairly static. However, in environments where the P2MP TE LSP placement needs to change in response to application demands, it is useful to support dynamic creation and tear down of P2MP TE LSPs. The ability for a PCE to trigger the creation of P2MP TE LSPs on demand can be seamlessly integrated into a controller-based network architecture, where intelligence in the controller can determine when and where to set up paths.

Section 3 of [I-D.ietf-pce-pce-initiated-lsp] further describes the motivation behind the PCE-Initiation capability, which are equally applicable for P2MP TE LSPs.

### 3.2. Operation Overview

A PCC or PCE indicates its ability to support PCE provisioned dynamic P2MP LSPs during the PCEP Initialization Phase via mechanism described in Section 4.

As per section 5.1 of [I-D.ietf-pce-pce-initiated-lsp], the PCE sends a Path Computation LSP Initiate Request (PCInitiate) message to the PCC to suggest instantiation or deletion of a P2P TE LSP. This document extends the PCInitiate message to support P2MP TE LSP (see details in Section 6.1).

P2MP TE LSP suggested instantiation and deletion operations are same as P2P LSP as described in section 5.3 and 5.4 of [I-D.ietf-pce-pce-initiated-lsp]. This document focuses on

extensions needed for further handling of P2MP TE LSP (see details in Section 6.2).

#### 4. Support of PCE Initiated P2MP TE LSPs

During PCEP Initialization Phase, as per Section 7.1.1 of [I-D.ietf-pce-stateful-pce], PCEP speakers advertises Stateful capability via Stateful PCE Capability TLV in open message. A new flag is defined for the STATEFUL-PCE-CAPABILITY TLV defined in [I-D.ietf-pce-stateful-pce] and updated in [I-D.ietf-pce-pce-initiated-lsp], [I-D.ietf-pce-stateful-sync-optimizations], and [I-D.palle-pce-stateful-pce-p2mp].

A new bit P (P2MP-LSP-INSTANTIATION-CAPABILITY) is added in this document:

P (P2MP-LSP-INSTANTIATION-CAPABILITY - 1 bit): If set to 1 by a PCC, the P Flag indicates that the PCC allows suggested instantiation of an P2MP LSP by a PCE. If set to 1 by a PCE, the P flag indicates that the PCE will suggest P2MP LSP instantiation. The P2MP-LSP-INSTANTIATION-CAPABILITY flag must be set by both PCC and PCE in order to support PCE-initiated P2MP LSP instantiation.

A PCEP speaker should continue to advertise the basic P2MP capability via mechanisms as described in [RFC6006].

#### 5. IGP Extensions for PCE-Initiation for P2MP Capabilities Advertisement

When PCCs are LSRs participating in the IGP (OSPF or IS-IS), and PCEs are either LSRs or servers also participating in the IGP, an effective mechanism for PCE discovery within an IGP routing domain consists of utilizing IGP advertisements. Extensions for the advertisement of PCE Discovery Information are defined for OSPF and for IS-IS in [RFC5088] and [RFC5089] respectively.

The PCE-CAP-FLAGS sub-TLV, defined in [RFC5089], is an optional sub-TLV used to advertise PCE capabilities. It MAY be present within the PCED sub-TLV carried by OSPF or IS-IS. [RFC5088] and [RFC5089] provide the description and processing rules for this sub-TLV when carried within OSPF and IS-IS, respectively.

The format of the PCE-CAP-FLAGS sub-TLV is included below for easy reference:

Type: 5

Length: Multiple of 4.

Value: This contains an array of units of 32 bit flags with the most significant bit as 0. Each bit represents one PCE capability.

PCE capability bits are defined in [RFC5088]. This document defines a new capability bit for the PCE-Initiation with P2MP as follows:

Bit	Capability
TBD	PCE-Initiation with P2MP

Note that while PCE-Initiation for P2MP capability may be advertised during discovery, PCEP Speakers that wish to use stateful PCEP MUST negotiate stateful PCE-Initiation capabilities during PCEP session setup, as specified in the current document.

## 6. PCE-initiated P2MP TE LSP Operations

### 6.1. The PCInitiate message

As defined in section 5.1 of [I-D.ietf-pce-pce-initiated-lsp], PCE sends a PCInitiate message to a PCC to recommend instantiation of a P2P TE LSP, this document extends the format of PCInitiate message for the creation of P2MP TE LSPs but the creation and deletion operations of P2MP TE LSP are same to the P2P TE LSP.

The format of PCInitiate message is 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-point-path-pair-list>
                                       [<attribute-list>]
```

```
<PCE-initiated-lsp-deletion> ::= <SRP>
                                   <LSP>
```

Where:

```
<end-point-path-pair-list> ::=
    [<END-POINTS>]
    <path>
    [<end-point-path-pair-list>]
```

```
<path> ::= (<ERO> | <SERO>)
            [<path>]
```

<attribute-list> is defined in [RFC5440] and extended by PCEP extensions.

The PCInitiate message with an LSP object with N bit (P2MP) set is used to convey operation on a P2MP TE LSP. The SRP object is used to correlate between initiation requests sent by the PCE and the error reports and state reports sent by the PCC as described in [I-D.ietf-pce-stateful-pce].

The END-POINTS object MUST be carried in PCInitiate message when N bit is set in LSP object for P2MP TE LSP. If the END-POINTS object is missing, the receiving PCC MUST send a PCErr message with Error-type=6 (Mandatory Object missing) and Error-value=3 (END-POINTS object missing) (defined in [RFC5440]).

## 6.2. P2MP TE LSP Instantiation

The Instantiation operation of P2MP TE LSP is same as defined in section 5.3 of [I-D.ietf-pce-pce-initiated-lsp] including handling of PLSP-ID, SYMBOLIC-PATH-NAME TLV etc. Rules of processing and error codes remains unchanged. Further, as defined in section 6.1 of [I-D.palle-pce-stateful-pce-p2mp], N bit MUST be set in LSP object in PCInitiate message by PCE to specify the instantiation is for P2MP TE LSP and the PCC or PCE MUST follow the mechanism defined in [I-D.palle-pce-stateful-pce-p2mp] for delegation and updation of P2MP TE LSPs.

Though N bit is set in the LSP object, P2MP-LSP-IDENTIFIER TLV defined in section 6.2 of [I-D.palle-pce-stateful-pce-p2mp] MUST NOT be included in the LSP object in PCInitiate message as it SHOULD be generated by PCC and carried in PCRpt message.

## 6.3. P2MP TE LSP Deletion

The deletion operation of P2MP TE LSP is same as defined in section 5.4 of [I-D.ietf-pce-pce-initiated-lsp] by sending an LSP Initiate Message with an LSP object carrying the PLSP-ID of the LSP to be removed and an SRP object with the R flag set (LSP-REMOVE as per section 5.2 of [I-D.ietf-pce-pce-initiated-lsp]). Rules of processing and error codes remains unchanged.

## 6.4. Adding and Pruning Leaves for the P2MP TE LSP

Adding of new leaves and Pruning of old Leaves for the PCE initiated P2MP TE LSP MUST be carried in PCUpd message and SHOULD refer [I-D.palle-pce-stateful-pce-p2mp] for P2MP TE LSP extensions. As defined in [RFC6006], leaf type = 1 for adding of new leaves, leaf type = 2 for pruning of old leaves of P2MP END-POINTS Object are used in PCUpd message.

PCC MAY use the Incremental State Update mechanisms as described in [RFC4875] to signal adding and pruning of leaves.

## 6.5. P2MP TE LSP Delegation and Cleanup

P2MP TE LSP delegation and cleanup operations are same as defined in section 6 of [I-D.ietf-pce-pce-initiated-lsp]. Rules of processing and error codes remains unchanged.



## 7. PCInitiate Message Fragmentation

The total PCEP message length, including the common header, is 16 bytes. In certain scenarios the P2MP LSP Initiate may not fit into a single PCEP message (e.g. initial PCInitiate message). The F-bit is used in the LSP object to signal that the initial PCInitiate was too large to fit into a single message and will be fragmented into multiple messages.

Fragmentation procedure described below for PCInitiate message is similar to [RFC6006] which describes request and response message fragmentation.

### 7.1. PCInitiate Fragmentation Procedure

Once the PCE initiates to set up the P2MP TE LSP, a PCInitiate message is sent to the PCC. If the PCInitiate is too large to fit into a single PCInitiate message, the PCE will split the PCInitiate over multiple messages. Each PCInitiate message sent by the PCE, except the last one, will have the F-bit set in the LSP object to signify that the PCInitiate has been fragmented into multiple messages. In order to identify that a series of PCInitiate messages represents a single Initiate, each message will use the same PLSP-ID (in this case 0) and SRP-ID-number.

To indicate P2MP message fragmentation errors associated with a P2MP PCInitiate, a Error-Type (18) and a new error-value TBD is used if a PCC has not received the last piece of the fragmented message, it should send an error message to the PCE to signal that it has received an incomplete message (i.e., "Fragmented Instantiation failure").

## 8. Non-Support of P2MP TE LSP Instantiation for Stateful PCE

The PCEP protocol extensions described in this document for PCC or PCE with instantiation capability for P2MP TE LSPs MUST NOT be used if PCC or PCE has not advertised its stateful capability with Instantiation and P2MP capability as per Section 4. If the PCEP Speaker on the PCC supports the extensions of this draft (understands the P (P2MP-LSP-INstantiation-CAPABILITY) flag in the LSP object) but did not advertise this capability, then upon receipt of PCInitiate message from the PCE, it SHOULD generate a PCErr with error-type 19 (Invalid Operation), error-value TBD (Attempted LSP Instantiation Request for P2MP if stateful PCE instantiation capability for P2MP was not advertised).

## 9. Security Considerations

The stateful operations on P2MP TE LSP are more CPU-intensive and also utilize more link bandwidth. In the event of an unauthorized stateful P2MP operations, or a denial of service attack, the subsequent PCEP operations may be disruptive to the network. Consequently, it is important that implementations conform to the relevant security requirements of [RFC5440], [RFC6006], [I-D.ietf-pce-stateful-pce] and [I-D.ietf-pce-pce-initiated-lsp].

## 10. Manageability Considerations

All manageability requirements and considerations listed in [RFC5440], [RFC6006], [I-D.ietf-pce-stateful-pce] and [I-D.ietf-pce-pce-initiated-lsp] apply to PCEP protocol extensions defined in this document. In addition, requirements and considerations listed in this section apply.

### 10.1. Control of Function and Policy

A PCE or PCC implementation **MUST** allow configuring the stateful Initiation capability for P2MP LSPs.

### 10.2. Information and Data Models

The PCEP MIB module **SHOULD** be extended to include advertised P2MP stateful PCE-Initiation capability etc.

### 10.3. 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].

### 10.4. Verify Correct Operations

Mechanisms defined in this document do not imply any new operation verification requirements in addition to those already listed in [RFC5440], [RFC6006] and [I-D.ietf-pce-stateful-pce].

### 10.5. Requirements On Other Protocols

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

## 10.6. 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], [RFC6006] and [I-D.ietf-pce-stateful-pce].

## 11. IANA Considerations

This document requests IANA actions to allocate code points for the protocol elements defined in this document. Values shown here are suggested for use by IANA.

### 11.1. PCE Capabilities in IGP Advertisements

IANA is requested to allocate a new bit in "PCE Capability Flags" registry for PCE-Initiation for P2MP capability as follows:

Bit	Meaning	Reference
TBD	Stateful PCE Initiation with P2MP	[This I-D]

### 11.2. STATEFUL-PCE-CAPABILITY TLV

The following values are defined in this document for the Flags field in the STATEFUL-PCE-CAPABILITY-TLV (defined in [I-D.ietf-pce-stateful-pce]) in the OPEN object:

Bit	Description	Reference
TBD	P2MP-LSP- INSTANTIATION- CAPABILITY	This.I-D

### 11.3. Extension of PCEP-Error Object

A error types 19 (recommended values) is defined in section 8.4 of [I-D.ietf-pce-stateful-pce]. The error-type 18 is deined in [RFC6006]. This document extend the new Error-Values for the error type for the following error conditions:

Error-Type	Meaning
18	P2MP Fragmentation Error Error-value= TBD. Fragmented Instantiation failure
19	Invalid Operation Error-value= TBD. Attempted LSP Instantiation Request for P2MP if stateful PCE instantiation capability for P2MP was not advertised

Upon approval of this document, IANA is requested to make the assignment of a new error value for the existing "PCEP-ERROR Object Error Types and Values" registry located at <http://www.iana.org/assignments/pcep/pcep.xhtml#pcep-error-object>.

## 12. Security Considerations

The security considerations described in [I-D.ietf-pce-stateful-pce] and [I-D.ietf-pce-pce-initiated-lsp] apply to the extensions described in this document. The stateful operations on P2MP TE LSP are more CPU-intensive and also utilize more link bandwidth. In the event of an unauthorized stateful P2MP operations, or a denial of service attack, the subsequent PCEP operations may be disruptive to the network. Consequently, it is important that implementations conform to the relevant security requirements of [RFC5440], [RFC6006], [I-D.ietf-pce-stateful-pce], and [I-D.ietf-pce-pce-initiated-lsp].

## 13. Acknowledgments

Thanks to Quintin Zhao, Avantika and Venugopal Reddy for his comments.

## 14. References

### 14.1. 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, <<http://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, <<http://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, <<http://www.rfc-editor.org/info/rfc5089>>.
- [RFC5440] Vasseur, JP., Ed. and JL. Le Roux, Ed., "Path Computation Element (PCE) Communication Protocol (PCEP)", RFC 5440, DOI 10.17487/RFC5440, March 2009, <<http://www.rfc-editor.org/info/rfc5440>>.
- [RFC6006] Zhao, Q., Ed., King, D., Ed., Verhaeghe, F., Takeda, T., Ali, Z., and J. Meuric, "Extensions to the Path Computation Element Communication Protocol (PCEP) for Point-to-Multipoint Traffic Engineering Label Switched Paths", RFC 6006, DOI 10.17487/RFC6006, September 2010, <<http://www.rfc-editor.org/info/rfc6006>>.
- [I-D.ietf-pce-stateful-pce]  
Crabbe, E., Minei, I., Medved, J., and R. Varga, "PCEP Extensions for Stateful PCE", draft-ietf-pce-stateful-pce-13 (work in progress), December 2015.
- [I-D.ietf-pce-pce-initiated-lsp]  
Crabbe, E., Minei, I., Sivabalan, S., and R. Varga, "PCEP Extensions for PCE-initiated LSP Setup in a Stateful PCE Model", draft-ietf-pce-pce-initiated-lsp-05 (work in progress), October 2015.
- [I-D.ietf-pce-stateful-sync-optimizations]  
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", draft-ietf-pce-stateful-sync-optimizations-04 (work in progress), November 2015.
- [I-D.palle-pce-stateful-pce-p2mp]  
Palle, U., Dhody, D., Tanaka, Y., Ali, Z., and V. Beeram, "Path Computation Element (PCE) Protocol Extensions for Stateful PCE usage for Point-to-Multipoint Traffic Engineering Label Switched Paths", draft-palle-pce-stateful-pce-p2mp-07 (work in progress), June 2015.

## 14.2. 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, <<http://www.rfc-editor.org/info/rfc4655>>.

- [RFC4857] Fogelstroem, E., Jonsson, A., and C. Perkins, "Mobile IPv4 Regional Registration", RFC 4857, DOI 10.17487/RFC4857, June 2007, <<http://www.rfc-editor.org/info/rfc4857>>.
- [RFC4875] Aggarwal, R., Ed., Papadimitriou, D., Ed., and S. Yasukawa, Ed., "Extensions to Resource Reservation Protocol - Traffic Engineering (RSVP-TE) for Point-to-Multipoint TE Label Switched Paths (LSPs)", RFC 4875, DOI 10.17487/RFC4875, May 2007, <<http://www.rfc-editor.org/info/rfc4875>>.
- [RFC5671] Yasukawa, S. and A. Farrel, Ed., "Applicability of the Path Computation Element (PCE) to Point-to-Multipoint (P2MP) MPLS and GMPLS Traffic Engineering (TE)", RFC 5671, DOI 10.17487/RFC5671, October 2009, <<http://www.rfc-editor.org/info/rfc5671>>.
- [I-D.ietf-pce-stateful-pce-app]  
Zhang, X. and I. Minei, "Applicability of a Stateful Path Computation Element (PCE)", draft-ietf-pce-stateful-pce-app-05 (work in progress), October 2015.

Appendix A. Contributor Addresses

Yuji Kamite  
NTT Communications Corporation  
Granpark Tower  
3-4-1 Shibaura, Minato-ku  
Tokyo 108-8118  
Japan

EMail: y.kamite@ntt.com

Authors' Addresses

Udayasree Palle  
Huawei Technologies  
Divyashree Techno Park, Whitefield  
Bangalore, Karnataka 560037  
India

EMail: udayasree.palle@huawei.com

Dhruv Dhody  
Huawei Technologies  
Divyashree Techno Park, Whitefield  
Bangalore, Karnataka 560037  
India

EMail: dhruv.ietf@gmail.com

Yosuke Tanaka  
NTT Communications Corporation  
Granpark Tower  
3-4-1 Shibaura, Minato-ku  
Tokyo 108-8118  
Japan

EMail: yosuke.tanaka@ntt.com

Zafar Ali  
Cisco Systems

EMail: zali@cisco.com

Vishnu Pavan Beeram  
Juniper Networks

EMail: [vbeeram@juniper.net](mailto:vbeeram@juniper.net)



PCE Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: December 11, 2016

U. Palle  
D. Dhody  
Huawei Technologies  
Y. Tanaka  
NTT Communications  
Z. Ali  
Cisco Systems  
V. Beeram  
Juniper Networks  
June 9, 2016

Path Computation Element (PCE) Protocol Extensions for Stateful PCE  
usage for Point-to-Multipoint Traffic Engineering Label Switched Paths  
draft-palle-pce-stateful-pce-p2mp-09

#### Abstract

The Path Computation Element (PCE) has been identified as an appropriate technology for the determination of the paths of point-to-multipoint (P2MP) TE LSPs. This document provides extensions required for Path Computation Element communication Protocol (PCEP) so as to enable the usage of a stateful PCE capability in supporting P2MP TE LSPs.

#### Status of This Memo

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

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

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

This Internet-Draft will expire on December 11, 2016.

#### Copyright Notice

Copyright (c) 2016 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
1.1. Requirements Language . . . . .	4
2. Terminology . . . . .	4
3. Supporting P2MP TE LSP for Stateful PCE . . . . .	4
3.1. Motivation . . . . .	4
3.2. Objectives . . . . .	5
4. Functions to Support P2MP TE LSPs for Stateful PCEs . . . . .	5
5. Architectural Overview of Protocol Extensions . . . . .	6
5.1. Extension of PCEP Messages . . . . .	6
5.2. Capability Advertisement . . . . .	6
5.3. IGP Extensions for Stateful PCE P2MP Capabilities Advertisement . . . . .	7
5.4. State Synchronization . . . . .	8
5.5. LSP Delegation . . . . .	8
5.6. LSP Operations . . . . .	8
5.6.1. Passive Stateful PCE . . . . .	8
5.6.2. Active Stateful PCE . . . . .	9
5.6.3. PCE-Initiated LSP . . . . .	9
5.6.3.1. P2MP TE LSP Instantiation . . . . .	9
5.6.3.2. P2MP TE LSP Deletion . . . . .	9
5.6.3.3. Adding and Pruning Leaves for the P2MP TE LSP . . . . .	9
5.6.3.4. P2MP TE LSP Delegation and Cleanup . . . . .	10
6. PCEP Message Extensions . . . . .	10
6.1. The PCRpt Message . . . . .	10
6.2. The PCUpd Message . . . . .	12
6.3. The PCReq Message . . . . .	13
6.4. The PCRep Message . . . . .	14
6.5. The PCInitiate message . . . . .	15
6.6. Example . . . . .	17
6.6.1. P2MP TE LSP Update Request . . . . .	17
6.6.2. P2MP TE LSP Report . . . . .	17
7. PCEP Object Extensions . . . . .	18
7.1. Extension of LSP Object . . . . .	18
7.2. P2MP-LSP-IDENTIFIER TLV . . . . .	19
7.3. S2LS Object . . . . .	21
8. Message Fragmentation . . . . .	22

8.1.	Report Fragmentation Procedure . . . . .	22
8.2.	Update Fragmentation Procedure . . . . .	23
8.3.	PCInitiate Fragmentation Procedure . . . . .	23
9.	Non-Support of P2MP TE LSPs for Stateful PCE . . . . .	23
10.	Manageability Considerations . . . . .	24
10.1.	Control of Function and Policy . . . . .	24
10.2.	Information and Data Models . . . . .	24
10.3.	Liveness Detection and Monitoring . . . . .	25
10.4.	Verify Correct Operations . . . . .	25
10.5.	Requirements On Other Protocols . . . . .	25
10.6.	Impact On Network Operations . . . . .	25
11.	IANA Considerations . . . . .	25
11.1.	PCE Capabilities in IGP Advertisements . . . . .	25
11.2.	STATEFUL-PCE-CAPABILITY TLV . . . . .	26
11.3.	Extension of LSP Object . . . . .	26
11.4.	Extension of PCEP-Error Object . . . . .	26
11.5.	PCEP TLV Type Indicators . . . . .	27
12.	Security Considerations . . . . .	28
13.	Acknowledgments . . . . .	28
14.	References . . . . .	28
14.1.	Normative References . . . . .	28
14.2.	Informative References . . . . .	29
Appendix A.	Contributor Addresses . . . . .	31
Authors'	Addresses . . . . .	31

## 1. Introduction

As per [RFC4655], the Path Computation Element (PCE) is an entity that is capable of computing a network path or route based on a network graph, and applying computational constraints. A Path Computation Client (PCC) may make requests to a PCE for paths to be computed.

[RFC4857] describes how to set up point-to-multipoint (P2MP) Traffic Engineering Label Switched Paths (TE LSPs) for use in Multiprotocol Label Switching (MPLS) and Generalized MPLS (GMPLS) networks. The PCE has been identified as a suitable application for the computation of paths for P2MP TE LSPs ([RFC5671]).

The PCEP is designed as a communication protocol between PCCs and PCEs for point-to-point (P2P) path computations and is defined in [RFC5440]. The extensions of PCEP to request path computation for P2MP TE LSPs are described in [RFC6006].

Stateful PCEs are shown to be helpful in many application scenarios, in both MPLS and GMPLS networks, as illustrated in [I-D.ietf-pce-stateful-pce-app]. These scenarios apply equally to P2P and P2MP TE LSPs. [I-D.ietf-pce-stateful-pce] provides the

fundamental extensions needed for stateful PCE to support general functionality for P2P TE LSP. [I-D.ietf-pce-pce-initiated-lsp] provides the an extensions needed for stateful PCE-initiated P2P TE LSP. Complementarily, this document focuses on the extensions that are necessary in order for the deployment of stateful PCEs to support P2MP TE LSPs. This document describes the setup, maintenance and teardown of PCE-initiated P2MP LSPs under the stateful PCE model.

### 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].

## 2. Terminology

Terminology used in this document is same as terminology used in [I-D.ietf-pce-stateful-pce], [I-D.ietf-pce-pce-initiated-lsp], and [RFC6006].

## 3. Supporting P2MP TE LSP for Stateful PCE

### 3.1. Motivation

[I-D.ietf-pce-stateful-pce-app] presents several use cases, demonstrating scenarios that benefit from the deployment of a stateful PCE including optimization, recovery, etc which are equally applicable to P2MP TE LSPs. [I-D.ietf-pce-stateful-pce] defines the extensions to PCEP for P2P TE LSPs. Complementarily, this document focuses on the extensions that are necessary in order for the deployment of stateful PCEs to support P2MP TE LSPs.

In addition to that, the stateful nature of a PCE simplifies the information conveyed in PCEP messages since it is possible to refer to the LSPs via PLSP-ID ([I-D.ietf-pce-stateful-pce]). For P2MP this is an added advantage, where the size of message is much larger. In case of stateless PCE, a modification of P2MP tree requires encoding of all leaves along with the paths in PCReq message, but using a stateful PCE with P2MP capability, the PCEP message can be used to convey only the modifications (the other information can be retrieved from the P2MP LSP identifier in the LSP database (LSPDB)).

In environments where the P2MP TE LSP placement needs to change in response to application demands, it is useful to support dynamic creation and tear down of P2MP TE LSPs. The ability for a PCE to trigger the creation of P2MP TE LSPs on demand can be seamlessly integrated into a controller-based network architecture, where intelligence in the controller can determine when and where to set up

paths. Section 3 of [I-D.ietf-pce-pce-initiated-lsp] further describes the motivation behind the PCE-Initiation capability, which are equally applicable for P2MP TE LSPs.

### 3.2. Objectives

The objectives for the protocol extensions to support P2MP TE LSP for stateful PCE are same as the objectives described in section 3.2 of [I-D.ietf-pce-stateful-pce].

## 4. Functions to Support P2MP TE LSPs for Stateful PCEs

[I-D.ietf-pce-stateful-pce] specifies new functions to support a stateful PCE. It also specifies that a function can be initiated either from a PCC towards a PCE (C-E) or from a PCE towards a PCC (E-C).

This document extends these functions to support P2MP TE LSPs.

Capability Advertisement (E-C,C-E): both the PCC and the PCE must announce during PCEP session establishment that they support PCEP Stateful PCE extensions for P2MP using mechanisms defined in Section 5.2.

LSP State Synchronization (C-E): after the session between the PCC and a stateful PCE with P2MP capability is initialized, the PCE must learn the state of a PCC's P2MP TE LSPs before it can perform path computations or update LSP attributes in a PCC.

LSP Update Request (E-C): a stateful PCE with P2MP capability requests modification of attributes on a PCC's P2MP TE LSP.

LSP State Report (C-E): a PCC sends an LSP state report to a PCE whenever the state of a P2MP TE LSP changes.

LSP Control Delegation (C-E,E-C): a PCC grants to a PCE the right to update LSP attributes on one or more P2MP TE LSPs; the PCE becomes the authoritative source of the LSP's attributes as long as the delegation is in effect (See Section 5.7 of [I-D.ietf-pce-stateful-pce]); the PCC may withdraw the delegation or the PCE may give up the delegation at any time.

PCE-initiated LSP instantiation (E-C): a PCE sends an LSP Initiate Message to a PCC to instantiate or delete a P2MP TE LSP.

## 5. Architectural Overview of Protocol Extensions

### 5.1. Extension of PCEP Messages

New PCEP messages are defined in [I-D.ietf-pce-stateful-pce] to support stateful PCE for P2P TE LSPs. In this document these messages are extended to support P2MP TE LSPs.

**Path Computation State Report (PCRpt):** Each P2MP TE LSP State Report in a PCRpt message can contain actual P2MP TE LSP path attributes, LSP status, etc. An LSP State Report carried on a PCRpt message is also used in delegation or revocation of control of a P2MP TE LSP to/from a PCE. The extension of PCRpt message is described in Section 6.1.

**Path Computation Update Request (PCUpd):** Each P2MP TE LSP Update Request in a PCUpd message MUST contain all LSP parameters that a PCE wishes to set for a given P2MP TE LSP. An LSP Update Request carried on a PCUpd message is also used to return LSP delegations if at any point PCE no longer desires control of a P2MP TE LSP. The PCUpd message is described in Section 6.2.

A new PCEP message is defined in [I-D.ietf-pce-pce-initiated-lsp] to support stateful PCE instantiation of P2P TE LSPs. In this document this message is extended to support P2MP TE LSPs.

**Path Computation LSP Initiate Message (PCInitiate):** is a PCEP message sent by a PCE to a PCC to trigger P2MP TE LSP instantiation or deletion. The PCInitiate message is described in Section 6.5.

### 5.2. Capability Advertisement

During PCEP Initialization Phase, as per Section 7.1.1 of [I-D.ietf-pce-stateful-pce], PCEP speakers advertises Stateful capability via Stateful PCE Capability TLV in open message. Two new flags are defined for the STATEFUL-PCE-CAPABILITY TLV defined in [I-D.ietf-pce-stateful-pce] and updated in [I-D.ietf-pce-pce-initiated-lsp] and [I-D.ietf-pce-stateful-sync-optimizations].

Three new bits N (P2MP-CAPABILITY), M (P2MP-LSP-UPDATE-CAPABILITY), and P (P2MP-LSP-INSTANTIATION-CAPABILITY) are added in this document:

N (P2MP-CAPABILITY - 1 bit): if set to 1 by a PCC, the N Flag indicates that the PCC is willing to send P2MP LSP State Reports whenever P2MP LSP parameters or operational status changes.; if set to 1 by a PCE, the N Flag indicates that the PCE is interested

in receiving LSP State Reports whenever LSP parameters or operational status changes. The P2MP-CAPABILITY Flag must be advertised by both a PCC and a PCE for PCRpt messages P2MP extension to be allowed on a PCEP session.

M (P2MP-LSP-UPDATE-CAPABILITY - 1 bit): if set to 1 by a PCC, the M Flag indicates that the PCC allows modification of P2MP LSP parameters; if set to 1 by a PCE, the M Flag indicates that the PCE is capable of updating P2MP LSP parameters. The P2MP-LSP-UPDATE-CAPABILITY Flag must be advertised by both a PCC and a PCE for PCUpd messages P2MP extension to be allowed on a PCEP session.

P (P2MP-LSP-INSTITIATION-CAPABILITY - 1 bit): If set to 1 by a PCC, the P Flag indicates that the PCC allows instantiation of an P2MP LSP by a PCE. If set to 1 by a PCE, the P flag indicates that the PCE supports P2MP LSP instantiation. The P2MP-LSP-INSTITIATION-CAPABILITY flag must be set by both PCC and PCE in order to support PCE-initiated P2MP LSP instantiation.

A PCEP speaker should continue to advertise the basic P2MP capability via mechanisms as described in [RFC6006].

### 5.3. IGP Extensions for Stateful PCE P2MP Capabilities Advertisement

When PCCs are LSRs participating in the IGP (OSPF or IS-IS), and PCEs are either LSRs or servers also participating in the IGP, an effective mechanism for PCE discovery within an IGP routing domain consists of utilizing IGP advertisements. Extensions for the advertisement of PCE Discovery Information are defined for OSPF and for IS-IS in [RFC5088] and [RFC5089] respectively.

The PCE-CAP-FLAGS sub-TLV, defined in [RFC5089], is an optional sub-TLV used to advertise PCE capabilities. It MAY be present within the PCED sub-TLV carried by OSPF or IS-IS. [RFC5088] and [RFC5089] provide the description and processing rules for this sub-TLV when carried within OSPF and IS-IS, respectively.

The format of the PCE-CAP-FLAGS sub-TLV is included below for easy reference:

Type: 5

Length: Multiple of 4.

Value: This contains an array of units of 32 bit flags with the most significant bit as 0. Each bit represents one PCE capability.

PCE capability bits are defined in [RFC5088]. This document defines new capability bits for the stateful PCE with P2MP as follows:

Bit	Capability
TBD	Active Stateful PCE with P2MP
TBD	Passive Stateful PCE with P2MP
TBD	PCE-Initiation with P2MP

Note that while active, passive or initiation stateful PCE with P2MP capabilities may be advertised during discovery, PCEP Speakers that wish to use stateful PCEP MUST advertise stateful PCEP capabilities during PCEP session setup, as specified in the current document. A PCC MAY initiate stateful PCEP P2MP capability advertisement at PCEP session setup even if it did not receive any IGP PCE capability advertisements.

#### 5.4. State Synchronization

State Synchronization operations described in Section 5.6 of [I-D.ietf-pce-stateful-pce] are applicable for P2MP TE LSPs as well.

#### 5.5. LSP Delegation

LSP delegation operations described in Section 5.7 of [I-D.ietf-pce-stateful-pce] are applicable for P2MP TE LSPs as well.

#### 5.6. LSP Operations

##### 5.6.1. Passive Stateful PCE

LSP operations for passive stateful PCE described in Section 5.8.1 of [I-D.ietf-pce-stateful-pce] are applicable for P2MP TE LSPs as well.

The Path Computation Request and Response message format for P2MP TE LSPs is described in Section 3.4 and Section 3.5 of [RFC6006] respectively.

The Request and Response message for P2MP TE LSPs are extended to support encoding of LSP object, so that it is possible to refer to a LSP with a unique identifier and simplify the PCEP message exchange. For example, incase of modification of one leaf in a P2MP tree, there should be no need to carry the full P2MP tree in PCReq message.

The extension for the Request and Response message for passive stateful operations on P2MP TE LSPs are described in Section 6.3 and Section 6.4. The extension for the Path Computation LSP State Report (PCRpt) message is described in Section 6.1.



### 5.6.2. Active Stateful PCE

LSP operations for active stateful PCE described in Section 5.8.2 of [I-D.ietf-pce-stateful-pce] are applicable for P2MP TE LSPs as well.

The extension for the Path Computation LSP Update (PCUpd) message for active stateful operations on P2MP TE LSPs are described in Section 6.2.

### 5.6.3. PCE-Initiated LSP

As per section 5.1 of [I-D.ietf-pce-pce-initiated-lsp], the PCE sends a Path Computation LSP Initiate Request (PCInitiate) message to the PCC to suggest instantiation or deletion of a P2P TE LSP. This document extends the PCInitiate message to support P2MP TE LSP (see details in Section 6.5).

P2MP TE LSP suggested instantiation and deletion operations are same as P2P LSP as described in section 5.3 and 5.4 of [I-D.ietf-pce-pce-initiated-lsp].

#### 5.6.3.1. P2MP TE LSP Instantiation

The Instantiation operation of P2MP TE LSP is same as defined in section 5.3 of [I-D.ietf-pce-pce-initiated-lsp] including handling of PLSP-ID, SYMBOLIC-PATH-NAME TLV etc. Rules of processing and error codes remains unchanged. The N bit MUST be set in LSP object in PCInitiate message by PCE to specify the instantiation is for P2MP TE LSP.

Though N bit is set in the LSP object, P2MP-LSP-IDENTIFIER TLV MUST NOT be included in the LSP object in PCInitiate message as it SHOULD be generated by PCC and carried in PCRpt message.

#### 5.6.3.2. P2MP TE LSP Deletion

The deletion operation of P2MP TE LSP is same as defined in section 5.4 of [I-D.ietf-pce-pce-initiated-lsp] by sending an LSP Initiate Message with an LSP object carrying the PLSP-ID of the LSP to be removed and an SRP object with the R flag set (LSP-REMOVE as per section 5.2 of [I-D.ietf-pce-pce-initiated-lsp]). Rules of processing and error codes remains unchanged.

#### 5.6.3.3. Adding and Pruning Leaves for the P2MP TE LSP

Adding of new leaves and Pruning of old Leaves for the PCE initiated P2MP TE LSP MUST be carried in PCUpd message and SHOULD refer Section 6.2 for P2MP TE LSP extensions. As defined in [RFC6006],

leaf type = 1 for adding of new leaves, leaf type = 2 for pruning of old leaves of P2MP END-POINTS Object are used in PCUpd message.

PCC MAY use the Incremental State Update mechanisms as described in [RFC4875] to signal adding and pruning of leaves.

#### 5.6.3.4. P2MP TE LSP Delegation and Cleanup

P2MP TE LSP delegation and cleanup operations are same as defined in section 6 of [I-D.ietf-pce-pce-initiated-lsp]. Rules of processing and error codes remains unchanged.

### 6. PCEP Message Extensions

#### 6.1. The PCRpt Message

As per Section 6.1 of [I-D.ietf-pce-stateful-pce], PCRpt message is used to report the current state of a P2P TE LSP. This document extends the PCRpt message in reporting the status of P2MP TE LSP.

The format of PCRpt message is as follows:

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

Where:

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

<state-report> ::= [<SRP>]  
                            <LSP>  
                            <end-point-path-pair-list>  
                            <attribute-list>

Where:

<end-point-path-pair-list> ::=  
                            [<END-POINTS>]  
                            [<S2LS>]  
                            <intended\_path>  
                            [<actual\_path>]  
                            [<end-point-path-pair-list>]

<intended\_path> ::= (<ERO>|<SERO>)  
                            [<intended\_path>]

<actual\_path> ::= (<RRO>|<SRRO>)  
                            [<actual\_path>]

<attribute-list> is defined in [RFC5440] and  
extended by PCEP extensions.

The P2MP END-POINTS object defined in [RFC6006] is mandatory for  
specifying address of P2MP leaves grouped based on leaf types.

- o New leaves to add (leaf type = 1)
- o Old leaves to remove (leaf type = 2)
- o Old leaves whose path can be modified/reoptimized (leaf type = 3)
- o Old leaves whose path must be left unchanged (leaf type = 4)

When reporting the status of a P2MP TE LSP, the destinations are  
grouped in END-POINTS object based on the operational status (O field  
in S2LS object) and leaf type (in END-POINTS). This way the leaves  
that share the same operational status are grouped together. For  
reporting the status of delegated P2MP TE LSP, leaf-type = 3, where  
as for non-delegated P2MP TE LSP, leaf-type = 4 is used.

For delegated P2MP TE LSP configuration changes are reported via PCRpt message. For example, adding of new leaves END-POINTS (leaf-type = 1) is used where as removing of old leaves (leaf-type = 2) is used.

Note that we preserve compatibility with the [I-D.ietf-pce-stateful-pce] definition of <state-report>. At least one instance of <END-POINTS> MUST be present in this message for P2MP LSP.

During state synchronization, the PCRpt message must report the status of the full P2MP TE LSP.

The S2LS object MUST be carried in PCRpt message along with END-POINTS object when N bit is set in LSP object for P2MP TE LSP. If the S2LS object is missing, the receiving PCE MUST send a PCErr message with Error-type=6 (Mandatory Object missing) and Error-value=TBD (S2LS object missing). If the END-POINTS object is missing, the receiving PCE MUST send a PCErr message with Error-type=6 (Mandatory Object missing) and Error-value=3 (END-POINTS object missing) (defined in [RFC5440]).

## 6.2. The PCUpd Message

As per Section 6.2 of [I-D.ietf-pce-stateful-pce], PCUpd message is used to update P2P TE LSP attributes. This document extends the PCUpd message in updating the attributes of P2MP TE LSP.

The format of a PCUpd message is as follows:

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

Where:

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

```
<update-request> ::= <SRP>
                        <LSP>
                        <end-point-path-pair-list>
```

```
<attribute-list>
```

Where:

```
<end-point-path-pair-list> ::=
                        [<END-POINTS>]
                        <path>
                        [<end-point-path-pair-list>]
```

```
<path> ::= (<ERO>|<SERO>)
            [<path>]
```

<attribute-list> is defined in [RFC5440] and extended by PCEP extensions.

Note that we preserve compatibility with the [I-D.ietf-pce-stateful-pce] definition of <update-request>.

The PCC MAY use the make-before-break or sub-group-based procedures described in [RFC4875] based on a local policy decision.

The END-POINTS object MUST be carried in PCUpd message when N bit is set in LSP object for P2MP TE LSP. If the END-POINTS object is missing, the receiving PCC MUST send a PCErr message with Error-type=6 (Mandatory Object missing) and Error-value=3 (END-POINTS object missing) (defined in [RFC5440]).

### 6.3. The PCReq Message

As per Section 3.4 of [RFC6006], PCReq message is used for a P2MP path computation request. This document extends the PCReq message such that a PCC MAY include the LSP object in the PCReq message if the stateful PCE P2MP capability has been negotiated on a PCEP session between the PCC and a PCE.

The format of PCReq message is as follows:

```

<PCReq Message> ::= <Common Header>
                    <request>

```

where:

```

<request> ::= <RP>
              <end-point-rro-pair-list>
              [<LSP>]
              [<OF>]
              [<LSPA>]
              [<BANDWIDTH>]
              [<metric-list>]
              [<IRO>]
              [<LOAD-BALANCING>]

```

where:

```

<end-point-rro-pair-list> ::= <END-POINTS> [<RRO-List>] [<BANDWIDTH>]
                             [<end-point-rro-pair-list>]

```

```

<RRO-List> ::= (<RRO> | <SRRO>) [<BANDWIDTH>] [<RRO-List>]

```

```

<metric-list> ::= <METRIC> [<metric-list>]

```

#### 6.4. The PCRep Message

As per Section 3.5 of [RFC6006], PCRep message is used for a P2MP path computation reply. This document extends the PCRep message such that a PCE MAY include the LSP object in the PCRep message if the stateful PCE P2MP capability has been negotiated on a PCEP session between the PCC and a PCE.

The format of PCRep message is as follows:

```
<PCRep Message> ::= <Common Header>
                        <response>
```

```
<response> ::= <RP>
                [<end-point-path-pair-list>]
                [<NO-PATH>]
                [<attribute-list>]
```

where:

```
<end-point-path-pair-list> ::=
    [<END-POINTS>]<path> [<end-point-path-pair-list>]
```

```
<path> ::= (<ERO> | <SERO>) [<path>]
```

```
<attribute-list> ::= [<LSP>]
                     [<OF>]
                     [<LSPA>]
                     [<BANDWIDTH>]
                     [<metric-list>]
                     [<IRO>]
```

#### 6.5. The PCInitiate message

As defined in section 5.1 of [I-D.ietf-pce-pce-initiated-lsp], PCE sends a PCInitiate message to a PCC to recommend instantiation of a P2P TE LSP, this document extends the format of PCInitiate message for the creation of P2MP TE LSPs but the creation and deletion operations of P2MP TE LSP are same to the P2P TE LSP.

The format of PCInitiate message is 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-point-path-pair-list>
                                       [<attribute-list>]
```

```
<PCE-initiated-lsp-deletion> ::= <SRP>
                                   <LSP>
```

Where:

```
<end-point-path-pair-list> ::=
    [<END-POINTS>]
    <path>
    [<end-point-path-pair-list>]
```

```
<path> ::= (<ERO> | <SERO>)
            [<path>]
```

<attribute-list> is defined in [RFC5440] and extended by PCEP extensions.

The PCInitiate message with an LSP object with N bit (P2MP) set is used to convey operation on a P2MP TE LSP. The SRP object is used to correlate between initiation requests sent by the PCE and the error reports and state reports sent by the PCC as described in [I-D.ietf-pce-stateful-pce].

The END-POINTS object MUST be carried in PCInitiate message when N bit is set in LSP object for P2MP TE LSP. If the END-POINTS object is missing, the receiving PCC MUST send a PCErr message with Error-type=6 (Mandatory Object missing) and Error-value=3 (END-POINTS object missing) (defined in [RFC5440]).



## 6.6. Example

### 6.6.1. P2MP TE LSP Update Request

LSP Update Request message is sent by an active stateful PCE to update the P2MP TE LSP parameters or attributes. An example of a PCUpd message for P2MP TE LSP is described below:

```
Common Header
SRP
LSP with P2MP flag set
END-POINTS for leaf type 3
ERO list
```

In this example, a stateful PCE request updation of path taken by some of the leaves in a P2MP tree. The update request uses the END-POINT type 3 (modified/reoptimized). The ERO list represents the S2LS path after modification. The update message does not need to encode the full P2MP tree in this case.

### 6.6.2. P2MP TE LSP Report

LSP State Report message is sent by a PCC to report or delegate the P2MP TE LSP. An example of a PCRpt message for a delegated P2MP TE LSP is described below to add new leaves to an existing P2MP TE LSP:

```
Common Header
LSP with P2MP flag set
END-POINTS for leaf type 1
S2LS (O=DOWN)
ERO list (empty)
```

An example of a PCRpt message for P2MP TE LSP is described below to prune leaves from an existing P2MP TE LSP:

```
Common Header
LSP with P2MP flag set
END-POINTS for leaf type 2
S2LS (O=UP)
ERO list
```

An example of a PCRpt message for a delegated P2MP TE LSP is described below to report status of leaves in an existing P2MP TE LSP:

```
Common Header
LSP with P2MP flag set
END-POINTS for leaf type 3
  S2LS (O=UP)
  ERO list
END-POINTS for leaf type 3
  S2LS (O=DOWN)
  ERO list
```

An example of a PCRpt message for a non-delegated P2MP TE LSP is described below to report status of leaves:

```
Common Header
LSP with P2MP flag set
END-POINTS for leaf type 4
  S2LS (O=ACTIVE)
  ERO list
END-POINTS for leaf type 4
  S2LS (O=DOWN)
  ERO list
```

## 7. PCEP Object Extensions

The PCEP TLV defined in this document is compliant with the PCEP TLV format defined in [RFC5440].

### 7.1. Extension of LSP Object

LSP Object is defined in Section 7.3 of [I-D.ietf-pce-stateful-pce]. It specifies PLSP-ID to uniquely identify an LSP that is constant for the life time of a PCEP session. Similarly for P2MP tunnel, PLSP-ID identify a P2MP TE LSP uniquely. This document adds the following flags to the LSP Object:

N (P2MP bit): If the bit is set to 1, it specifies the message is for P2MP TE LSP which MUST be set in PCRpt or PCUpd message for a P2MP TE LSP.

F (Fragmentation bit): If the bit is set to 1, it specifies the message is fragmented.

If P2MP bit is set, the following P2MP-LSP-IDENTIFIER TLV MUST be present in LSP object.

## 7.2. P2MP-LSP-IDENTIFIER TLV

The P2MP LSP Identifier TLV MUST be included in the LSP object in PCRpt message for RSVP-TE signaled P2MP TE LSPs. If the TLV is missing, the PCE will generate an error with error-type 6 (mandatory object missing) and error-value TBD (P2MP-LSP-IDENTIFIERS TLV missing) and close the PCEP session.

The P2MP LSP Identifier TLV MAY be included in the LSP object in PCUpd message for RSVP-TE signaled P2MP TE LSPs. The special value of all zeros for this TLV is used to refer to all paths pertaining to a particular PLSP-ID.

There are two P2MP LSP Identifier TLVs, one for IPv4 and one for IPv6.

The format of the IPV4-P2MP-LSP-IDENTIFIER TLV is shown in the following figure:

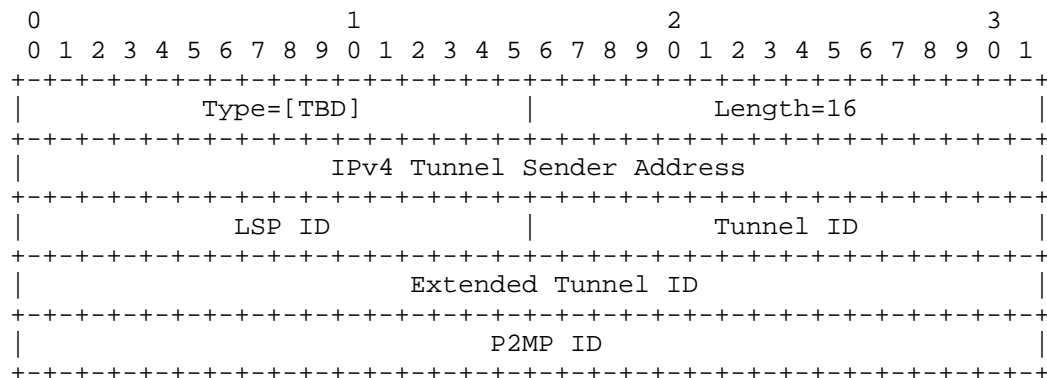


Figure 6: IPV4-P2MP-LSP-IDENTIFIER TLV format

The type (16-bit) of the TLV is [TBD] to be assigned by IANA. The length (16-bit) has a fixed value of 16 octets. The value contains the following fields:

IPv4 Tunnel Sender Address: contains the sender node's IPv4 address, as defined in [RFC3209], Section 4.6.2.1 for the LSP\_TUNNEL\_IPv4 Sender Template Object.

LSP ID: contains the 16-bit 'LSP ID' identifier defined in [RFC3209], Section 4.6.2.1 for the LSP\_TUNNEL\_IPv4 Sender Template Object.

Tunnel ID: contains the 16-bit 'Tunnel ID' identifier defined in [RFC3209], Section 4.6.1.1 for the LSP\_TUNNEL\_IPv4 Session Object.

Extended Tunnel ID: contains the 32-bit 'Extended Tunnel ID' identifier defined in [RFC3209], Section 4.6.1.1 for the LSP\_TUNNEL\_IPv4 Session Object.

P2MP ID: contains the 32-bit 'P2MP ID' identifier defined in Section 19.1.1 of [RFC4875] for the P2MP LSP Tunnel IPv4 SESSION Object.

The format of the IPV6-P2MP-LSP-IDENTIFIER TLV is shown in the following figure:

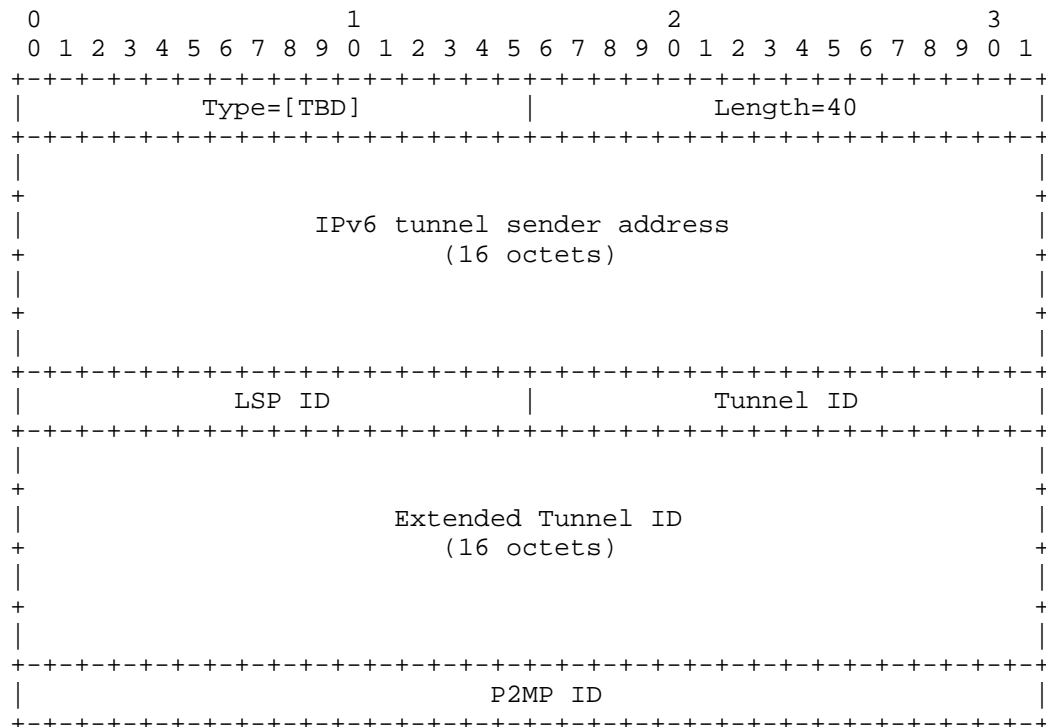


Figure 7: IPV6-P2MP-LSP-IDENTIFIER TLV format

The type of the TLV is [TBD] to be assigned by IANA. The length (16-bit) has a fixed length of 40 octets. The value contains the following fields:

IPv6 Tunnel Sender Address: contains the sender node's IPv6 address, as defined in [RFC3209], Section 4.6.2.2 for the LSP\_TUNNEL\_IPv6 Sender Template Object.

LSP ID: contains the 16-bit 'LSP ID' identifier defined in [RFC3209], Section 4.6.2.2 for the LSP\_TUNNEL\_IPv6 Sender Template Object.

Tunnel ID: contains the 16-bit 'Tunnel ID' identifier defined in [RFC3209], Section 4.6.1.2 for the LSP\_TUNNEL\_IPv6 Session Object.

Extended Tunnel ID: contains the 128-bit 'Extended Tunnel ID' identifier defined in [RFC3209], Section 4.6.1.2 for the LSP\_TUNNEL\_IPv6 Session Object.

P2MP ID: As defined above in IPV4-P2MP-LSP-IDENTIFIERS TLV.

Tunnel ID remains constant over the life time of a tunnel.

### 7.3. S2LS Object

The S2LS (Source-to-Leaves) Object is used to report RSVP-TE state of one or more destinations (leaves) encoded within the END-POINTS object for a P2MP TE LSP. It MUST be carried in PCRpt message along with END-POINTS object when N bit is set in LSP object.

S2LS Object-Class is [TBD].

S2LS Object-Types is 1.

The format of the S2LS object is shown in the following figure:

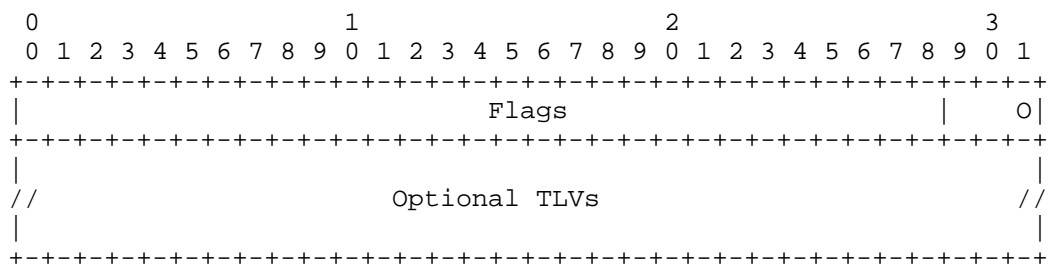


Figure 8: S2LS object format

Flags(32 bits):

O(Operational - 3 bits) the O Field represents the operational status of the group of destinations. The values are as per Operational field in LSP object defined in Section 7.3 of [I-D.ietf-pce-stateful-pce].

When N bit is set in LSP object then the O field in LSP object represents the operational status of the full P2MP TE LSP and the O field in S2LS object represents the operational status of a group of destinations encoded within the END-POINTS object.

Future documents MAY define optional TLVs that MAY be included in the S2LS Object.

## 8. Message Fragmentation

The total PCEP message length, including the common header, is 16 bytes. In certain scenarios the P2MP report and update request may not fit into a single PCEP message (e.g. initial report or update). The F-bit is used in the LSP object to signal that the initial report, update, or initiate message was too large to fit into a single message and will be fragmented into multiple messages. In order to identify the single report or update each message will use the same PLSP-ID. In order to identify that a series of PCInitiate messages represents a single Initiate, each message will use the same PLSP-ID (in this case 0) and SRP-ID-number.

Fragmentation procedure described below for report or update message is similar to [RFC6006] which describes request and response message fragmentation.

### 8.1. Report Fragmentation Procedure

If the initial report is too large to fit into a single report message, the PCC will split the report over multiple messages. Each message sent to the PCE, except the last one, will have the F-bit set in the LSP object to signify that the report has been fragmented into multiple messages. In order to identify that a series of report messages represents a single report, each message will use the same PLSP-ID.

To indicate P2MP message fragmentation errors associated with a P2MP Report, a Error-Type (18) and a new error-value TBD is used if a PCE has not received the last piece of the fragmented message, it should send an error message to the PCC to signal that it has received an incomplete message (i.e., "Fragmented Report failure").

## 8.2. Update Fragmentation Procedure

Once the PCE computes and updates a path for some or all leaves in a P2MP TE LSP, an update message is sent to the PCC. If the update is too large to fit into a single update message, the PCE will split the update over multiple messages. Each update message sent by the PCE, except the last one, will have the F-bit set in the LSP object to signify that the update has been fragmented into multiple messages. In order to identify that a series of update messages represents a single update, each message will use the same PLSP-ID and SRP-ID-number.

To indicate P2MP message fragmentation errors associated with a P2MP Update request, a Error-Type (18) and a new error-value TBD is used if a PCC has not received the last piece of the fragmented message, it should send an error message to the PCE to signal that it has received an incomplete message (i.e., "Fragmented Update failure").

## 8.3. PCInitiate Fragmentation Procedure

Once the PCE initiates to set up the P2MP TE LSP, a PCInitiate message is sent to the PCC. If the PCInitiate is too large to fit into a single PCInitiate message, the PCE will split the PCInitiate over multiple messages. Each PCInitiate message sent by the PCE, except the last one, will have the F-bit set in the LSP object to signify that the PCInitiate has been fragmented into multiple messages. In order to identify that a series of PCInitiate messages represents a single Initiate, each message will use the same PLSP-ID (in this case 0) and SRP-ID-number.

To indicate P2MP message fragmentation errors associated with a P2MP PCInitiate, a Error-Type (18) and a new error-value TBD is used if a PCC has not received the last piece of the fragmented message, it should send an error message to the PCE to signal that it has received an incomplete message (i.e., "Fragmented Instantiation failure").

## 9. Non-Support of P2MP TE LSPs for Stateful PCE

The PCEP protocol extensions described in this document for stateful PCEs with P2MP capability MUST NOT be used if PCE has not advertised its stateful capability with P2MP as per Section 5.2. If the PCEP Speaker on the PCC supports the extensions of this draft (understands the P2MP flag in the LSP object) but did not advertise this capability, then upon receipt of PCUpd message from the PCE, it SHOULD generate a PCErr with error-type 19 (Invalid Operation), error-value TBD (Attempted LSP Update Request for P2MP if active stateful PCE capability for P2MP was not advertised). If the PCEP

Speaker on the PCE supports the extensions of this draft (understands the P2MP flag in the LSP object) but did not advertise this capability, then upon receipt of a PCRpt message from the PCC, it SHOULD generate a PCErr with error-type 19 (Invalid Operation), error-value TBD (Attempted LSP State Report for P2MP if stateful PCE capability for P2MP was not advertised) and it will terminate the PCEP session.

If a Stateful PCE receives a P2MP TE LSP report message and the PCE does not understand the P2MP flag in the LSP object, and therefore the PCEP extensions described in this document, then the Stateful PCE would act as per [I-D.ietf-pce-stateful-pce].

The PCEP protocol extensions described in this document for PCC or PCE with instantiation capability for P2MP TE LSPs MUST NOT be used if PCC or PCE has not advertised its stateful capability with Instantiation and P2MP capability as per Section 5.2. If the PCEP Speaker on the PCC supports the extensions of this draft (understands the P (P2MP-LSP-INSTANTIATION-CAPABILITY) flag in the LSP object) but did not advertise this capability, then upon receipt of PCInitiate message from the PCE, it SHOULD generate a PCErr with error-type 19 (Invalid Operation), error-value TBD (Attempted LSP Instantiation Request for P2MP if stateful PCE instantiation capability for P2MP was not advertised).

## 10. Manageability Considerations

All manageability requirements and considerations listed in [RFC5440], [RFC6006], [I-D.ietf-pce-stateful-pce], and [I-D.ietf-pce-pce-initiated-lsp] apply to PCEP protocol extensions defined in this document. In addition, requirements and considerations listed in this section apply.

### 10.1. Control of Function and Policy

A PCE or PCC implementation MUST allow configuring the stateful PCEP capability, the LSP Update capability, and the LSP Initiation capability for P2MP LSPs.

### 10.2. Information and Data Models

The PCEP MIB module SHOULD be extended to include advertised P2MP stateful capabilities, P2MP synchronization status, and P2MP delegation status etc.



### 10.3. 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].

### 10.4. Verify Correct Operations

Mechanisms defined in this document do not imply any new operation verification requirements in addition to those already listed in [RFC5440], [RFC6006], [I-D.ietf-pce-stateful-pce], and [I-D.ietf-pce-pce-initiated-lsp].

### 10.5. Requirements On Other Protocols

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

### 10.6. 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], [RFC6006], [I-D.ietf-pce-stateful-pce], and [I-D.ietf-pce-pce-initiated-lsp].

## 11. IANA Considerations

This document requests IANA actions to allocate code points for the protocol elements defined in this document.

### 11.1. PCE Capabilities in IGP Advertisements

IANA is requested to allocate new bits in "PCE Capability Flags" registry for stateful PCE with P2MP capability as follows:

Bit	Meaning	Reference
TBD	Active Stateful PCE with P2MP	[This I-D]
TBD	Passive Stateful PCE with P2MP	[This I-D]
TBD	Stateful PCE Initiation with P2MP	[This I-D]

### 11.2. STATEFUL-PCE-CAPABILITY TLV

The following values are defined in this document for the Flags field in the STATEFUL-PCE-CAPABILITY-TLV (defined in [I-D.ietf-pce-stateful-pce]) in the OPEN object:

Bit	Description	Reference
TBD	P2MP-CAPABILITY	This.I-D
TBD	P2MP-LSP-UPDATE-CAPABILITY	This.I-D
TBD	P2MP-LSP-INSTANTIATION-CAPABILITY	This.I-D

### 11.3. Extension of LSP Object

This document requests that a registry is created to manage the Flags field of the LSP object (defined in [I-D.ietf-pce-stateful-pce]). New values are to be assigned by Standards Action [RFC5226]. Each bit should be tracked with the following qualities:

- o Bit number (counting from bit 0 as the most significant bit)
- o Capability description
- o Defining RFC

The following values are defined in this document:

Bit	Description	Reference
TBD	P2MP	This.I-D
TBD	Fragmentation	This.I-D

### 11.4. Extension of PCEP-Error Object

A new 19 (recommended values) defined in section 8.5 of [I-D.ietf-pce-stateful-pce]. The error-type 6 is defined in [RFC5440] and error-type 18 in [RFC6006]. This document extend the

new Error-Values for those error types for the following error conditions:

Error-Type	Meaning
6	Mandatory Object missing Error-value=TBD: S2LS object missing Error-value=TBD: P2MP-LSP-IDENTIFIER TLV missing
18	P2MP Fragmentation Error Error-value= TBD. Fragmented Report failure Error-value= TBD. Fragmented Update failure Error-value= TBD. Fragmented Instantiation failure
19	Invalid Operation Error-value= TBD. Attempted LSP State Report for P2MP if stateful PCE capability for P2MP was not advertised Error-value= TBD. Attempted LSP Update Request for P2MP if active stateful PCE capability for P2MP was not advertised Error-value= TBD. Attempted LSP Instantiation Request for P2MP if stateful PCE instantiation capability for P2MP was not advertised

Upon approval of this document, IANA is requested to make the assignment of a new error value for the existing "PCEP-ERROR Object Error Types and Values" registry located at <http://www.iana.org/assignments/pcep/pcep.xhtml#pcep-error-object>.

#### 11.5. PCEP TLV Type Indicators

Upon approval of this document, IANA is requested to make the assignment of a new value for the existing "PCEP TLV Type Indicators" registry located at <http://www.iana.org/assignments/pcep/pcep.xhtml#pcep-tlv-type-indicators>. This document defines the following new PCEP TLVs:

Value	Meaning	Reference
TBD	P2MP-IPV4-LSP-IDENTIFIERS	This.I-D
TBD	P2MP-IPV6-LSP-IDENTIFIERS	This.I-D

## 12. Security Considerations

The stateful operations on P2MP TE LSP are more CPU-intensive and also utilize more link bandwidth. In the event of an unauthorized stateful P2MP operations, or a denial of service attack, the subsequent PCEP operations may be disruptive to the network. Consequently, it is important that implementations conform to the relevant security requirements of [RFC5440], [RFC6006] and [I-D.ietf-pce-stateful-pce], and [I-D.ietf-pce-pce-initiated-lsp]. Further [I-D.ietf-pce-pceps] discusses an experimental approach to provide secure transport for PCEP.

## 13. Acknowledgments

Thanks to Quintin Zhao, Avantika and Venugopal Reddy for his comments.

## 14. References

### 14.1. 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, <<http://www.rfc-editor.org/info/rfc2119>>.
- [RFC3209] Awduche, D., Berger, L., Gan, D., Li, T., Srinivasan, V., and G. Swallow, "RSVP-TE: Extensions to RSVP for LSP Tunnels", RFC 3209, DOI 10.17487/RFC3209, December 2001, <<http://www.rfc-editor.org/info/rfc3209>>.
- [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, <<http://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, <<http://www.rfc-editor.org/info/rfc5089>>.
- [RFC5440] Vasseur, JP., Ed. and JL. Le Roux, Ed., "Path Computation Element (PCE) Communication Protocol (PCEP)", RFC 5440, DOI 10.17487/RFC5440, March 2009, <<http://www.rfc-editor.org/info/rfc5440>>.

[RFC6006] Zhao, Q., Ed., King, D., Ed., Verhaeghe, F., Takeda, T., Ali, Z., and J. Meuric, "Extensions to the Path Computation Element Communication Protocol (PCEP) for Point-to-Multipoint Traffic Engineering Label Switched Paths", RFC 6006, DOI 10.17487/RFC6006, September 2010, <<http://www.rfc-editor.org/info/rfc6006>>.

[I-D.ietf-pce-stateful-pce]  
Crabbe, E., Minei, I., Medved, J., and R. Varga, "PCEP Extensions for Stateful PCE", draft-ietf-pce-stateful-pce-14 (work in progress), March 2016.

[I-D.ietf-pce-stateful-sync-optimizations]  
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", draft-ietf-pce-stateful-sync-optimizations-05 (work in progress), April 2016.

[I-D.ietf-pce-pce-initiated-lsp]  
Crabbe, E., Minei, I., Sivabalan, S., and R. Varga, "PCEP Extensions for PCE-initiated LSP Setup in a Stateful PCE Model", draft-ietf-pce-pce-initiated-lsp-05 (work in progress), October 2015.

#### 14.2. 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, <<http://www.rfc-editor.org/info/rfc4655>>.

[RFC4857] Fogelstroem, E., Jonsson, A., and C. Perkins, "Mobile IPv4 Regional Registration", RFC 4857, DOI 10.17487/RFC4857, June 2007, <<http://www.rfc-editor.org/info/rfc4857>>.

[RFC4875] Aggarwal, R., Ed., Papadimitriou, D., Ed., and S. Yasukawa, Ed., "Extensions to Resource Reservation Protocol - Traffic Engineering (RSVP-TE) for Point-to-Multipoint TE Label Switched Paths (LSPs)", RFC 4875, DOI 10.17487/RFC4875, May 2007, <<http://www.rfc-editor.org/info/rfc4875>>.

[RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, DOI 10.17487/RFC5226, May 2008, <<http://www.rfc-editor.org/info/rfc5226>>.

[RFC5671] Yasukawa, S. and A. Farrel, Ed., "Applicability of the Path Computation Element (PCE) to Point-to-Multipoint (P2MP) MPLS and GMPLS Traffic Engineering (TE)", RFC 5671, DOI 10.17487/RFC5671, October 2009, <<http://www.rfc-editor.org/info/rfc5671>>.

[I-D.ietf-pce-stateful-pce-app]  
Zhang, X. and I. Minei, "Applicability of a Stateful Path Computation Element (PCE)", draft-ietf-pce-stateful-pce-app-05 (work in progress), October 2015.

[I-D.ietf-pce-pceps]  
Lopez, D., Dios, O., Wu, W., and D. Dhody, "Secure Transport for PCEP", draft-ietf-pce-pceps-09 (work in progress), March 2016.

Appendix A. Contributor Addresses

Yuji Kamite  
NTT Communications Corporation  
Granpark Tower  
3-4-1 Shibaura, Minato-ku  
Tokyo 108-8118  
Japan

EMail: y.kamite@ntt.com

Authors' Addresses

Udayasree Palle  
Huawei Technologies  
Divyashree Techno Park, Whitefield  
Bangalore, Karnataka 560066  
India

EMail: udayasree.palle@huawei.com

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

EMail: dhruv.ietf@gmail.com

Yosuke Tanaka  
NTT Communications Corporation  
Granpark Tower  
3-4-1 Shibaura, Minato-ku  
Tokyo 108-8118  
Japan

EMail: yosuke.tanaka@ntt.com

Zafar Ali  
Cisco Systems

EMail: zali@cisco.com

Vishnu Pavan Beeram  
Juniper Networks

EMail: [vbeeram@juniper.net](mailto:vbeeram@juniper.net)



PCE Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: December 29, 2017

Q. Wu  
D. Dhody  
Huawei  
M. Boucadair  
C. Jacquenet  
Orange  
J. Tantsura  
June 27, 2017

PCEP Extensions for Service Function Chaining (SFC)  
draft-wu-pce-traffic-steering-sfc-12

## Abstract

This document provides an overview of the usage of Path Computation Element (PCE) to dynamically structure service function chains. Service Function Chaining (SFC) is a technique that is meant to facilitate the dynamic enforcement of differentiated traffic forwarding policies within a domain. Service function chains are composed of an ordered set of elementary Service Functions (such as firewalls, load balancers) that need to be invoked according to the design of a given service. Corresponding traffic is thus forwarded along a Service Function Path (SFP) that can be computed by means of PCE.

This document specifies extensions to the Path Computation Element Protocol (PCEP) that allow a stateful PCE to compute and instantiate Service Function Paths.

## Status of This Memo

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

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

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

This Internet-Draft will expire on December 29, 2017.

## Copyright Notice

Copyright (c) 2017 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 . . . . .	2
2. Conventions used in this document . . . . .	3
3. Service Function Paths and PCE . . . . .	4
4. Overview of PCEP Operation in SFC-Enabled Networks . . . . .	6
4.1. SFP Instantiation . . . . .	6
4.2. SFP Withdrawal . . . . .	6
4.3. SFP Delegation and Cleanup . . . . .	7
4.4. SFP State Synchronization . . . . .	7
4.5. SFP Update and Report . . . . .	7
5. Object Formats . . . . .	7
5.1. The OPEN Object . . . . .	7
5.2. The LSP Object . . . . .	8
5.2.1. SFP Identifiers TLV . . . . .	8
6. Backward Compatibility . . . . .	9
7. SFP Instantiation Signaling and Forwarding Considerations . . . . .	9
8. Security Considerations . . . . .	10
9. IANA Considerations . . . . .	10
10. Acknowledgements . . . . .	10
11. References . . . . .	10
11.1. Normative References . . . . .	10
11.2. Informative References . . . . .	11
Authors' Addresses . . . . .	12

## 1. Introduction

Service Function Chaining (SFC) enables the creation of composite services that consist of an ordered set of Service Functions (SF) that must be applied to packets and/or frames and/or flows selected as a result of service-inferred traffic classification as described in [RFC7665]. A Service Function Path (SFP) is a path along which traffic that is bound to a specific service function chain will be

forwarded. Packets typically follow a Service Function Path from a classifier through the Service Functions (SF) that need to be invoked according to the SFC instructions. Forwarding decisions are made by Service Function Forwarders (SFF) according to such instructions.

[RFC5440] describes the Path Computation Element Protocol (PCEP) as the protocol used by a Path Computation Client (PCC) and a Path Control Element (PCE) to exchange information, thereby enabling the computation of Multiprotocol Label Switching (MPLS) for Traffic Engineering Label Switched Path (TE LSP), in particular.

[I-D.ietf-pce-stateful-pce] specifies extensions to PCEP to enable a stateful control of MPLS TE LSPs. [I-D.ietf-pce-pce-initiated-lsp] provides the extensions needed for stateful PCE-initiated LSP instantiation.

This document specifies PCEP extensions that allow a stateful PCE to compute and instantiate traffic-engineered Service Function Paths (SFP).

## 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 RFC2119 [RFC2119].

This document makes use of these acronyms:

PCC: Path Computation Client.

PCE: Path Computation Element.

PCEP: Path Computation Element Protocol.

PDP: Policy Decision Point.

SF: Service Function.

SFC: Service Function Chain.

SFP: Service Function Path.

RSP: Rendered Service Path.

SFF: Service Function Forwarder.

UNI: User-Network Interface.

### 3. Service Function Paths and PCE

Service function chains are constructed as a sequence of SFs, where a SF can be virtualized or embedded in a physical network element. One or several SFs may be supported by the same physical network element. A SFC creates an abstracted view of a service and specifies the set of required SFs as well as the order in which they must be executed.

When an SFC is created, it is necessary to select the specific instances of SFs that will be used. A service function path for that SFC will then be established (notion of rendered service path) or can be precomputed, based upon the sequence of SFs that need to be invoked by the corresponding traffic, i.e., the traffic that is bound to the corresponding SFC. Note that a SF instance can be serviced by one or multiple SFFs. One or multiple SF instances can be serviced by one SFF. Thus, the instantiation of an SFC results in the establishment of a Service Function Path, either in a hop-by-hop fashion, or by means of traffic-engineering capabilities. In the latter case, the SFP is precomputed, i.e., an SFP is an instantiation of the defined SFC as described in [RFC7665].

The computation, the selection, and the establishment of a traffic-engineered SFP can rely upon a set of (service-specific) policies (forwarding and routing, QoS, security, etc., or a combination thereof). Stateful PCE with appropriate SFC-aware PCEP extensions can be used to compute traffic-engineered SFPs.

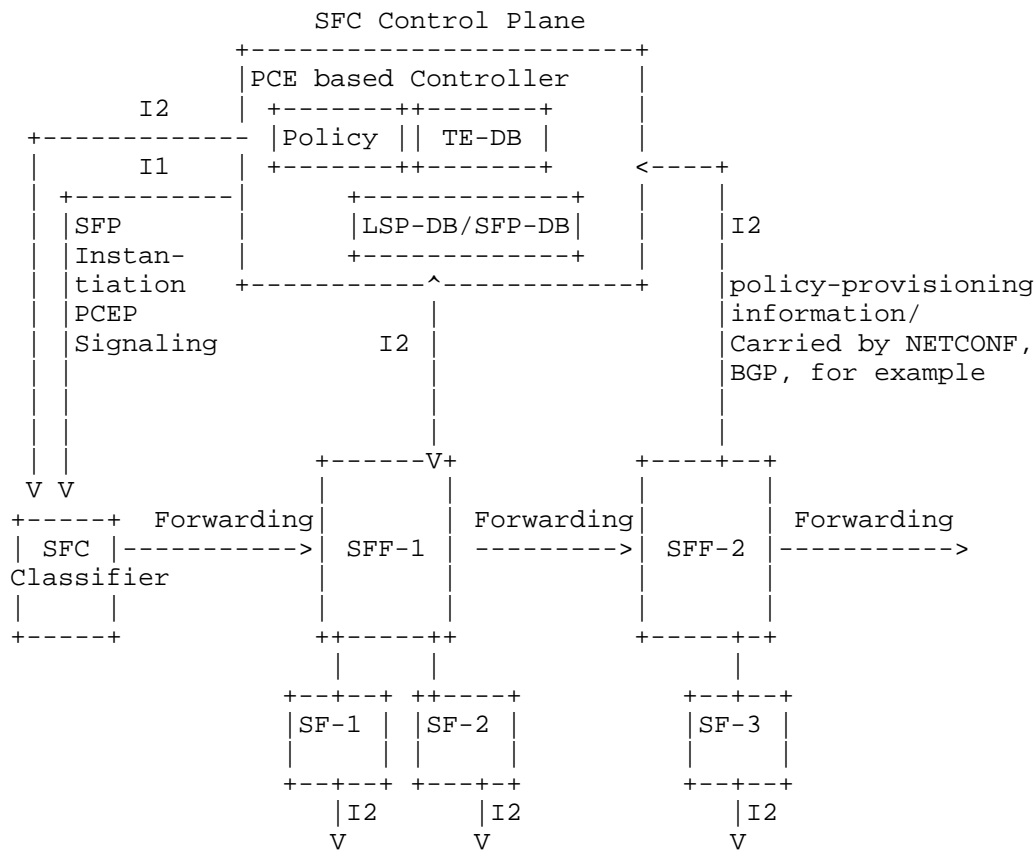


Figure 1: PCE-based SFP instantiation

In Figure 1, the PCE-based Controller [I-D.ietf-teas-pce-central-control] in the SFC Control plane is responsible for computing the path for a given service function chain. This PCE-based controller can operate as a stateful PCE ([I-D.draft\_ietf-stateful-pce]) that will provide a classifier (a headend from a PCE standpoint) with the PCEP-formatted information to instantiate a given SFP. As a consequence, the PCE-based controller derives the set of policy-provisioning information (namely SFP configuration information and traffic classification rules) that will be provided to the various elements (Classifier, SFF) involved in the establishment of the SFP.

By doing so, SFC Classifier can bind a flow to a service function chain and forward such flow along the corresponding SFP. The SFC Control Plane [I-D.ietf-sfc-control-plane] is also responsible for defining the appropriate policies (traffic classification, forwarding and routing, etc.) that will be enforced by SFC Classifiers, SFF Nodes

and SF Nodes, as described in [RFC7665]. From that standpoint, the SFC Control Plane embeds a Policy Decision Point that is responsible for defining the SFC policies. SFC policies will be provided by the PDP and enforced by SFC components like classifiers and SFFs by means of policy-provision information. A protocol like NETCONF, BGP can be used to carry such policy-provisioning information.

#### 4. Overview of PCEP Operation in SFC-Enabled Networks

A PCEP speaker indicates its ability to support PCE-computed SFP paths during the PCEP Initialization phase via a mechanism described in Section 5.1. A PCE may initiate SFPs only for PCCs that advertised this capability; a PCC follows the procedures described in this document only for sessions where the PCE advertised this capability.

As per Section 5.1 of [I-D.ietf-pce-pce-initiated-lsp], the PCE sends a Path Computation LSP Initiate Request (PCInitiate) message to the PCC to instantiate or delete a LSP. The Explicit Route Object (ERO) is used to encode either a full sequence of SF instances or a specific sequence of SFFs and SFs to establish an SFP. If the said SFFs and SFs are identified with an IP address, the IP sub-object can be used as a SF/SFF identification means. This document makes no change to the PCInitiate message format but extends LSP objects described in Section 5.2.

Editor's note: In case a PCE-Initiated signaling mechanism is used to set up the service function path, does the classifier / PCE-Initiated signaling protocol need to understand whether an IP address is assigned to a SFF or a SF, or the signaling protocol is only used to signal IP addresses for SFs?

To prevent multiple classifiers assign the same SFP ID to one Service Function Path(SFP ID assignment conflict), in this document, we assume SFP ID can be predetermined and assigned by stateful PCE when stateful PCE can be used to compute traffic-engineered SFPs.

##### 4.1. SFP Instantiation

The instantiation of a SFP is the same as defined in Section 5.3 of [I-D.ietf-pce-pce-initiated-lsp]. Rules for processing and error codes remain unchanged.

##### 4.2. SFP Withdrawal

The withdrawal of an SFP is the same as defined in Section 5.4 of [I-D.ietf-pce-pce-initiated-lsp]: the PCE sends an LSP Initiate Message with an LSP object carrying the PLSP-ID of the SFP and the

SFP Identifier to be removed, as well as an SRP object with the R flag set (LSP-REMOVE as per Section 5.2 of [I-D.ietf-pce-pce-initiated-lsp]). Rules for processing and error codes remain unchanged.

#### 4.3. SFP Delegation and Cleanup

SFP delegation and cleanup operations are similar to those defined in Section 6 of [I-D.ietf-pce-pce-initiated-lsp]. Rules for processing and error codes remain unchanged.

#### 4.4. SFP State Synchronization

State Synchronization operations described in Section 5.4 of [I-D.ietf-pce-stateful-pce] can be applied to SFP state maintenance as well.

#### 4.5. SFP Update and Report

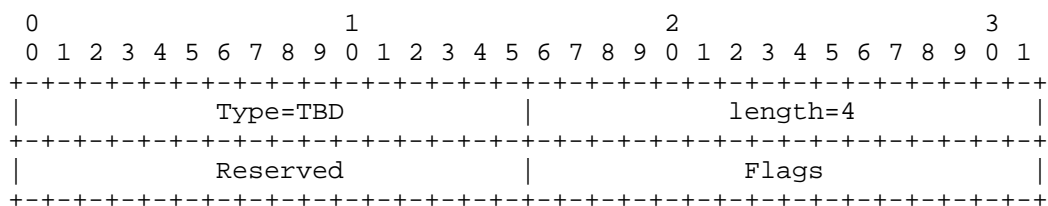
A PCE can send an SFP Update request to a PCC to update one or more attributes of an SFP and to re-signal the SFP with the updated attributes. A PCC can send an SFP state report to a PCE, and which contains the SFP State information. The mechanism is described in [I-D.ietf-pce-stateful-pce] and can be applied to SFPs as well.

### 5. Object Formats

#### 5.1. The OPEN Object

The optional TLV shown in Figure 2 is defined for use in the OPEN Object to indicate the PCEP speaker's Service Function Chaining capability.

The SFC-PCE-CAPABILITY TLV is an optional TLV to be carried in the OPEN Object to advertise the SFC capability during the PCEP session.



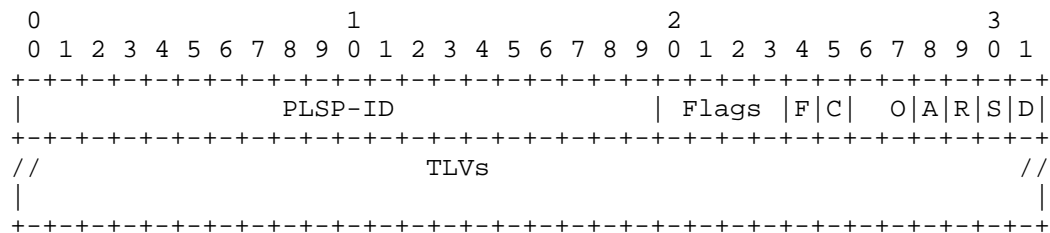
SFC-PCE-CAPABILITY TLV Format

The code point for the TLV type is to be defined by IANA (see Section 9). The TLV length is 4 octets.

As per [I-D.ietf-pce-stateful-pce], a PCEP speaker advertises the capability of instantiating PCE-initiated LSPs via the Stateful PCE Capability TLV (LSP-INSTANTIATION-CAPABILITY bit) carried in an Open message. The inclusion of the SFC-PCE-CAPABILITY TLV in an OPEN object indicates that the sender is SFC-capable. Both mechanisms indicate the SFP instantiation capability of the PCEP speaker.

## 5.2. The LSP Object

The LSP object is defined in [I-D.ietf-pce-pce-initiated-lsp] and included here for reference (Figure 3).



LSP Object Format

A new flag, called the SFC flag (F-bit), is introduced. The F-bit set to "1" indicates that this LSP is actually an SFP. The C flag will also be set to indicate it was created via a PCInitiate message.

### 5.2.1. SFP Identifiers TLV

As described in section 4, SFP ID is predetermined and assigned by stateful PCE. The SFP Identifiers TLV MUST be included in the LSP object for SFPs. The SFP Identifier TLV is used by the classifier to select the SFP along which some traffic will be forwarded, according to the traffic classification rules applied by the classifier [RFC7665]. The SFP Identifier is part of the SFC metadata carried in packets and is used by the SFF to invoke service functions and identify the next SFF.

The format of the SFP Identifier TLV is shown in Figure 4.



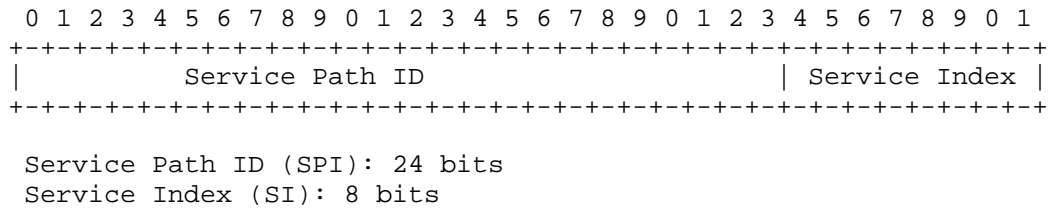


Figure 4

SPI: identifies a service path. The same ID is used by the participating nodes for path setup/selection. An administrator can use the SPI for reporting and troubleshooting packets along a specific path. SPI along with PLSP-ID is used by PCEP to identify the Service Path.

SI: provides location within the service path.

## 6. Backward Compatibility

The SFP instantiation capability defined as a PCEP extension and documented in this draft MUST NOT be used if PCCs or the PCE did not advertise their stateful SFP instantiation capability, Section 5.1. If this is not the case and stateful operations on SFPs are attempted, then a PCErr message with error-type 19 (Invalid Operation) and error-value TBD needs to be generated.

[Editor's note: more information on exact error value is needed]

## 7. SFP Instantiation Signaling and Forwarding Considerations

The PCE-initiated SFP instantiation signaling described in this document is exchanged between PCE server and SFC Classifier and does not assume any specific mechanism to exchange SFP information (e.g., path identification information, metadata [I-D.ietf-sfc-nsh]) between SFFs or between SFF and SF, or between the controller and SFF and establish SFP in the data plane throughout a SFC domain. For example, such mechanism can rely upon the use of the SFC Encapsulation defined in [I-D.ietf-sfc-nsh] to exchange SFP information between SFFs or rely upon the use of BGP Control plane defined in [I-D.ietf-bess-nsh-bgp-control-plane] to exchange SFP information between the Controller and SFF.

Likewise, [I-D.ietf-teas-pce-central-control] can use the signaling mechanism described in this draft to enforce SFC-inferred traffic engineering policies and provide load balancing between service function nodes. The approach that relies upon the Segment Routing technique [I-D.ietf-pce-segment-routing] can also take advantage of

the signaling mechanism described in this document to support Service Path instantiation, which does not require any additional specific extension to the Segment Routing machinery.

## 8. Security Considerations

The security considerations described in [RFC5440] and [I-D.ietf-pce-pce-initiated-lsp] are applicable to this specification. This document does not raise any additional security issue.

## 9. IANA Considerations

IANA is requested to allocate a new code point in the PCEP TLV Type Indicators registry, as follows:

Value	Meaning	Reference
TBD	SFC-PCE-CAPABILITY	This document

## 10. Acknowledgements

Many thanks to Ron Parker, Hao Wang, Dave Dolson, Jing Huang, and Joel M. Halpern for the discussion about the content for the document.

## 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, <<http://www.rfc-editor.org/info/rfc2119>>.
- [I-D.ietf-pce-stateful-pce] Crabbe, E., Minei, I., Medved, J., and R. Varga, "PCEP Extensions for Stateful PCE", draft-ietf-pce-stateful-pce-21 (work in progress), June 2017.
- [RFC5440] Vasseur, JP., Ed. and JL. Le Roux, Ed., "Path Computation Element (PCE) Communication Protocol (PCEP)", RFC 5440, DOI 10.17487/RFC5440, March 2009, <<http://www.rfc-editor.org/info/rfc5440>>.

- [I-D.ietf-pce-pce-initiated-lsp]  
Crabbe, E., Minei, I., Sivabalan, S., and R. Varga, "PCEP Extensions for PCE-initiated LSP Setup in a Stateful PCE Model", draft-ietf-pce-pce-initiated-lsp-10 (work in progress), June 2017.
- [I-D.ietf-teas-pce-central-control]  
Farrel, A., Zhao, Q., Li, Z., and C. Zhou, "An Architecture for Use of PCE and PCEP in a Network with Central Control", draft-ietf-teas-pce-central-control-03 (work in progress), June 2017.

## 11.2. Informative References

- [RFC2753] Yavatkar, R., Pendarakis, D., and R. Guerin, "A Framework for Policy-based Admission Control", RFC 2753, DOI 10.17487/RFC2753, January 2000, <<http://www.rfc-editor.org/info/rfc2753>>.
- [RFC7665] Halpern, J., Ed. and C. Pignataro, Ed., "Service Function Chaining (SFC) Architecture", RFC 7665, DOI 10.17487/RFC7665, October 2015, <<http://www.rfc-editor.org/info/rfc7665>>.
- [RFC5394] Bryskin, I., Papadimitriou, D., Berger, L., and J. Ash, "Policy-Enabled Path Computation Framework", RFC 5394, DOI 10.17487/RFC5394, December 2008, <<http://www.rfc-editor.org/info/rfc5394>>.
- [I-D.ietf-sfc-control-plane]  
Boucadair, M., "Service Function Chaining (SFC) Control Plane Components & Requirements", draft-ietf-sfc-control-plane-08 (work in progress), October 2016.
- [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-09 (work in progress), April 2017.
- [I-D.ietf-sfc-nsh]  
Quinn, P. and U. Elzur, "Network Service Header", draft-ietf-sfc-nsh-12 (work in progress), February 2017.
- [I-D.ietf-bess-nsh-bgp-control-plane]  
Farrel, A., Drake, J., Rosen, E., Uttaro, J., and L. Jalil, "BGP Control Plane for NSH SFC", draft-ietf-bess-nsh-bgp-control-plane-00 (work in progress), March 2017.

Authors' Addresses

Qin Wu  
Huawei  
101 Software Avenue, Yuhua District  
Nanjing, Jiangsu 210012  
China

EMail: bill.wu@huawei.com

Dhruv Dhody  
Huawei  
Leela Palace  
Bangalore, Karnataka 560008  
INDIA

EMail: dhruv.ietf@gmail.com

Mohamed Boucadair  
Orange  
Rennes 35000  
France

EMail: mohamed.boucadair@orange.com

Christian Jacquenet  
Orange  
Rennes  
France

EMail: christian.jacquenet@orange.com

Jeff Tantsura  
2330 Central Expressway  
Santa Clara, CA 95050  
US

EMail: jefftant.ietf@gmail.com

PCE Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: December 20, 2018

Q. Zhao  
Z. Li  
D. Dhody  
S. Karunanithi  
Huawei Technologies  
A. Farrel  
Juniper Networks, Inc  
C. Zhou  
Cisco Systems  
June 18, 2018

PCEP Procedures and Protocol Extensions for Using PCE as a Central  
Controller (PCECC) of LSPs  
draft-zhao-pce-pcep-extension-for-pce-controller-08

Abstract

The Path Computation Element (PCE) is a core component of Software-Defined Networking (SDN) systems. It can compute optimal paths for traffic across a network and can also update the paths to reflect changes in the network or traffic demands.

PCE was developed to derive paths for MPLS Label Switched Paths (LSPs), which are supplied to the head end of the LSP using the Path Computation Element Communication Protocol (PCEP). But SDN has a broader applicability than signaled (G)MPLS traffic-engineered (TE) networks, and the PCE may be used to determine paths in a range of use cases. PCEP has been proposed as a control protocol for use in these environments to allow the PCE to be fully enabled as a central controller.

A PCE-based central controller (PCECC) can simplify the processing of a distributed control plane by blending it with elements of SDN and without necessarily completely replacing it. Thus, the LSP can be calculated/setup/initiated and the label forwarding entries can also be downloaded through a centralized PCE server to each network devices along the path while leveraging the existing PCE technologies as much as possible.

This document specifies the procedures and PCEP protocol extensions for using the PCE as the central controller.

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 20, 2018.

#### Copyright Notice

Copyright (c) 2018 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. Requirements Language . . . . .	5
2. Terminology . . . . .	5
3. Basic PCECC Mode . . . . .	5
4. PCEP Requirements . . . . .	5
5. Procedures for Using the PCE as the Central Controller (PCECC) . . . . .	6
5.1. Stateful PCE Model . . . . .	6
5.2. New LSP Functions . . . . .	6
5.3. PCECC Capability Advertisement . . . . .	7
5.4. LSP Operations . . . . .	8
5.4.1. Basic PCECC LSP Setup . . . . .	8
5.4.2. Central Control Instructions . . . . .	10
5.4.2.1. Label Download . . . . .	10
5.4.2.2. Label Cleanup . . . . .	11
5.4.3. PCE Initiated PCECC LSP . . . . .	12
5.4.4. PCECC LSP Update . . . . .	14
5.4.5. Re Delegation and Cleanup . . . . .	16
5.4.6. Synchronization of Central Controllers Instructions . . . . .	16

5.4.7. PCECC LSP State Report . . . . .	16
6. PCEP messages . . . . .	16
6.1. The PCInitiate message . . . . .	17
6.2. The PCRpt message . . . . .	18
7. PCEP Objects . . . . .	19
7.1. OPEN Object . . . . .	19
7.1.1. PCECC Capability sub-TLV . . . . .	19
7.2. PATH-SETUP-TYPE TLV . . . . .	20
7.3. CCI Object . . . . .	20
7.3.1. Address TLVs . . . . .	21
8. Security Considerations . . . . .	23
8.1. Malicious PCE . . . . .	23
9. Manageability Considerations . . . . .	23
9.1. Control of Function and Policy . . . . .	23
9.2. Information and Data Models . . . . .	23
9.3. Liveness Detection and Monitoring . . . . .	23
9.4. Verify Correct Operations . . . . .	23
9.5. Requirements On Other Protocols . . . . .	23
9.6. Impact On Network Operations . . . . .	24
10. IANA Considerations . . . . .	24
10.1. PCEP TLV Type Indicators . . . . .	24
10.2. New Path Setup Type Registry . . . . .	24
10.3. PCEP Object . . . . .	24
10.4. CCI Object Flag Field . . . . .	24
10.5. PCEP-Error Object . . . . .	25
11. Acknowledgments . . . . .	25
12. References . . . . .	25
12.1. Normative References . . . . .	25
12.2. Informative References . . . . .	26
Appendix A. Contributor Addresses . . . . .	29
Authors' Addresses . . . . .	29

## 1. Introduction

The Path Computation Element (PCE) [RFC4655] was developed to offload path computation function from routers in an MPLS traffic-engineered network. Since then, the role and function of the PCE has grown to cover a number of other uses (such as GMPLS [RFC7025]) and to allow delegated control [RFC8231] and PCE-initiated use of network resources [RFC8281].

According to [RFC7399], Software-Defined Networking (SDN) refers to a separation between the control elements and the forwarding components so that software running in a centralized system, called a controller, can act to program the devices in the network to behave in specific ways. A required element in an SDN architecture is a component that plans how the network resources will be used and how the devices will be programmed. It is possible to view this

component as performing specific computations to place traffic flows within the network given knowledge of the availability of network resources, how other forwarding devices are programmed, and the way that other flows are routed. This is the function and purpose of a PCE, and the way that a PCE integrates into a wider network control system (including an SDN system) is presented in [RFC7491].

In early PCE implementations, where the PCE was used to derive paths for MPLS Label Switched Paths (LSPs), paths were requested by network elements (known as Path Computation Clients (PCCs)), and the results of the path computations were supplied to network elements using the Path Computation Element Communication Protocol (PCEP) [RFC5440]. This protocol was later extended to allow a PCE to send unsolicited requests to the network for LSP establishment [RFC8281].

[RFC8283] introduces the architecture for PCE as a central controller as an extension of the architecture described in [RFC4655] and assumes the continued use of PCEP as the protocol used between PCE and PCC. [RFC8283] further examines the motivations and applicability for PCEP as a Southbound Interface (SBI), and introduces the implications for the protocol. [I-D.ietf-teas-pcecc-use-cases] describes the use cases for the PCECC architecture.

A PCE-based central controller (PCECC) can simplify the processing of a distributed control plane by blending it with elements of SDN and without necessarily completely replacing it. Thus, the LSP can be calculated/setup/initiated and the label forwarding entries can also be downloaded through a centralized PCE server to each network devices along the path while leveraging the existing PCE technologies as much as possible.

This draft specify the procedures and PCEP protocol extensions for using the PCE as the central controller for static LSPs, where LSPs can be provisioned as explicit label instructions at each hop on the end-to-end path. Each router along the path must be told what label-forwarding instructions to program and what resources to reserve. The PCE-based controller keeps a view of the network and determines the paths of the end-to-end LSPs, and the controller uses PCEP to communicate with each router along the path of the end-to-end LSP.

The extension for PCECC in Segment Routing (SR) is specified in a separate draft [I-D.zhao-pce-pcep-extension-pce-controller-sr].



### 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. Terminology

Terminologies used in this document is same as described in the draft [RFC8283] and [I-D.ietf-teas-pcecc-use-cases].

## 3. Basic PCECC Mode

In this mode LSPs are provisioned as explicit label instructions at each hop on the end-to-end path. Each router along the path must be told what label forwarding instructions to program and what resources to reserve. The controller uses PCEP to communicate with each router along the path of the end-to-end LSP.

Note that the PCE-based controller will take responsibility for managing some part of the MPLS label space for each of the routers that it controls, and may take wider responsibility for partitioning the label space for each router and allocating different parts for different uses. This is also described in section 3.1.2. of [RFC8283]. For the purpose of this document, it is assumed that label range to be used by a PCE is known and set on both PCEP peers. A future extension could add this capability to advertise the range via possible PCEP extensions as well. The rest of processing is similar to the existing stateful PCE mechanism.

## 4. PCEP Requirements

Following key requirements associated PCECC should be considered when designing the PCECC based solution:

1. PCEP speaker supporting this draft MUST have the capability to advertise its PCECC capability to its peers.
2. PCEP speaker not supporting this draft MUST be able to reject PCECC related extensions with a error reason code that indicates that this feature is not supported.
3. PCEP speaker MUST provide a means to identify PCECC based LSP in the PCEP messages.

4. PCEP procedures SHOULD provide a means to update (or cleanup) the label- download entry to the PCC.
  5. PCEP procedures SHOULD provide a means to synchronize the labels between PCE to PCC in PCEP messages.
5. Procedures for Using the PCE as the Central Controller (PCECC)
- 5.1. Stateful PCE Model

Active stateful PCE is described in [RFC8231]. PCE as a central controller (PCECC) reuses existing Active stateful PCE mechanism as much as possible to control the LSP.

5.2. New LSP Functions

This document defines the following new PCEP messages and extends the existing messages to support PCECC:

(PCRpt): a PCEP message described in [RFC8231]. PCRpt message is used to send PCECC LSP Reports. It is also extended to report the set of Central Controller's Instructions (CCI) (label forwarding instructions in the context of this document) received from the PCE. See Section 5.4.6 for more details.

(PCInitiate): a PCEP message described in [RFC8281]. PCInitiate message is used to setup PCE-Initiated LSP based on PCECC mechanism. It is also extended for Central Controller's Instructions (CCI) (download or cleanup the Label forwarding instructions in the context of this document) on all nodes along the path.

(PCUpd): a PCEP message described in [RFC8231]. PCUpd message is used to send PCECC LSP Update.

The new LSP functions defined in this document are mapped onto the messages as shown in the following table.

Function	Message
PCECC Capability advertisement	Open
Label entry Add	PCInitiate
Label entry Cleanup	PCInitiate
PCECC Initiated LSP	PCInitiate
PCECC LSP Update	PCUpd
PCECC LSP State Report	PCRpt
PCECC LSP Delegation	PCRpt
PCECC Label Report	PCRpt

This document specifies a new object CCI (see Section 7.3) for the encoding of central controller's instructions. In the scope of this document this is limited to Label forwarding instructions. The CC-ID is the unique identifier for the central controller's instructions in PCEP. The PCEP messages are extended in this document to handle the PCECC operations.

### 5.3. PCECC Capability Advertisement

During PCEP Initialization Phase, PCEP Speakers (PCE or PCC) advertise their support of PCECC extensions.

This document defines a new Path Setup Type (PST) [I-D.ietf-pce-lsp-setup-type] for PCECC, as follows:

- o PST = TBD: Path is setup via PCECC mode.

A PCEP speaker MUST indicate its support of the function described in this document by sending a PATH-SETUP-TYPE-CAPABILITY TLV in the OPEN object with this new PST included in the PST list.

This document also defines the PCECC Capability sub-TLV Section 7.1.1. PCEP speakers use this sub-TLV to exchange information about their PCECC capability. If a PCEP speaker includes PST=TBD in the PST List of the PATH-SETUP-TYPE-CAPABILITY TLV then it MUST also include the PCECC Capability sub-TLV inside the PATH-SETUP-TYPE-CAPABILITY TLV.

The presence of the PST and PCECC Capability sub-TLV in PCC's OPEN Object indicates that the PCC is willing to function as a PCECC client.

The presence of the PST and PCECC Capability sub-TLV in PCE's OPEN message indicates that the PCE is interested in function as a PCECC server.

The PCEP protocol extensions for PCECC MUST NOT be used if one or both PCEP Speakers have not included the PST or the PCECC Capability sub-TLV in their respective OPEN message. If the PCEP Speakers support the extensions of this draft but did not advertise this capability then a PCerr message with Error-Type=19(Invalid Operation) and Error-Value=TBD (Attempted PCECC operations when PCECC capability was not advertised) will be generated and the PCEP session will be terminated.

A PCC or a PCE MUST include both PCECC-CAPABILITY sub-TLV and STATEFUL-PCE-CAPABILITY TLV ([RFC8231]) (with I flag set [RFC8281]) in OPEN Object to support the extensions defined in this document. If PCECC-CAPABILITY sub-TLV is advertised and STATEFUL-PCE-CAPABILITY TLV is not advertised in OPEN Object, it SHOULD send a PCerr message with Error-Type=19 (Invalid Operation) and Error-value=TBD (stateful PCE capability was not advertised) and terminate the session.

#### 5.4. LSP Operations

The PCEP messages pertaining to PCECC MUST include PATH-SETUP-TYPE TLV [I-D.ietf-pce-lsp-setup-type] in the SRP object to clearly identify the PCECC LSP is intended.

##### 5.4.1. Basic PCECC LSP Setup

In order to setup a LSP based on PCECC mechanism, a PCC MUST delegate the LSP by sending a PCRpt message with PST set for PCECC (see Section 7.2) and D (Delegate) flag (see [RFC8231]) set in the LSP object.

LSP-IDENTIFIER TLV MUST be included for PCECC LSP, the tuple uniquely identifies the LSP in the network. The LSP object is included in central controller's instructions (label download) to identify the PCECC LSP for this instruction. The PLSP-ID is the original identifier used by the ingress PCC, so the transit LSR could have multiple central controller instructions that have the same PLSP-ID. The PLSP-ID in combination with the source (in LSP-IDENTIFIER TLV) MUST be unique. The PLSP-ID is included for maintainability reasons. As per [RFC8281], the LSP object could include SPEAKER-ENTITY-ID TLV to identify the PCE that initiated these instructions. Also the CC-ID is unique on the PCEP session as described in Section 7.3.

When a PCE receives PCRpt message with D flags and PST Type set, it calculates the path and assigns labels along the path; and set up the

path by sending PCInitiate message to each node along the path of the LSP. The PCC generates a Path Computation State Report (PCRpt) and include the central controller's instruction (CCI) and the identified LSP. The CC-ID is uniquely identify the central controller's instruction within PCEP. The PCC further responds with the PCRpt messages including the CCI and LSP objects.

Once the central controller's instructions (label operations) are completed, the PCE SHOULD send the PCUpd message to the Ingress PCC. The PCUpd message is as per [RFC8231] SHOULD include the path information as calculated by the PCE.

Note that the PCECC LSPs MUST be delegated to a PCE at all times.

LSP deletion operation for PCECC LSP is same as defined in [RFC8231]. If the PCE receives PCRpt message for LSP deletion then it does Label cleanup operation as described in Section 5.4.2.2 for the corresponding LSP.

The Basic PCECC LSP setup sequence is as shown below.

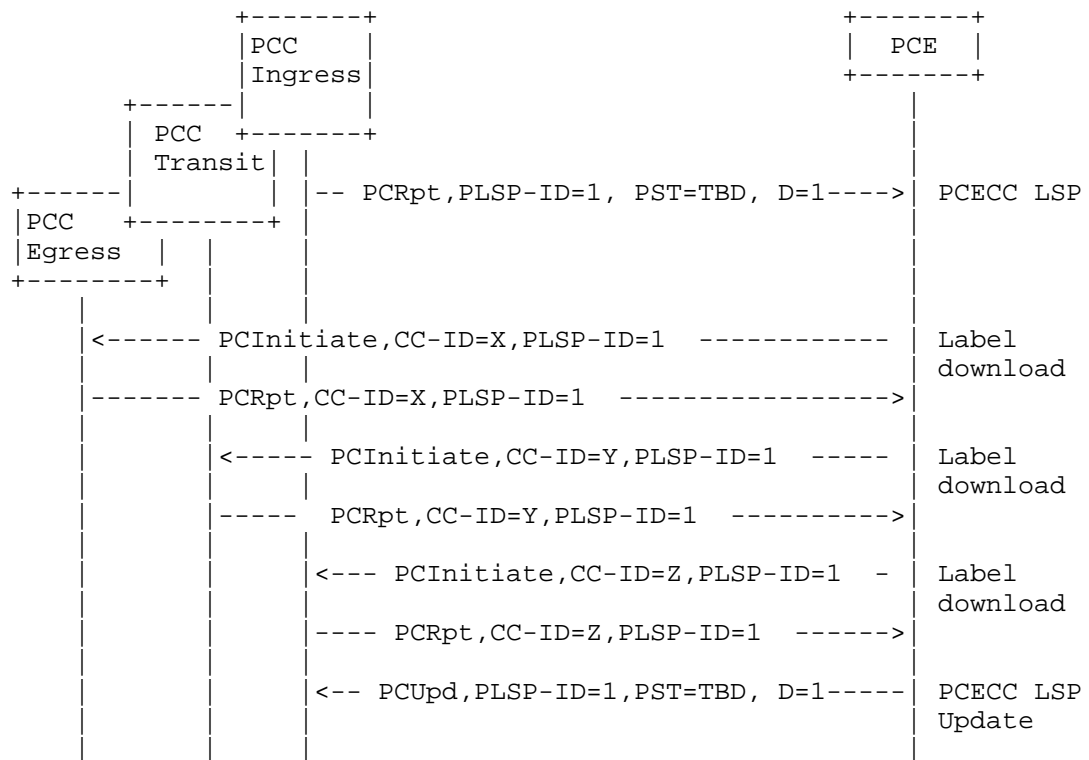


Figure 2: Basic PCECC LSP setup

The PCECC LSP are considered to be 'up' by default (on receipt of PCUpd message from PCE). The Ingress MAY further choose to deploy a data plane check mechanism and report the status back to the PCE via PCRpt message.

#### 5.4.2. Central Control Instructions

The new central controller's instructions (CCI) for the label operations in PCEP is done via the PCInitiate message, by defining a new PCEP Objects for CCI operations. Local label range of each PCC is assumed to be known at both the PCC and the PCE.

##### 5.4.2.1. Label Download

In order to setup an LSP based on PCECC, the PCE sends a PCInitiate message to each node along the path to download the Label instruction as described in Section 5.4.1.

The CCI object MUST be included, along with the LSP object in the PCInitiate message. The LSP-IDENTIFIER TLV MUST be included in LSP object. The SPEAKER-ENTITY-ID TLV SHOULD be included in LSP object.

If a node (PCC) receives a PCInitiate message which includes a Label to download as part of CCI, that is out of the range set aside for the PCE, it MUST send a PCErr message with Error-type=TBD (PCECC failure) and Error-value=TBD (Label out of range) and MUST include the SRP object to specify the error is for the corresponding label update via PCInitiate message. If a PCC receives a PCInitiate message but failed to download the Label entry, it MUST send a PCErr message with Error-type=TBD (PCECC failure) and Error-value=TBD (instruction failed) and MUST include the SRP object to specify the error is for the corresponding label update via PCInitiate message.

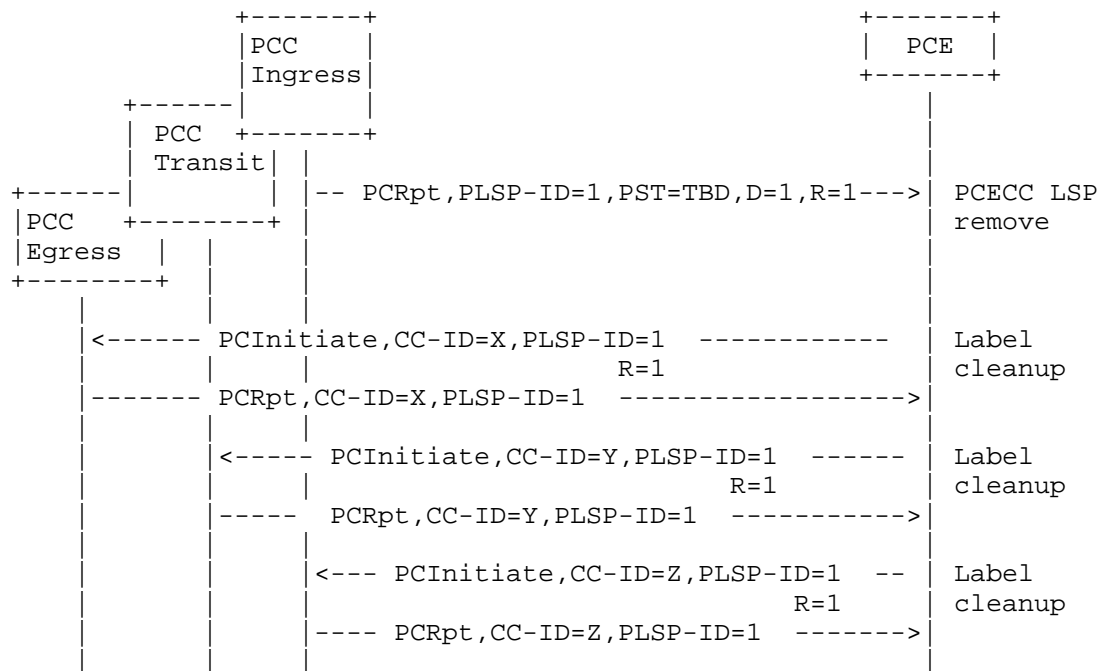
New PCEP object for central control instructions (CCI) is defined in Section 7.3.

#### 5.4.2.2. Label Cleanup

In order to delete an LSP based on PCECC, the PCE sends a central controller instructions via a PCInitiate message to each node along the path of the LSP to cleanup the Label forwarding instruction.

If the PCC receives a PCInitiate message but does not recognize the label in the CCI, the PCC MUST generate a PCErr message with Error-Type 19(Invalid operation) and Error-Value=TBD, "Unknown Label" and MUST include the SRP object to specify the error is for the corresponding label cleanup (via PCInitiate message).

The R flag in the SRP object defined in [RFC8281] specifies the deletion of Label Entry in the PCInitiate message.



As per [RFC8281], following the removal of the Label forwarding instruction, the PCC MUST send a PCRpt message. The SRP object in the PCRpt MUST include the SRP-ID-number from the PCInitiate message that triggered the removal. The R flag in the SRP object MUST be set.

#### 5.4.3. PCE Initiated PCECC LSP

The LSP Instantiation operation is same as defined in [RFC8281].

In order to setup a PCE Initiated LSP based on the PCECC mechanism, a PCE sends PCInitiate message with Path Setup Type set for PCECC (see Section 7.2) to the Ingress PCC.

The Ingress PCC MUST also set D (Delegate) flag (see [RFC8231]) and C (Create) flag (see [RFC8281]) in LSP object of PCRpt message. The PCC responds with first PCRpt message with the status as "GOING-UP" and assigned PLSP-ID.

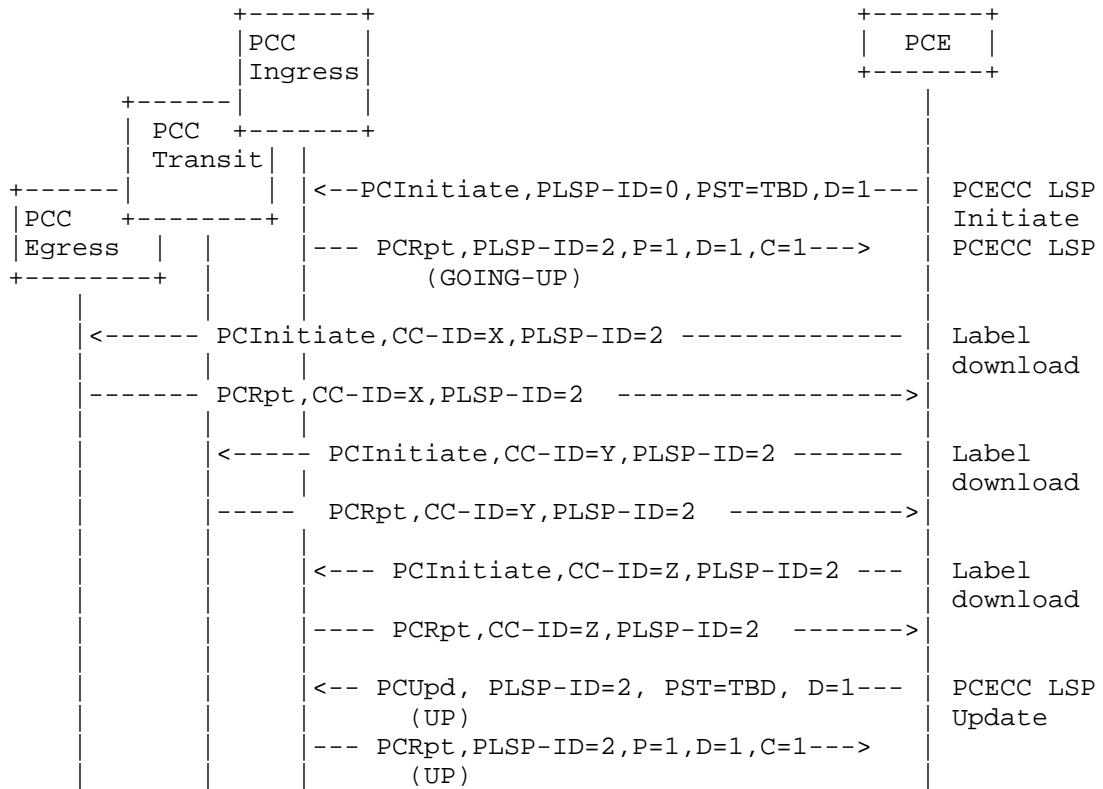
Note that the label forwarding instructions from PCECC are send after the initial PCInitiate and PCRpt exchange. This is done so that the PLSP-ID and other LSP identifiers can be obtained from the ingress and can be included in the label forwarding instruction in the next



PCInitiate message. The rest of the PCECC LSP setup operations are same as those described in Section 5.4.1.

The LSP deletion operation for PCE Initiated PCECC LSP is same as defined in [RFC8281]. The PCE should further perform Label entry cleanup operation as described in Section 5.4.2.2 for the corresponding LSP.

The PCE Initiated PCECC LSP setup sequence is shown below -

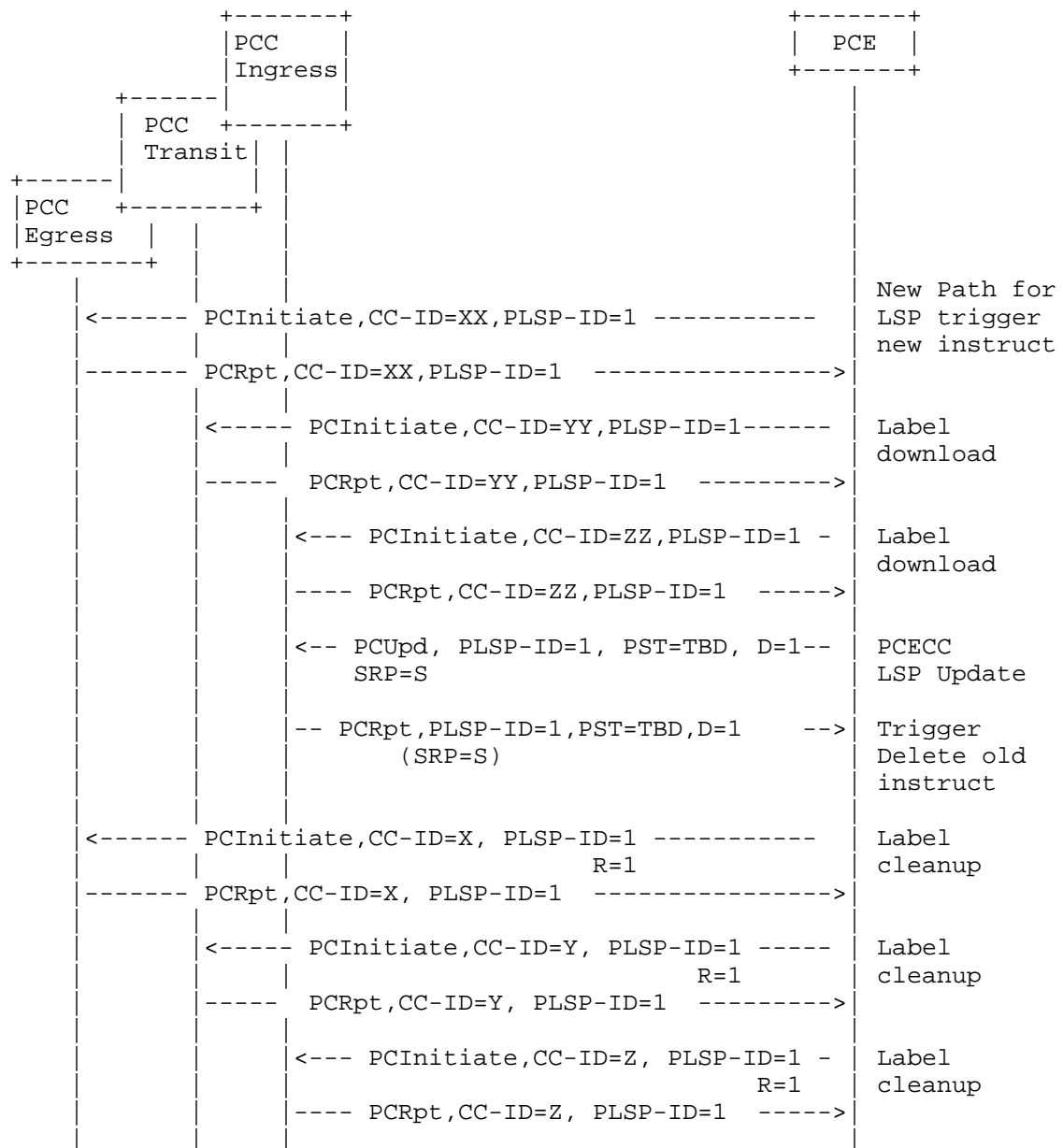


Once the label operations are completed, the PCE SHOULD send the PCUpd message to the Ingress PCC. The PCUpd message is as per [RFC8231].

#### 5.4.4. PCECC LSP Update

In case of a modification of PCECC LSP with a new path, a PCE sends a PCUpd message to the Ingress PCC. But to follow the make-before-break procedures, the PCECC first update new instructions based on the updated LSP and then update to ingress to switch traffic, before cleaning up the old instructions. A new CC-ID is used to identify the updated instruction, the existing identifiers in the LSP object identify the existing LSP. Once new instructions are downloaded, the PCE further updates the new path at the ingress which triggers the traffic switch on the updated path. The Ingress PCC acknowledges with a PCRpt message, on receipt of PCRpt message, the PCE does cleanup operation for the old LSP as described in Section 5.4.2.2.

The PCECC LSP Update sequence is shown below -



The modified PCECC LSP are considered to be 'up' by default. The Ingress MAY further choose to deploy a data plane check mechanism and report the status back to the PCE via PCRpt message.

#### 5.4.5. Re Delegation and Cleanup

As described in [RFC8281], a new PCE can gain control over the orphaned LSP. In case of PCECC LSP, the new PCE MUST also gain control over the central controllers instructions in the same way by sending a PCInitiate message that includes the SRP, LSP and CCI objects and carries the CC-ID and PLSP-ID identifying the instruction, it wants to take control of.

Further, as described in [RFC8281], the State Timeout Interval timer ensures that a PCE crash does not result in automatic and immediate disruption for the services using PCE-initiated LSPs. Similarly the central controller instructions are not removed immediately upon PCE failure. Instead, they are cleaned up on the expiration of this timer. This allows for network cleanup without manual intervention. The PCC MUST support removal of CCI as one of the behaviors applied on expiration of the State Timeout Interval timer.

#### 5.4.6. Synchronization of Central Controllers Instructions

The purpose of Central Controllers Instructions synchronization (labels in the context of this document) is to make sure that the PCE's view of CCI (Labels) matches with the PCC's Label allocation. This synchronization is performed as part of the LSP state synchronization as described in [RFC8231] and [RFC8233].

As per LSP State Synchronization [RFC8231], a PCC reports the state of its LSPs to the PCE using PCRpt messages and as per [RFC8281], PCE would initiate any missing LSPs and/or remove any LSPs that are not wanted. The same PCEP messages and procedure is also used for the Central Controllers Instructions synchronization. The PCRpt message includes the CCI and the LSP object to report the label forwarding instructions. The PCE would further remove any unwanted instructions or initiate any missing instructions.

#### 5.4.7. PCECC LSP State Report

As mentioned before, an Ingress PCC MAY choose to apply any OAM mechanism to check the status of LSP in the Data plane and MAY further send its status in PCRpt message to the PCE.

### 6. PCEP messages

As defined in [RFC5440], a PCEP message consists of a common header followed by a variable-length body made of a set of objects that can be either mandatory or optional. An object is said to be mandatory in a PCEP message when the object must be included for the message to be considered valid. For each PCEP message type, a set of rules is

defined that specify the set of objects that the message can carry. An implementation MUST form the PCEP messages using the object ordering specified in this document.

LSP-IDENTIFIERS TLV MUST be included in the LSP object for PCECC LSP.

### 6.1. The PCInitiate message

The PCInitiate message [RFC8281] can be used to download or remove the labels, the message has been extended as shown below -

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

Where:

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

The LSP and SRP object is defined in [RFC8231].

When PCInitiate message is used for central controller's instructions (labels), the SRP, LSP and CCI objects MUST be present. The SRP object is defined in [RFC8231] and if the SRP object is missing, the receiving PCC MUST send a PCErr message with Error-type=6 (Mandatory Object missing) and Error-value=10 (SRP object missing). The LSP object is defined in [RFC8231] and if the LSP object is missing, the receiving PCC MUST send a PCErr message with Error-type=6 (Mandatory Object missing) and Error-value=8 (LSP object missing). The CCI

object is defined in Section 7.3 and if the CCI object is missing, the receiving PCC MUST send a PCErr message with Error-type=6 (Mandatory Object missing) and Error-value=TBD (CCI object missing). More than one CCI object MAY be included in the PCInitiate message for the transit LSR.

To cleanup the SRP object must set the R (remove) bit.

At max two instances of CCI object would be included in case of transit LSR to encode both in-coming and out-going label forwarding instructions. Other instances MUST be ignored.

## 6.2. The PCRpt message

The PCRpt message can be used to report the labels that were allocated by the PCE, to be used during the state synchronization phase.

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

Where:

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

When PCRpt message is used to report the central controller's instructions (labels), the LSP and CCI objects MUST be present. The LSP object is defined in [RFC8231] and if the LSP object is missing, the receiving PCE MUST send a PCErr message with Error-type=6 (Mandatory Object missing) and Error-value=8 (LSP object missing).

The CCI object is defined in Section 7.3 and if the CCI object is missing, the receiving PCC MUST send a PCErr message with Error-type=6 (Mandatory Object missing) and Error-value=TBD (CCI object missing). Two CCI object can be included in the PCRpt message for the transit LSR.

## 7. PCEP Objects

The PCEP objects defined in this document are compliant with the PCEP object format defined in [RFC5440].

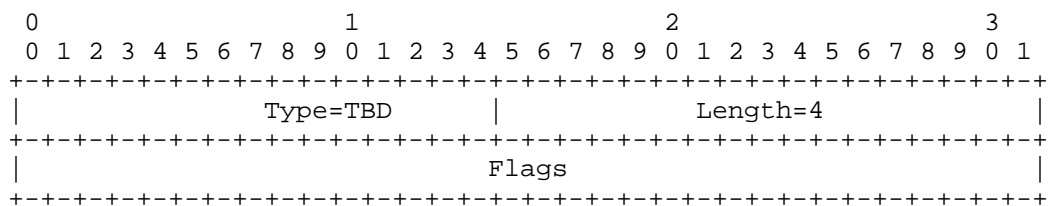
### 7.1. OPEN Object

This document defines a new optional TLVs for use in the OPEN Object.

#### 7.1.1. PCECC Capability sub-TLV

The PCECC-CAPABILITY sub-TLV is an optional TLV for use in the OPEN Object for PCECC capability advertisement in PATH-SETUP-TYPE-CAPABILITY TLV. Advertisement of the PCECC capability implies support of LSPs that are setup through PCECC as per PCEP extensions defined in this document.

Its format is shown in the following figure:



The type of the TLV is TBD and it has a fixed length of 4 octets.

The value comprises a single field - Flags (32 bits).

No flags are assigned right now.

Unassigned bits are considered reserved. They MUST be set to 0 on transmission and MUST be ignored on receipt.

## 7.2. PATH-SETUP-TYPE TLV

The PATH-SETUP-TYPE TLV is defined in [I-D.ietf-pce-lsp-setup-type]; this document defines a new PST value:

- o PST = TBD: Path is setup via PCECC mode.

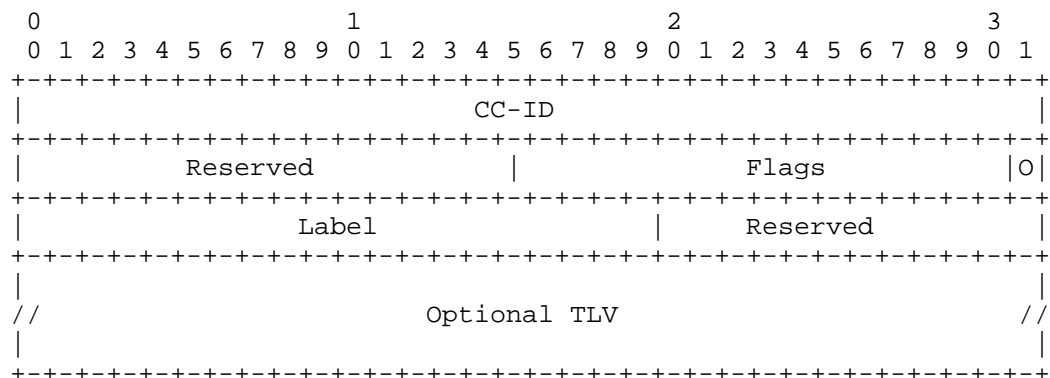
On a PCRpt/PCUpd/PCInitiate message, the PST=TBD in PATH-SETUP-TYPE TLV in SRP object indicates that this LSP was setup via a PCECC based mechanism.

## 7.3. CCI Object

The Central Control Instructions (CCI) Object is used by the PCE to specify the forwarding instructions (Label information in the context of this document) to the PCC, and MAY be carried within PCInitiate or PCRpt message for label download.

CCI Object-Class is TBD.

CCI Object-Type is 1 for the MPLS Label.



The fields in the CCI object are as follows:

**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:** is used to carry any additional information pertaining to the CCI. Currently, the following flag bit is defined:



- \* O bit(Out-label) : If the bit is set, it specifies the label is the OUT label and it is mandatory to encode the next-hop information (via IPV4-ADDRESS TLV or IPV6-ADDRESS TLV or UNNUMBERED-IPV4-ID-ADDRESS TLV in the CCI object). If the bit is not set, it specifies the label is the IN label and it is optional to encode the local interface information (via IPV4-ADDRESS TLV or IPV6-ADDRESS TLV or UNNUMBERED-IPV4-ID-ADDRESS TLV in the CCI object).

Label (20-bit): The Label information.

Reserved (12 bit): Set to zero while sending, ignored on receive.

#### 7.3.1. Address TLVs

This document defines the following TLVs for the CCI object to associate the next-hop information in case of an outgoing label and local interface information in case of an incoming label.

## IPv4-ADDRESS TLV:

```

      0                   1                   2                   3
      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                               |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                                     IPv4 address                           |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

## IPv6-ADDRESS TLV:

```

      0                   1                   2                   3
      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                               |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                                     //                                     //
|                                     IPv6 address (16 bytes)                 |
|                                     //                                     //
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

## UNNUMBERED-IPv4-ID-ADDRESS TLV:

```

      0                   1                   2                   3
      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                               |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                                     Node-ID                                |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                                     Interface ID                           |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

The address TLVs are as follows:

IPv4-ADDRESS TLV: an IPv4 address.

IPv6-ADDRESS TLV: an IPv6 address.

UNNUMBERED-IPv4-ID-ADDRESS TLV: a pair of Node ID / Interface ID tuples.

## 8. Security Considerations

The security considerations described in [RFC8231] and [RFC8281] apply to the extensions described in this document. Additional considerations related to a malicious PCE are introduced.

### 8.1. Malicious PCE

PCE has complete control over PCC to update the labels and can cause the LSP's to behave inappropriate and cause cause major impact to the network. As a general precaution, 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].

## 9. Manageability Considerations

### 9.1. Control of Function and Policy

A PCE or PCC implementation SHOULD allow to configure to enable/disable PCECC capability as a global configuration.

### 9.2. Information and Data Models

[RFC7420] describes the PCEP MIB, this MIB can be extended to get the PCECC capability status.

The PCEP YANG module [I-D.ietf-pce-pcep-yang] could be extended to enable/disable PCECC capability.

### 9.3. 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].

### 9.4. 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].

### 9.5. Requirements On Other Protocols

PCEP extensions defined in this document do not put new requirements on other protocols.

## 9.6. Impact On Network Operations

PCEP extensions defined in this document do not put new requirements on network operations.

## 10. IANA Considerations

### 10.1. PCEP TLV Type Indicators

IANA is requested to confirm the early allocation of the following TLV Type Indicator values within the "PCEP TLV Type Indicators" sub-registry of the PCEP Numbers registry, and to update the reference in the registry to point to this document, when it is an RFC:

Value	Meaning	Reference
TBD	PCECC-CAPABILITY	This document
TBD	IPV4-ADDRESS TLV	This document
TBD	IPV6-ADDRESS TLV	This document
TBD	UNNUMBERED-IPV4-ID-ADDRESS TLV	This document

### 10.2. New Path Setup Type Registry

IANA is requested to allocate new PST Field in PATH- SETUP-TYPE TLV. The allocation policy for this new registry should be by IETF Consensus. The new registry should contain the following value:

Value	Description	Reference
TBD	Traffic engineering path is setup using PCECC mode	This document

### 10.3. PCEP Object

IANA is requested to allocate new registry for CCI PCEP object.

Object-Class Value	Name	Reference
TBD	CCI Object-Type	This document
	1	MPLS Label

### 10.4. CCI Object Flag Field

IANA is requested to create a registry to manage the Flag field of the CCI object.

One bit to be defined for the CCI Object flag field in this document:

Codespace of the Flag field (CCI Object)

Bit	Description	Reference
7	Specifies label is out label	This document

### 10.5. PCEP-Error Object

IANA is requested to allocate new error types and error values within the "PCEP-ERROR Object Error Types and Values" sub-registry of the PCEP Numbers registry for the following errors:

Error-Type	Meaning	
-----	-----	
19	Invalid operation.	
	Error-value = TBD :	Attempted PCECC operations when PCECC capability was not advertised
	Error-value = TBD :	Stateful PCE capability was not advertised
	Error-value = TBD :	Unknown Label
6	Mandatory Object missing.	
	Error-value = TBD :	CCI object missing
TBD	PCECC failure.	
	Error-value = TBD :	Label out of range.
	Error-value = TBD :	Instruction failed.

### 11. Acknowledgments

We would like to thank Robert Tao, Changjing Yan, Tieying Huang and Avantika for their useful comments and suggestions.

### 12. References

#### 12.1. 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>>.
- [RFC7420] Koushik, A., Stephan, E., Zhao, Q., King, D., and J. Hardwick, "Path Computation Element Communication Protocol (PCEP) Management Information Base (MIB) Module", RFC 7420, DOI 10.17487/RFC7420, December 2014, <<https://www.rfc-editor.org/info/rfc7420>>.
- [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>>.
- [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>>.
- [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>>.
- [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>>.

## 12.2. 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>>.

- [RFC7025] Otani, T., Ogaki, K., Caviglia, D., Zhang, F., and C. Margaria, "Requirements for GMPLS Applications of PCE", RFC 7025, DOI 10.17487/RFC7025, September 2013, <<https://www.rfc-editor.org/info/rfc7025>>.
- [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>>.
- [RFC7491] King, D. and A. Farrel, "A PCE-Based Architecture for Application-Based Network Operations", RFC 7491, DOI 10.17487/RFC7491, March 2015, <<https://www.rfc-editor.org/info/rfc7491>>.
- [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>>.
- [RFC8283] Farrel, A., Ed., Zhao, Q., Ed., Li, Z., and C. Zhou, "An Architecture for Use of PCE and the PCE Communication Protocol (PCEP) in a Network with Central Control", RFC 8283, DOI 10.17487/RFC8283, December 2017, <<https://www.rfc-editor.org/info/rfc8283>>.
- [I-D.ietf-teas-pcecc-use-cases]  
Zhao, Q., Li, Z., Khasanov, B., Ke, Z., Fang, L., Zhou, C., Communications, T., and A. Rachitskiy, "The Use Cases for Using PCE as the Central Controller(PCECC) of LSPs", draft-ietf-teas-pcecc-use-cases-01 (work in progress), May 2017.
- [I-D.ietf-pce-lsp-setup-type]  
Sivabalan, S., Tantsura, J., Minei, I., Varga, R., and J. Hardwick, "Conveying path setup type in PCEP messages", draft-ietf-pce-lsp-setup-type-10 (work in progress), May 2018.
- [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-07 (work in progress), March 2018.

[I-D.zhao-pce-pcep-extension-pce-controller-sr]

Zhao, Q., Li, Z., Dhody, D., Karunanithi, S., Farrel, A.,  
and C. Zhou, "PCEP Procedures and Protocol Extensions for  
Using PCE as a Central Controller (PCECC) of SR-LSPs",  
draft-zhao-pce-pcep-extension-pce-controller-sr-02 (work  
in progress), March 2018.



## Appendix A. Contributor Addresses

Udayasree Palle  
Huawei Technologies  
Divyashree Techno Park, Whitefield  
Bangalore, Karnataka 560066  
India

EMail: udayasreereddy@gmail.com

Mahendra Singh Negi  
Huawei Technologies  
Divyashree Techno Park, Whitefield  
Bangalore, Karnataka 560066  
India

EMail: mahendrasingh@huawei.com

Katherine Zhao  
Huawei Technologies  
2330 Central Expressway  
Santa Clara, CA 95050  
USA

EMail: katherine.zhao@huawei.com

Boris Zhang  
Telus Ltd.  
Toronto  
Canada

EMail: boris.zhang@telus.com

## Authors' Addresses

Quintin Zhao  
Huawei Technologies  
125 Nagog Technology Park  
Acton, MA 01719  
USA

EMail: quintin.zhao@huawei.com

Zhenbin Li  
Huawei Technologies  
Huawei Bld., No.156 Beiqing Rd.  
Beijing 100095  
China

EMail: lizhenbin@huawei.com

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

EMail: dhruv.ietf@gmail.com

Satish Karunanithi  
Huawei Technologies  
Divyashree Techno Park, Whitefield  
Bangalore, Karnataka 560066  
India

EMail: satishk@huawei.com

Adrian Farrel  
Juniper Networks, Inc  
UK

EMail: adrian@olddog.co.uk

Chao Zhou  
Cisco Systems

EMail: choa.zhou@cisco.com