PCE Working Group D. Dhody
Internet-Draft U. Palle
Intended status: Standards Track Huawei Technologies
Expires: September 6, 2015 R. Singh
Juniper Networks
March 5, 2015

PCEP Extensions for MPLS-TE LSP Automatic Bandwidth Adjustment with stateful PCE
draft-dhody-pce-stateful-pce-auto-bandwidth-04

Abstract

The Path Computation Element Communication Protocol (PCEP) provides mechanisms for Path Computation Elements (PCEs) to perform path computations in response to Path Computation Clients (PCCs) requests. The stateful PCE extensions provide stateful control of Multiprotocol Label Switching (MPLS) Traffic Engineering Label Switched Paths (TE LSP) via PCEP, for a model where the PCC delegates control over one or more locally configured LSPs to the PCE.

This document describes the automatic bandwidth adjustment of such LSPs under the Active Stateful PCE model.

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 6, 2015.

Copyright Notice

Copyright (c) 2015 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

[RFC5440] describes the Path Computation Element Protocol (PCEP) as 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).

[I-D.ietf-pce-stateful-pce] specifies extensions to PCEP to enable stateful control of MPLS TE LSPs. In this document focus is on Active Stateful PCE where LSPs are configured on the PCC and control over them is delegated to the PCE.

Over time, based on the varying traffic pattern, an LSP established with certain bandwidth may require to adjust the reserved bandwidth over time automatically. Ingress Label Switch Router (LSR) samples the traffic rate at each sample-interval (BwSample) to determine the traffic information as Maximum Average Bandwidth (MaxAvgBw). Further adjustment to the reserved bandwidth should be made at every adjustment-interval automatically.

Enabling Auto-Bandwidth on a LSP results in the LSP automatically adjusting its bandwidth based on the actual traffic flowing through the LSP. A LSP can therefore be setup with some arbitrary (or zero) bandwidth value such that the LSP automatically monitors the traffic flow and adjusts its bandwidth every adjustment-interval period. The bandwidth adjustment uses the make-before-break signaling method so that there is no interruption to traffic flow. This is described in detail in Section 4.1. [I-D.ietf-pce-stateful-pce-app] describes the usecase for auto-bandwidth adjustment for passive and active stateful PCE.

There are two approaches to automatic bandwidth adjustments in case of active stateful PCE -

- PCE to decide adjusted bandwidth:

  * Active stateful PCE can use other information such as historical trending data, application-specific information about expected demands and central policy information along with realtime actual flow volumes to make smarter bandwidth adjustment to delegated LSPs. Since LSP has delegated control to the PCE, it is inherently suited that it should be stateful PCE that decides the bandwidth adjustments. But this requires PCC to report the realtime bandwidth usage as well as the configuration knobs etc.

- PCC to decide adjusted bandwidth:

  * This approach would be similar to passive stateful PCE model, where the headend (PCC) monitor and calculate the new adjusted bandwidth and request the computed adjusted bandwidth to be updated. The passive stateful PCE would use path request/reply

mechanism where as in active stateful PCE report/update mechanism is used to adjust the bandwidth. This approach only require PCC to report the calculated bandwidth to be adjusted. But this approach does not utilize the optimization advantages offered by the active stateful PCE.

This document defines extensions needed to support Auto-Bandwidth feature along with mechanism to provide traffic information of the LSPs in a stateful PCE model using PCEP.

This document does not exclude use of any other mechanism employed by stateful PCE to learn real time traffic information etc. But at the same time, using the same protocol (PCEP in this case) for updating and reporting the LSP parameters as well as to support automatic bandwidth adjustment is operationally beneficial.

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 following terminology is used in this document.

Active Stateful PCE: PCE that uses tunnel state information learned from PCCs to optimize path computations. Additionally, it actively updates tunnel parameters in those PCCs that delegated control over their tunnels to the PCE.

Delegation: :An operation to grant a PCE temporary rights to modify a subset of tunnel parameters on one or more PCC’s tunnels. Tunnels are delegated from a PCC to a PCE.

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.

TE LSP: Traffic Engineering Label Switched Path.

Note the additional terms defined in Section 4.1.

3. Motivation

An active stateful PCE can update the bandwidth for a delegated LSP via mechanisms described in [I-D.ietf-pce-stateful-pce]. Note that further extension are needed because of following reasons:

1. To identify the LSPs that would like to use this feature. Not all LSPs in some deployments would like their bandwidth to be dependent on the live traffic but be constant as set by the operator. Incase of PCC initiated LSP, they would be configured at PCC and PCEP should support a mechanism to identify the LSP with auto bandwidth feature enabled at the PCE. Where as for PCE initiated LSP, PCEP should support mechanims to request PCC to provide live traffic information.

2. Further for LSP with auto bandwidth feature enabled, operator should be able to specify the knobs to control this feature like the bandwidth-range etc and PCEP should support their encoding.

3. PCC would need to report the live traffic information using the same protocol (PCEP in this case) making the network operations easier.

Extensions as specified in this document is one of the way for PCE to learn this information. But at the same time a stateful PCE MAY choose to learn this information from other means like management, performance tools.

4. Architectural Overview

4.1. Auto-Bandwidth Overview

Auto-Bandwidth feature allows an LSP to automatically and dynamically adjust its reserved bandwidth over time, i.e. without network operator intervention. The bandwidth adjustment uses the make-before-break adaptive signaling method so that there is no interruption to traffic flow.

The new bandwidth reservation is determined by sampling the actual traffic flowing through the LSP. If the traffic flowing through the LSP is lower than the configured or current bandwidth of the LSP, the extra bandwidth is being reserved needlessly and being wasted. Conversely, if the actual traffic flowing through the LSP is higher than the configured or current bandwidth of the LSP, it can potentially cause congestion or packet loss. With Auto-Bandwidth feature, the LSP bandwidth can be set to some arbitrary value (even zero) during initial setup time, and it will be periodically adjusted over time based on the actual bandwidth requirement.

Note the following terms:

Maximum Average Bandwidth (MaxAvgBw): The maximum average bandwidth is the unit to measure the current traffic demand between a time interval. This is the maximum value of the averaged traffic pattern in a particular time interval.

Sample-Interval: The time interval in which the traffic rate is collected as a sample.

Adjustment-Interval: The time interval in which the bandwidth adjustment should be made based on the MaxAvgBw.

Minimum Bandwidth: The minimum bandwidth that should be reserved for the LSP.

Maximum Bandwidth: The maximum bandwidth that can be reserved for the LSP.

Report-Threshold: This value indicates when the current live traffic bandwidth sample (BwSample) must be reported to stateful PCE via PCRpt message. Only if the percentage difference between the current BwSample and the last BwSample is greater than or equal to the threshold percentage the LSP bandwidth is reported to PCE.

Adjust-Threshold: This value indicates when the bandwidth must be adjusted. Only if the percentage difference between the current MaxAvgBw and the current bandwidth allocation is greater than or equal to the threshold percentage the LSP bandwidth is adjusted to the current bandwidth demand.

4.2. Deploying Auto-Bandwidth Feature

The traffic rate is repeatedly sampled at each sample-interval (which can be configured by the user and the default value as 5 minutes). The sampled traffic rates are accumulated over the adjustment-interval period (which can be configured by the user and the default value as 24 hours).

The ingress LSR reports the live traffic information to the stateful PCE via the PCRpt message, to avoid multiple reports, the Report-Threshold percentage is used. Only if the percentage difference between the current BwSample and the last BwSample is greater than or equal to the threshold percentage the LSP bandwidth is reported to PCE.

Stateful PCE will adjust the bandwidth of the LSP to the highest sampled traffic rate amongst the set of samples taken over the adjustment-interval. Note that the highest sampled traffic rate could be higher or lower than the current LSP bandwidth. Only if the current MaxAvgBw and the current bandwidth allocation is greater than or equal to the Adjust-Threshold percentage the LSP bandwidth is adjusted to the current bandwidth demand.

Also to avoid multiple LSP re-signaling, sometimes operator set up longer adjustment intervals. However long adjustment-interval can also result in an undesirable effect of masking sudden changes in traffic patterns. To avoid this, the stateful PCE MAY pre-maturely expire the adjustment-interval to accommodate sudden bursts in traffic.

5. Extensions to the PCEP

5.1. AUTO-BANDWIDTH-ATTRIBUTE TLV

The AUTO-BANDWIDTH-ATTRIBUTE TLV can be included as an optional TLV in the LSP object as described in [I-D.ietf-pce-stateful-pce]. Whenever the LSP with Auto-Bandwidth feature enabled is delegated, AUTO-BANDWIDTH-ATTRIBUTE TLV is carried in PCRpt message. The TLV provides PCE with the 'local configurable knobs' of this feature. In case of PCE Initiated LSP ([I-D.ietf-pce-pce-initiated-lsp]) with this feature enabled, this TLV is included in LSP object with PCInitiate message.

The format of the AUTO-BANDWIDTH-ATTRIBUTE TLV is shown in the following figure:

```
 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]          |           Length=16           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Sample Int   |    Adj Int    | Rpt Threshold | Adj Threshold |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Minimum Bandwidth                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Maximum Bandwidth                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Reserved             |            Flags            |L|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

AUTO-BANDWIDTH-ATTRIBUTE TLV format

The type of the TLV is [TBD] and it has a fixed length of 16 octets.

The value contains the following fields:

Sample Int (8 bits): The Sample-Interval, time interval in which the traffic rate is collected at the PCC.

Adj Int (8 bits): The Adjustment-Interval, time interval in which the bandwidth adjustment should be made.

Rpt Threshold (8 bits): The Report-Threshold value is encoded in percentage. Only if the percentage difference between the between the current BwSample and the last BwSample is greater than or equal to the threshold percentage the real time bandwidth sample is reported to PCE.

Adj Threshold (8 bits): The Adjust-Threshold value is encoded in percentage. Only if the percentage difference between the current MaxAvgBw and the current bandwidth allocation is greater than or equal to the threshold percentage the LSP bandwidth is adjusted to the current bandwidth demand.

Minimum Bandwidth (32 bits): The minimum bandwidth allowed is encoded in IEEE floating point format (see [IEEE.754.1985]), expressed in bytes per second. Refer to Section 3.1.2 of [RFC3471] for a table of commonly used values.

Maximum Bandwidth (32 bits): The maximum bandwidth allowed is encoded in IEEE floating point format (see [IEEE.754.1985]), expressed in bytes per second. Refer to Section 3.1.2 of [RFC3471] for a table of commonly used values.

Flags (16 bits): One flag is currently defined:

* L (Live-Traffic - 1 bit): If set, PCC SHOULD report the live traffic information flowing on the LSP as per the Report-Threshold set. Otherwise PCC only reports the calculated bandwidth to be adjusted to the PCE.

Unassigned flags MUST be set to zero on transmission and MUST be ignored on receipt.

Reserved (16 bits): This field MUST be set to zero on transmission and MUST be ignored on receipt.

If the above parameters are not specified by the user, based on the local policy at Ingress (PCC) the default value can be encoded.

If no default value is specified at Ingress, value 'zero' can be encoded for the particular field. The stateful PCE can then apply its own default value based on the local policy.

5.2. BANDWIDTH Object

As per [RFC5440], the BANDWIDTH object is defined with two Object-Type values:

- o Requested Bandwidth: BANDWIDTH Object-Type is 1.

- o Re-optimization Bandwidth: Bandwidth of an existing TE LSP for which a reoptimization is requested. BANDWIDTH Object-Type is 2.

The new BANDWIDTH object type 3 [TBD] is used to specify the BwSample determined from the existing TE LSP Traffic flow at every sample-interval when L bit is set in AUTO-BANDWIDTH-ATTRIBUTE TLV. The Report-Threshold percentage is used to determine if there is a need to report the current BwSample.

If Live-Traffic (L-Bit) is not set, PCC only reports the calculated bandwidth to be adjusted (MaxAvgBw) to the PCE. This is done via the existing 'Requested Bandwidth with BANDWIDTH Object-Type as 1'.

5.3. The PCRpt Message

When the delegated LSP is enabled with the Auto-Bandwidth adjustment feature with Live-Traffic (L-Bit) set, PCC SHOULD include the BANDWIDTH object of type 3 [TBD] in the PCRpt message. The definition of the PCRpt message (see [I-D.ietf-pce-stateful-pce]) is unchanged.

When LSP is delegated to a PCE for the very first time, BANDWIDTH object of type 1 is used to specify the requested bandwidth in the PCRpt message. To report the live traffic flow information (as the BwSample) the BANDWIDTH object of type 3 [TBD] is encoded in further PCRpt message.

If Live-Traffic (L-Bit) is not set, PCC SHOULD include the BANDWIDTH object of type 1 to specify the he calculated bandwidth to be adjusted to the PCE.

5.4. The PCInitiate Message

For PCE Initiated LSP ([I-D.ietf-pce-pce-initiated-lsp]) with Auto-Bandwidth feature enabled, AUTO-BANDWIDTH-ATTRIBUTE TLV is included in LSP object with the PCInitiate message. The rest of the processing remains unchanged.

6. Security Considerations

This document defines a new BANDWIDTH type and AUTO-BANDWIDTH-ATTRIBUTE TLV which does not add any new security concerns beyond those discussed in [RFC5440] and [I-D.ietf-pce-stateful-pce] in itself. Some deployments may find the live traffic bandwidth information as extra sensitive and thus should employ suitable PCEP security mechanisms like TCP-AO or [I-D.ietf-pce-pceps].

7. Manageability Considerations

7.1. Control of Function and Policy

The Auto-Bandwidth feature MUST BE controlled per tunnel at Ingress (PCC), the values for parameters like sample-interval, adjustment-interval, minimum-bandwidth, maximum-bandwidth, report-threshold, adjust-threshold, Live-Traffic (L-Bit) SHOULD BE configurable by an operator.

7.2. Information and Data Models

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

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].

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

8. IANA Considerations

8.1. PCEP TLV Type Indicators

This document defines the following new PCEP TLVs; IANA is requested to make the following allocations from this registry.

Value	Meaning	Reference
TBD	AUTO-BANDWIDTH-ATTRIBUTE	[This I.D.]

8.2. AUTO-BANDWIDTH-ATTRIBUTE

This document requests that a registry is created to manage the Flags field in the AUTO-BANDWIDTH-ATTRIBUTE TLV in the LSP object. 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
31	Live-Traffic (L-Bit)	[This I.D.]

8.3. BANDWIDTH Object

This document defines new object type for the BANDWIDTH object; IANA is requested to make the following allocations from this registry.

Object-Class Value	Name	Reference
5	BANDWIDTH Object-Type 3: MaxAvgBw determined from the existing TE LSP Traffic flow.	[This I.D.]

9. Acknowledgments

We would like to thank Venugopal Reddy, Reeja Paul, Sandeep Boina and Avantika for their useful comments and suggestions.

10. References

10.1. Normative References

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

[RFC5440] Vasseur, JP. and JL. Le Roux, "Path Computation Element (PCE) Communication Protocol (PCEP)", RFC 5440, March 2009.

[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-10 (work in progress), October 2014.

[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-02 (work in progress), October 2014.

10.2. Informative References

[RFC3471] Berger, L., "Generalized Multi-Protocol Label Switching (GMPLS) Signaling Functional Description", RFC 3471, January 2003.

[RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, May 2008.

[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, December 2014.

[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-03 (work in progress), October 2014.

[I-D.ietf-pce-pceps] Lopez, D., Dios, O., Wu, W., and D. Dhody, "Secure Transport for PCEP", draft-ietf-pce-pceps-03 (work in progress), March 2015.

[IEEE.754.1985]
Institute of Electrical and Electronics Engineers, "Standard for Binary Floating-Point Arithmetic", IEEE Standard 754, August 1985.

Appendix A. Contributor Addresses

He Zekun
Tencent Holdings Ltd,
Shenzhen P.R.China

Email: kinghe@tencent.com

Xian Zhang
Huawei Technologies
Research Area F3-1B,
Huawei Industrial Base,
Shenzhen, 518129, China

Phone: +86-755-28972645
Email: zhang.xian@huawei.com

Young Lee
Huawei Technologies
1700 Alma Drive, Suite 100
Plano, TX 75075
US

Phone: +1 972 509 5599 x2240
Fax: +1 469 229 5397
EMail: leeyoung@huawei.com

Authors' Addresses

Dhruv Dhody
Huawei Technologies
Divyashree Techno Park, Whitefield
Bangalore, Karnataka 560037
India

EMail: dhruv.ietf@gmail.com

Udayasree Palle
Huawei Technologies
Divyashree Techno Park, Whitefield
Bangalore, Karnataka 560037
India

EMail: udayasree.palle@huawei.com

Ravi Singh
Juniper Networks
1194 N. Mathilda Ave.
Sunnyvale, CA 94089
USA

EMail: ravis@juniper.net

PCE Working Group
Internet-Draft
Intended status: Standards Track
Expires: May 20, 2017

D. Dhody
U. Palle
Huawei Technologies
R. Singh
Juniper Networks
R. Gandhi
Individual Contributor
L. Fang
eBay

November 16, 2016

PCEP Extensions for MPLS-TE LSP Automatic Bandwidth Adjustment with Stateful PCE

draft-dhody-pce-stateful-pce-auto-bandwidth-09

Abstract

The Path Computation Element Communication Protocol (PCEP) provides mechanisms for Path Computation Elements (PCEs) to perform path computations in response to Path Computation Clients (PCCs) requests. The stateful PCE extensions allow stateful control of Multi-Protocol Label Switching (MPLS) Traffic Engineering Label Switched Paths (TE LSPs) using PCEP.

This document describes PCEP extensions for automatic bandwidth adjustment when employing an Active Stateful PCE for both PCE-initiated and PCC-initiated 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."

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

[RFC5440] describes the Path Computation Element Protocol (PCEP) as a communication mechanism between a Path Computation Client (PCC) and a Path Control Element (PCE), or between PCE and PCE, that enables computation of Multi-Protocol Label Switching (MPLS) Traffic Engineering Label Switched Paths (TE LSPs).

[I-D.ietf-pce-stateful-pce] specifies extensions to PCEP to enable stateful control of MPLS TE LSPs. It describes two mode of operations - Passive stateful PCE and Active stateful PCE. In this document, the focus is on Active stateful PCE where LSPs are configured at the PCC and control over them is delegated to the PCE. Further [I-D.ietf-pce-pce-initiated-lsp] describes the setup, maintenance and teardown of PCE-initiated LSPs for the stateful PCE model.

Over time, based on the varying traffic pattern, an LSP established with certain bandwidth may require to adjust the bandwidth, reserved in the network automatically. Ingress Label Switch Router (LSR) collects the traffic rate at each sample interval to determine the bandwidth demand of the LSP. This bandwidth information is then used to adjust the LSP bandwidth periodically. This feature is commonly referred to as Auto-Bandwidth.

Enabling Auto-Bandwidth feature on an LSP results in the LSP automatically adjusting its bandwidth reservation based on the actual traffic flowing through the LSP. The initial LSP bandwidth can be set to an arbitrary value (including zero), in practice, it can be operator expected value based on design and planning. Once the LSP is set-up, the LSP monitors the traffic flow and adjusts its bandwidth every adjustment-interval period. The bandwidth adjustment uses the make-before-break signaling method so that there is no interruption to the traffic flow. The Auto-Bandwidth is described in detail in Section 4.1. [I-D.ietf-pce-stateful-pce-app] describes the use-case for Auto-Bandwidth adjustment for passive and active stateful PCE.

- o The PCC (head-end of the LSP) monitors and calculates the new adjusted bandwidth. The PCC reports the calculated bandwidth to be adjusted to the PCE.

- o This approach would be similar to passive stateful PCE model, while the passive stateful PCE uses path request/reply mechanism, the active stateful PCE uses report/update mechanism to adjust the LSP bandwidth.

- o For PCE-initiated LSP, the PCC is requested during the LSP initiation to monitor and calculate the new adjusted bandwidth.

This document defines extensions needed to support Auto-Bandwidth feature on the LSPs in a active stateful PCE model using PCEP.

2. Conventions Used in This Document

2.1. 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].

2.2. Terminology

The following terminology is used in this document.

Active Stateful PCE: PCE that uses tunnel state information learned from PCCs to optimize path computations. Additionally, it actively updates tunnel parameters in those PCCs that delegated control over their tunnels to the PCE.

Delegation: An operation to grant a PCE temporary rights to modify a subset of tunnel parameters on one or more PCC’s tunnels. Tunnels are delegated from a PCC to a PCE.

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.

TE LSP: Traffic Engineering Label Switched Path.

Note the Auto-Bandwidth feature specific terms defined in Section 4.1.

3. Requirements for PCEP Extensions

The PCEP speaker supporting this document MUST have a mechanism to advertise the automatic bandwidth adjustment capability.

PCEP extensions required are summarized in the following table.

PCC Initiated	PCE Initiated
PCC monitors the traffic and reports the calculated bandwidth to be adjusted to the PCE.	At the time of initiation, PCE request PCC to monitor the traffic and report the calculated bandwidth to be adjusted to the PCE.
No new extensions are needed.	Extension is needed for PCE to pass on the adjustment parameters at the time of Initiation.

Table 1: Auto-Bandwidth PCEP extensions

Further Auto-Bandwidth deployment considerations are summarized below:

o It is required to identify and inform the PCEP peer, the LSP that are enabled with Auto-Bandwidth feature. Not all LSPs in some deployments would like their bandwidth to be dependent on the real-time bandwidth usage but be constant as set by the operator.

- Further for the LSP with Auto-Bandwidth feature enabled, an operator should be able to specify the adjustment parameters (i.e. configuration knobs) to control this feature (e.g. minimum/ maximum bandwidth range) and PCEP peer should be informed.

4. Architectural Overview

4.1. Auto-Bandwidth Overview

Auto-Bandwidth feature allows an LSP to automatically and dynamically adjust its reserved bandwidth over time, i.e. without network operator intervention. The bandwidth adjustment uses the make-before-break signaling method so that there is no interruption to the traffic flow.

The new bandwidth reservation is determined by sampling the actual traffic flowing through the LSP. If the traffic flowing through the LSP is lower than the configured or current bandwidth of the LSP, the extra bandwidth is being reserved needlessly and being wasted. Conversely, if the actual traffic flowing through the LSP is higher than the configured or current bandwidth of the LSP, it can potentially cause congestion or packet loss in the network. With Auto-Bandwidth feature, the LSP bandwidth can be set to some arbitrary value (including zero) during initial setup time, and it will be periodically adjusted over time based on the actual bandwidth requirement.

Note the following definitions of the Auto-Bandwidth terms:

Maximum Average Bandwidth (MaxAvgBw): The maximum average bandwidth represents the current traffic bandwidth demand during a time interval. This is the maximum value of the averaged traffic bandwidth rate in a given adjustment-interval.

Adjusted Bandwidth: This is the Auto-Bandwidth computed bandwidth that needs to be adjusted for the LSP.

Sample-Interval: The periodic time interval at which the traffic rate is collected as a sample.

Bandwidth-Sample (BwSample): The bandwidth sample collected at every sample interval to measure the traffic rate.

Adjustment-Interval: The periodic time interval at which the bandwidth adjustment should be made using the MaxAvgBw.

Maximum-Bandwidth: The maximum bandwidth that can be reserved for the LSP.

Minimum-Bandwidth: The minimum bandwidth that can be reserved for the LSP.

Adjustment-Threshold: This value is used to decide when the bandwidth should be adjusted. If the percentage or absolute difference between the current MaxAvgBw and the current bandwidth reservation is greater than or equal to the threshold value, the LSP bandwidth is adjusted to the current bandwidth demand (Adjusted Bandwidth) at the adjustment-interval expiry.

Overflow-Count: This value is used to decide when the bandwidth should be adjusted when there is a sudden increase in traffic demand. This value indicates how many times consecutively, the percentage or absolute difference between the current MaxAvgBw and the current bandwidth reservation is greater than or equal to the Overflow-Threshold value.

Overflow-Threshold: This value is used to decide when the bandwidth should be adjusted when there is a sudden increase in traffic demand. If the percentage or absolute difference between the current MaxAvgBw and the current bandwidth reservation is greater than or equal to the threshold value, the overflow-condition is set to be met. The LSP bandwidth is adjusted to the current bandwidth demand bypassing the adjustment-interval if the overflow-condition is met consecutively for the Overflow-Count.

Underflow-Count: This value is used to decide when the bandwidth should be adjusted when there is a sudden decrease in traffic demand. This value indicates how many times consecutively, the percentage or absolute difference between the current MaxAvgBw and the current bandwidth reservation is greater than or equal to the Underflow-Threshold value.

Underflow-Threshold: This value is used to decide when the bandwidth should be adjusted when there is a sudden decrease in traffic demand. If the percentage or absolute difference between the current MaxAvgBw and the current bandwidth reservation is greater than or equal to the threshold value, the underflow-condition is set to be met. The LSP bandwidth is adjusted to the current bandwidth demand bypassing the adjustment-interval if the underflow-condition is met consecutively for the Underflow-Count.

4.2. Auto-bandwidth Theory of Operation

The traffic rate is periodically sampled at each sample-interval (which can be configured by the user and the default value as 5 minutes) by the head-end node of the LSP. The sampled traffic rates are accumulated over the adjustment-interval period (which can be configured by the user and the default value as 24 hours). The PCEP peer which is in-charge of calculating the bandwidth to be adjusted, will adjust the bandwidth of the LSP to the highest sampled traffic rate (MaxAvgBw) amongst the set of bandwidth samples collected over the adjustment-interval.

Note that the highest sampled traffic rate could be higher or lower than the current LSP bandwidth. Only if the difference between the current bandwidth demand (MaxAvgBw) and the current bandwidth reservation is greater than or equal to the Adjustment-Threshold (percentage or absolute value), the LSP bandwidth is adjusted to the current bandwidth demand (MaxAvgBw). Some LSPs are less eventful while other LSPs may encounter a lot of changes in the traffic pattern. PCE sets the intervals for adjustment based on the traffic pattern of the LSP.

In order to avoid frequent re-signaling, an operator may set a longer adjustment-interval value. However, longer adjustment-interval can result in an undesirable effect of masking sudden changes in traffic demands of an LSP. To avoid this, the Auto-Bandwidth feature may pre-maturely expire the adjustment-interval and adjust the LSP bandwidth to accommodate the sudden bursts of increase in traffic demand as an overflow condition or decrease in traffic demand as an underflow condition.

All thresholds in this document could be represented in both absolute value and percentage, and could be used together.

4.3. Scaling Considerations

It should be noted that any bandwidth change would require re-signaling of an LSP in a make-before-break fashion, which can further trigger preemption of lower priority LSPs in the network. When deployed under scale, this can lead to a signaling churn in the network. The Auto-bandwidth application algorithm is thus advised to take this into consideration before adjusting the LSP bandwidth. Operators are advised to set the values of various auto-bandwidth adjustment parameters appropriate for the deployed LSP scale.

If a PCE gets overwhelmed, it can notify the PCC to temporarily suspend its auto-bandwidth reporting (see Section 5.6). Similarly if a PCC gets overwhelmed due to signaling churn, it can notify the PCE to temporarily suspend the LSP bandwidth adjustment.

5. Extensions to the PCEP

5.1. Capability Advertisement

During PCEP Initialization Phase, PCEP Speakers (PCE or PCC) advertise their support of Automatic Bandwidth Adjustment. A PCEP Speaker includes the "Auto-Bandwidth Capability" TLV, in the OPEN Object to advertise its support for PCEP Auto-Bandwidth extensions. The presence of the "Auto-Bandwidth Capability" TLV in the OPEN Object indicates that the Automatic Bandwidth Adjustment is supported as described in this document.

The PCEP protocol extensions for Auto-Bandwidth adjustments MUST NOT be used if one or both PCEP Speakers have not included the "Auto-Bandwidth Capability" TLV in their respective OPEN message. If the PCEP speaker that supports the extensions of this draft but did not advertise this capability, then upon receipt of AUTO-BANDWIDTH-ATTRIBUTE TLV in LSPA object, it SHOULD generate a PCErr with error-type 19 (Invalid Operation), error-value TBD4 (Auto-Bandwidth capability was not advertised) and it will terminate the PCEP session.

5.1.1 AUTO-BANDWIDTH-CAPABILITY TLV

The AUTO-BANDWIDTH-CAPABILITY TLV is an optional TLV for use in the OPEN Object for Automatic Bandwidth Adjustment via PCEP capability advertisement. Its format is shown in the following figure:

```
 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=[TBD5]     |            Length=4           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                             Flags                             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

AUTO-BANDWIDTH-CAPABILITY TLV format

The type of the TLV is (TBD5) and it has a fixed length of 4 octets.

The value comprises a single field - Flags (32 bits). Currently no flags are defined for this TLV.

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

Advertisement of the Auto-Bandwidth capability TLV implies support of auto-bandwidth adjustment, as well as the objects, TLVs and procedures defined in this document.

5.2. AUTO-BANDWIDTH-ATTRIBUTE TLV

The AUTO-BANDWIDTH-ATTRIBUTE TLV provides the 'configurable knobs' of the feature and it can be included as an optional TLV in the LSPA Object (as described in [RFC5440]).

For PCE-Initiated LSP ([I-D.ietf-pce-pce-initiated-lsp]), this TLV is included in the LSPA Object with PCInitiate message. For delegated LSPs, this TLV is carried in PCRpt message in LSPA Object.

The TLV is encoded in all PCEP messages for the LSP till the auto bandwidth adjustment feature is enabled, the absence of the TLV indicate the PCEP speaker wish to disable the feature.

The format of the AUTO-BANDWIDTH-ATTRIBUTE TLV is shown in the following figure:

```
 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=[TBD1]         |           Length              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
//                          sub-TLVs                           //
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

AUTO-BANDWIDTH-ATTRIBUTE TLV format

Type: TBD1

Length: Variable

Value: This comprises one or more sub-TLVs.

Following sub-TLVs are defined in this document:

```
Type Len Name
-------------------------------------------------------------------
 1   4   Sample-Interval sub-TLV
```

```
 2   4   Adjustment-Interval sub-TLV
 3   4   Adjustment-Threshold sub-TLV
 4   4   Adjustment-Threshold-Percentage sub-TLV
 5   4   Minimum-Bandwidth sub-TLV
 6   4   Maximum-Bandwidth sub-TLV
 7   8   Overflow-Threshold sub-TLV
 8   4   Overflow-Threshold-Percentage sub-TLV
 9   8   Underflow-Threshold sub-TLV
10   4   Underflow-Threshold-Percentage sub-TLV
```

Future specification can define additional sub-TLVs.

The presence of AUTO-BANDWIDTH-ATTRIBUTE TLV in LSPA Object means that the automatic bandwidth adjustment feature is enabled. All sub-TLVs are optional and any unrecognized sub-TLV MUST be silently ignored. If a sub-TLV of same type appears more than once, only the first occurrence is processed and all others MUST be ignored.

The AUTO-BANDWIDTH-ATTRIBUTE TLV can also be carried in PCUpd message in LSPA Object in order to make updates to auto-bandwidth attributes such as Adjustment-Interval.

If sub-TLVs are not present, the default values based on the local policy are assumed.

The sub-TLVs are encoded to inform the PCEP peer the various sampling and adjustment parameters.

The following sub-sections describe the sub-TLVs which are currently defined to be carried within the AUTO-BANDWIDTH-ATTRIBUTE TLV.

5.2.1. Sample-Interval sub-TLV

The Sample-Interval sub-TLV specifies a time interval in seconds at which traffic samples are collected at the PCC.

The Type is 1, Length is 4, and the value comprises of 4-octet time interval, the valid range is from 1 to 604800, in seconds. The default value is 300 seconds.

```
 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              |           Length=4            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Sample-Interval                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Sample-Interval sub-TLV format

5.2.2. Adjustment-Interval sub-TLV

The Adjustment-Interval sub-TLV specifies a time interval in seconds at which bandwidth adjustment should be made.

The Type is 2, Length is 4, and the value comprises of 4-octet time interval, the valid range is from 1 to 604800, in seconds. The default value is 300 seconds.

```
 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            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     Adjustment-Interval                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Adjustment-Interval sub-TLV format

5.2.3. Adjustment Threshold

The sub-TLVs in this section are encoded to inform the PCEP peer the adjustment threshold parameters. An implementation MAY include both sub-TLVs for the absolute value and the percentage, in which case the bandwidth is adjusted when either of the adjustment threshold conditions are met.

5.2.3.1. Adjustment-Threshold sub-TLV

The Adjustment-Threshold sub-TLV is used to decide when the LSP bandwidth should be adjusted.

```
 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            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Adjustment Threshold                     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Adjustment-Threshold sub-TLV format

The Type is 3, Length is 4, and the value comprises of -

o Adjustment Threshold: The absolute Adjustment-Threshold bandwidth value, encoded in IEEE floating point format (see

[IEEE.754.1985]), expressed in bytes per second. Refer to Section 3.1.2 of [RFC3471] for a table of commonly used values.

If the difference between the current MaxAvgBw and the current bandwidth reservation is greater than or equal to the threshold value, the LSP bandwidth is adjusted to the current bandwidth demand.

5.2.3.2. Adjustment-Threshold-Percentage sub-TLV

The Adjustment-Threshold-Percentage sub-TLV is used to decide when the LSP bandwidth should be adjusted.

```
 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            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                   Reserved                    |  Percentage   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Adjustment-Threshold-Percentage sub-TLV format

The Type is 4, Length is 4, and the value comprises of -

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

o Percentage: The Adjustment-Threshold value, encoded in percentage (an integer from 0 to 100). If the percentage difference between the current MaxAvgBw and the current bandwidth reservation is greater than or equal to the threshold percentage, the LSP bandwidth is adjusted to the current bandwidth demand.

5.2.4. Minimum and Maximum Bandwidth Values

5.2.4.1. Minimum-Bandwidth sub-TLV

The Minimum-Bandwidth sub-TLV specify the minimum bandwidth allowed for the LSP, and is expressed in bytes per second. The LSP bandwidth cannot be adjusted below the minimum bandwidth value.

The Type is 5, Length is 4, and the value comprises of 4-octet bandwidth value encoded in IEEE floating point format (see [IEEE.754.1985]), expressed in bytes per second. Refer to Section 3.1.2 of [RFC3471] for a table of commonly used values.

```
 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=5              |           Length=4            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Minimum-Bandwidth                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Minimum-Bandwidth sub-TLV format

5.2.4.2. Maximum-Bandwidth sub-TLV

The Maximum-Bandwidth sub-TLV specify the maximum bandwidth allowed for the LSP, and is expressed in bytes per second. The LSP bandwidth cannot be adjusted above the maximum bandwidth value.

The Type is 6, Length is 4, and the value comprises of 4-octet bandwidth value encoded in IEEE floating point format (see [IEEE.754.1985]), expressed in bytes per second. Refer to Section 3.1.2 of [RFC3471] for a table of commonly used values.

```
 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=6              |           Length=4            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Maximum-Bandwidth                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Maximum-Bandwidth sub-TLV format

5.2.5. Overflow and Underflow Conditions

The sub-TLVs in this section are encoded to inform the PCEP peer the overflow and underflow threshold parameters. An implementation MAY include sub-TLVs for the absolute value and the percentage for the threshold, in which case the bandwidth is immediately adjusted when either of the adjustment threshold conditions are met consecutively for the given count.

5.2.5.1. Overflow-Threshold sub-TLV

The Overflow-Threshold sub-TLV is used to decide if the bandwidth should be adjusted immediately.

```
 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=7              |           Length=8            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                  Reserved                     |     Count     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       Overflow Threshold                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Overflow-Threshold sub-TLV format

The Type is 7, Length is 8, and the value comprises of -

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

- o Count: The Overflow-Count value, encoded in integer. The value 0 is considered to be invalid. The number of consecutive samples for which the overflow condition MUST be met for the LSP bandwidth to be immediately adjusted to the current bandwidth demand, bypassing the adjustment-interval.

- o Overflow Threshold: The absolute Overflow-Threshold bandwidth value, encoded in IEEE floating point format (see [IEEE.754.1985]), expressed in bytes per second. Refer to Section 3.1.2 of [RFC3471] for a table of commonly used values. If the increase of the current MaxAvgBw from the current bandwidth reservation is greater than or equal to the threshold value, the overflow condition is met.

5.2.5.2. Overflow-Threshold-Percentage sub-TLV

The Overflow-Threshold-Percentage sub-TLV is used to decide if the bandwidth should be adjusted immediately.

```
 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=8              |           Length=4            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Percentage |    Reserved                     |     Count     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Overflow-Threshold-Percentage sub-TLV format

The Type is 8, Length is 4, and the value comprises of -

- o Percentage: The Overflow-Threshold value, encoded in percentage (an integer from 0 to 100). If the percentage increase of the current MaxAvgBw from the current bandwidth reservation is greater than or equal to the threshold percentage, the overflow condition is met.

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

- o Count: The Overflow-Count value, encoded in integer. The value 0 is considered to be invalid. The number of consecutive samples for which the overflow condition MUST be met for the LSP bandwidth to be immediately adjusted to the current bandwidth demand, bypassing the adjustment-interval.

5.2.5.3. Underflow-Threshold sub-TLV

The Underflow-Threshold sub-TLV is used to decide if the bandwidth should be adjusted immediately.

```
 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=8            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       Reserved                |     Count     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Underflow Threshold                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Underflow-Threshold sub-TLV format

The Type is 9, Length is 8, and the value comprises of -

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

- o Count: The Underflow-Count value, encoded in integer. The value 0 is considered to be invalid. The number of consecutive samples for which the underflow condition MUST be met for the LSP bandwidth to be immediately adjusted to the current bandwidth demand, bypassing the adjustment-interval.

- o Underflow Threshold: The absolute Underflow-Threshold bandwidth value, encoded in IEEE floating point format (see [IEEE.754.1985]), expressed in bytes per second. Refer to Section 3.1.2 of [RFC3471] for a table of commonly used values. If the decrease of the current MaxAvgBw from the current bandwidth reservation is greater than or equal to the threshold value, the underflow condition is met.

5.2.5.4. Underflow-Threshold-Percentage sub-TLV

The Underflow-Threshold-Percentage sub-TLV is used to decide if the bandwidth should be adjusted immediately.

```
 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=4            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Percentage |    Reserved                     |      Count    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Underflow-Threshold-Percentage sub-TLV format

The Type is 10, Length is 4, and the value comprises of -

- o Percentage: The Underflow-Threshold value, encoded in percentage (an integer from 0 to 100). If the percentage decrease of the current MaxAvgBw from the current bandwidth reservation is greater than or equal to the threshold percentage, the underflow condition is met.

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

- o Count: The Underflow-Count value, encoded in integer. The value 0 is considered to be invalid. The number of consecutive samples for which the underflow condition MUST be met for the LSP bandwidth to be immediately adjusted to the current bandwidth demand, bypassing the adjustment-interval.

5.3. BANDWIDTH Object

As per [RFC5440], the BANDWIDTH object (Object-Class value 5) is defined with two Object-Type values as following:

- o Requested Bandwidth: BANDWIDTH Object-Type value is 1.

- o Re-optimization Bandwidth: Bandwidth of an existing TE LSP for which a re-optimization is requested. BANDWIDTH Object-Type value is 2.

PCC reports the calculated bandwidth to be adjusted (MaxAvgBw) to the

PCE using existing ’Requested Bandwidth with BANDWIDTH Object-Type as 1.

5.4. The PCInitiate Message

A PCInitiate message is a PCEP message sent by a PCE to a PCC to trigger LSP instantiation or deletion [I.D.ietf-pce-pce-initiated-lsp].

For the PCE-initiated LSP [I-D.ietf-pce-pce-initiated-lsp] with Auto-Bandwidth feature enabled, AUTO-BANDWIDTH-ATTRIBUTE TLV MUST be included in the LSPA object with the PCInitiate message. The rest of the processing remains unchanged.

5.5. The PCRpt Message

As specified in [I.D.ietf-pce-pce-initiated-lsp], the PCC creates the LSP using the attributes communicated by the PCE, and local values for the unspecified parameters. After the successful instantiation of the LSP, PCC automatically delegates the LSP to the PCE and generates an LSP State Report (PCRpt) for the LSP.

When LSP is delegated to a PCE for the very first time, BANDWIDTH object of type 1 is used to specify the requested bandwidth in the PCRpt message.

When the LSP is enabled with the Auto-Bandwidth feature, PCC SHOULD include the BANDWIDTH object of type 1 to specify the calculated bandwidth to be adjusted to the PCE in the PCRpt message.

The definition of the PCRpt message (see [I-D.ietf-pce-stateful-pce]) is unchanged by this document.

5.6. The PCNtf Message

As per [RFC5440], the PCEP Notification message (PCNtf) can be sent by a PCEP speaker to notify its peer of a specific event. As described in Section 4.3 of this document, a PCEP speaker SHOULD notify its PCEP peer that it is overwhelmed, and on receipt of such notification the peer SHOULD NOT send any PCEP messages related to auto-bandwidth adjustment. If a PCEP message related to auto-bandwidth adjustment is received, it MUST be silently ignored.

When a PCEP speaker is overwhelmed, it SHOULD notify its peer by sending a PCNtf message with Notification Type = TBD6 (Auto-bandwidth

Overwhelm State) and Notification Value = 1 (Entering auto-bandwidth overwhelm state). Optionally, OVERLOADED-DURATION TLV [RFC5440] MAY be included that specifies the time period during which no further PCEP messages related to auto-bandwidth adjustment should be sent. When the PCEP speaker is no longer in the overwhelm state and is available to process the auto-bandwidth adjustment, it SHOULD notify its peer by sending a PCNtf message with Notification Type = TBD6 (Auto-bandwidth Overwhelm State) and Notification Value = 2 (Clearing auto-bandwidth overwhelm state).

When Auto-Bandwidth feature is deployed, a PCE can send this notification to PCC when a PCC is reporting frequent auto-bandwidth adjustments. If a PCC is overwhelmed with re-signaling/re-routing, it can also notify the PCE to not adjust the LSP bandwidth while in overwhelm state.

6. Security Considerations

This document defines AUTO-BANDWIDTH-CAPABILITY TLV, AUTO-BANDWIDTH-ATTRIBUTE TLV which do not add any new security concerns beyond those discussed in [RFC5440] and [I-D.ietf-pce-stateful-pce].

Some deployments may find the reporting of the auto-bandwidth information as extra sensitive and thus SHOULD employ suitable PCEP security mechanisms like TCP-AO or [I-D.ietf-pce-pceps].

7. Manageability Considerations

7.1. Control of Function and Policy

The Auto-Bandwidth feature SHOULD be controlled per tunnel (at ingress (PCC) or PCE), the values for parameters like sample-interval, adjustment-interval, minimum-bandwidth, maximum-bandwidth, adjustment-threshold SHOULD be configurable by an operator.

7.2. Information and Data Models

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

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].

7.5. Requirements On Other Protocols

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

7.6. 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].

8. IANA Considerations

8.1. PCEP TLV Type Indicators

This document defines the following new PCEP TLVs; IANA is requested to make the following allocations from this registry. <http://www.iana.org/assignments/pcep/pcep.xhtml#pcep-tlv-type-indicators>.

Value	Name	Reference
TBD5	AUTO-BANDWIDTH-CAPABILITY	[This I.D.]
TBD1	AUTO-BANDWIDTH-ATTRIBUTE	[This I.D.]

8.2. AUTO-BANDWIDTH-CAPABILITY TLV Flag Field

IANA is requested to create a registry to manage the Flag field of the AUTO-BANDWIDTH-CAPABILITY TLV.

New bit numbers may be allocated only by an IETF Consensus action. 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

No bit is defined for the AUTO-BANDWIDTH-CAPABILITY TLV Object flag field in this document.

8.3. AUTO-BANDWIDTH-ATTRIBUTE Sub-TLV

This document specifies the AUTO-BANDWIDTH-ATTRIBUTE Sub-TLVs. IANA is requested to create an "AUTO-BANDWIDTH-ATTRIBUTE Sub-TLV Types" sub-registry in the "PCEP TLV Type Indicators" for the sub-TLVs carried in the AUTO-BANDWIDTH-ATTRIBUTE TLV. This document defines the following types:

Type	Name	Reference
0	Reserved	[This I.D.]
1	Sample-Interval sub-TLV	[This I.D.]
2	Adjustment-Interval sub-TLV	[This I.D.]
3	Adjustment-Threshold sub-TLV	[This I.D.]
4	Adjustment-Threshold-Percentage sub-TLV	[This I.D.]

5	Minimum-Bandwidth sub-TLV	[This I.D.]
6	Maximum-Bandwidth sub-TLV	[This I.D.]
7	Overflow-Threshold sub-TLV	[This I.D.]
8	Overflow-Threshold-Percentage sub-TLV	[This I.D.]
9	Underflow-Threshold sub-TLV	[This I.D.]
10	Underflow-Threshold-Percentage sub-TLV	[This I.D.]
11-65535	Unassigned	[This I.D.]

8.4. Error Object

This document defines a new Error-Value for PCErr message of type 19 (Invalid Operation) [I-D.ietf-pce-stateful-pce]); IANA is requested to make the following allocation from this registry. <http://www.iana.org/assignments/pcep/pcep.xhtml#pcep-error-object>

Error-Value	Meaning	Reference
TBD4	Auto-Bandwidth Capability was not Advertised	[This I.D.]

8.5. Notification Object

IANA is requested to allocate new Notification Types and Notification Values within the "Notification Object" sub-registry of the PCEP Numbers registry, as follows:

Type	Meaning		Reference
TBD6	Auto-Bandwidth Overwhelm State		[This I.D.]
	Notification-value=1:	Entering Auto-Bandwidth overwhelm state	
	Notification-value=2:	Clearing Auto-Bandwidth overwhelm state	

9. References

9.1. Normative References

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

[RFC5440] Vasseur, JP. and JL. Le Roux, "Path Computation Element (PCE) Communication Protocol (PCEP)", RFC 5440, March 2009.

[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 (work in progress).

[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 (work in progress).

9.2. Informative References

[RFC3471] Berger, L., "Generalized Multi-Protocol Label Switching (GMPLS) Signaling Functional Description", RFC 3471, January 2003.

[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, December 2014.

[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 (work in progress).

[I-D.ietf-pce-pceps] Lopez, D., Dios, O., Wu, W., and D. Dhody, "Secure Transport for PCEP", draft-ietf-pce-pceps (work in progress).

[IEEE.754.1985] Institute of Electrical and Electronics Engineers, "Standard for Binary Floating-Point Arithmetic", IEEE Standard 754, August 1985.

Acknowledgments

Authors would like to thank Robert Varga, Venugopal Reddy, Reeja Paul, Sandeep Boina, Avantika, JP Vasseur and Himanshu Shah for their useful comments and suggestions.

Contributors' Addresses

He Zekun
Tencent Holdings Ltd,
Shenzhen P.R.China

EMail: kinghe@tencent.com

Xian Zhang
Huawei Technologies
Research Area F3-1B,
Huawei Industrial Base,
Shenzhen, 518129
China

Phone: +86-755-28972645
EMail: zhang.xian@huawei.com

Young Lee
Huawei Technologies
1700 Alma Drive, Suite 100
Plano, TX 75075
USA

Phone: +1 972 509 5599 x2240
Fax: +1 469 229 5397
EMail: leeyoung@huawei.com

Authors' Addresses

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

EMail: dhruv.ietf@gmail.com

Udayasree Palle
Huawei Technologies
Divyashree Techno Park, Whitefield
Bangalore, Karnataka 560037
India

EMail: udayasree.palle@huawei.com

Ravi Singh
Juniper Networks
1194 N. Mathilda Ave.
Sunnyvale, CA 94089
USA

EMail: ravis@juniper.net

Rakesh Gandhi
Individual Contributor

EMail: rgandhi.ietf@gmail.com

Luyuan Fang
eBay
USA

EMail: luyuanf@gmail.com

PCE Working Group
Internet-Draft
Intended status: Standards Track
Expires: November 5, 2018

S. Sivabalan
Cisco Systems, Inc.
J. Tantsura
Nuage Networks
I. Minei
Google, Inc.
R. Varga
Pantheon Technologies SRO
J. Hardwick
Metaswitch Networks

May 4, 2018

Conveying path setup type in PCEP messages
draft-ietf-pce-lsp-setup-type-10

Abstract

A Path Computation Element (PCE) can compute Traffic Engineering (TE) paths through a network that are subject to various constraints. Currently, TE paths are Label Switched Paths (LSPs) which are set up using the RSVP-TE signaling protocol. However, other TE path setup methods are possible within the PCE architecture. This document proposes an extension to the PCE communication protocol (PCEP) to allow support for different path setup methods over a given PCEP session.

Status of This Memo

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

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

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

This Internet-Draft will expire on November 5, 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

[RFC5440] describes the Path Computation Element communication Protocol (PCEP) for communication between a Path Computation Client (PCC) and a Path Computation Element (PCE), or between a PCE and a PCE. A PCC requests a path subject to various constraints and optimization criteria from a PCE. The PCE responds to the PCC with a hop-by-hop path in an Explicit Route Object (ERO). The PCC uses the ERO to set up the path in the network.

[RFC8231] specifies extensions to PCEP that allow a PCC to delegate its LSPs to a PCE. The PCE can then update the state of LSPs delegated to it. In particular, the PCE may modify the path of an LSP by sending a new ERO. The PCC uses this ERO to re-route the LSP in a make-before-break fashion. [RFC8281] specifies a mechanism allowing a PCE to dynamically instantiate an LSP on a PCC by sending the ERO and the characteristics of the LSP. The PCC creates the LSP using the ERO and other attributes sent by the PCE.

So far, PCEP and its extensions have assumed that the TE paths are label switched and are established via the RSVP-TE protocol. However, other methods of LSP setup are possible in the PCE architecture (see [RFC4655] and [RFC4657]). This document generalizes PCEP to allow other LSP setup methods to be used. It defines two new TLVs and specifies the base procedures to facilitate this, as follows.

- The PATH-SETUP-TYPE-CAPABILITY TLV, which allows a PCEP speaker to announce which LSP setup methods it supports when the PCEP session is established.

- The PATH-SETUP-TYPE TLV, which allows a PCEP speaker to specify which setup method should be used for a given LSP. When multiple path setup types are deployed in a network, a given PCEP session may have to simultaneously support more than one path setup type. A PCEP speaker uses the PATH-SETUP-TYPE TLV to explicitly indicate the intended path setup type in the appropriate PCEP messages, unless the path setup type is RSVP-TE (which is assumed to be the path setup type if no other setup type is indicated). This is so that both the PCC and the PCE can take the necessary steps to set up the path.

This document defines a path setup type code for RSVP-TE. When a new path setup type (other than RSVP-TE) is introduced for setting up a path, a path setup type code and, optionally, a sub-TLV pertaining to the new path setup type will be defined by the document that specifies the new path setup type.

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.

LSR: Label Switching Router.

PCC: Path Computation Client.

PCE: Path Computation Element.

PCEP: Path Computation Element Protocol.

PST: Path Setup Type.

TLV: Type, Length, and Value.

3. Path Setup Type Capability TLV

A PCEP speaker indicates which PSTs it supports during the PCEP initialization phase, as follows. When the PCEP session is created, it sends an Open message with an OPEN object containing the PATH-SETUP-TYPE-CAPABILITY TLV. The format of this 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Type (TBD1)         |             Length            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                            Reserved           |  Num of PSTs  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     PST#1     |      ...      |     PST#N     |    Padding    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
//               Optional sub-TLVs (variable)                  //
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 1: PATH-SETUP-TYPE-CAPABILITY TLV

The TLV type is TBD1 (to be assigned by IANA). Its reserved field MUST be set to zero by the sender and MUST be ignored by the receiver. The other fields in the TLV are as follows.

Length: The total length in bytes of the remainder of the TLV, that is, excluding the Type and Length fields.

Number of PSTs: The number of PSTs in the following list, excluding padding.

List of PSTs: A list of the PSTs that the PCEP speaker supports. Each PST is a single byte in length. Duplicate entries in this list MUST be ignored. The PCEP speaker MUST pad the list with zeros so that it is a muliple of four bytes in length. This document defines the following PST value:

* PST = 0: Path is setup using the RSVP-TE signaling protocol.

Optional sub-TLVs: A list of sub-TLVs associated with the supported PSTs. Each PST has zero or one sub-TLVs associated with it, and each sub-TLV is associated with exactly one PST. Each sub-TLV MUST obey the rules for TLV formatting defined in ([RFC5440]). That is, each sub-TLV is padded to a four byte alignment, and the length field of each sub-TLV does not include the padding bytes. This document does not define any sub-TLVs; an example can be found in [I-D.ietf-pce-segment-routing].

A PCEP speaker MUST check that this TLV is correctly formatted, as follows.

o If there are no sub-TLVs, then the TLV length field MUST be equal to four bytes plus the size of the PST list, excluding any padding bytes.

o If there are sub-TLVs then the TLV Length field MUST be equal to four bytes plus the size of the PST list (rounded up to the nearest multiple of four) plus the size of the appended sub-TLVs excluding any padding bytes in the final sub-TLV.

o The Number of PSTs field MUST be greater than zero.

If a PCEP speaker receives a PATH-SETUP-TYPE-CAPABILITY TLV which violates these rules, then the PCEP speaker MUST send a PCErr message with Error-Type = 10 (Reception of an invalid object) and Error-Value = 11 (Malformed object) and MUST close the PCEP session. The PCEP speaker MAY include the malformed OPEN object in the PCErr message as well.

If a PCEP speaker receives an OPEN object with more than one PATH-SETUP-TYPE-CAPABILITY TLV then it MUST ignore all but the first instance of this TLV.

The absence of the PATH-SETUP-TYPE-CAPABILITY TLV from the OPEN object is equivalent to a PATH-SETUP-TYPE-CAPABILITY TLV containing a single PST of 0 (RSVP-TE signaling protocol) and no sub-TLVs. A PCEP speaker MAY omit the PATH-SETUP-TYPE-CAPABILITY TLV if the only PST it supports is RSVP-TE. If a PCEP speaker supports other PSTs besides RSVP-TE, then it SHOULD include the PATH-SETUP-TYPE-CAPABILITY TLV in its OPEN object.

If a PCEP speaker does not recognize the PATH-SETUP-TYPE-CAPABILITY TLV, it will ignore the TLV in accordance with [RFC5440].

4. Path Setup Type TLV

When a PCEP session is used to set up TE paths using different methods, the corresponding PCE and PCC must be aware of the path setup method used. That means, a PCE must be able to specify paths in the correct format and a PCC must be able to take control plane and forwarding plane actions appropriate to the PST.

```
 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 (28)           |           Length (4)          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           Reserved            |      PST      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 2: PATH-SETUP-TYPE TLV

PATH-SETUP-TYPE TLV is an optional TLV associated with the RP ([RFC5440]) and the SRP ([RFC8231]) objects. Its format is shown in the above figure. The TLV type is 28. Its reserved field MUST be set to zero. The one byte value contains the PST as defined for the PATH-SETUP-TYPE-CAPABILITY TLV.

The absence of the PATH-SETUP-TYPE TLV is equivalent to a PATH-SETUP-TYPE TLV with a PST value of 0 (RSVP-TE). A PCEP speaker MAY omit the TLV if the PST is RSVP-TE. If the RP or SRP object contains more than one PATH-SETUP-TYPE TLV, only the first TLV MUST be processed and the rest MUST be ignored.

If a PCEP speaker does not recognize the PATH-SETUP-TYPE TLV, it will ignore the TLV in accordance with [RFC5440], and will use RSVP-TE to set up the path.

5. Operation

During the PCEP initialization phase, if a PCEP speaker receives a PATH-SETUP-TYPE-CAPABILITY TLV from its peer, it MUST assume that the peer supports only the PSTs listed in the TLV. If the PCEP speaker and its peer have no PSTs in common, then the PCEP speaker MUST send a PCErr message with Error-Type = 21 (Invalid traffic engineering path setup type) and Error-Value = 2 (Mismatched path setup type) and close the PCEP session.

If the peer has sent no PATH-SETUP-TYPE-CAPABILITY TLV, then the PCEP speaker MUST infer that the peer supports path setup using at least RSVP-TE. The PCEP speaker MAY also infer that the peer supports other path setup types, but the means of inference are outside the scope of this document.

When a PCC sends a PCReq message to a PCE ([RFC5440]), it MUST include the PATH-SETUP-TYPE TLV in the RP object, unless the intended PST is RSVP-TE, in which case it MAY omit the PATH-SETUP-TYPE TLV. If the PCE is capable of expressing the path in a format appropriate to the intended PST, it MUST use the appropriate ERO format in the PCRep message.

When a PCE sends a PCRep message to a PCC ([RFC5440]), it MUST include the PATH-SETUP-TYPE TLV in the RP object, unless the PST is RSVP-TE, in which case it MAY omit the PATH-SETUP-TYPE TLV. If the PCE does not support the intended PST, it MUST send a PCErr message with Error-Type = 21 (Invalid traffic engineering path setup type) and Error-Value = 1 (Unsupported path setup type) and close the PCEP session. If the PSTs corresponding to the PCReq and PCRep messages do not match, the PCC MUST send a PCErr message with Error-Type = 21 (Invalid traffic engineering path setup type) and Error-Value = 2 (Mismatched path setup type) and close the PCEP session.

When a stateful PCE sends a PCUpd message ([RFC8231]) or a PCInitiate message ([RFC8281]) to a PCC, it MUST include the PATH-SETUP-TYPE TLV in the SRP object, unless the intended PST is RSVP-TE, in which case it MAY omit the PATH-SETUP-TYPE TLV. If the PCC does not support the PST associated with the PCUpd or PCInitiate message, it MUST send a PCErr message with Error-Type = 21 (Invalid traffic engineering path setup type) and Error-Value = 1 (Unsupported path setup type) and close the PCEP session.

When a PCC sends a PCRpt message to a stateful PCE ([RFC8231]), it MUST include the PATH-SETUP-TYPE TLV in the SRP object, unless the PST is RSVP-TE, in which case it MAY omit the PATH-SETUP-TYPE TLV. The PCC MUST include the SRP object in the PCRpt message if the PST is not RSVP-TE, even when the SRP-ID-number is the reserved value of 0x00000000. If the PCRpt message is triggered by a PCUpd or PCInitiate message, then the PST that the PCC indicates in the PCRpt MUST match the PST that the stateful PCE intended in the PCUpd or PCInitiate. If it does not, then the PCE MUST send a PCErr message with Error-Type = 21 (Invalid traffic engineering path setup type) and Error-Value = 2 (Mismatched path setup type) and close the PCEP session.

6. Manageability Considerations

This document generalises PCEP to allow path setup methods other than RSVP-TE to be used by the network (but does not define any new path setup types, besides RSVP-TE). It is possible that, in a given network, multiple path setup methods will be used. It is also possible that not all devices will support the same set of path setup methods. Managing networks that combine multiple path setup methods may therefore raise some challenges from a configuration and observability point of view.

Each document that defines a new Path Setup Type in the Path Setup Type Registry (Section 8.2) must include a manageability section. The manageability section must explain how operators can manage PCEP with the new path setup type. It must address the following questions, which are generally applicable when working with multiple path setup types in PCEP.

- What are the criteria for when devices will use the new path setup type in PCEP, and how can the operator control this?

- How can the network be migrated to the new path setup type, and are there any backwards compatibility issues that operators need to be aware of?

- Are paths set up using the new path setup type intended to coexist with other paths over the long term and, if so, how is this situation managed with PCEP?

- How can operators verify the correct operation of PCEP in the network with respect to the new path setup type? Which fault conditions must be reported to the operators?

- Are there any existing management interfaces (such as YANG models) that must be extended to model the operation of PCEP in the network with respect to the new path setup type?

See [RFC5706] for further guidance on how to write manageability sections in standards-track documents.

7. Security Considerations

The security considerations described in [RFC5440] and [RFC8281] are applicable to this specification. No additional security measure is required.

Note that, if the security mechanisms of [RFC5440] and [RFC8281] are not used, then the protocol described by this draft could be attacked in the following new way. An attacker, using a TCP man-in-the-middle attack, could inject error messages into the PCEP session when a particular PST is (or is not) used. By doing so, the attacker could potentially force the use of a specific PST, which may allow them to subsequently attack a weakness in that PST.

8. IANA Considerations

8.1. PCEP TLV Type Indicators

IANA is requested to confirm the early allocation of the following code point in the PCEP TLV Type Indicators registry.

Value	Description	Reference
28	PATH-SETUP-TYPE	This document

IANA is requested to allocate a new code point for the following TLV in the PCEP TLV Type Indicators registry.

Value	Description	Reference
TBD1	PATH-SETUP-TYPE-CAPABILITY	This document

Note to IANA: The above TLV type was not part of the early code point allocation that was done for this draft. It was added to the draft after the early code point allocation had taken place. Please assign a code point from the indicated registry and replace each instance of "TBD1" in this document with the allocated code point.

8.2. New Path Setup Type Registry

IANA is requested to create a new sub-registry within the "Path Computation Element Protocol (PCEP) Numbers" registry called "PCEP Path Setup Types". The allocation policy for this new registry should be by IETF Review. The new registry should contain the following value:

Value	Description	Reference
0	Path is setup using the RSVP-TE signaling protocol.	This document

8.3. PCEP-Error Object

IANA is requested to confirm the early allocation of the following code-points in the PCEP-ERROR Object Error Types and Values registry.

```
 Error-Type  Meaning
    10       Reception of an invalid object

               Error-value=11: Malformed object

 Error-Type  Meaning
    21       Invalid traffic engineering path setup type

               Error-value=0:  Unassigned
               Error-value=1:  Unsupported path setup type
               Error-value=2:  Mismatched path setup type
```

Note to IANA: the early allocation for Error-Type=10, Error-value=11 was originally done by draft-ietf-pce-segment-routing. However, we have since moved its definition into this document. Therefore, please update the reference for this Error-value in the indicated registry to point to RFC.ietf-pce-lsp-setup-type.

9. Contributors

The following people contributed to this document:

- Jan Medved
- Edward Crabbe

10. Acknowledgements

We like to thank Marek Zavodsky for valuable comments.

11. References

11.1. Normative References

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

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

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

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

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

11.2. 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.

[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>.

[RFC5706] Harrington, D., "Guidelines for Considering Operations and Management of New Protocols and Protocol Extensions", RFC 5706, DOI 10.17487/RFC5706, November 2009, <https://www.rfc-editor.org/info/rfc5706>.

Authors' Addresses

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

Email: msiva@cisco.com

Jeff Tantsura
Nuage Networks
755 Ravendale Drive
Mountain View, CA 94043
USA

Email: jefftant.ietf@gmail.com

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

Email: inaminei@google.com

Robert Varga
Pantheon Technologies SRO
Mlynske Nivy 56
Bratislava, 821 05
Slovakia

Email: nite@hq.sk

Jon Hardwick
Metaswitch Networks
100 Church Street
Enfield, Middlesex
UK

Email: jonathan.hardwick@metaswitch.com

PCE
Internet-Draft
Updates: 8408 (if approved)
Intended status: Standards Track
Expires: September 5, 2019

S. Sivabalan
C. Filsfils
Cisco Systems, Inc.
J. Tantsura
Apstra, Inc.
W. Henderickx
Nokia
J. Hardwick
Metaswitch Networks
March 4, 2019

PCEP Extensions for Segment Routing
draft-ietf-pce-segment-routing-16

Abstract

Segment Routing (SR) enables any head-end node to select any path without relying on a hop-by-hop signaling technique (e.g., LDP or RSVP-TE). It depends only on "segments" that are advertised by link-state Interior Gateway Protocols (IGPs). A Segment Routing Path can be derived from a variety of mechanisms, including an IGP Shortest Path Tree (SPT), explicit configuration, or a Path Computation Element (PCE). This document specifies extensions to the Path Computation Element Communication Protocol (PCEP) that allow a stateful PCE to compute and initiate Traffic Engineering (TE) paths, as well as a PCC to request a path subject to certain constraints and optimization criteria in SR networks.

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 September 5, 2019.

Copyright Notice

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

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

Table of Contents

1. Introduction

Segment Routing (SR) leverages the source routing paradigm. Using SR, a source node steers a packet through a path without relying on hop-by-hop signaling protocols such as LDP or RSVP-TE. Each path is specified as an ordered list of instructions called "segments". Each segment is an instruction to route the packet to a specific place in the network, or to perform a function on the packet. A database of segments can be distributed through the network using a routing protocol (such as IS-IS or OSPF) or by any other means. Several types of segment are defined. A node segment uniquely identifies a specific node in the SR domain. Each router in the SR domain associates a node segment with an ECMP-aware shortest path to the node that it identifies. An adjacency segment represents a unidirectional adjacency. An adjacency segment is local to the node which advertises it. Both node segments and adjacency segments can be used for SR.

[RFC8402] describes the SR architecture. The corresponding IS-IS and OSPF extensions are specified in [I-D.ietf-isis-segment-routing-extensions] and [I-D.ietf-ospf-segment-routing-extensions], respectively.

The SR architecture can be implemented using either an MPLS forwarding plane [I-D.ietf-spring-segment-routing-mpls] or an IPv6 forwarding plane [I-D.ietf-6man-segment-routing-header]. The MPLS forwarding plane can be applied to SR without any change, in which case an SR path corresponds to an MPLS Label Switching Path (LSP). This document is relevant to the MPLS forwarding plane only. In this document, "Node-SID" and "Adjacency-SID" denote Node Segment Identifier and Adjacency Segment Identifier respectively.

A Segment Routing path (SR path) can be derived from an IGP Shortest Path Tree (SPT). SR-TE paths may not follow an IGP SPT. Such paths may be chosen by a suitable network planning tool and provisioned on the ingress node of the SR-TE path.

[RFC5440] describes the Path Computation Element Communication Protocol (PCEP) for communication between a Path Computation Client (PCC) and a Path Computation Element (PCE) or between a pair of PCEs. A PCE computes paths for MPLS Traffic Engineering LSPs (MPLS-TE LSPs) based on various constraints and optimization criteria. [RFC8231] specifies extensions to PCEP that allow a stateful PCE to compute and recommend network paths in compliance with [RFC4657] and defines objects and TLVs for MPLS-TE LSPs. Stateful PCEP extensions provide synchronization of LSP state between a PCC and a PCE or between a pair of PCEs, delegation of LSP control, reporting of LSP state from a PCC to a PCE, controlling the setup and path routing of an LSP from a PCE to a PCC. Stateful PCEP extensions are intended for an operational model in which LSPs are configured on the PCC, and control over them is delegated to the PCE.

A mechanism to dynamically initiate LSPs on a PCC based on the requests from a stateful PCE or a controller using stateful PCE is specified in [RFC8281]. This mechanism is useful in Software Defined Networking (SDN) applications, such as on-demand engineering, or bandwidth calendaring [RFC8413].

It is possible to use a stateful PCE for computing one or more SR-TE paths taking into account various constraints and objective functions. Once a path is chosen, the stateful PCE can initiate an SR-TE path on a PCC using PCEP extensions specified in [RFC8281] using the SR specific PCEP extensions specified in this document. Additionally, using procedures described in this document, a PCC can request an SR path from either a stateful or a stateless PCE.

This specification relies on the procedures specified in [RFC8408] to exchange the segment routing capability and to specify that the path setup type of an LSP is segment routing. This specification also updates [RFC8408] to clarify the use of sub-TLVs in the PATH-SETUP-TYPE-CAPABILITY TLV. See Section 4.1.1 for details.

This specification provides a mechanism for a network controller (acting as a PCE) to instantiate candidate paths for an SR Policy onto a head-end node (acting as a PCC) using PCEP. For more information on the SR Policy Architecture, see [I-D.ietf-spring-segment-routing-policy].

2. Terminology

The following terminologies are used in this document:

ERO: Explicit Route Object

IGP: Interior Gateway Protocol

IS-IS: Intermediate System to Intermediate System

LSR: Label Switching Router

MSD: Base MPLS Imposition Maximum SID Depth, as defined in [RFC8491]

NAI: Node or Adjacency Identifier

OSPF: Open Shortest Path First

PCC: Path Computation Client

PCE: Path Computation Element

PCEP: Path Computation Element Communication Protocol

RRO: Record Route Object

SID: Segment Identifier

SR: Segment Routing

SR-DB: Segment Routing Database: the collection of SRGBs, SRLBs and SIDs and the objects they map to, advertised by a link state IGP

SRGB: Segment Routing Global Block

SRLB: Segment Routing Local Block

SR-TE: Segment Routing Traffic Engineering

3. Overview of PCEP Operation in SR Networks

In an SR network, the ingress node of an SR path prepends an SR header to all outgoing packets. The SR header consists of a list of SIDs (or MPLS labels in the context of this document). The header has all necessary information so that, in combination with the information distributed by the IGP, the packets can be guided from the ingress node to the egress node of the path; hence, there is no need for any signaling protocol.

In PCEP messages, LSP route information is carried in the Explicit Route Object (ERO), which consists of a sequence of subobjects. SR-TE paths computed by a PCE can be represented in an ERO in one of the following forms:

- o An ordered set of IP addresses representing network nodes/links.
- o An ordered set of SIDs, with or without the corresponding IP addresses.
- o An ordered set of MPLS labels, with or without corresponding IP address.

The PCC converts these into an MPLS label stack and next hop, as described in Section 5.2.2.

This document defines a new ERO subobject denoted by "SR-ERO subobject" capable of carrying a SID as well as the identity of the node/adjacency represented by the SID. SR-capable PCEP speakers should be able to generate and/or process such ERO subobject. An ERO containing SR-ERO subobjects can be included in the PCEP Path Computation Reply (PCRep) message defined in [RFC5440], the PCEP LSP Initiate Request message (PCInitiate) defined in [RFC8281], as well as in the PCEP LSP Update Request (PCUpd) and PCEP LSP State Report (PCRpt) messages defined in [RFC8231].

When a PCEP session between a PCC and a PCE is established, both PCEP speakers exchange their capabilities to indicate their ability to support SR-specific functionality.

A PCE can update an LSP that is initially established via RSVP-TE signaling to use an SR-TE path, by sending a PCUpd to the PCC that delegated the LSP to it ([RFC8231]). A PCC can update an undelegated LSP that is initially established via RSVP-TE signaling to use an SR-TE path as follows. First, it requests an SR-TE Path from a PCE by sending a PCReq message. If it receives a suitable path, it establishes the path in the data plane, and then tears down the original RSVP-TE path. If the PCE is stateful, then the PCC sends PCRpt messages indicating that the new path is set up and the old path is torn down, per [RFC8231].

Similarly, a PCE or PCC can update an LSP initially created with an SR-TE path to use RSVP-TE signaling, if necessary. This capability is useful for rolling back a change when a network is migrated from RSVP-TE to SR-TE technology.

A PCC MAY include an RRO containing the recorded LSP in PCReq and PCRpt messages as specified in [RFC5440] and [RFC8231], respectively.

This document defines a new RRO subobject for SR networks. The methods used by a PCC to record the SR-TE LSP are outside the scope of this document.

In summary, this document:

- o Defines a new ERO subobject, a new RRO subobject and new PCEP error codes.
- o Specifies how two PCEP speakers can establish a PCEP session that can carry information about SR-TE paths.
- o Specifies processing rules for the ERO subobject.
- o Defines a new path setup type to be used in the PATH-SETUP-TYPE and PATH-SETUP-TYPE-CAPABILITY TLVs ([RFC8408]).
- o Defines a new sub-TLV for the PATH-SETUP-TYPE-CAPABILITY TLV.

The extensions specified in this document complement the existing PCEP specifications to support SR-TE paths. As such, the PCEP messages (e.g., Path Computation Request, Path Computation Reply, Path Computation Report, Path Computation Update, Path Computation Initiate, etc.,) are formatted according to [RFC5440], [RFC8231], [RFC8281], and any other applicable PCEP specifications.

4. Object Formats

4.1. The OPEN Object

4.1.1. The Path Setup Type Capability TLV

[RFC8408] defines the PATH-SETUP-TYPE-CAPABILITY TLV for use in the OPEN object. The PATH-SETUP-TYPE-CAPABILITY TLV contains an optional list of sub-TLVs which are intended to convey parameters that are associated with the path setup types supported by a PCEP speaker.

This specification updates [RFC8408], as follows. It creates a new registry which defines the valid type indicators of the sub-TLVs of the PATH-SETUP-TYPE-CAPABILITY TLV (see Section 8.6). A PCEP speaker MUST NOT include a sub-TLV in the PATH-SETUP-TYPE-CAPABILITY TLV unless it appears in this registry. If a PCEP speaker receives a sub-TLV whose type indicator does not match one of those from the registry, or else is not recognised by the speaker, then the speaker MUST ignore the sub-TLV.

4.1.2. The SR PCE Capability sub-TLV

This document defines a new Path Setup Type (PST) for SR, as follows:

- PST = 1: Path is setup using Segment Routing Traffic Engineering.

A PCEP speaker SHOULD 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 SR-PCE-CAPABILITY sub-TLV. PCEP speakers use this sub-TLV to exchange information about their SR capability. If a PCEP speaker includes PST=1 in the PST List of the PATH-SETUP-TYPE-CAPABILITY TLV then it MUST also include the SR-PCE-CAPABILITY sub-TLV inside the PATH-SETUP-TYPE-CAPABILITY TLV.

The format of the SR-PCE-CAPABILITY sub-TLV is shown in the following figure:

```
 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=TBD11            |            Length=4           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Reserved              |   Flags   |N|X|      MSD      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 1: SR-PCE-CAPABILITY sub-TLV format

The code point for the TLV type is TBD11. The TLV length is 4 octets.

The 32-bit value is formatted as follows.

Reserved: MUST be set to zero by the sender and MUST be ignored by the receiver.

Flags: This document defines the following flag bits. The other bits MUST be set to zero by the sender and MUST be ignored by the receiver.

- N: A PCC sets this flag bit to 1 to indicate that it is capable of resolving a Node or Adjacency Identifier (NAI) to a SID.

- X: A PCC sets this flag bit to 1 to indicate that it does not impose any limit on the MSD.

Maximum SID Depth (MSD): specifies the maximum number of SIDs (MPLS label stack depth in the context of this document) that a PCC is capable of imposing on a packet. Section 5.1 explains the relationship between this field and the X flag.

4.2. The RP/SRP Object

To set up an SR-TE LSP using SR, the RP (Request Parameters) or SRP (Stateful PCE Request Parameters) object MUST include the PATH-SETUP-TYPE TLV, specified in [RFC8408], with the PST set to 1 (path setup using SR-TE).

The LSP-IDENTIFIERS TLV MAY be present for the above PST type.

4.3. ERO

An SR-TE path consists of one or more SIDs where each SID MAY be associated with the identifier that represents the node or adjacency corresponding to the SID. This identifier is referred to as the 'Node or Adjacency Identifier' (NAI). As described later, a NAI can be represented in various formats (e.g., IPv4 address, IPv6 address, etc). Furthermore, a NAI is used for troubleshooting purposes and, if necessary, to derive SID value as described below.

The ERO specified in [RFC5440] is used to carry SR-TE path information. In order to carry SID and/or NAI, this document defines a new ERO subobject referred to as "SR-ERO subobject" whose format is specified in the following section. An ERO carrying an SR-TE path consists of one or more ERO subobjects, and MUST carry only SR-ERO subobjects. Note that an SR-ERO subobject does not need to have both SID and NAI. However, at least one of them MUST be present.

When building the MPLS label stack from ERO, a PCC MUST assume that SR-ERO subobjects are organized as a last-in-first-out stack. The first subobject relative to the beginning of ERO contains the information about the topmost label. The last subobject contains information about the bottommost label.

4.3.1. SR-ERO Subobject

An SR-ERO subobject is formatted as shown in the following diagram.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|L|   Type=36   |     Length    |  NT   |     Flags     |F|S|C|M|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         SID (optional)                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
//                   NAI (variable, optional)                  //
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 2: SR-ERO subobject format

The fields in the SR-ERO Subobject are as follows:

The 'L' Flag: Indicates whether the subobject represents a loose-hop in the LSP [RFC3209]. If this flag is set to zero, a PCC MUST NOT overwrite the SID value present in the SR-ERO subobject. Otherwise, a PCC MAY expand or replace one or more SID values in the received SR-ERO based on its local policy.

Type: Set to 36.

Length: Contains the total length of the subobject in octets. The Length MUST be at least 8, and MUST be a multiple of 4. An SR-ERO subobject MUST contain at least one of a SID or an NAI. The flags described below indicate whether the SID or NAI fields are absent.

NAI Type (NT): Indicates the type and format of the NAI contained in the object body, if any is present. If the F bit is set to zero (see below) then the NT field has no meaning and MUST be ignored by the receiver. This document describes the following NT values:

NT=0 The NAI is absent.

NT=1 The NAI is an IPv4 node ID.

NT=2 The NAI is an IPv6 node ID.

NT=3 The NAI is an IPv4 adjacency.

NT=4 The NAI is an IPv6 adjacency with global IPv6 addresses.

NT=5 The NAI is an unnumbered adjacency with IPv4 node IDs.

NT=6 The NAI is an IPv6 adjacency with link-local IPv6 addresses.

Flags: Used to carry additional information pertaining to the SID. This document defines the following flag bits. The other bits MUST be set to zero by the sender and MUST be ignored by the receiver.

* M: If this bit is set to 1, the SID value represents an MPLS label stack entry as specified in [RFC3032]. Otherwise, the SID value is an administratively configured value which represents an index into an MPLS label space (either SRGB or SRLB) per [RFC8402].

* C: If the M bit and the C bit are both set to 1, then the TC, S, and TTL fields in the MPLS label stack entry are specified by the PCE. However, a PCC MAY choose to override these values according its local policy and MPLS forwarding rules. If the M bit is set to 1 but the C bit is set to zero, then the TC, S, and TTL fields MUST be ignored by the PCC. The PCC MUST set these fields according to its local policy and MPLS forwarding rules. If the M bit is set to zero then the C bit MUST be set to zero.

* S: When this bit is set to 1, the SID value in the subobject body is absent. In this case, the PCC is responsible for choosing the SID value, e.g., by looking up in the SR-DB using the NAI which, in this case, MUST be present in the subobject. If the S bit is set to 1 then the M and C bits MUST be set to zero.

* F: When this bit is set to 1, the NAI value in the subobject body is absent. The F bit MUST be set to 1 if NT=0, and otherwise MUST be set to zero. The S and F bits MUST NOT both be set to 1.

SID: The Segment Identifier. Depending on the M bit, it contains either:

* A 4 octet index defining the offset into an MPLS label space per [RFC8402].

* A 4 octet MPLS Label Stack Entry, where the 20 most significant bits encode the label value per [RFC3032].

NAI: The NAI associated with the SID. The NAI's format depends on the value in the NT field, and is described in the following section.

At least one of the SID and the NAI MUST be included in the SR-ERO subobject, and both MAY be included.

4.3.2. NAI Associated with SID

This document defines the following NAIs:

'IPv4 Node ID' is specified as an IPv4 address. In this case, the NT value is 1 and the NAI field length is 4 octets.

'IPv6 Node ID' is specified as an IPv6 address. In this case, the NT value is 2 and the NAI field length is 16 octets.

'IPv4 Adjacency' is specified as a pair of IPv4 addresses. In this case, the NT value is 3 and the NAI field length is 8 octets. The format of the NAI is shown in the following figure:

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Local IPv4 address                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     Remote IPv4 address                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 3: NAI for IPv4 adjacency

'IPv6 Global Adjacency' is specified as a pair of global IPv6 addresses. It is used to describe an IPv6 adjacency for a link that uses global IPv6 addresses. Each global IPv6 address is configured on a specific router interface, so together they identify an adjacency between a pair of routers. In this case, the NT value is 4 and the NAI field length is 32 octets. The format of the NAI is shown in the following figure:

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
//               Local IPv6 address (16 octets)                //
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
//               Remote IPv6 address (16 octets)               //
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 4: NAI for IPv6 global adjacency

'Unnumbered Adjacency with IPv4 NodeIDs' is specified as a pair of (node ID, interface ID) tuples. In this case, the NT value is 5 and the NAI field length is 16 octets. The format of the NAI is shown in the following figure:

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Local Node-ID                            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Local Interface ID                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Remote Node-ID                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                   Remote Interface ID                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 5: NAI for Unnumbered adjacency with IPv4 Node IDs

'IPv6 Link-Local Adjacency' is specified as a pair of (global IPv6 address, interface ID) tuples. It is used to describe an IPv6 adjacency for a link that uses only link local IPv6 addresses. Each global IPv6 address is configured on a specific router, so together they identify a pair of adjacent routers. The interface IDs identify the link that the adjacency is formed over. In this case, the NT value is 6 and the NAI field length is 40 octets. The format of the NAI is shown in the following figure:

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
//               Local IPv6 address (16 octets)                //
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Local Interface ID                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
//               Remote IPv6 address (16 octets)               //
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                   Remote Interface ID                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 6: NAI for IPv6 link-local adjacency

4.4. RRO

A PCC reports an SR-TE LSP to a PCE by sending a PCRpt message, per [RFC8231]. The RRO on this message represents the SID list that was applied by the PCC, that is, the actual path taken by the LSP. The procedures of [RFC8231] with respect to the RRO apply equally to this specification without change.

An RRO contains one or more subobjects called "SR-RRO subobjects" whose format 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=36    |     Length    |  NT   |     Flags     |F|S|C|M|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                              SID                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
//                        NAI (variable)                       //
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 7: SR-RRO Subobject format

The format of the SR-RRO subobject is the same as that of the SR-ERO subobject, but without the L flag.

A PCC MUST order the SR-RRO subobjects such that the first subobject relative to the beginning of the RRO identifies the first segment visited by the SR-TE LSP, and the last subobject identifies the final segment of the SR-TE LSP, that is, its endpoint.

4.5. METRIC Object

A PCC MAY request that PCE optimizes an individual path computation request to minimize the SID depth of the computed path by using the METRIC object defined in [RFC5440]. This document defines a new type for the METRIC object to be used for this purpose, as follows:

o T = 11: Maximum SID Depth of the requested path.

If the PCC includes a METRIC object of this type on a path computation request, then the PCE minimizes the SID depth of the computed path. If the B (bound) bit is set to to 1 in the METRIC object, then the PCE MUST NOT return a path whose SID depth exceeds the given metric-value. If the PCC did not set the X flag in its SR-PCE-CAPABILITY TLV, then it MUST set the B bit to 1. If the PCC set the X flag in its SR-PCE-CAPABILITY TLV, then it MAY set the B bit to 1 or zero.

If a PCEP session is established with a non-zero default MSD value, then the PCC MUST NOT send an MSD METRIC object with an MSD greater than the session’s default MSD. If the PCE receives a path computation request with an MSD METRIC object on such a session that is greater than the session’s default MSD, then it MUST consider the request invalid and send a PCErr with Error-Type = 10 ("Reception of an invalid object") and Error-Value 9 ("MSD exceeds the default for the PCEP session").

5. Procedures

5.1. Exchanging the SR PCE Capability

A PCC indicates that it is capable of supporting the head-end functions for SR-TE LSP by including the SR-PCE-CAPABILITY sub-TLV in the Open message that it sends to a PCE. A PCE indicates that it is capable of computing SR-TE paths by including the SR-PCE-CAPABILITY sub-TLV in the Open message that it sends to a PCC.

If a PCEP speaker receives a PATH-SETUP-TYPE-CAPABILITY TLV with a PST list containing PST=1, and supports that path setup type, then it checks for the presence of the SR-PCE-CAPABILITY sub-TLV. If that sub-TLV is absent, then the PCEP speaker MUST send a PCErr message with Error-Type 10 (Reception of an invalid object) and Error-Value TBD1 (Missing PCE-SR-CAPABILITY sub-TLV) and MUST then close the PCEP session. If a PCEP speaker receives a PATH-SETUP-TYPE-CAPABILITY TLV with a SR-PCE-CAPABILITY sub-TLV, but the PST list does not contain PST=1, then the PCEP speaker MUST ignore the SR-PCE-CAPABILITY sub-TLV.

If a PCC sets the N flag to 1, then the PCE MAY send an SR-ERO subobject containing NAI and no SID (see Section 5.2). Otherwise, the PCE MUST NOT send an SR-ERO subobject containing NAI and no SID.

The number of SIDs that can be imposed on a packet depends on the PCC's data plane's capability. If a PCC sets the X flag to 1 then the MSD is not used and MUST be set to zero. If a PCE receives an SR-PCE-CAPABILITY sub-TLV with the X flag set to 1 then it MUST ignore the MSD field and assumes that the sender can impose a SID stack of any depth. If a PCC sets the X flag to zero, then it sets the MSD field to the maximum number of SIDs that it can impose on a packet. In this case, the PCC MUST set the MSD to a number greater than zero. If a PCE receives an SR-PCE-CAPABILITY sub-TLV with the X flag and MSD both set to zero then it MUST send a PCErr message with Error-Type 10 (Reception of an invalid object) and Error-Value TBD10 (Maximum SID depth must be nonzero) and MUST then close the PCEP session.

Note that the MSD value exchanged via the SR-PCE-CAPABILITY sub-TLV indicates the SID/label imposition limit for the PCC node. It is anticipated that, in many deployments, the PCCs will have network interfaces that are homogeneous with respect to MSD (that is, each interface has the same MSD). In such cases, having a per-node MSD on the PCEP session is sufficient; the PCE SHOULD interpret this to mean that all network interfaces on the PCC have the given MSD. However, the PCE MAY also learn a per-node MSD and a per-interface MSD from the routing protocols, as specified in: [RFC8491]; [RFC8476];

[I-D.ietf-idr-bgp-ls-segment-routing-msd]. If the PCE learns the per-node MSD of a PCC from a routing protocol, then it MUST ignore the per-node MSD value in the SR-PCE-CAPABILITY sub-TLV and use the per-node MSD learned from the routing protocol instead. If the PCE learns the MSD of a network interface on a PCC from a routing protocol, then it MUST use the per-interface MSD instead of the MSD value in the SR-PCE-CAPABILITY sub-TLV when it computes a path that uses that interface.

Once an SR-capable PCEP session is established with a non-zero MSD value, the corresponding PCE MUST NOT send SR-TE paths with a number of SIDs exceeding that MSD value. If a PCC needs to modify the MSD value, it MUST close the PCEP session and re-establish it with the new MSD value. If a PCEP session is established with a non-zero MSD value, and the PCC receives an SR-TE path containing more SIDs than specified in the MSD value, the PCC MUST send a PCErr message with Error-Type 10 (Reception of an invalid object) and Error-Value 3 (Unsupported number of Segment ERO subobjects). If a PCEP session is established with an MSD value of zero, then the PCC MAY specify an MSD for each path computation request that it sends to the PCE, by including a "maximum SID depth" metric object on the request, as defined in Section 4.5.

The N flag, X flag and MSD value inside the SR-PCE-CAPABILITY sub-TLV are meaningful only in the Open message sent from a PCC to a PCE. As such, a PCE MUST set the N flag to zero, the X flag to 1 and MSD value to zero in an outbound message to a PCC. Similarly, a PCC MUST ignore any MSD value received from a PCE. If a PCE receives multiple SR-PCE-CAPABILITY sub-TLVs in an Open message, it processes only the first sub-TLV received.

5.2. ERO Processing

5.2.1. SR-ERO Validation

If a PCC does not support the SR PCE Capability and thus cannot recognize the SR-ERO or SR-RRO subobjects, it will respond according to the rules for a malformed object per [RFC5440].

On receiving an SR-ERO, a PCC MUST validate that the Length field, the S bit, the F bit and the NT field are consistent, as follows.

- o If NT=0, the F bit MUST be 1, the S bit MUST be zero and the Length MUST be 8.
- o If NT=1, the F bit MUST be zero. If the S bit is 1, the Length MUST be 8, otherwise the Length MUST be 12.

- o If NT=2, the F bit MUST be zero. If the S bit is 1, the Length MUST be 20, otherwise the Length MUST be 24.
- o If NT=3, the F bit MUST be zero. If the S bit is 1, the Length MUST be 12, otherwise the Length MUST be 16.
- o If NT=4, the F bit MUST be zero. If the S bit is 1, the Length MUST be 36, otherwise the Length MUST be 40.
- o If NT=5, the F bit MUST be zero. If the S bit is 1, the Length MUST be 20, otherwise the Length MUST be 24.
- o If NT=6, the F bit MUST be zero. If the S bit is 1, the Length MUST be 44, otherwise the Length MUST be 48.

If a PCC finds that the NT field, Length field, S bit and F bit are not consistent, it MUST consider the entire ERO invalid and MUST send a PCErr message with Error-Type = 10 ("Reception of an invalid object") and Error-Value = 11 ("Malformed object").

If a PCC does not recognise or support the value in the NT field, it MUST consider the entire ERO invalid and MUST send a PCErr message with Error-Type = 10 ("Reception of an invalid object") and Error-Value = TBD2 ("Unsupported NAI Type in Segment ERO subobject").

If a PCC receives an SR-ERO subobject in which the S and F bits are both set to 1 (that is, both the SID and NAI are absent), it MUST consider the entire ERO invalid and send a PCErr message with Error-Type = 10 ("Reception of an invalid object") and Error-Value = 6 ("Both SID and NAI are absent in SR-ERO subobject").

If a PCC receives an SR-ERO subobject in which the S bit is set to 1 and the F bit is set to zero (that is, the SID is absent and the NAI is present), but the PCC does not support NAI resolution, it MUST consider the entire ERO invalid and send a PCErr message with Error-Type = 4 ("Not supported object") and Error-Value = 4 ("Unsupported parameter").

If a PCC receives an SR-ERO subobject in which the S bit is set to 1 and either or both of the M or C bits is set to 1, it MUST consider the entire ERO invalid and send a PCErr message with Error-Type = 10 ("Reception of an invalid object") and Error-Value = 11 ("Malformed object").

If a PCC receives an SR-ERO subobject in which the S bit is set to zero and the M bit is set to 1, then the subobject contains an MPLS label. The PCC MAY choose not to accept a label provided by the PCE, based on it local policy. The PCC MUST NOT accept MPLS label value 3

(Implicit NULL), but it MAY accept other special purpose MPLS label values. If the PCC decides not to accept an MPLS label value, it MUST send a PCErr message with Error-Type = 10 ("Reception of an invalid object") and Error Value = 2 ("Bad label value").

If both M and C bits of an SR-ERO subobject are set to 1, and if a PCC finds erroneous setting in one or more of TC, S, and TTL fields, it MAY overwrite those fields with values chosen according to its own policy. If the PCC does not overwrite them, it MUST send a PCErr message with Error-Type = 10 ("Reception of an invalid object") and Error-Value = 4 ("Bad label format").

If the M bit of an SR-ERO subobject is set to zero but the C bit is set to 1, then the PCC MUST consider the entire ERO invalid and MUST send a PCErr message with Error-Type = 10 ("Reception of an invalid object") and Error-Value = 11 ("Malformed object").

If a PCC receives an SR-ERO subobject in which the S bit is set to zero and the M bit is set to zero, then the subobject contains a SID index value. If the SID is an Adjacency-SID then the L flag MUST NOT be set. If the L flag is set for an Adjacency-SID then the PCC MUST send a PCErr message with Error-Type = 10 ("Reception of an invalid object") and Error-Value = 11 ("Malformed object").

If a PCC detects that the subobjects of an ERO are a mixture of SR-ERO subobjects and subobjects of other types, then it MUST send a PCErr message with Error-Type = 10 ("Reception of an invalid object") and Error-Value = 5 ("ERO mixes SR-ERO subobjects with other subobject types").

The SR-ERO subobjects can be classified according to whether they contain a SID representing an MPLS label value, a SID representing an index value, or no SID. If a PCC detects that the SR-ERO subobjects are a mixture of more than one of these types, then it MUST send a PCErr message with Error-Type = 10 ("Reception of an invalid object") and Error-Value = TBD9 ("Inconsistent SIDs in SR-ERO / SR-RRO subobjects").

If an ERO specifies a new SR-TE path for an existing LSP and the PCC determines that the ERO contains SR-ERO subobjects that are not valid, then the PCC MUST NOT update the LSP.

5.2.2. Interpreting the SR-ERO

The SR-ERO contains a sequence of subobjects. Each SR-ERO subobject in the sequence identifies a segment that the traffic will be directed to, in the order given. That is, the first subobject identifies the first segment the traffic will be directed to, the second subobject represents the second segment, and so on.

The PCC interprets the SR-ERO by converting it to an MPLS label stack plus a next hop. The PCC sends packets along the segment routed path by prepending the MPLS label stack onto the packets and sending the resulting, modified packet to the next hop.

The PCC uses a different procedure to do this conversion, depending on the information that the PCE has provided in the subobjects.

- o If the subobjects contain SID index values, then the PCC converts them into the corresponding MPLS labels by following the procedure defined in [I-D.ietf-spring-segment-routing-mpls].

- o If the subobjects contain NAI only, the PCC first converts each NAI into a SID index value and then proceeds as above. To convert an NAI to a SID index, the PCC looks for a fully-specified prefix or adjacency matching the fields in the NAI. If the PCC finds a matching prefix/adjacency, and the matching prefix/adjacency has a SID associated with it, then the PCC uses that SID. If the PCC cannot find a matching prefix/adjacency, or if the matching prefix/adjacency has no SID associated with it, the PCC behaves as specified in Section 5.2.2.1.

- o If the subobjects contain MPLS labels, then the PCC looks up the offset of the first subobject's label in its SRGB or SRLB. This gives the first SID. The PCC pushes the labels in any remaining subobjects onto the packet (with the final subobject specifying the bottom-of-stack label).

For all cases above, after the PCC has imposed the label stack on the packet, it sends the packet to the segment identified by the first SID.

5.2.2.1. Handling Errors During SR-ERO Conversion

There are several errors that can occur during the process of converting an SR-ERO sequence to an MPLS label stack and a next hop. The PCC deals with them as follows.

- o If the PCC cannot find a SID index in the SR-DB, it MUST send a PCErr message with Error-Type = 10 ("Reception of an invalid object") and Error-Value = TBD3 ("Unknown SID").

- o If the PCC cannot find an NAI in the SR-DB, it MUST send a PCErr message with Error-Type = 10 ("Reception of an invalid object") and Error-Value = TBD4 ("NAI cannot be resolved to a SID").

- o If the PCC needs to convert a SID into an MPLS label value but cannot find the corresponding router’s SRGB in the SR-DB, it MUST send a PCErr message with Error-Type = 10 ("Reception of an invalid object") and Error-Value = TBD5 ("Could not find SRGB").

- o If the PCC finds that a router’s SRGB is not large enough for a SID index value, it MUST send a PCErr message with Error-Type = 10 ("Reception of an invalid object") and Error-Value = TBD6 ("SID index exceeds SRGB size").

- o If the PCC needs to convert a SID into an MPLS label value but cannot find the corresponding router’s SRLB in the SR-DB, it MUST send a PCErr message with Error-Type = 10 ("Reception of an invalid object") and Error-Value = TBD7 ("Could not find SRLB").

- o If the PCC finds that a router’s SRLB is not large enough for a SID index value, it MUST send a PCErr message with Error-Type = 10 ("Reception of an invalid object") and Error-Value = TBD8 ("SID index exceeds SRLB size").

- o If the number of labels in the computed label stack exceeds the maximum number of SIDs that the PCC can impose on the packet, it MUST send a PCErr message with Error-Type = 10 ("Reception of an invalid object") and Error-Value = 3 ("Unsupported number of Segment ERO subobjects").

If an ERO specifies a new SR-TE path for an existing LSP and the PCC encounters an error while processing the ERO, then the PCC MUST NOT update the LSP.

5.3. RRO Processing

The syntax checking rules that apply to the SR-RRO subobject are identical to those of the SR-ERO subobject, except as noted below.

If a PCEP speaker receives an SR-RRO subobject in which both SID and NAI are absent, it MUST consider the entire RRO invalid and send a PCErr message with Error-Type = 10 ("Reception of an invalid object") and Error-Value = 7 ("Both SID and NAI are absent in SR-RRO subobject").

If a PCE detects that the subobjects of an RRO are a mixture of SR-RRO subobjects and subobjects of other types, then it MUST send a PCErr message with Error-Type = 10 ("Reception of an invalid object") and Error-Value = 10 ("RRO mixes SR-RRO subobjects with other subobject types").

The SR-RRO subobjects can be classified according to whether they contain a SID representing an MPLS label value or a SID representing an index value, or no SID. If a PCE detects that the SR-RRO subobjects are a mixture of more than one of these types, then it MUST send a PCErr message with Error-Type = 10 ("Reception of an invalid object") and Error-Value = TBD9 ("Inconsistent SIDs in SR-ERO / SR-RRO subobjects").

6. Management Considerations

This document adds a new path setup type to PCEP to allow LSPs to be set up using segment routing techniques. This path setup type may be used with PCEP alongside other path setup types, such as RSVP-TE, or it may be used exclusively.

6.1. Controlling the Path Setup Type

The following factors control which path setup type is used for a given LSP.

- o The available path setup types are constrained to those that are supported by, or enabled on, the PCEP speakers. The PATH-SETUP-TYPE-CAPABILITY TLV indicates which path setup types a PCEP speaker supports. To use segment routing as a path setup type, it is a prerequisite that the PCC and PCE both include PST=1 in the list of supported path setup types in this TLV, and also include the SR-PCE-CAPABILITY sub-TLV.

- o When a PCE initiates an LSP, it proposes which path setup type to use by including it in the PATH-SETUP-TYPE TLV in the SRP object of the PCInitiate message. The PCE chooses the path setup type based on the capabilities of the network nodes on the path and on its local policy. The PCC MAY choose to accept the proposed path setup type, or to reject the PCInitiate request, based on its local policy.

- o When a PCC requests a path for an LSP, it can nominate a preferred path setup type by including it in the PATH-SETUP-TYPE TLV in the RP object of the PCReq message. The PCE MAY choose to reply with a path of the requested type, or to reply with a path of a different type, or to reject the request, based on the capabilities of the network nodes on the path and on its local policy.

The operator can influence the path setup type as follows.

- o Implementations MUST allow the operator to enable and disable the segment routing path setup type on a PCEP-speaking device.

Implementations MAY also allow the operator to enable and disable the RSVP-TE path setup type.

- o PCE implementations MUST allow the operator to specify that an LSP should be instantiated using segment routing or RSVP-TE as the proposed path setup type.

- o PCE implementations MAY allow the operator to configure a preference for the PCE to propose paths using segment routing or RSVP-TE in the absence of a specified path setup type.

- o PCC implementations MUST allow the operator to specify that a path requested for an LSP nominates segment routing or RSVP-TE as the path setup type.

- o PCC implementations MAY allow the operator to configure a preference for the PCC to nominate segment routing or RSVP-TE as the path setup type if none is specified for an LSP.

- o PCC implementations SHOULD allow the operator to configure a PCC to refuse to set up an LSP using an undesired path setup type.

6.2. Migrating a Network to Use PCEP Segment Routed Paths

This section discusses the steps that the operator takes when migrating a network to enable PCEP to set up paths using segment routing as the path setup type.

- o The operator enables the segment routing PST on the PCE servers.

- o The operator enables the segment routing PST on the PCCs.

- o The operator resets each PCEP session. The PCEP sessions come back up with segment routing enabled.

- o If the operator detects a problem, they can roll the network back to its initial state by disabling the segment routing PST on the PCEP speakers and resetting the PCEP sessions.

Note that the data plane is unaffected if a PCEP session is reset. Any LSPs that were set up before the session reset will remain in place and will still be present after the session comes back up.

An implementation SHOULD allow the operator to manually trigger a PCEP session to be reset.

An implementation MAY automatically reset a PCEP session when an operator reconfigures the PCEP speaker's capabilities. However, note that if the capabilities at both ends of the PCEP session are not reconfigured simultaneously, then the session could be reset twice, which could lead to unnecessary network traffic. Therefore, such implementations SHOULD allow the operator to override this behaviour and wait instead for a manual reset.

Once segment routing is enabled on a PCEP session, it can be used as the path setup type for future LSPs.

User traffic is not automatically migrated from existing LSPs onto segment routed LSPs just by enabling the segment routing PST in PCEP. The migration of user traffic from existing LSPs onto segment routing LSPs is beyond the scope of this document.

6.3. Verification of Network Operation

The operator needs the following information to verify that PCEP is operating correctly with respect to the segment routing path setup type.

- o An implementation SHOULD allow the operator to view whether the PCEP speaker sent the segment routing PST capability to its peer. If the PCEP speaker is a PCC, then the implementation SHOULD also allow the operator to view the values of the L and N flags that were sent, and the value of the MSD field that was sent.

- o An implementation SHOULD allow the operator to view whether the peer sent the segment routing PST capability. If the peer is a PCC, then the implementation SHOULD also allow the operator to view the values of the L and N flags and MSD fields that the peer sent.

- o An implementation SHOULD allow the operator to view whether the segment routing PST is enabled on the PCEP session.

- o If one PCEP speaker advertises the segment routing PST capability, but the other does not, then the implementation SHOULD create a log to inform the operator of the capability mismatch.

- o An implementation SHOULD allow the operator to view the PST that was proposed, or requested, for an LSP, and the PST that was actually used.

- o If a PCEP speaker decides to use a different PST to the one that was proposed, or requested, for an LSP, then the implementation SHOULD create a log to inform the operator that the expected PST has not been used. The log SHOULD give the reason for this choice (local policy, equipment capability etc.)

- o If a PCEP speaker rejects a segment routing path, then it SHOULD create a log to inform the operator, giving the reason for the decision (local policy, MSD exceeded etc.)

6.4. Relationship to Existing Management Models

The PCEP YANG module is defined in [I-D.ietf-pce-pcep-yang]. In future, this YANG module should be extended or augmented to provide the following additional information relating to segment routing:

- o The advertised PST capabilities and MSD per PCEP session.

- o The PST configured for, and used by, each LSP.

The PCEP MIB [RFC7420] could also be updated to include this information.

7. Security Considerations

The security considerations described in [RFC5440], [RFC8231], [RFC8281] and [RFC8408] are applicable to this specification. No additional security measure is required.

Note that this specification enables a network controller to instantiate a path in the network without the use of a hop-by-hop signaling protocol (such as RSVP-TE). This creates an additional vulnerability if the security mechanisms of [RFC5440], [RFC8231] and [RFC8281] are not used. If there is no integrity protection on the session, then an attacker could create a path which is not subjected to the further verification checks that would be performed by the signaling protocol.

Note that this specification adds the MSD field to the OPEN message (see Section 4.1.2) which discloses how many MPLS labels the sender can push onto packets that it forwards into the network. If the security mechanisms of [RFC8231] and [RFC8281] are not used with strong encryption, then an attacker could use this new field to gain intelligence about the capabilities of the edge devices in the network.

8. IANA Considerations

8.1. PCEP ERO and RRO subobjects

This document defines a new subobject type for the PCEP explicit route object (ERO), and a new subobject type for the PCEP record route object (RRO). The code points for subobject types of these objects is maintained in the RSVP parameters registry, under the

EXPLICIT_ROUTE and ROUTE_RECORD objects. IANA is requested to confirm the early allocation of the following code points in the RSVP Parameters registry for each of the new subobject types defined in this document.

Object	Subobject	Subobject Type
EXPLICIT_ROUTE	SR-ERO (PCEP-specific)	36
ROUTE_RECORD	SR-RRO (PCEP-specific)	36

8.2. New NAI Type Registry

IANA is requested to create a new sub-registry within the "Path Computation Element Protocol (PCEP) Numbers" registry called "PCEP SR-ERO NAI Types". The allocation policy for this new registry should be by IETF Review. The new registry should contain the following values:

Value	Description	Reference
0	NAI is absent.	This document
1	NAI is an IPv4 node ID.	This document
2	NAI is an IPv6 node ID.	This document
3	NAI is an IPv4 adjacency.	This document
4	NAI is an IPv6 adjacency with global IPv6 addresses.	This document
5	NAI is an unnumbered adjacency with IPv4 node IDs.	This document
6	NAI is an IPv6 adjacency with link-local IPv6 addresses.	This document

8.3. New SR-ERO Flag Registry

IANA is requested to create a new sub-registry, named "SR-ERO Flag Field", within the "Path Computation Element Protocol (PCEP) Numbers" registry to manage the Flag field of the SR-ERO subobject. New values are to be assigned by Standards Action [RFC8126]. 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
0-7	Unassigned	
8	NAI is absent (F)	This document
9	SID is absent (S)	This document
10	SID specifies TC, S and TTL in addition to an MPLS label (C)	This document
11	SID specifies an MPLS label (M)	This document

8.4. PCEP-Error Object

IANA is requested to confirm the early allocation of the code-points in the PCEP-ERROR Object Error Types and Values registry for the following new error-values:

Error-Type	Meaning	
10	Reception of an invalid object.	
	Error-value = 2:	Bad label value
	Error-value = 3:	Unsupported number of SR-ERO subobjects
	Error-value = 4:	Bad label format
	Error-value = 5:	ERO mixes SR-ERO subobjects with other subobject types
	Error-value = 6:	Both SID and NAI are absent in SR-ERO subobject
	Error-value = 7:	Both SID and NAI are absent in SR-RRO subobject
	Error-value = 9:	MSD exceeds the default for the PCEP session
	Error-value = 10:	RRO mixes SR-RRO subobjects with other subobject types
	Error-value = TBD1:	Missing PCE-SR-CAPABILITY sub-TLV

Error-value = TBD2:	Unsupported NAI Type in SR-ERO subobject
Error-value = TBD3:	Unknown SID
Error-value = TBD4:	NAI cannot be resolved to a SID
Error-value = TBD5:	Could not find SRGB
Error-value = TBD6:	SID index exceeds SRGB size
Error-value = TBD7:	Could not find SRLB
Error-value = TBD8:	SID index exceeds SRLB size
Error-value = TBD9:	Inconsistent SIDs in SR-ERO / SR-RRO subobjects
Error-value = TBD10:	MSD must be nonzero

Note to IANA: this draft originally had an early allocation for Error-value=11 (Malformed object) in the above list. However, we have since moved the definition of that code point to RFC8408.

Note to IANA: some Error-values in the above list were defined after the early allocation took place, and so do not currently have a code point assigned. Please assign code points from the indicated registry and replace each instance of "TBD1", "TBD2" etc. in this document with the respective code points.

Note to IANA: some of the Error-value descriptive strings above have changed since the early allocation. Please refresh the registry.

8.5. PCEP TLV Type Indicators

IANA is requested to confirm the early allocation of the following code point in the PCEP TLV Type Indicators registry. Note that this TLV type indicator is deprecated but retained in the registry to ensure compatibility with early implementations of this specification. See Appendix A for details.

Value	Meaning	Reference
26	SR-PCE-CAPABILITY (deprecated)	This document

8.6. PATH-SETUP-TYPE-CAPABILITY Sub-TLV Type Indicators

IANA is requested to create a new sub-registry, named "PATH-SETUP-TYPE-CAPABILITY Sub-TLV Type Indicators", within the "Path Computation Element Protocol (PCEP) Numbers" registry to manage the type indicator space for sub-TLVs of the PATH-SETUP-TYPE-CAPABILITY TLV. New values are to be assigned by Standards Action [RFC8126]. The valid range of values in the registry is 0-65535. IANA is requested to initialize the registry with the following values. All other values in the registry should be marked as "Unassigned".

Value	Meaning	Reference
0	Reserved	This document
TBD11 (recommended 26)	SR-PCE-CAPABILITY	This document

Note to IANA: Please replace each instance of "TBD11" in this document with the allocated code point. We have recommended that value 26 be used for consistency with the deprecated value in the PCEP TLV Type Indicators registry.

8.7. New Path Setup Type

[RFC8408] created a sub-registry within the "Path Computation Element Protocol (PCEP) Numbers" registry called "PCEP Path Setup Types". IANA is requested to allocate a new code point within this registry, as follows:

Value	Description	Reference
1	Traffic engineering path is setup using Segment Routing.	This document

8.8. New Metric Type

IANA is requested to confirm the early allocation of the following code point in the PCEP METRIC object T field registry:

Value	Description	Reference
11	Segment-ID (SID) Depth.	This document

8.9. SR PCE Capability Flags

IANA is requested to create a new sub-registry, named "SR Capability Flag Field", within the "Path Computation Element Protocol (PCEP) Numbers" registry to manage the Flag field of the SR-PCE-CAPABILITY TLV. New values are to be assigned by Standards Action [RFC8126]. 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
0-5	Unassigned	
6	Node or Adjacency Identifier (NAI) is supported (N)	This document
7	Unlimited Maximum SID Depth (X)	This document

Note to IANA: The name of bit 7 has changed from "Unlimited Maximum SID Depth (L)" to "Unlimited Maximum SID Depth (X)".

9. Contributors

The following people contributed to this document:

- Lakshmi Sharma
- Jan Medved
- Edward Crabbe
- Robert Raszuk
- Victor Lopez

10. Acknowledgements

We thank Ina Minei, George Swallow, Marek Zavodsky, Dhruv Dhody, Ing-Wher Chen and Tomas Janciga for the valuable comments.

11. References

11.1. Normative References

[I-D.ietf-spring-segment-routing-mpls]
Bashandy, A., Filsfils, C., Previdi, S., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing with MPLS data plane", draft-ietf-spring-segment-routing-mpls-18 (work in progress), December 2018.

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

[RFC3032] Rosen, E., Tappan, D., Fedorkow, G., Rekhter, Y., Farinacci, D., Li, T., and A. Conta, "MPLS Label Stack Encoding", RFC 3032, DOI 10.17487/RFC3032, January 2001, <https://www.rfc-editor.org/info/rfc3032>.

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

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

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

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

[RFC8402] Filsfils, C., Ed., Previdi, S., Ed., Ginsberg, L., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing Architecture", RFC 8402, DOI 10.17487/RFC8402, July 2018, <https://www.rfc-editor.org/info/rfc8402>.

[RFC8408] Sivabalan, S., Tantsura, J., Minei, I., Varga, R., and J. Hardwick, "Conveying Path Setup Type in PCE Communication Protocol (PCEP) Messages", RFC 8408, DOI 10.17487/RFC8408, July 2018, <https://www.rfc-editor.org/info/rfc8408>.

[RFC8491] Tantsura, J., Chunduri, U., Aldrin, S., and L. Ginsberg, "Signaling Maximum SID Depth (MSD) Using IS-IS", RFC 8491, DOI 10.17487/RFC8491, November 2018, <https://www.rfc-editor.org/info/rfc8491>.

11.2. Informative References

[I-D.ietf-6man-segment-routing-header] Filsfils, C., Previdi, S., Leddy, J., Matsushima, S., and d. daniel.voyer@bell.ca, "IPv6 Segment Routing Header (SRH)", draft-ietf-6man-segment-routing-header-16 (work in progress), February 2019.

[I-D.ietf-idr-bgp-ls-segment-routing-msd]
Tantsura, J., Chunduri, U., Mirsky, G., and S. Sivabalan, "Signaling MSD (Maximum SID Depth) using Border Gateway Protocol Link-State", draft-ietf-idr-bgp-ls-segment-routing-msd-02 (work in progress), August 2018.

[I-D.ietf-isis-segment-routing-extensions]
Previdi, S., Ginsberg, L., Filsfils, C., Bashandy, A., Gredler, H., and B. Decraene, "IS-IS Extensions for Segment Routing", draft-ietf-isis-segment-routing-extensions-22 (work in progress), December 2018.

[I-D.ietf-ospf-segment-routing-extensions]
Psenak, P., Previdi, S., Filsfils, C., Gredler, H., Shakir, R., Henderickx, W., and J. Tantsura, "OSPF Extensions for Segment Routing", draft-ietf-ospf-segment-routing-extensions-27 (work in progress), December 2018.

[I-D.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-09 (work in progress), October 2018.

[I-D.ietf-spring-segment-routing-policy]
Filsfils, C., Sivabalan, S., daniel.voyer@bell.ca, d., bogdanov@google.com, b., and P. Mattes, "Segment Routing Policy Architecture", draft-ietf-spring-segment-routing-policy-02 (work in progress), October 2018.

[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>.

[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>.

[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>.

[RFC8413] Zhuang, Y., Wu, Q., Chen, H., and A. Farrel, "Framework for Scheduled Use of Resources", RFC 8413, DOI 10.17487/RFC8413, July 2018, <https://www.rfc-editor.org/info/rfc8413>.

[RFC8476] Tantsura, J., Chunduri, U., Aldrin, S., and P. Psenak, "Signaling Maximum SID Depth (MSD) Using OSPF", RFC 8476, DOI 10.17487/RFC8476, December 2018, <https://www.rfc-editor.org/info/rfc8476>.

Appendix A. Compatibility with Early Implementations

An early implementation of this specification will send the SR-CAPABILITY-TLV as a top-level TLV in the OPEN object instead of sending the PATH-SETUP-TYPE-CAPABILITY TLV in the OPEN object. Implementations that wish to interoperate with such early implementations should also send the SR-CAPABILITY-TLV as a top-level TLV in their OPEN object and should interpret receiving this top-level TLV as though the sender had sent a PATH-SETUP-TYPE-CAPABILITY TLV with a PST list of (0, 1) (that is, both RSVP-TE and SR-TE PSTs are supported) with the SR-CAPABILITY-TLV as a sub-TLV. If a PCEP speaker receives an OPEN object in which both the SR-CAPABILITY-TLV and PATH-SETUP-TYPE-CAPABILITY TLV appear as top-level TLVs, then it should ignore the top-level SR-CAPABILITY-TLV and process only the PATH-SETUP-TYPE-CAPABILITY TLV.

Authors' Addresses

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

Email: msiva@cisco.com

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

Email: cfilsfil@cisco.com

Jeff Tantsura
Apstra, Inc.
333 Middlefield Rd #200
Menlo Park, CA 94025
USA

Email: jefftant.ietf@gmail.com

Wim Henderickx
Nokia
Copernicuslaan 50
Antwerp 2018, CA 95134
BELGIUM

Email: wim.henderickx@alcatel-lucent.com

Jon Hardwick
Metaswitch Networks
100 Church Street
Enfield, Middlesex
UK

Email: jonathan.hardwick@metaswitch.com

PCE Working Group R. Pobbathi
Internet-Draft V. KumarS
Intended status: Standards Track D. Dhody, Ed.
Expires: August 27, 2015 Huawei Technologies
J. Hardwick
Metaswitch
February 23, 2015

A YANG Data Model for Path Computation Element Communications Protocol (PCEP)
draft-pkd-pce-pcep-yang-02

Abstract

This document defines a YANG data model for the management of Path Computation Element communications Protocol (PCEP) for communications between a Path Computation Client (PCC) and a Path Computation Element (PCE), or between two PCEs. The data model includes configuration data and state data (status information and counters for the collection of statistics).

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 27, 2015.

Copyright Notice

Copyright (c) 2015 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

The Path Computation Element (PCE) defined in [RFC4655] 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.

PCEP is the communication protocol between a PCC and PCE and is defined in [RFC5440]. PCEP interactions include path computation requests and path computation replies as well as notifications of specific states related to the use of a PCE in the context of

Multiprotocol Label Switching (MPLS) and Generalized MPLS (GMPLS) Traffic Engineering (TE).

This document defines a YANG [RFC6020] data model for the management of PCEP speakers. It is important to establish a common data model for how PCEP speakers are identified, configured, and monitored. The data model includes configuration data and state data (status information and counters for the collection of statistics).

This document contains a specification of the base PCEP YANG module, "ietf-pcep" which provides the basic PCEP [RFC5440] data model.

[Editor’s Note: Further modules augmenting the data model with advanced features maybe handled in a future revision or a separate document.]

2. 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].

3. Terminology and Notation

This document uses the terminology defined in [RFC4655] and [RFC5440]. In particular, it uses the following acronyms.

- o Path Computation Request message (PCReq).
- o Path Computation Reply message (PCRep).
- o Notification message (PCNtf).
- o Error message (PCErr).
- o Request Parameters object (RP).
- o Synchronization Vector object (SVEC).
- o Explicit Route object (ERO).

This document also uses the following terms defined in [RFC7420]:

- o PCEP entity: a local PCEP speaker.
- o PCEP peer: to refer to a remote PCEP speaker.

- PCEP speaker: where it is not necessary to distinguish between local and remote.

3.1. Tree Diagrams

A graphical representation of the complete data tree is presented in Section 5. The meaning of the symbols in these diagrams is as follows and as per [I-D.ietf-netmod-rfc6087bis]:

- Brackets "[" and "]" enclose list keys.
- Curly braces "{" and "}" contain names of optional features that make the corresponding node conditional.
- Abbreviations before data node names: "rw" means configuration (read-write), and "ro" state data (read-only).
- Symbols after data node names: "?" means an optional node and "*" denotes a "list" or "leaf-list".
- Parentheses enclose choice and case nodes, and case nodes are also marked with a colon (":").
- Ellipsis ("...") stands for contents of subtrees that are not shown.

3.2. Prefixes in Data Node Names

In this document, names of data nodes and other data model objects are often used without a prefix, as long as it is clear from the context in which YANG module each name is defined. Otherwise, names are prefixed using the standard prefix associated with the corresponding YANG module, as shown in Table 1.

Prefix	YANG module	Reference
yang	ietf-yang-types	[RFC6991]
inet	ietf-inet-types	[RFC6991]

Table 1: Prefixes and corresponding YANG modules

4. Objectives

This section describes some of the design objectives for the model:

o In case of existing implementations, it needs to map the data model defined in this document to their proprietary native data model. To facilitate such mappings, the data model should be simple.

o The data model should be suitable for new implementations to use as is.

o Mapping to the PCEP MIB Module should be clear.

o The data model should allow for static configurations of peers.

o The data model should include read-only counters in order to gather statistics for sent and received PCEP messages, received messages with errors, and messages that could not be sent due to errors.

o It should be fairly straightforward to augment the base data model for advanced PCE features.

5. The Design of PCEP Data Model

The module, "ietf-pcep", defines the basic components of a PCE speaker.

```
module: ietf-pcep
   +--rw pcep
   |  +--rw entity* [addr]
   |     +--rw addr                           inet:ip-address
   |     +--rw enabled?                       boolean
   |     +--rw role                           pcep-role
   |     +--rw description?                   string
   |     +--rw pce-info
   |     |  +--rw scope
   |     |  |  +--rw intra-area-scope?           boolean
   |     |  |  +--rw intra-area-pref?            uint8
   |     |  |  +--rw inter-area-scope?           boolean
   |     |  |  +--rw inter-area-scope-default?   boolean
   |     |  |  +--rw inter-area-pref?            uint8
   |     |  |  +--rw inter-as-scope?             boolean
   |     |  |  +--rw inter-as-scope-default?     boolean
   |     |  |  +--rw inter-as-pref?              uint8
   |     |  |  +--rw inter-layer-scope?          boolean
   |     |  |  +--rw inter-layer-pref?           uint8
   |     |  +--rw domain
   |     |  |  +--rw domain-type?   pce-domain-type
   |     |  |  +--rw domain?        pce-domain
   |     |  +--rw neigh-domains
```

```
     |      |  |  +--rw domain* [domain-type domain]
     |      |  |     +--rw domain-type    pce-domain-type
     |      |  |     +--rw domain         pce-domain
     |      |  +--rw capability
     |      |     +--rw gmpls?                  boolean
     |      |     +--rw bi-dir?                 boolean
     |      |     +--rw diverse?                boolean
     |      |     +--rw load-balance?           boolean
     |      |     +--rw synchronize?            boolean
     |      |     +--rw objective-function?     boolean
     |      |     +--rw add-path-constraint?    boolean
     |      |     +--rw prioritization?         boolean
     |      |     +--rw multi-request?          boolean
     |      |     +--rw gco?                    boolean
     |      |     +--rw p2mp?                   boolean
     |      +--rw connect-timer?                 uint32
     |      +--rw connect-max-retry?             uint32
     |      +--rw init-backoff-timer?            uint32
     |      +--rw max-backoff-timer?             uint32
     |      +--rw open-wait-timer?               uint32
     |      +--rw keep-wait-timer?               uint32
     |      +--rw keep-alive-timer?              uint32
     |      +--rw dead-timer?                    uint32
     |      +--rw allow-negotiation?             boolean
     |      +--rw max-keep-alive-timer?          uint32
     |      +--rw max-dead-timer?                uint32
     |      +--rw min-keep-alive-timer?          uint32
     |      +--rw min-dead-timer?                uint32
     |      +--rw sync-timer?                    uint32
     |      +--rw request-timer?                 uint32
     |      +--rw max-sessions?                  uint32
     |      +--rw max-unknown-reqs?              uint32
     |      +--rw max-unknown-msgs?              uint32
     |      +--rw pcep-notification-max-rate     uint32
     |      +--rw peers
     |         +--rw peer* [addr]
     |            +--rw addr           inet:ip-address
     |            +--rw description?   string
     |            +--rw pce-info
     |               +--rw scope
     |               |  +--rw intra-area-scope?           boolean
     |               |  +--rw intra-area-pref?            uint8
     |               |  +--rw inter-area-scope?           boolean
     |               |  +--rw inter-area-scope-default?   boolean
     |               |  +--rw inter-area-pref?            uint8
     |               |  +--rw inter-as-scope?             boolean
     |               |  +--rw inter-as-scope-default?     boolean
     |               |  +--rw inter-as-pref?              uint8
```

```
     |              |  +--rw inter-layer-scope?          boolean
     |              |  +--rw inter-layer-pref?           uint8
     |              +--rw domain
     |              |  +--rw domain-type?   pce-domain-type
     |              |  +--rw domain?        pce-domain
     |              +--rw neigh-domains
     |              |  +--rw domain* [domain-type domain]
     |              |     +--rw domain-type    pce-domain-type
     |              |     +--rw domain         pce-domain
     |              +--rw capability
     |                 +--rw gmpls?                 boolean
     |                 +--rw bi-dir?                boolean
     |                 +--rw diverse?               boolean
     |                 +--rw load-balance?          boolean
     |                 +--rw synchronize?           boolean
     |                 +--rw objective-function?    boolean
     |                 +--rw add-path-constraint?   boolean
     |                 +--rw prioritization?        boolean
     |                 +--rw multi-request?         boolean
     |                 +--rw gco?                   boolean
     |                 +--rw p2mp?                  boolean
     +--ro pcep-state
        +--ro entity* [addr]
           +--ro index?                 uint32
           +--ro addr                   inet:ip-address
           +--ro admin-status?          pcep-admin-status
           +--ro oper-status?           pcep-admin-status
           +--ro role?                  pcep-role
           +--ro pce-info
           |  +--ro scope
           |  |  +--ro intra-area-scope?           boolean
           |  |  +--ro intra-area-pref?            uint8
           |  |  +--ro inter-area-scope?           boolean
           |  |  +--ro inter-area-scope-default?   boolean
           |  |  +--ro inter-area-pref?            uint8
           |  |  +--ro inter-as-scope?             boolean
           |  |  +--ro inter-as-scope-default?     boolean
           |  |  +--ro inter-as-pref?              uint8
           |  |  +--ro inter-layer-scope?          boolean
           |  |  +--ro inter-layer-pref?           uint8
           |  +--ro domain
           |  |  +--ro domain-type?   pce-domain-type
           |  |  +--ro domain?        pce-domain
           |  +--ro neigh-domains
           |  |  +--ro domain* [domain-type domain]
           |  |     +--ro domain-type    pce-domain-type
           |  |     +--ro domain         pce-domain
           |  +--ro capability
```

```
           |     +--ro gmpls?                  boolean
           |     +--ro bi-dir?                 boolean
           |     +--ro diverse?                boolean
           |     +--ro load-balance?           boolean
           |     +--ro synchronize?            boolean
           |     +--ro objective-function?     boolean
           |     +--ro add-path-constraint?    boolean
           |     +--ro prioritization?         boolean
           |     +--ro multi-request?          boolean
           |     +--ro gco?                    boolean
           |     +--ro p2mp?                   boolean
           +--ro connect-timer?          uint32
           +--ro connect-max-retry?      uint32
           +--ro init-backoff-timer?     uint32
           +--ro max-backoff-timer?      uint32
           +--ro open-wait-timer?        uint32
           +--ro keep-wait-timer?        uint32
           +--ro keep-alive-timer?       uint32
           +--ro dead-timer?             uint32
           +--ro allow-negotiation?      boolean
           +--ro max-keep-alive-timer?   uint32
           +--ro max-dead-timer?         uint32
           +--ro min-keep-alive-timer?   uint32
           +--ro min-dead-timer?         uint32
           +--ro sync-timer?             uint32
           +--ro request-timer?          uint32
           +--ro max-sessions?           uint32
           +--ro max-unknown-reqs?       uint32
           +--ro max-unknown-msgs?       uint32
           +--ro peers
              +--ro peer* [addr]
                 +--ro addr                         inet:ip-address
                 +--ro role?                        pcep-role
                 +--ro pce-info
                 |  +--ro scope
                 |  |  +--ro intra-area-scope?           boolean
                 |  |  +--ro intra-area-pref?            uint8
                 |  |  +--ro inter-area-scope?           boolean
                 |  |  +--ro inter-area-scope-default?   boolean
                 |  |  +--ro inter-area-pref?            uint8
                 |  |  +--ro inter-as-scope?             boolean
                 |  |  +--ro inter-as-scope-default?     boolean
                 |  |  +--ro inter-as-pref?              uint8
                 |  |  +--ro inter-layer-scope?          boolean
                 |  |  +--ro inter-layer-pref?           uint8
                 |  +--ro domain
                 |  |  +--ro domain-type?   pce-domain-type
                 |  |  +--ro domain?        pce-domain
```

```
                 |  +--ro neigh-domains
                 |  |  +--ro domain* [domain-type domain]
                 |  |     +--ro domain-type    pce-domain-type
                 |  |     +--ro domain         pce-domain
                 |  +--ro capability
                 |     +--ro gmpls?                boolean
                 |     +--ro bi-dir?               boolean
                 |     +--ro diverse?              boolean
                 |     +--ro load-balance?         boolean
                 |     +--ro synchronize?          boolean
                 |     +--ro objective-function?   boolean
                 |     +--ro add-path-constraint?  boolean
                 |     +--ro prioritization?       boolean
                 |     +--ro multi-request?        boolean
                 |     +--ro gco?                  boolean
                 |     +--ro p2mp?                 boolean
                 +--ro discontinuity-time?        yang:timestamp
                 +--ro initiate-session?          boolean
                 +--ro session-exists?            boolean
                 +--ro num-sess-setup-ok?         yang:counter32
                 +--ro num-sess-setup-fail?       yang:counter32
                 +--ro session-up-time?           yang:timestamp
                 +--ro session-fail-time?         yang:timestamp
                 +--ro session-fail-up-time?      yang:timestamp
                 +--ro avg-rsp-time?              uint32
                 +--ro lwm-rsp-time?              uint32
                 +--ro hwm-rsp-time?              uint32
                 +--ro num-pcreq-sent?            yang:counter32
                 +--ro num-pcreq-rcvd?            yang:counter32
                 +--ro num-pcrep-sent?            yang:counter32
                 +--ro num-pcrep-rcvd?            yang:counter32
                 +--ro num-pcerr-sent?            yang:counter32
                 +--ro num-pcerr-rcvd?            yang:counter32
                 +--ro num-pcntf-sent?            yang:counter32
                 +--ro num-pcntf-rcvd?            yang:counter32
                 +--ro num-keepalive-sent?        yang:counter32
                 +--ro num-keepalive-rcvd?        yang:counter32
                 +--ro num-unknown-rcvd?          yang:counter32
                 +--ro num-corrupt-rcvd?          yang:counter32
                 +--ro num-req-sent?              yang:counter32
                 +--ro num-svec-sent?             yang:counter32
                 +--ro num-svec-req-sent?         yang:counter32
                 +--ro num-req-sent-pend-rep?     yang:counter32
                 +--ro num-req-sent-ero-rcvd?     yang:counter32
                 +--ro num-req-sent-nopath-rcvd?  yang:counter32
                 +--ro num-req-sent-cancel-rcvd?  yang:counter32
                 +--ro num-req-sent-error-rcvd?   yang:counter32
                 +--ro num-req-sent-timeout?      yang:counter32
```

```
                 +--ro num-req-sent-cancel-sent?   yang:counter32
                 +--ro num-req-rcvd?               yang:counter32
                 +--ro num-svec-rcvd?              yang:counter32
                 +--ro num-svec-req-rcvd?          yang:counter32
                 +--ro num-req-rcvd-pend-rep?      yang:counter32
                 +--ro num-req-rcvd-ero-sent?      yang:counter32
                 +--ro num-req-rcvd-nopath-sent?   yang:counter32
                 +--ro num-req-rcvd-cancel-sent?   yang:counter32
                 +--ro num-req-rcvd-error-sent?    yang:counter32
                 +--ro num-req-rcvd-cancel-rcvd?   yang:counter32
                 +--ro num-rep-rcvd-unknown?       yang:counter32
                 +--ro num-req-rcvd-unknown?       yang:counter32
                 +--ro num-req-sent-closed?        yang:counter32
                 +--ro num-req-rcvd-closed?        yang:counter32
                 +--ro sessions
                    +--ro session* [initiator]
                       +--ro initiator                   pcep-initiator
                       +--ro state-last-change?          yang:timestamp
                       +--ro state?                      pcep-sess-state
                       +--ro connect-retry?              yang:counter32
                       +--ro local-id?                   uint32
                       +--ro remote-id?                  uint32
                       +--ro keepalive-timer?            uint32
                       +--ro peer-keepalive-timer?       uint32
                       +--ro dead-timer?                 uint32
                       +--ro peer-dead-timer?            uint32
                       +--ro ka-hold-time-rem?           uint32
                       +--ro overloaded?                 boolean
                       +--ro overload-time?              uint32
                       +--ro peer-overloaded?            boolean
                       +--ro peer-overload-time?         uint32
                       +--ro discontinuity-time?         yang:timestamp
                       +--ro avg-rsp-time?               uint32
                       +--ro lwm-rsp-time?               uint32
                       +--ro hwm-rsp-time?               uint32
                       +--ro num-pcreq-sent?             yang:counter32
                       +--ro num-pcreq-rcvd?             yang:counter32
                       +--ro num-pcrep-sent?             yang:counter32
                       +--ro num-pcrep-rcvd?             yang:counter32
                       +--ro num-pcerr-sent?             yang:counter32
                       +--ro num-pcerr-rcvd?             yang:counter32
                       +--ro num-pcntf-sent?             yang:counter32
                       +--ro num-pcntf-rcvd?             yang:counter32
                       +--ro num-keepalive-sent?         yang:counter32
                       +--ro num-keepalive-rcvd?         yang:counter32
                       +--ro num-unknown-rcvd?           yang:counter32
                       +--ro num-corrupt-rcvd?           yang:counter32
                       +--ro num-req-sent?               yang:counter32
```

```
                       +--ro num-svec-sent?              yang:counter32
                       +--ro num-svec-req-sent?          yang:counter32
                       +--ro num-req-sent-pend-rep?      yang:counter32
                       +--ro num-req-sent-ero-rcvd?      yang:counter32
                       +--ro num-req-sent-nopath-rcvd?   yang:counter32
                       +--ro num-req-sent-cancel-rcvd?   yang:counter32
                       +--ro num-req-sent-error-rcvd?    yang:counter32
                       +--ro num-req-sent-timeout?       yang:counter32
                       +--ro num-req-sent-cancel-sent?   yang:counter32
                       +--ro num-req-rcvd?               yang:counter32
                       +--ro num-svec-rcvd?              yang:counter32
                       +--ro num-svec-req-rcvd?          yang:counter32
                       +--ro num-req-rcvd-pend-rep?      yang:counter32
                       +--ro num-req-rcvd-ero-sent?      yang:counter32
                       +--ro num-req-rcvd-nopath-sent?   yang:counter32
                       +--ro num-req-rcvd-cancel-sent?   yang:counter32
                       +--ro num-req-rcvd-error-sent?    yang:counter32
                       +--ro num-req-rcvd-cancel-rcvd?   yang:counter32
                       +--ro num-rep-rcvd-unknown?       yang:counter32
                       +--ro num-req-rcvd-unknown?       yang:counter32
  notifications:
     +---n pcep-session-up
     |  +--ro entity-addr?         leafref
     |  +--ro peer-addr?           leafref
     |  +--ro session-initiator?   leafref
     |  +--ro state-last-change?   yang:timestamp
     |  +--ro state?               pcep-sess-state
     +---n pcep-session-down
     |  +--ro entity-addr?         leafref
     |  +--ro peer-addr?           leafref
     |  +--ro session-initiator?   pcep-initiator
     |  +--ro state-last-change?   yang:timestamp
     |  +--ro state?               pcep-sess-state
     +---n pcep-session-local-overload
     |  +--ro entity-addr?         leafref
     |  +--ro peer-addr?           leafref
     |  +--ro session-initiator?   leafref
     |  +--ro overloaded?          boolean
     |  +--ro overload-time?       uint32
     +---n pcep-session-local-overload-clear
     |  +--ro entity-addr?   leafref
     |  +--ro peer-addr?     leafref
     |  +--ro overloaded?    boolean
     +---n pcep-session-peer-overload
     |  +--ro entity-addr?          leafref
     |  +--ro peer-addr?            leafref
     |  +--ro session-initiator?    leafref
     |  +--ro peer-overloaded?      boolean
```

```
     |  +--ro peer-overload-time?   uint32
     +---n pcep-session-peer-overload-clear
        +--ro entity-addr?       leafref
        +--ro peer-addr?         leafref
        +--ro peer-overloaded?   boolean
```

5.1. The Entity Lists

The PCEP yang module may contain status information for multiple logical local PCEP entities. There are several scenarios in which there may be more than one local PCEP entity, it is listed in [RFC7420].

The data model for PCEP presented in this document uses a flat list of entities. Each entity in the list is identified by its IP address (addr-type, addr). Furthermore, each entity has a mandatory "role" leaf (the local entity PCEP role). The ietf-inet-types [RFC6991] is used.

There is one list for configuration ("/pcep/entity"), and a separate list for the operational state of all entities ("/pcep-state/entity").

The PCEP MIB module [RFC7420] uses a system generated entity index as a primary index to the read only entity table. If the device implements the PCEP MIB, the "index" leaf MUST contain the value of the corresponding pcePcepEntityIndex.

5.2. The Peer Lists

The peer list contains peer(s) that the local PCEP entity knows about. A PCEP speaker is identified by its IP address. If there is a PCEP speaker in the network that uses multiple IP addresses then it looks like multiple distinct peers to the other PCEP speakers in the network.

Since PCEP sessions can be ephemeral, the peer list tracks a peer even when no PCEP session currently exists to that peer. The statistics contained are an aggregate of the statistics for all successive sessions to that peer.

To limit the quantity of information that is stored, an implementation MAY choose to discard this information if and only if no PCEP session exists to the corresponding peer.

The data model for PCEP peer presented in this document uses a flat list of peers. Each peer in the list is identified by its IP address (addr-type, addr).

There is one list for static peer configuration ("/pcep/entity/peers"), and a separate list for the operational state of all peers (i.e. static as well as discovered)("/pcep-state/entity/peers").

5.3. The Session Lists

The session list contains PCEP session that the PCEP entity (PCE or PCC) is currently participating in. The statistics in session are semantically different from those in peer since the former applies to the current session only, whereas the latter is the aggregate for all sessions that have existed to that peer.

Although [RFC5440] forbids more than one active PCEP session between a given pair of PCEP entities at any given time, there is a window during session establishment where two sessions may exist for a given pair, one representing a session initiated by the local PCEP entity and the other representing a session initiated by the peer. If either of these sessions reaches active state first, then the other is discarded.

The data model for PCEP session presented in this document uses a flat list of sessions. Each session in the list is identified by its initiator. This index allows two sessions to exist transiently for a given peer, as discussed above.

There is only one list for the operational state of all sessions ("/pcep-state/entity/peers/peer/sessions/session").

5.4. Notifications

This YANG model defines a list of notifications to inform client of important events detected during the protocol operation. The notifications defined cover the PCEP MIB notifications.

6. Advanced PCE Features

This document contains a specification of the base PCEP YANG module, "ietf-pcep" which provides the basic PCEP [RFC5440] data model.

A means and procedure to handle to following PCE features needs to be decided:

o Capability and Scope

- o Domain information (local/neighbour)
- o Path-Key
- o OF
- o GCO
- o P2MP
- o GMPLS
- o Inter-Layer
- o Stateful PCE

7. PCEP YANG Module

RFC Ed.: In this section, replace all occurrences of 'XXXX' with the actual RFC number and all occurrences of the revision date below with the date of RFC publication (and remove this note).

```
<CODE BEGINS> file "ietf-pcep@2015-02-xx.yang"




module ietf-pcep {
    namespace "urn:ietf:params:xml:ns:yang:ietf-pcep";
    prefix pcep;

    import ietf-inet-types {
        prefix inet;
    }

    import ietf-yang-types {
        prefix yang;
    }

    organization
        "IETF PCE (Path Computation Element) Working Group";

    contact
        "WG Web:   <http://tools.ietf.org/wg/pce/>
         WG List:  <mailto:pce@ietf.org>
```

```
         WG Chair: JP Vasseur
                   <mailto:jpv@cisco.com>

         WG Chair: Julien Meuric
                   <mailto:julien.meuric@orange.com>

         Editor:   Dhruv Dhody
                   <mailto:dhruv.ietf@gmail.com>";

     description
         "The YANG module defines a generic configuration and
          operational model for PCEP common across all of the
          vendor implementations.";


     revision 2015-02-23 {
         description "Initial revision.";
         reference
             "RFC XXXX:  A YANG Data Model for Path Computation
                         Element Communications Protocol
                         (PCEP)";
     }

     /*
      * Identities
      */

     identity pcep {
         description "Identity for the PCEP protocol.";
     }

     /*
      * Typedefs
      */
     typedef pcep-role {
         type enumeration {
             enum unknown {
                 value "0";
                 description
                 "An unknown role";
             }
             enum pcc {
                 value "1";
                 description
                 "The role of a Path Computation Client";
             }
             enum pce {
                 value "2";
```

```
                description
                "The role of Path Computation Element";
            }
            enum pcc-and-pce {
                value "3";
                description
                "The role of both Path Computation Client and
                 Path Computation Element";
            }
        }

        description
            "The role of a PCEP speaker.
             Takes one of the following values
             - unknown(0): the role is not known.
             - pcc(1): the role is of a Path Computation
               Client (PCC).
             - pce(2): the role is of a Path Computation
               Server (PCE).
             - pccAndPce(3): the role is of both a PCC and
               a PCE.";

    }

    typedef pcep-admin-status {
        type enumeration {
                enum admin-status-up {
                value "1";
                description
                "Admin Status is Up";
            }
            enum admin-status-down {
                value "2";
                description
                "Admin Status is Down";
            }
        }

        description
        "The Admin Status of the PCEP entity.
         Takes one of the following values
             - admin-status-up(1): Admin Status is Up.
             - admin-status-down(2): Admin Status is Down";
    }

    typedef pcep-oper-status {
        type enumeration {
            enum oper-status-up {
```

```
                value "1";
                description
                "The PCEP entity is active";
            }
            enum oper-status-down {
                value "2";
                description
                "The PCEP entity is inactive";
            }
            enum oper-status-going-up {
                value "3";
                description
                "The PCEP entity is activating";
            }
            enum oper-status-going-down {
                value "4";
                description
                "The PCEP entity is deactivating";
            }
            enum oper-status-failed {
                value "5";
                description
                "The PCEP entity has failed and will recover
                 when possible.";
            }
            enum oper-status-failed-perm {
                value "6";
                description
                "The PCEP entity has failed and will not recover
                 without operator intervention";
            }
        }
        description
        "The operational status of the PCEP entity.
         Takes one of the following values
             - oper-status-up(1): Active
             - oper-status-down(2): Inactive
             - oper-status-going-up(3): Activating
             - oper-status-going-down(4): Deactivating
             - oper-status-failed(5): Failed
             - oper-status-failed-perm(6): Failed Permanantly";
    }

    typedef pcep-initiator {
        type enumeration {
            enum local {
                value "1";
                description
```

```
                    "The local PCEP entity initiated the session";
                }

                enum remote {
                    value "2";
                    description
                    "The remote PCEP peer initiated the session";
                }
            }
            description
            "The initiator of the session, that is, whether the TCP
             connection was initiated by the local PCEP entity or
             the remote peer.
             Takes one of the following values
                 - local(1): Initiated locally
                 - remote(2): Initiated remotely";
        }

        typedef pcep-sess-state {
            type enumeration {
                enum tcp-pending {
                    value "1";
                    description
                        "The tcp-pending state of PCEP session.";
                }

                enum open-wait {
                    value "2";
                    description
                        "The open-wait state of PCEP session.";
                }

                enum keep-wait {
                    value "3";
                    description
                        "The keep-wait state of PCEP session.";
                }

                enum session-up {
                    value "4";
                    description
                        "The session-up state of PCEP session.";
                }
            }
            description
                "The current state of the session.

                 The set of possible states excludes the idle state
```

```
             since entries do not exist in the idle state.
             Takes one of the following values
                - tcp-pending(1): PCEP TCP Pending state
                - open-wait(2): PCEP Open Wait state
                - keep-wait(3): PCEP Keep Wait state
                - session-up(4): PCEP Session Up state";
    }

    typedef pce-domain-type {
        type enumeration {
            enum ospf-area {
                value "1";
                description
                    "The OSPF area.";
            }
            enum isis-area {
                value "2";
                description
                    "The IS-IS area.";
            }
            enum as {
                value "3";
                description
                    "The Autonomous System (AS).";
            }
        }
        description
            "The PCE Domain Type";
    }

    typedef domain-ospf-area {
        type union {
            type uint32;
            type yang:dotted-quad;
       }
       description
            "OSPF Area ID.";
    }

    typedef domain-isis-area {
        type string {
            pattern '[0-9A-Fa-f]{2}\.([0-9A-Fa-f]{4}\.){0,3}';
        }
        description
            "IS-IS Area ID.";
    }

    typedef domain-as {
```

```
        type uint32;
        description
            "Autonomous System number.";

    }

    typedef pce-domain {
        type union {
            type domain-ospf-area;
            type domain-isis-area;
            type domain-as;
        }
        description
            "The PCE Domain";
    }



    /*
     * Features - TBD
     */

    /*
     * Groupings
     */



    grouping pcep-entity-info{
        description
            "This grouping defines the attributes for PCEP entity.";
        leaf connect-timer {
            type uint32 {
                range "1..65535";
            }
            units "seconds";
            default 60;
            description
                "The time in seconds that the PCEP entity will wait
                 to establish a TCP connection with a peer.  If a
                 TCP connection is not established within this time
                 then PCEP aborts the session setup attempt.";
            reference
                "RFC 5440: Path Computation Element (PCE)
                           Communication Protocol (PCEP)";
        }

        leaf connect-max-retry {
```

```
            type uint32;
            default 5;
            description
                "The maximum number of times the system tries to
                 establish a TCP connection to a peer before the
                 session with the peer transitions to the idle
                 state.";
            reference
                "RFC 5440: Path Computation Element (PCE)
                           Communication Protocol (PCEP)";
        }

        leaf init-backoff-timer {
            type uint32 {
                range "1..65535";
            }
            units "seconds";
            description
               "The initial back-off time in seconds for retrying
                a failed session setup attempt to a peer.

                The back-off time increases for each failed
                session setup attempt, until a maximum back-off
                time is reached.  The maximum back-off time is
                max-backoff-timer.";
        }

        leaf max-backoff-timer {
            type uint32;
            units "seconds";
            description
               "The maximum back-off time in seconds for retrying
                a failed session setup attempt to a peer.

                The back-off time increases for each failed session
                setup attempt, until this maximum value is reached.
                Session setup attempts then repeat periodically
                without any further increase in back-off time.";
        }

        leaf open-wait-timer {
            type uint32 {
                range "1..65535";
            }
            units "seconds";
            default 60;
            description
               "The time in seconds that the PCEP entity will wait
```

```
                to receive an Open message from a peer after the
                TCP connection has come up.

                If no Open message is received within this time then
                PCEP terminates the TCP connection and deletes the
                associated sessions.";
            reference
                "RFC 5440: Path Computation Element (PCE)
                           Communication Protocol (PCEP)";
        }

        leaf keep-wait-timer {
            type uint32 {
                range "1..65535";
            }
            units "seconds";
            default 60;
            description
                "The time in seconds that the PCEP entity will wait
                 to receive a Keepalive or PCErr message from a peer
                 during session initialization after receiving an
                 Open message.  If no Keepalive or PCErr message is
                 received within this time then PCEP terminates the
                 TCP connection and deletes the associated
                 sessions.";
            reference
                "RFC 5440: Path Computation Element (PCE)
                           Communication Protocol (PCEP)";
        }

        leaf keep-alive-timer {
            type uint32 {
                range "0..255";
            }
            units "seconds";
            default 30;
            description
                "The keep alive transmission timer that this PCEP
                 entity will propose in the initial OPEN message of
                 each session it is involved in.  This is the
                 maximum time between two consecutive messages sent
                 to a peer. Zero means that the PCEP entity prefers
                 not to send Keepalives at all.

                 Note that the actual Keepalive transmission
                 intervals, in either direction of an active PCEP
                 session, are determined by negotiation between the
                 peers as specified by RFC 5440, and so may differ
```

```
                 from this configured value.";
            reference
                "RFC 5440: Path Computation Element (PCE)
                           Communication Protocol (PCEP)";
        }

        leaf dead-timer {
            type uint32 {
                range "0..255";
            }
            units "seconds";
            must ". >= ../keep-alive-timer" {
                error-message "The dead timer must be "
                       + "larger than the keep alive timer";
                description
                    "This value MUST be greater than
                     keep-alive-timer.";

            }
            default 120;
            description
                "The dead timer that this PCEP entity will propose
                 in the initial OPEN message of each session it is
                 involved in. This is the time after which a peer
                 should declare a session down if it does not
                 receive any PCEP messages. Zero suggests that the
                 peer does not run a dead timer at all." ;
            reference
                "RFC 5440: Path Computation Element (PCE)
                           Communication Protocol (PCEP)";
        }

        leaf allow-negotiation{
            type boolean;
            description
                "Whether the PCEP entity will permit negotiation of
                 session parameters.";
        }

        leaf max-keep-alive-timer{
            type uint32 {
                range "0..255";
            }
            units "seconds";
            description
                "In PCEP session parameter negotiation in seconds,
                 the maximum value that this PCEP entity will
                 accept from a peer for the interval between
```

```
                 Keepalive transmissions. Zero means that the PCEP
                 entity will allow no Keepalive transmission at
                 all." ;
        }

        leaf max-dead-timer{
            type uint32 {
                range "0..255";
            }
            units "seconds";
            description
                "In PCEP session parameter negotiation in seconds,
                 the maximum value that this PCEP entity will accept
                 from a peer for the Dead timer.  Zero means that
                 the PCEP entity will allow not running a Dead
                 timer.";
        }

        leaf min-keep-alive-timer{
            type uint32 {
                range "0..255";
            }
            units "seconds";
            description
                "In PCEP session parameter negotiation in seconds,
                 the minimum value that this PCEP entity will
                 accept for the interval between Keepalive
                 transmissions. Zero means that the PCEP entity
                 insists on no Keepalive transmission at all.";
        }

        leaf min-dead-timer{
            type uint32 {
                range "0..255";
            }
            units "seconds";
            description
                 "In PCEP session parameter negotiationin in
                  seconds, the minimum value that this PCEP entity
                  will accept for the Dead timer.  Zero means that
                  the PCEP entity insists on not running a Dead
                  timer.";
        }

        leaf sync-timer{
            type uint32 {
                range "0..65535";
            }
```

```
            units "seconds";
            default 60;
            description
                "The value of SyncTimer in seconds is used in the
                 case of synchronized path computation request
                 using the SVEC object. Consider the case where a
                 PCReq message is received by a PCE that contains
                 the SVEC object referring to M synchronized path
                 computation requests.  If after the expiration of
                 the SyncTimer all the M path computation requests
                 have not been, received a protocol error is
                 triggered and the PCE MUST cancel the whole set
                 of path computation  requests.

                 The aim of the SyncTimer is to avoid the storage
                 of unused synchronized requests should one of
                 them get lost for some reasons (for example, a
                 misbehaving PCC).

                 Zero means that the PCEP entity does not use the
                 SyncTimer.";
            reference
                "RFC 5440: Path Computation Element (PCE)
                           Communication Protocol (PCEP)";
        }

        leaf request-timer{
            type uint32 {
                range "1..65535";
            }
            units "seconds";
            description
                "The maximum time that the PCEP entity will wait
                 for a response to a PCReq message.";
        }

        leaf max-sessions{
            type uint32;
            description
               "Maximum number of sessions involving this PCEP
                entity that can exist at any time.";
        }

        leaf max-unknown-reqs{
            type uint32;
            default 5;
            description
              "The maximum number of unrecognized requests and
```

```
               replies that any session on this PCEP entity is
               willing to accept per minute before terminating
               the session.

               A PCRep message contains an unrecognized reply
               if it contains an RP object whose request ID
               does not correspond to any in-progress request
               sent by this PCEP entity.

               A PCReq message contains an unrecognized request
               if it contains an RP object whose request ID is
               zero.";
            reference
                "RFC 5440: Path Computation Element (PCE)
                           Communication Protocol (PCEP)";
        }

        leaf max-unknown-msgs{
            type uint32;
            default 5;
            description
             "The maximum number of unknown messages that any
              session on this PCEP entity is willing to accept
              per minute before terminating the session.";
            reference
                "RFC 5440: Path Computation Element (PCE)
                           Communication Protocol (PCEP)";
        }
    }//pcep-entity-info

    grouping  pce-scope{
        description
            "This grouping defines PCE path computation scope
             information which maybe relevant to PCE selection.
             This information corresponds to PCE auto-discovery
             information.";
        reference
            "RFC 5088: OSPF Protocol Extensions for Path
                       Computation Element (PCE)
                       Discovery
             RFC 5089: IS-IS Protocol Extensions for Path
                       Computation Element (PCE)
                       Discovery";
        leaf intra-area-scope{
            type boolean;
            default true;
            description
                "PCE can compute intra-area paths.";
```

```
        }
        leaf intra-area-pref{
            type uint8{
                range "0..7";
            }
            description
              "The PCE’s preference for intra-area TE LSP
              computation.";
        }
        leaf inter-area-scope{
            type boolean;
            default false;
            description
                "PCE can compute inter-area paths.";
        }
        leaf inter-area-scope-default{
            type boolean;
            default false;
            description
                "PCE can act as a default PCE for inter-area
                 path computation.";
        }
        leaf inter-area-pref{
            type uint8{
                range "0..7";
            }
            description
              "The PCE’s preference for inter-area TE LSP
              computation.";
        }
        leaf inter-as-scope{
            type boolean;
            default false;
            description
                "PCE can compute inter-AS paths.";
        }
        leaf inter-as-scope-default{
            type boolean;
            default false;
            description
                "PCE can act as a default PCE for inter-AS
                 path computation.";
        }
        leaf inter-as-pref{
            type uint8{
                range "0..7";
            }
            description
```

```
              "The PCE's preference for inter-AS TE LSP
              computation.";
        }
        leaf inter-layer-scope{
            type boolean;
            default false;
            description
                "PCE can compute inter-layer paths.";
        }
        leaf inter-layer-pref{
            type uint8{
                range "0..7";
            }
            description
              "The PCE's preference for inter-layer TE LSP
              computation.";
        }
    }//pce-scope

    grouping  pce-domain{
        description
            "This grouping specifies a PCE-Domain where the
             PCE has topology visibility and through which
             the PCE can compute paths.";
        leaf domain-type{
            type pce-domain-type;
            description
              "The PCE domain type.";
        }
        leaf domain{
            type pce-domain;
            description
              "The PCE domain.";
        }
    }//pce-domain

    grouping  pce-capability{
        description
            "This grouping specifies a PCE-capability
             information which maybe relevant to PCE selection.
             This information corresponds to PCE auto-discovery
             information.";
        reference
            "RFC 5088: OSPF Protocol Extensions for Path
                       Computation Element (PCE)
                       Discovery
             RFC 5089: IS-IS Protocol Extensions for Path
                       Computation Element (PCE)
```

```
                     Discovery";
        leaf gmpls{
            type boolean;
            description
              "Path computation with GMPLS link
               constraints.";
        }
        leaf bi-dir{
            type boolean;
            description
              "Bidirectional path computation.";
        }
        leaf diverse{
            type boolean;
            description
              "Diverse path computation.";
        }
        leaf load-balance{
            type boolean;
            description
              "Load-balanced path computation.";
        }
        leaf synchronize{
            type boolean;
            description
              "Synchronized paths computation.";
        }
        leaf objective-function{
            type boolean;
            description
              "Support for multiple objective functions.";
        }
        leaf add-path-constraint{
            type boolean;
            description
              "Support for additive path constraints (max
               hop count, etc.).";
        }
        leaf prioritization{
            type boolean;
            description
              "Support for request prioritization.";
        }
        leaf multi-request{
            type boolean;
            description
              "Support for multiple requests per message.";
        }
```

```
        leaf gco{
            type boolean;
            description
              "Support for Global Concurrent Optimization
               (GCO).";
        }
        leaf p2mp{
            type boolean;
            description
              "Support for P2MP path computation.";
        }
    }//pce-capability

    grouping  pce-info{
        description
            "This grouping specifies all PCE information
             which maybe relevant to the PCE selection.
             This information corresponds to PCE auto-discovery
             information.";
        container scope{
            description
                "The path computation scope";
            uses pce-scope;
        }
        container domain{
            description
                "The PCE domain";
            uses pce-domain{
                description
                    "The PCE local domain.";
            }
        }
        container neigh-domains{
            description
                "The list of neighbour PCE-Domain
                 toward which a PCE can compute
                 paths";
            list domain{
                key "domain-type domain";

                description
                    "The neighbour domain.";
                uses pce-domain{
                    description
                        "The PCE neighbour domain.";
                }
            }
        }
```

```
        container capability{
            description
                "The PCE capability";
            uses pce-capability{
                description
                    "The PCE entity supported
                    capabilities.";
            }
        }
    }//pce-info

    grouping  pcep-stats{
        description
            "This grouping defines statistics for PCEP. It is used
             for both peer and current session.";
        leaf avg-rsp-time{
            type uint32;
            units "milliseconds";
            must "(/pcep-state/entity/peers/peer/role != 'pcc'" +
                 " or " +
                 "(/pcep-state/entity/peers/peer/role = 'pcc'" +
                 " and avg-rsp-time = 0))" {
                error-message
                    "Invalid average response time";
                description
                    "If role is pcc then this leaf is meaningless
                     and is set to zero.";
            }
            description
              "The average response time.

               If an average response time has not been
               calculated then this leaf has the value zero.";
        }

        leaf lwm-rsp-time{
            type uint32;
            units "milliseconds";
            must "(/pcep-state/entity/peers/peer/role != 'pcc'" +
                 " or " +
                 "(/pcep-state/entity/peers/peer/role = 'pcc'" +
                 " and lwm-rsp-time = 0))" {
                error-message
                    "Invalid smallest (low-water mark)
                     response time";
                description
                    "If role is pcc then this leaf is meaningless
                     and is set to zero.";
```

```
            }
            description
             "The smallest (low-water mark) response time seen.

              If no responses have been received then this
              leaf has the value zero.";
        }

        leaf hwm-rsp-time{
            type uint32;
            units "milliseconds";
            must "(/pcep-state/entity/peers/peer/role != 'pcc'" +
                 " or " +
                 "(/pcep-state/entity/peers/peer/role = 'pcc'" +
                 " and hwm-rsp-time = 0))" {
                error-message
                    "Invalid greatest (high-water mark)
                     response time seen";
                description
                    "If role is pcc then this field is
                     meaningless and is set to zero.";
            }
            description
             "The greatest (high-water mark) response time seen.

              If no responses have been received then this object
              has the value zero.";
        }

        leaf num-pcreq-sent{
            type yang:counter32;
            description
             "The number of PCReq messages sent.";
        }

        leaf num-pcreq-rcvd{
            type yang:counter32;
            description
              "The number of PCReq messages received.";
        }

        leaf num-pcrep-sent{
            type yang:counter32;
            description
              "The number of PCRep messages sent.";
        }

        leaf num-pcrep-rcvd{
```

```
            type yang:counter32;
            description
              "The number of PCRep messages received.";
        }

        leaf num-pcerr-sent{
            type yang:counter32;
            description
              "The number of PCErr messages sent.";
        }

        leaf num-pcerr-rcvd{
            type yang:counter32;
            description
              "The number of PCErr messages received.";
        }

        leaf num-pcntf-sent{
            type yang:counter32;
            description
              "The number of PCNtf messages sent.";
        }

        leaf num-pcntf-rcvd{
            type yang:counter32;
            description
              "The number of PCNtf messages received.";
        }

        leaf num-keepalive-sent{
            type yang:counter32;
            description
              "The number of Keepalive messages sent.";
        }

        leaf num-keepalive-rcvd{
            type yang:counter32;
            description
              "The number of Keepalive messages received.";
        }

        leaf num-unknown-rcvd{
            type yang:counter32;
            description
              "The number of unknown messages received.";
        }

        leaf num-corrupt-rcvd{
```

```
            type yang:counter32;
            description
              "The number of corrupted PCEP message received.";
        }

        leaf num-req-sent{
            type yang:counter32;
            description
              "The number of requests sent.  A request corresponds
               1:1 with an RP object in a PCReq message. This might
               be greater than num-pcreq-sent because multiple
               requests can be batched into a single PCReq
               message.";
        }

        leaf num-svec-sent{
            type yang:counter32;
            description
              "The number of SVEC objects sent in PCReq messages.
               An SVEC object represents a set of synchronized
               requests.";
        }

        leaf num-svec-req-sent{
            type yang:counter32;
            description
              "The number of requests sent that appeared in one
               or more SVEC objects.";
        }

        leaf num-req-sent-pend-rep{
            type yang:counter32;
            description
              "The number of requests that have been sent for
               which a response is still pending.";
        }

        leaf num-req-sent-ero-rcvd{
            type yang:counter32;
            description
               "The number of requests that have been sent for
                which a response with an ERO object was received.
                Such responses indicate that a path was
                successfully computed by the peer.";
        }

        leaf num-req-sent-nopath-rcvd{
            type yang:counter32;
```

```
            description
              "The number of requests that have been sent for
               which a response with a NO-PATH object was
               received. Such responses indicate that the peer
               could not  find a path to satisfy the
               request.";
        }

        leaf num-req-sent-cancel-rcvd{
            type yang:counter32;
            description
              "The number of requests that were cancelled with
               a PCNtf message.

               This might be different than num-pcntf-rcvd because
               not all PCNtf messages are used to cancel requests,
               and a single PCNtf message can cancel multiple
               requests.";
        }

        leaf num-req-sent-error-rcvd{
            type yang:counter32;
            description
             "The number of requests that were rejected with a
              PCErr message.

              This might be different than num-pcerr-rcvd because
              not all PCErr messages are used to reject requests,
              and a single PCErr message can reject multiple
              requests.";
        }

        leaf num-req-sent-timeout{
            type yang:counter32;
            description
             "The number of requests that have been sent to a peer
             and have been abandoned because the peer has taken too
             long to respond to them.";
        }

        leaf num-req-sent-cancel-sent{
            type yang:counter32;
            description
             "The number of requests that were sent to the peer and
              explicitly cancelled by the local PCEP entity sending
              a PCNtf.";
        }
```

```
        leaf num-req-rcvd{
            type yang:counter32;
            description
             "The number of requests received.  A request
              corresponds 1:1 with an RP object in a PCReq
              message.

              This might be greater than num-pcreq-rcvd because
              multiple requests can be batched into a single
              PCReq message.";
        }

        leaf num-svec-rcvd{
            type yang:counter32;
            description
             "The number of SVEC objects received in PCReq
              messages. An SVEC object represents a set of
              synchronized requests.";
        }

        leaf num-svec-req-rcvd{
            type yang:counter32;
            description
              "The number of requests received that appeared in one
               or more SVEC objects.";
        }

        leaf num-req-rcvd-pend-rep{
            type yang:counter32;
            description
             "The number of requests that have been received for
              which a response is still pending.";
        }

        leaf num-req-rcvd-ero-sent{
            type yang:counter32;
            description
             "The number of requests that have been received for
              which a response with an ERO object was sent.  Such
              responses indicate that a path was successfully
              computed by the local PCEP entity.";
        }

        leaf num-req-rcvd-nopath-sent{
            type yang:counter32;
            description
             "The number of requests that have been received for
              which a response with a NO-PATH object was sent. Such
```

```
              responses indicate that the local PCEP entity could
              not find a path to satisfy the request.";
        }

        leaf num-req-rcvd-cancel-sent{
            type yang:counter32;
            description
             "The number of requests received that were cancelled
              by the local PCEP entity sending a PCNtf message.

              This might be different than num-pcntf-sent because
              not all PCNtf messages are used to cancel requests,
              and a single PCNtf message can cancel multiple
              requests.";
        }

        leaf num-req-rcvd-error-sent{
            type yang:counter32;
            description
             "The number of requests received that were cancelled
              by the local PCEP entity sending a PCErr message.

              This might be different than num-pcerr-sent because
              not all PCErr messages are used to cancel requests,
              and a single PCErr message can cancel multiple
              requests.";
        }

        leaf  num-req-rcvd-cancel-rcvd{
            type yang:counter32;
            description
             "The number of requests that were received from the
              peer and explicitly cancelled by the peer sending
              a PCNtf.";
        }

        leaf  num-rep-rcvd-unknown{
            type yang:counter32;
            description
              "The number of responses to unknown requests
               received. A response to an unknown request is a
               response whose RP object does not contain the
               request ID of any request that is currently
               outstanding on the session.";
        }

        leaf  num-req-rcvd-unknown{
            type yang:counter32;
```

```
            description
              "The number of unknown requests that have been
               received. An unknown request is a request
               whose RP object contains a request ID of
               zero.";
        }

    }//pcep-stats

    grouping notification-instance-hdr {
        description
            "This group describes common instance specific data
             for notifications.";

        leaf entity-addr {
            type leafref {
                path "/pcep-state/entity/addr";
            }
            description
                "Reference to local entity address";
        }

        leaf peer-addr {
            type leafref {
                path "/pcep-state/entity/peers/peer/addr";
            }
            description
                "Reference to peer address";
        }

    }// notification-instance-hdr

    grouping notification-session-hdr {
        description
        "This group describes common session instance specific
         data for notifications.";

        leaf session-initiator {
            type leafref {
            path "/pcep-state/entity/peers/peer/sessions/" +
                 "session/initiator";
            }
            description
                "Reference to pcep session initiator leaf";
        }
    }// notification-session-hdr

    /*
```

```
     * Configuration data nodes
     */
    container pcep{

        description
            "Parameters for list of configured PCEP entities
             on the device.";

        list entity{
            key "addr";

            description
                "The configured PCEP entity on the device.";

            leaf addr {
                type inet:ip-address;
                description
                    "The local Internet address of this PCEP
                    entity.

                    If operating as a PCE server, the PCEP
                    entity listens on this address.

                    If operating as a PCC, the PCEP entity
                    binds outgoing TCP connections to this
                    address.

                    It is possible for the PCEP entity to
                    operate both as a PCC and a PCE Server, in
                    which case it uses this address both to
                    listen for incoming TCP connections and to
                    bind outgoing TCP connections.";
            }

            leaf enabled {
                type boolean;
                default true;
                description
                    "The administrative status of this PCEP
                     Entity.";
            }

            leaf role {
                type pcep-role;
                mandatory true;
                description
                    "The role that this entity can play.
                     Takes one of the following values.
```

```
                  - unknown(0): this PCEP Entity role is not
                    known.
                  - pcc(1): this PCEP Entity is a PCC.
                  - pce(2): this PCEP Entity is a PCE.
                  - pcc-and-pce(3): this PCEP Entity is both
                    a PCC and a PCE.";
         }

         leaf description {
             type string;
             description
                 "Description of the PCEP entity configured
                  by the user";
         }

         container pce-info {
             must "((../role == 'pce')" +
                  " or " +
                  "(../role == 'pcc-and-pce'))"
             {
                 error-message "The PCEP entity must be PCE";
                 description
                     "When PCEP entity is PCE";
             }
             uses pce-info {
                 description
                     "Local PCE information";
             }
             description
                 "The Local PCE Entity PCE information";
         }

         uses pcep-entity-info {
             description
                 "The configuration related to the PCEP
                  entity.";
         }



         leaf pcep-notification-max-rate {
             type uint32;
             mandatory true;
             description
                 "This variable indicates the maximum number of
                  notifications issued per second. If events occur
                  more rapidly, the implementation may simply fail
                  to emit these notifications during that period,
```

```
                     or may queue them until an appropriate time. A
                     value of 0 means no notifications are emitted
                     and all should be discarded (that is, not
                     queued).";
            }

            container peers{
                description
                        "The list of configured peers for the
                         entity";

                list peer{
                    key "addr";

                    description
                        "The peer configured for the entity.";

                    leaf addr {
                        type inet:ip-address;
                        description
                            "The local Internet address of this PCEP peer.";
                    }

                     leaf description {
                        type string;
                        description
                            "Description of the PCEP peer
                             configured by the user";
                    }
                    container pce-info {
                        uses pce-info {
                            description
                                "PCE Peer information";
                        }
                    description
                        "The Local PCE Entity PCE information";
                    }
                }//peer
            }//peers
        }//entity
    }//pcep

    /*
     * Operational data nodes
     */

    container pcep-state{
        config false;
```

```
        description
            "The list of operational PCEP entities on the
             device.";

        list entity{
            key "addr";
            unique "index";

            description
                "The operational PCEP entity on the device.";

            leaf index{
                type uint32;
                description
                    "The index of the operational PECP
                     entity";
            }

            leaf addr {
                type inet:ip-address;
                description
                    "The local Internet address of this PCEP
                    entity.

                    If operating as a PCE server, the PCEP
                    entity listens on this address.

                    If operating as a PCC, the PCEP entity
                    binds outgoing TCP connections to this
                    address.

                    It is possible for the PCEP entity to
                    operate both as a PCC and a PCE Server, in
                    which case it uses this address both to
                    listen for incoming TCP connections and to
                    bind outgoing TCP connections.";
            }

            leaf admin-status {
                type pcep-admin-status;
                description
                    "The administrative status of this PCEP Entity.
                     This is the desired operational status as
                     currently set by an operator or by default in
                     the implementation.  The value of enabled
                     represents the current status of an attempt
                     to reach this desired status.";
            }
```

```
            leaf oper-status {
                type pcep-admin-status;
                description
                   "The operational status of the PCEP entity.
                    Takes one of the following values.
                    - oper-status-up(1): the PCEP entity is
                      active.
                    - oper-status-down(2): the PCEP entity is
                      inactive.
                    - oper-status-going-up(3): the PCEP entity is
                      activating.
                    - oper-status-going-down(4): the PCEP entity is
                      deactivating.
                    - oper-status-failed(5): the PCEP entity has
                      failed and will recover when possible.
                    - oper-status-failed-perm(6): the PCEP entity
                      has failed and will not recover without
                      operator intervention.";
            }

            leaf role {
                type pcep-role;
                description
                    "The role that this entity can play.
                     Takes one of the following values.
                     - unknown(0): this PCEP entity role is
                       not known.
                     - pcc(1): this PCEP entity is a PCC.
                     - pce(2): this PCEP entity is a PCE.
                     - pcc-and-pce(3): this PCEP entity is
                       both a PCC and a PCE.";
            }

            container pce-info {
                when "((../role == 'pce')" +
                     " or " +
                     "(../role == 'pcc-and-pce'))"
                {
                    description
                        "When PCEP entity is PCE";
                }
                uses pce-info {
                    description
                        "Local PCE information";
                }
                description
                    "The Local PCE Entity PCE information";
            }
```

```
            uses pcep-entity-info{
                description
                    "The operational information related to the
                     PCEP entity.";
            }



            container peers{
                description
                        "The list of peers for the entity";

                list peer{
                    key "addr";

                    description
                        "The peer for the entity.";

                    leaf addr {
                        type inet:ip-address;
                        description
                            "The local Internet address of this PCEP peer.";
                    }

                    leaf role {
                        type pcep-role;
                        description
                            "The role of the PCEP Peer.
                             Takes one of the following values.
                             - unknown(0): this PCEP peer role
                               is not known.
                             - pcc(1): this PCEP peer is a PCC.
                             - pce(2): this PCEP peer is a PCE.
                             - pcc-and-pce(3): this PCEP peer
                               is both a PCC and a PCE.";
                    }

                    container pce-info {
                        uses pce-info {
                            description
                                "PCE Peer information";
                        }
                    description
                        "The Local PCE Entity PCE information";
                    }

                    leaf discontinuity-time {
                        type yang:timestamp;
```

```
                    description
                        "The timestamp of the time when the
                         information and statistics were
                         last reset.";
                }

                leaf initiate-session {
                    type boolean;
                    description
                        "Indicates whether the local PCEP
                         entity initiates sessions to this peer,
                         or waits for the peer to initiate a
                         session.";
                }

                leaf session-exists{
                    type boolean;
                    description
                        "Indicates whether a session with
                         this peer currently exists.";
                }

                leaf num-sess-setup-ok{
                    type yang:counter32;
                    description
                        "The number of PCEP sessions successfully
                         successfully established with the peer,
                         including any current session.  This
                         counter is incremented each time a
                         session with this peer is successfully
                         established.";
                }

                leaf num-sess-setup-fail{
                    type yang:counter32;
                    description
                       "The number of PCEP sessions with the peer
                        that have been attempted but failed
                        before being fully established. This
                        counter is incremented each time a
                        session retry to this peer fails.";
                }

                leaf session-up-time{
                    type yang:timestamp;
                    must "(../num-sess-setup-ok != 0  or " +
                        "(../num-sess-setup-ok = 0  and "  +
                        "session-up-time = 0))" {
```

```
                        error-message
                            "Invalid Session Up timestamp";
                        description
                            "If num-sess-setup-ok is zero,
                             then this leaf contains zero.";
                    }
                description
                   "The timestamp value of the last time a
                    session with this peer was successfully
                    established.";
            }

            leaf session-fail-time{
                type yang:timestamp;
                must "(../num-sess-setup-fail != 0  or " +
                    "(../num-sess-setup-fail = 0  and "  +
                    "session-fail-time = 0))" {
                        error-message
                            "Invalid Session Fail timestamp";
                        description
                            "If num-sess-setup-fail is zero,
                             then this leaf contains zero.";
                    }
                description
                  "The timestamp value of the last time a
                   session with this peer failed to be
                   established.";
            }

            leaf session-fail-up-time{
                type yang:timestamp;
                must "(../num-sess-setup-ok != 0  or "     +
                    "(../num-sess-setup-ok = 0  and "     +
                    "session-fail-up-time = 0))" {
                        error-message
                            "Invalid Session Fail from
                             Up timestamp";
                        description
                            "If num-sess-setup-ok is zero,
                             then this leaf contains zero.";
                    }
                description
                  "The timestamp value of the last time a
                   session with this peer failed from
                   active.";
            }

            uses pcep-stats{
```

```
                    description
                        "Since PCEP sessions can be ephemeral,
                         the peer statistics tracks a peer even
                         when no PCEP session currently exists
                         to that peer. The statistics contained
                         are an aggregate of the statistics for
                         all successive sessions to that peer.";
                }

                leaf num-req-sent-closed{
                    type yang:counter32;
                    description
                        "The number of requests that were sent
                         to the peer and implicitly cancelled
                         when the session they were sent over
                         was closed.";
                }

                leaf  num-req-rcvd-closed{
                    type yang:counter32;
                    description
                        "The number of requests that were
                         received from the peer and implicitly
                         cancelled when the session they were
                         received over was closed.";
                }

                container sessions {
                    description
                        "This entry represents a single PCEP
                         session in which the local PCEP entity
                         participates.

                         This entry exists only if the
                         corresponding PCEP session has been
                         initialized by some event, such as
                         manual user configuration, auto-
                         discovery of a peer, or an incoming
                         TCP connection.";

                    list session {
                        key "initiator";

                        description
                            "The list of sessions, note that
                             for a time being two sessions
                             may exist for a peer";
```

```
                            leaf initiator {
                                type pcep-initiator;
                                description
                                    "The initiator of the session,
                                     that is, whether the TCP
                                     connection was initiated by
                                     the local PCEP entity or the
                                     peer.

                                     There is a window during
                                     session initialization where
                                     two sessions can exist between
                                     a pair of PCEP speakers, each
                                     initiated by one of the
                                     speakers. One of these
                                     sessions is always discarded
                                     before it leaves OpenWait state.
                                     However, before it is discarded,
                                     two sessions to the given peer
                                     appear transiently in this MIB
                                     module. The sessions are
                                     distinguished by who initiated
                                     them, and so this field is the
                                     key.";
                            }

                            leaf state-last-change {
                                type yang:timestamp;
                                description
                                    "The timestamp value at the
                                     time this session entered its
                                     current state as denoted by
                                     the state leaf.";
                            }

                            leaf state {
                                type pcep-sess-state;
                                description
                                    "The current state of the
                                     session.

                                     The set of possible states
                                     excludes the idle state since
                                     entries do not exist in the
                                     idle state.";
                            }

                            leaf connect-retry {
```

```
                        type yang:counter32;
                        description
                             "The number of times that the
                              local PCEP entity has
                              attempted to establish a TCP
                              connection for this session
                              without success. The PCEP
                              entity gives up when this
                              reaches connect-max-retry.";
                    }

                    leaf local-id {
                        type uint32 {
                          range "0..255";
                        }
                        description
                             "The value of the PCEP session
                              ID used by the local PCEP
                              entity in the Open message
                              for this session.

                              If state is tcp-pending then
                              this is the session ID that
                              will be used in the Open
                              message. Otherwise, this is
                              the session ID that was sent
                              in the Open message.";
                    }

                    leaf remote-id {
                        type uint32 {
                            range "0..255";
                        }
                        must "((../state != 'tcp-pending'" +
                             "and " +
                             "../state != 'open-wait' )" +
                             "or " +
                             "((../state = 'tcp-pending'" +
                             " or " +
                             "../state = 'open-wait' )" +
                             "and remote-id = 0))" {
                                error-message
                                     "Invalid remote-id";
                                description
                                     "If state is tcp-pending
                                      or open-wait then this
                                      leaf is not used and
                                      MUST be set to zero.";
```

```
                        }
                    description
                         "The value of the PCEP session
                          ID used by the peer in its
                          Open message for this
                          session.";
                }

                leaf keepalive-timer {
                    type uint32 {
                      range "0..255";
                    }
                    units "seconds";
                    must "(../state = 'session-up'" +
                         "or " +
                         "(../state != 'session-up'" +
                         "and keepalive-timer = 0))" {
                           error-message
                               "Invalid keepalive
                                timer";
                           description
                               "This field is used if
                                and only if state is
                                session-up. Otherwise,
                                it is not used and
                                MUST be set to
                                zero.";
                        }
                    description
                         "The agreed maximum interval at
                          which the local PCEP entity
                          transmits PCEP messages on this
                          PCEP session.  Zero means that
                          the local PCEP entity never
                          sends Keepalives on this
                          session.";
                }

                leaf peer-keepalive-timer {
                    type uint32 {
                      range "0..255";
                    }
                    units "seconds";
                    must "(../state = 'session-up'" +
                         "or "   +
                         "(../state != 'session-up'" +
                         "and " +
                         "peer-keepalive-timer = 0))" {
```

```
                                error-message
                                    "Invalid Peer keepalive
                                     timer";
                                description
                                    "This field is used if
                                     and only if state is
                                     session-up. Otherwise,
                                     it is not used and MUST
                                     be set to zero.";
                            }
                        description
                             "The agreed maximum interval at
                              which the peer transmits PCEP
                              messages on this PCEP session.
                              Zero means that the peer never
                              sends Keepalives on this
                              session.";
                    }

                    leaf dead-timer {
                        type uint32 {
                          range "0..255";
                        }
                        units "seconds";
                        description
                             "The dead timer interval for
                              this PCEP session.";
                    }

                    leaf peer-dead-timer {
                        type uint32 {
                          range "0..255";
                        }
                        units "seconds";
                        must "((../state != 'tcp-pending'" +
                             "and " +
                             "../state != 'open-wait' )" +
                             "or " +
                             "((../state = 'tcp-pending'" +
                             " or " +
                             "../state = 'open-wait' )" +
                             "and " +
                             "peer-dead-timer = 0))" {
                                error-message
                                    "Invalid Peer Dead
                                     timer";
                                description
                                    "If state is tcp-
```

```
                                  pending or open-wait
                                  then this leaf is not
                                  used and MUST be set to
                                  zero.";
                         }
                     description
                          "The peer’s dead-timer interval
                           for this PCEP session.";
                 }

                 leaf ka-hold-time-rem {
                     type uint32 {
                       range "0..255";
                     }
                     units "seconds";
                     must "((../state != ’tcp-pending’" +
                          "and " +
                          "../state != ’open-wait’ ) " +
                          "or " +
                          "((../state = ’tcp-pending’" +
                          "or " +
                          "../state = ’open-wait’ )" +
                          "and " +
                          "ka-hold-time-rem = 0))" {
                             error-message
                                 "Invalid Keepalive hold
                                  time remaining";
                             description
                                 "If state is tcp-pending
                                  or open-wait then this
                                  field is not used and
                                  MUST be set to zero.";
                         }
                     description
                          "The keep alive hold time
                           remaining for this session.";
                 }

                 leaf overloaded {
                     type boolean;
                     description
                          "If the local PCEP entity has
                           informed the peer that it is
                           currently overloaded, then this
                           is set to true.  Otherwise, it
                           is set to false.";
                 }
```

```
                            leaf overload-time {
                                type uint32;
                                units "seconds";
                                must "(../overloaded = true or" +
                                    "(../overloaded != true and" +
                                    " overload-time = 0))" {
                                        error-message
                                            "Invalid overload-time";
                                        description
                                            "This field is only used
                                             if overloaded is set to
                                             true. Otherwise, it is
                                             not used and MUST be set
                                             to zero.";
                                    }
                                description
                                     "The interval of time that is
                                      remaining until the local PCEP
                                      entity will cease to be
                                      overloaded on this session.";
                            }

                            leaf peer-overloaded {
                                type boolean;
                                description
                                     "If the peer has informed the
                                      local PCEP entity that it is
                                      currently overloaded, then this
                                      is set to true. Otherwise, it
                                      is set to false.";
                            }

                            leaf peer-overload-time {
                                type uint32;
                                units "seconds";
                                must "(../peer-overloaded = true" +
                                     " or " +
                                     "(../peer-overloaded != true" +
                                     " and " +
                                     "peer-overload-time = 0))" {
                                        error-message
                                            "Invalid peer overload
                                             time";
                                        description
                                            "This field is only used
                                             if peer-overloaded is
                                             set to true. Otherwise,
                                             it is not used and MUST
```

```
                                          be set to zero.";
                                 }
                             description
                                  "The interval of time that is
                                   remaining until the peer will
                                   cease to be overloaded.  If it
                                   is not known how long the peer
                                   will stay in overloaded state,
                                   this leaf is set to zero.";
                         }

                         leaf discontinuity-time {
                             type yang:timestamp;
                             description
                                  "The timestamp value of the time
                                   when the statistics were last
                                   reset.";
                         }

                         uses pcep-stats{
                             description
                                 "The statistics contained are
                                  for the current sessions to that
                                  peer. These are lost when the
                                  session goes down.";

                         }

                     } // session
                 } // sessions
             }//peer
         }//peers
     }//entity
 }//pcep-state

 /*
  * Notifications
  */
 notification pcep-session-up {
     description
         "This notification is sent when the value of
          '/pcep/pcep-state/peers/peer/sessions/session/state'
          enters the 'session-up' state.";

     uses notification-instance-hdr;

     uses notification-session-hdr;
```

```
        leaf state-last-change {
            type yang:timestamp;
            description
                "The timestamp value at the time this session entered
                its current state as denoted by the state leaf.";
        }

        leaf state {
            type pcep-sess-state;
            description
                "The current state of the session.

                 The set of possible states excludes the idle state
                 since entries do not exist in the idle state.";
        }
    } //notification

    notification pcep-session-down {
        description
            "This notification is sent when the value of
             '/pcep/pcep-state/peers/peer/sessions/session/state'
             leaves the 'session-up' state.";

        uses notification-instance-hdr;

        leaf session-initiator {
            type pcep-initiator;
            description
                "The initiator of the session.";
        }

        leaf state-last-change {
            type yang:timestamp;
            description
                "The timestamp value at the time this session entered
                its current state as denoted by the state leaf.";
        }

        leaf state {
            type pcep-sess-state;
            description
                "The current state of the session.

                 The set of possible states excludes the idle state
                 since entries do not exist in the idle state.";
        }
    } //notification
```

```
    notification pcep-session-local-overload {
        description
            "This notification is sent when the local PCEP entity
            enters overload state for a peer.";

        uses notification-instance-hdr;

        uses notification-session-hdr;

        leaf overloaded {
            type boolean;
            description
                 "If the local PCEP entity has informed the peer that
                 it is currently overloaded, then this is set to
                 true. Otherwise, it is set to false.";
        }

        leaf overload-time {
            type uint32;
            units "seconds";
            must "(../overloaded = true or "    +
                "(../overloaded != true  and "  +
                "overload-time = 0))" {
                    error-message
                        "Invalid overload-time";
                    description
                        "This field is only used if overloaded is
                         set to true. Otherwise, it is not used
                         and MUST be set to zero.";
                }
            description
                 "The interval of time that is remaining until the
                 local PCEP entity will cease to be overloaded on
                 this session.";
        }
    } //notification

    notification pcep-session-local-overload-clear {
        description
            "This notification is sent when the local PCEP entity
            leaves overload state for a peer.";

        uses notification-instance-hdr;

        leaf overloaded {
            type boolean;
            description
                 "If the local PCEP entity has informed the peer
```

```
                 that it is currently overloaded, then this is set
                 to true.  Otherwise, it is set to false.";
        }
    } //notification

    notification pcep-session-peer-overload {
        description
            "This notification is sent when a peer enters overload
            state.";

        uses notification-instance-hdr;

        uses notification-session-hdr;

        leaf peer-overloaded {
            type boolean;
            description
                "If the peer has informed the local PCEP entity that
                it is currently overloaded, then this is set to true.
                Otherwise, it is set to false.";
        }

        leaf peer-overload-time {
            type uint32;
            units "seconds";
            must "(../peer-overloaded = true or "    +
                "(../peer-overloaded != true  and "  +
                "peer-overload-time = 0))" {
                    error-message
                        "Invalid peer-overload-time";
                    description
                        "This field is only used if
                         peer-overloaded is set to true.
                         Otherwise, it is not used and MUST
                         be set to zero.";
                }
            description
                "The interval of time that is remaining until the
                peer will cease to be overloaded.  If it is not known
                how long the peer will stay in overloaded state, this
                leaf is set to zero.";
        }
    } //notification

    notification pcep-session-peer-overload-clear {
        description
            "This notification is sent when a peer leaves overload
            state.";
```

```
        uses notification-instance-hdr;

        leaf peer-overloaded {
            type boolean;
            description
                "If the peer has informed the local PCEP entity that
                it is currently overloaded, then this is set to true.
                Otherwise, it is set to false.";
        }
    } //notification
}//module
```

<CODE ENDS>

8. Security Considerations

The YANG module defined in this memo is designed to be accessed via the NETCONF protocol [RFC6241]. The lowest NETCONF layer is the secure transport layer and the mandatory-to-implement secure transport is SSH [RFC6242]. The NETCONF access control model [RFC6536] provides the means to restrict access for particular NETCONF users to a pre-configured subset of all available NETCONF protocol operations and content.

There are a number of data nodes defined in the YANG module which are writable/creatable/deletable (i.e., config true, which is the default). These data nodes may be considered sensitive or vulnerable in some network environments. Write operations (e.g., <edit-config>) to these data nodes without proper protection can have a negative effect on network operations.

TBD: List specific Subtrees and data nodes and their sensitivity/ vulnerability.

9. Manageability Considerations

9.1. Control of Function and Policy

9.2. Information and Data Models

9.3. Liveness Detection and Monitoring

9.4. Verify Correct Operations

9.5. Requirements On Other Protocols

9.6. Impact On Network Operations

10. IANA Considerations

This document registers a URI in the "IETF XML Registry" [RFC3688]. Following the format in RFC 3688, the following registration has been made.

URI: urn:ietf:params:xml:ns:yang:ietf-pcep

Registrant Contact: The PCE WG of the IETF.

XML: N/A; the requested URI is an XML namespace.

This document registers a YANG module in the "YANG Module Names" registry [RFC6020].

```
Name:         ietf-pcep
Namespace:    urn:ietf:params:xml:ns:yang:ietf-pcep
Prefix:       pcep
Reference:    This I-D
```

11. Acknowledgements

The initial document is based on the PCEP MIB [RFC7420]. Further this document structure is based on Routing Yang Module [I-D.ietf-netmod-routing-cfg]. We would like to thank the authors of aforementioned documents.

12. References

12.1. Normative References

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

[RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, January 2004.

[RFC5440] Vasseur, JP. and JL. Le Roux, "Path Computation Element (PCE) Communication Protocol (PCEP)", RFC 5440, March 2009.

[RFC6020] Bjorklund, M., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, October 2010.

[RFC6991] Schoenwaelder, J., "Common YANG Data Types", RFC 6991, July 2013.

12.2. Informative References

[RFC4655] Farrel, A., Vasseur, J., and J. Ash, "A Path Computation Element (PCE)-Based Architecture", RFC 4655, August 2006.

[RFC6241] Enns, R., Bjorklund, M., Schoenwaelder, J., and A. Bierman, "Network Configuration Protocol (NETCONF)", RFC 6241, June 2011.

[RFC6242] Wasserman, M., "Using the NETCONF Protocol over Secure Shell (SSH)", RFC 6242, June 2011.

[RFC6536] Bierman, A. and M. Bjorklund, "Network Configuration Protocol (NETCONF) Access Control Model", RFC 6536, March 2012.

[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, December 2014.

[I-D.ietf-netmod-routing-cfg]
Lhotka, L., "A YANG Data Model for Routing Management", draft-ietf-netmod-routing-cfg-16 (work in progress), October 2014.

[I-D.ietf-netmod-rfc6087bis]
Bierman, A., "Guidelines for Authors and Reviewers of YANG Data Model Documents", draft-ietf-netmod-rfc6087bis-01 (work in progress), October 2014.

Appendix A. Contributor Addresses

Young Lee
Huawei Technologies
5340 Legacy Drive, Building 3
Plano, TX 75023, USA

Phone: (469) 277-5838
EMail: leeyoung@huawei.com

Udayasree Palle
Huawei Technologies
Divyashree Techno Park, Whitefield
Bangalore, Karnataka 560037
India

EMail: udayasree.palle@huawei.com

Vishnu Pavan Beeram
Juniper Networks
USA

EMail: vbeeram@juniper.net

Xian Zhang
Huawei Technologies
Bantian, Longgang District
Shenzhen 518129
P.R.China

EMail: zhang.xian@huawei.com

Authors' Addresses

Rohit Pobbathi
Huawei Technologies
Divyashree Techno Park, Whitefield
Bangalore, Karnataka 560037
India

EMail: rohit.pobbathi@huawei.com

Vinod KumarS
Huawei Technologies
Divyashree Techno Park, Whitefield
Bangalore, Karnataka 560037
India

EMail: vinods.kumar@huawei.com

Dhruv Dhody (editor)
Huawei Technologies
Divyashree Techno Park, Whitefield
Bangalore, Karnataka 560037
India

EMail: dhruv.ietf@gmail.com

Jonathan Hardwick
Metaswitch
100 Church Street
Enfield EN2 6BQ
UK

EMail: jonathan.hardwick@metaswitch.com

PCE Working Group
Internet-Draft
Intended status: Standards Track
Expires: January 8, 2017

D. Dhody, Ed.
Huawei Technologies
J. Hardwick
Metaswitch
V. Beeram
Juniper Networks
J. Tantsura

July 7, 2016

A YANG Data Model for Path Computation Element Communications Protocol (PCEP)

draft-pkd-pce-pcep-yang-06

Abstract

This document defines a YANG data model for the management of Path Computation Element communications Protocol (PCEP) for communications between a Path Computation Client (PCC) and a Path Computation Element (PCE), or between two PCEs. The data model includes configuration data and state data (status information and counters for the collection of statistics).

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 January 8, 2017.

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

The Path Computation Element (PCE) defined in [RFC4655] 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.

PCEP is the communication protocol between a PCC and PCE and is defined in [RFC5440]. PCEP interactions include path computation requests and path computation replies as well as notifications of specific states related to the use of a PCE in the context of Multiprotocol Label Switching (MPLS) and Generalized MPLS (GMPLS) Traffic Engineering (TE). [I-D.ietf-pce-stateful-pce] specifies extensions to PCEP to enable stateful control of MPLS TE LSPs.

This document defines a YANG [RFC6020] data model for the management of PCEP speakers. It is important to establish a common data model for how PCEP speakers are identified, configured, and monitored. The data model includes configuration data and state data (status information and counters for the collection of statistics).

This document contains a specification of the PCEP YANG module, "ietf-pcep" which provides the PCEP [RFC5440] data model.

2. 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].

3. Terminology and Notation

This document uses the terminology defined in [RFC4655] and [RFC5440]. In particular, it uses the following acronyms.

- o Path Computation Request message (PCReq).
- o Path Computation Reply message (PCRep).
- o Notification message (PCNtf).
- o Error message (PCErr).
- o Request Parameters object (RP).
- o Synchronization Vector object (SVEC).
- o Explicit Route object (ERO).

This document also uses the following terms defined in [RFC7420]:

- o PCEP entity: a local PCEP speaker.

- o PCEP peer: to refer to a remote PCEP speaker.

- o PCEP speaker: where it is not necessary to distinguish between local and remote.

Further, this document also uses the following terms defined in [I-D.ietf-pce-stateful-pce] :

- o Stateful PCE, Passive Stateful PCE, Active Stateful PCE

- o Delegation, Revocation, Redelegation

- o LSP State Report, Path Computation Report message (PCRpt).

- o LSP State Update, Path Computation Update message (PCUpd).

[I-D.ietf-pce-pce-initiated-lsp] :

- o PCE-initiated LSP, Path Computation LSP Initiate Message (PCInitiate).

[I-D.ietf-pce-lsp-setup-type] :

- o Path Setup Type (PST).

[I-D.ietf-pce-segment-routing] :

- o Segment Routing (SR).

- o Segment Identifier (SID).

- o Maximum SID Depth (MSD).

3.1. Tree Diagrams

A graphical representation of the complete data tree is presented in Section 5. The meaning of the symbols in these diagrams is as follows and as per [I-D.ietf-netmod-rfc6087bis]:

- o Brackets "[" and "]" enclose list keys.

- o Curly braces "{" and "}" contain names of optional features that make the corresponding node conditional.

- o Abbreviations before data node names: "rw" means configuration (read-write), and "ro" state data (read-only).

- Symbols after data node names: "?" means an optional node and "*" denotes a "list" or "leaf-list".

- Parentheses enclose choice and case nodes, and case nodes are also marked with a colon (":").

- Ellipsis ("...") stands for contents of subtrees that are not shown.

3.2. Prefixes in Data Node Names

In this document, names of data nodes and other data model objects are often used without a prefix, as long as it is clear from the context in which YANG module each name is defined. Otherwise, names are prefixed using the standard prefix associated with the corresponding YANG module, as shown in Table 1.

Prefix	YANG module	Reference
yang	ietf-yang-types	[RFC6991]
inet	ietf-inet-types	[RFC6991]

Table 1: Prefixes and corresponding YANG modules

4. Objectives

This section describes some of the design objectives for the model:

- In case of existing implementations, it needs to map the data model defined in this document to their proprietary native data model. To facilitate such mappings, the data model should be simple.

- The data model should be suitable for new implementations to use as is.

- Mapping to the PCEP MIB Module should be clear.

- The data model should allow for static configurations of peers.

- The data model should include read-only counters in order to gather statistics for sent and received PCEP messages, received messages with errors, and messages that could not be sent due to errors.

o It should be fairly straightforward to augment the base data model for advanced PCE features.

5. The Design of PCEP Data Model

The module, "ietf-pcep", defines the basic components of a PCE speaker.

```
module: ietf-pcep
   +--rw pcep!
   |  +--rw entity
   |     +--rw addr                           inet:ip-address
   |     +--rw enabled?                       boolean
   |     +--rw role                           pcep-role
   |     +--rw description?                   string
   |     +--rw domain
   |     |  +--rw domain* [domain-type domain]
   |     |     +--rw domain-type    domain-type
   |     |     +--rw domain         domain
   |     +--rw capability
   |     |  +--rw gmpls?                   boolean {gmpls}?
   |     |  +--rw bi-dir?                  boolean
   |     |  +--rw diverse?                 boolean
   |     |  +--rw load-balance?            boolean
   |     |  +--rw synchronize?             boolean {svec}?
   |     |  +--rw objective-function?      boolean {obj-fn}?
   |     |  +--rw add-path-constraint?     boolean
   |     |  +--rw prioritization?          boolean
   |     |  +--rw multi-request?           boolean
   |     |  +--rw gco?                     boolean {gco}?
   |     |  +--rw p2mp?                    boolean {p2mp}?
   |     |  +--rw stateful {stateful}?
   |     |  |  +--rw enabled?         boolean
   |     |  |  +--rw active?          boolean
   |     |  |  +--rw pce-initiated?   boolean {pce-initiated}?
   |     |  +--rw sr {sr}?
   |     |     +--rw enabled?   boolean
   |     |     +--rw msd?       uint8
   |     +--rw pce-info
   |     |  +--rw scope
   |     |  |  +--rw intra-area-scope?           boolean
   |     |  |  +--rw intra-area-pref?            uint8
   |     |  |  +--rw inter-area-scope?           boolean
   |     |  |  +--rw inter-area-scope-default?   boolean
   |     |  |  +--rw inter-area-pref?            uint8
   |     |  |  +--rw inter-as-scope?             boolean
   |     |  |  +--rw inter-as-scope-default?     boolean
   |     |  |  +--rw inter-as-pref?              uint8
```

```
   |     |  |  +--rw inter-layer-scope?          boolean
   |     |  |  +--rw inter-layer-pref?           uint8
   |     |  +--rw neigh-domains
   |     |  |  +--rw domain* [domain-type domain]
   |     |  |     +--rw domain-type    domain-type
   |     |  |     +--rw domain         domain
   |     |  +--rw (auth-type-selection)?
   |     |     +--:(auth-key-chain)
   |     |     |  +--rw key-chain?          key-chain:key-chain-ref
   |     |     +--:(auth-key)
   |     |     |  +--rw key?                string
   |     |     |  +--rw crypto-algorithm
   |     |     |     +--rw (algorithm)?
   |     |     |        +--:(hmac-sha-1-12) {crypto-hmac-sha-1-12}?
   |     |     |        |  +--rw hmac-sha1-12?             empty
   |     |     |        +--:(aes-cmac-prf-128) {aes-cmac-prf-128}?
   |     |     |        |  +--rw aes-cmac-prf-128?         empty
   |     |     |        +--:(md5)
   |     |     |        |  +--rw md5?                      empty
   |     |     |        +--:(sha-1)
   |     |     |        |  +--rw sha-1?                    empty
   |     |     |        +--:(hmac-sha-1)
   |     |     |        |  +--rw hmac-sha-1?               empty
   |     |     |        +--:(hmac-sha-256)
   |     |     |        |  +--rw hmac-sha-256?             empty
   |     |     |        +--:(hmac-sha-384)
   |     |     |        |  +--rw hmac-sha-384?             empty
   |     |     |        +--:(hmac-sha-512)
   |     |     |        |  +--rw hmac-sha-512?             empty
   |     |     |        +--:(clear-text) {clear-text}?
   |     |     |        |  +--rw clear-text?               empty
   |     |     |        +--:(replay-protection-only) {replay-protection-only}?
   |     |     |           +--rw replay-protection-only?   empty
   |     |     +--:(auth-tls) {tls}?
   |     |        +--rw tls
   |     +--rw connect-timer?               uint32
   |     +--rw connect-max-retry?           uint32
   |     +--rw init-backoff-timer?          uint32
   |     +--rw max-backoff-timer?           uint32
   |     +--rw open-wait-timer?             uint32
   |     +--rw keep-wait-timer?             uint32
   |     +--rw keep-alive-timer?            uint32
   |     +--rw dead-timer?                  uint32
   |     +--rw allow-negotiation?           boolean
   |     +--rw max-keep-alive-timer?        uint32
   |     +--rw max-dead-timer?              uint32
   |     +--rw min-keep-alive-timer?        uint32
   |     +--rw min-dead-timer?              uint32
```

```
   |     +--rw sync-timer?                   uint32 {svec}?
   |     +--rw request-timer?                uint32
   |     +--rw max-sessions?                 uint32
   |     +--rw max-unknown-reqs?             uint32
   |     +--rw max-unknown-msgs?             uint32
   |     +--rw pcep-notification-max-rate    uint32
   |     +--rw stateful-parameter {stateful}?
   |     |  +--rw state-timeout?          uint32
   |     |  +--rw redelegation-timeout?   uint32
   |     |  +--rw rpt-non-pcep-lsp?       boolean
   |     +--rw peers
   |        +--rw peer* [addr]
   |           +--rw addr                 inet:ip-address
   |           +--rw description?         string
   |           +--rw domain
   |           |  +--rw domain* [domain-type domain]
   |           |     +--rw domain-type    domain-type
   |           |     +--rw domain         domain
   |           +--rw capability
   |           |  +--rw gmpls?                 boolean {gmpls}?
   |           |  +--rw bi-dir?                boolean
   |           |  +--rw diverse?               boolean
   |           |  +--rw load-balance?          boolean
   |           |  +--rw synchronize?           boolean {svec}?
   |           |  +--rw objective-function?    boolean {obj-fn}?
   |           |  +--rw add-path-constraint?   boolean
   |           |  +--rw prioritization?        boolean
   |           |  +--rw multi-request?         boolean
   |           |  +--rw gco?                   boolean {gco}?
   |           |  +--rw p2mp?                  boolean {p2mp}?
   |           |  +--rw stateful {stateful}?
   |           |  |  +--rw enabled?         boolean
   |           |  |  +--rw active?          boolean
   |           |  |  +--rw pce-initiated?   boolean {pce-initiated}?
   |           |  +--rw sr {sr}?
   |           |     +--rw enabled?   boolean
   |           |     +--rw msd?       uint8
   |           +--rw scope
   |           |  +--rw intra-area-scope?           boolean
   |           |  +--rw intra-area-pref?            uint8
   |           |  +--rw inter-area-scope?           boolean
   |           |  +--rw inter-area-scope-default?   boolean
   |           |  +--rw inter-area-pref?            uint8
   |           |  +--rw inter-as-scope?             boolean
   |           |  +--rw inter-as-scope-default?     boolean
   |           |  +--rw inter-as-pref?              uint8
   |           |  +--rw inter-layer-scope?          boolean
   |           |  +--rw inter-layer-pref?           uint8
```

```
   |            +--rw neigh-domains
   |            |  +--rw domain* [domain-type domain]
   |            |     +--rw domain-type    domain-type
   |            |     +--rw domain         domain
   |            +--rw delegation-pref?    uint8 {stateful}?
   |            +--rw (auth-type-selection)?
   |               +--:(auth-key-chain)
   |               |  +--rw key-chain?          key-chain:key-chain-ref
   |               +--:(auth-key)
   |               |  +--rw key?                string
   |               |  +--rw crypto-algorithm
   |               |     +--rw (algorithm)?
   |               |        +--:(hmac-sha-1-12) {crypto-hmac-sha-1-12}?
   |               |        |  +--rw hmac-sha1-12?             empty
   |               |        +--:(aes-cmac-prf-128) {aes-cmac-prf-128}?
   |               |        |  +--rw aes-cmac-prf-128?         empty
   |               |        +--:(md5)
   |               |        |  +--rw md5?                      empty
   |               |        +--:(sha-1)
   |               |        |  +--rw sha-1?                    empty
   |               |        +--:(hmac-sha-1)
   |               |        |  +--rw hmac-sha-1?               empty
   |               |        +--:(hmac-sha-256)
   |               |        |  +--rw hmac-sha-256?             empty
   |               |        +--:(hmac-sha-384)
   |               |        |  +--rw hmac-sha-384?             empty
   |               |        +--:(hmac-sha-512)
   |               |        |  +--rw hmac-sha-512?             empty
   |               |        +--:(clear-text) {clear-text}?
   |               |        |  +--rw clear-text?               empty
   |               |        +--:(replay-protection-only) {replay-protection-only}
?
   |               |           +--rw replay-protection-only?   empty
   |               +--:(auth-tls) {tls}?
   |                  +--rw tls
   +--ro pcep-state
      +--ro entity
         +--ro addr?                    inet:ip-address
         +--ro index?                   uint32
         +--ro admin-status?            pcep-admin-status
         +--ro oper-status?             pcep-admin-status
         +--ro role?                    pcep-role
         +--ro domain
         |  +--ro domain* [domain-type domain]
         |     +--ro domain-type    domain-type
         |     +--ro domain         domain
         +--ro capability
         |  +--ro gmpls?                  boolean {gmpls}?
         |  +--ro bi-dir?                 boolean
```

```
         |  +--ro diverse?               boolean
         |  +--ro load-balance?          boolean
         |  +--ro synchronize?           boolean {svec}?
         |  +--ro objective-function?    boolean {obj-fn}?
         |  +--ro add-path-constraint?   boolean
         |  +--ro prioritization?        boolean
         |  +--ro multi-request?         boolean
         |  +--ro gco?                   boolean {gco}?
         |  +--ro p2mp?                  boolean {p2mp}?
         |  +--ro stateful {stateful}?
         |  |  +--ro enabled?         boolean
         |  |  +--ro active?          boolean
         |  |  +--ro pce-initiated?   boolean {pce-initiated}?
         |  +--ro sr {sr}?
         |     +--ro enabled?   boolean
         |     +--ro msd?       uint8
         +--ro pce-info
         |  +--ro scope
         |  |  +--ro intra-area-scope?           boolean
         |  |  +--ro intra-area-pref?            uint8
         |  |  +--ro inter-area-scope?           boolean
         |  |  +--ro inter-area-scope-default?   boolean
         |  |  +--ro inter-area-pref?            uint8
         |  |  +--ro inter-as-scope?             boolean
         |  |  +--ro inter-as-scope-default?     boolean
         |  |  +--ro inter-as-pref?              uint8
         |  |  +--ro inter-layer-scope?          boolean
         |  |  +--ro inter-layer-pref?           uint8
         |  +--ro neigh-domains
         |  |  +--ro domain* [domain-type domain]
         |  |     +--ro domain-type    domain-type
         |  |     +--ro domain         domain
         |  +--ro (auth-type-selection)?
         |     +--:(auth-key-chain)
         |     |  +--ro key-chain?          key-chain:key-chain-ref
         |     +--:(auth-key)
         |     |  +--ro key?                string
         |     |  +--ro crypto-algorithm
         |     |     +--ro (algorithm)?
         |     |        +--:(hmac-sha-1-12) {crypto-hmac-sha-1-12}?
         |     |        |  +--ro hmac-sha1-12?             empty
         |     |        +--:(aes-cmac-prf-128) {aes-cmac-prf-128}?
         |     |        |  +--ro aes-cmac-prf-128?         empty
         |     |        +--:(md5)
         |     |        |  +--ro md5?                      empty
         |     |        +--:(sha-1)
         |     |        |  +--ro sha-1?                    empty
         |     |        +--:(hmac-sha-1)
```

```
         |     |        |  +--ro hmac-sha-1?               empty
         |     |        +--:(hmac-sha-256)
         |     |        |  +--ro hmac-sha-256?             empty
         |     |        +--:(hmac-sha-384)
         |     |        |  +--ro hmac-sha-384?             empty
         |     |        +--:(hmac-sha-512)
         |     |        |  +--ro hmac-sha-512?             empty
         |     |        +--:(clear-text) {clear-text}?
         |     |        |  +--ro clear-text?               empty
         |     |        +--:(replay-protection-only) {replay-protection-only}?
         |     |           +--ro replay-protection-only?   empty
         |     +--:(auth-tls) {tls}?
         |        +--ro tls
         +--ro connect-timer?          uint32
         +--ro connect-max-retry?      uint32
         +--ro init-backoff-timer?     uint32
         +--ro max-backoff-timer?      uint32
         +--ro open-wait-timer?        uint32
         +--ro keep-wait-timer?        uint32
         +--ro keep-alive-timer?       uint32
         +--ro dead-timer?             uint32
         +--ro allow-negotiation?      boolean
         +--ro max-keep-alive-timer?   uint32
         +--ro max-dead-timer?         uint32
         +--ro min-keep-alive-timer?   uint32
         +--ro min-dead-timer?         uint32
         +--ro sync-timer?             uint32 {svec}?
         +--ro request-timer?          uint32
         +--ro max-sessions?           uint32
         +--ro max-unknown-reqs?       uint32
         +--ro max-unknown-msgs?       uint32
         +--ro stateful-parameter {stateful}?
         |  +--ro state-timeout?          uint32
         |  +--ro redelegation-timeout?   uint32
         |  +--ro rpt-non-pcep-lsp?       boolean
         +--ro lsp-db {stateful}?
         |  +--ro association-list* [id source global-source extended-id]
         |  |  +--ro type?            assoc-type
         |  |  +--ro id               uint16
         |  |  +--ro source           inet:ip-address
         |  |  +--ro global-source    uint32
         |  |  +--ro extended-id      string
         |  |  +--ro lsp* [plsp-id pcc-id]
         |  |     +--ro plsp-id    -> /pcep-state/entity/lsp-db/lsp/plsp-id
         |  |     +--ro pcc-id     -> /pcep-state/entity/lsp-db/lsp/pcc-id
         |  +--ro lsp* [plsp-id pcc-id]
         |     +--ro plsp-id              uint32
         |     +--ro pcc-id               inet:ip-address
```

```
         |     +--ro lsp-ref
         |     |  +--ro source?               -> /te:te/lsps-state/lsp/source
         |     |  +--ro destination?          -> /te:te/lsps-state/lsp/destinati
on
         |     |  +--ro tunnel-id?            -> /te:te/lsps-state/lsp/tunnel-id
         |     |  +--ro lsp-id?               -> /te:te/lsps-state/lsp/lsp-id
         |     |  +--ro extended-tunnel-id?   -> /te:te/lsps-state/lsp/extended-
tunnel-id
         |     |  +--ro type?                 -> /te:te/lsps-state/lsp/type
         |     +--ro admin-state?          boolean
         |     +--ro operational-state?    operational-state
         |     +--ro delegated
         |     |  +--ro enabled?   boolean
         |     |  +--ro pce?       -> /pcep-state/entity/peers/peer/addr
         |     |  +--ro srp-id?    uint32
         |     +--ro initiation {pce-initiated}?
         |     |  +--ro enabled?   boolean
         |     |  +--ro pce?       -> /pcep-state/entity/peers/peer/addr
         |     +--ro symbolic-path-name?   string
         |     +--ro last-error?           lsp-error
         |     +--ro pst?                  pst
         |     +--ro association-list* [id source global-source extended-id]
         |        +--ro id               -> /pcep-state/entity/lsp-db/associatio
n-list/id
         |        +--ro source           -> /pcep-state/entity/lsp-db/associatio
n-list/source
         |        +--ro global-source    -> /pcep-state/entity/lsp-db/associatio
n-list/global-source
         |        +--ro extended-id      -> /pcep-state/entity/lsp-db/associatio
n-list/extended-id
         +--ro peers
            +--ro peer* [addr]
               +--ro addr                     inet:ip-address
               +--ro role?                    pcep-role
               +--ro domain
               |  +--ro domain* [domain-type domain]
               |     +--ro domain-type    domain-type
               |     +--ro domain         domain
               +--ro capability
               |  +--ro gmpls?                  boolean {gmpls}?
               |  +--ro bi-dir?                 boolean
               |  +--ro diverse?                boolean
               |  +--ro load-balance?           boolean
               |  +--ro synchronize?            boolean {svec}?
               |  +--ro objective-function?     boolean {obj-fn}?
               |  +--ro add-path-constraint?    boolean
               |  +--ro prioritization?         boolean
               |  +--ro multi-request?          boolean
               |  +--ro gco?                    boolean {gco}?
               |  +--ro p2mp?                   boolean {p2mp}?
               |  +--ro stateful {stateful}?
               |  |  +--ro enabled?          boolean
               |  |  +--ro active?           boolean
               |  |  +--ro pce-initiated?    boolean {pce-initiated}?
```

```
               |  +--ro sr {sr}?
               |     +--ro enabled?   boolean
               |     +--ro msd?       uint8
               +--ro pce-info
               |  +--ro scope
               |  |  +--ro intra-area-scope?           boolean
               |  |  +--ro intra-area-pref?            uint8
               |  |  +--ro inter-area-scope?           boolean
               |  |  +--ro inter-area-scope-default?   boolean
               |  |  +--ro inter-area-pref?            uint8
               |  |  +--ro inter-as-scope?             boolean
               |  |  +--ro inter-as-scope-default?     boolean
               |  |  +--ro inter-as-pref?              uint8
               |  |  +--ro inter-layer-scope?          boolean
               |  |  +--ro inter-layer-pref?           uint8
               |  +--ro neigh-domains
               |     +--ro domain* [domain-type domain]
               |        +--ro domain-type    domain-type
               |        +--ro domain         domain
               +--ro delegation-pref?        uint8 {stateful}?
               +--ro (auth-type-selection)?
               |  +--:(auth-key-chain)
               |  |  +--ro key-chain?             key-chain:key-chain-ref
               |  +--:(auth-key)
               |  |  +--ro key?                   string
               |  |  +--ro crypto-algorithm
               |  |     +--ro (algorithm)?
               |  |        +--:(hmac-sha-1-12) {crypto-hmac-sha-1-12}?
               |  |        |  +--ro hmac-sha1-12?            empty
               |  |        +--:(aes-cmac-prf-128) {aes-cmac-prf-128}?
               |  |        |  +--ro aes-cmac-prf-128?        empty
               |  |        +--:(md5)
               |  |        |  +--ro md5?                     empty
               |  |        +--:(sha-1)
               |  |        |  +--ro sha-1?                   empty
               |  |        +--:(hmac-sha-1)
               |  |        |  +--ro hmac-sha-1?              empty
               |  |        +--:(hmac-sha-256)
               |  |        |  +--ro hmac-sha-256?            empty
               |  |        +--:(hmac-sha-384)
               |  |        |  +--ro hmac-sha-384?            empty
               |  |        +--:(hmac-sha-512)
               |  |        |  +--ro hmac-sha-512?            empty
               |  |        +--:(clear-text) {clear-text}?
               |  |        |  +--ro clear-text?              empty
               |  |        +--:(replay-protection-only) {replay-protection-only}
?
               |  |           +--ro replay-protection-only?   empty
               |  +--:(auth-tls) {tls}?
```

```
|     +--ro tls
+--ro discontinuity-time?     yang:timestamp
+--ro initiate-session?       boolean
+--ro session-exists?         boolean
+--ro num-sess-setup-ok?      yang:counter32
+--ro num-sess-setup-fail?    yang:counter32
+--ro session-up-time?        yang:timestamp
+--ro session-fail-time?      yang:timestamp
+--ro session-fail-up-time?   yang:timestamp
+--ro pcep-stats
|  +--ro avg-rsp-time?                uint32
|  +--ro lwm-rsp-time?                uint32
|  +--ro hwm-rsp-time?                uint32
|  +--ro num-pcreq-sent?              yang:counter32
|  +--ro num-pcreq-rcvd?              yang:counter32
|  +--ro num-pcrep-sent?              yang:counter32
|  +--ro num-pcrep-rcvd?              yang:counter32
|  +--ro num-pcerr-sent?              yang:counter32
|  +--ro num-pcerr-rcvd?              yang:counter32
|  +--ro num-pcntf-sent?              yang:counter32
|  +--ro num-pcntf-rcvd?              yang:counter32
|  +--ro num-keepalive-sent?          yang:counter32
|  +--ro num-keepalive-rcvd?          yang:counter32
|  +--ro num-unknown-rcvd?            yang:counter32
|  +--ro num-corrupt-rcvd?            yang:counter32
|  +--ro num-req-sent?                yang:counter32
|  +--ro num-req-sent-pend-rep?       yang:counter32
|  +--ro num-req-sent-ero-rcvd?       yang:counter32
|  +--ro num-req-sent-nopath-rcvd?    yang:counter32
|  +--ro num-req-sent-cancel-rcvd?    yang:counter32
|  +--ro num-req-sent-error-rcvd?     yang:counter32
|  +--ro num-req-sent-timeout?        yang:counter32
|  +--ro num-req-sent-cancel-sent?    yang:counter32
|  +--ro num-req-rcvd?                yang:counter32
|  +--ro num-req-rcvd-pend-rep?       yang:counter32
|  +--ro num-req-rcvd-ero-sent?       yang:counter32
|  +--ro num-req-rcvd-nopath-sent?    yang:counter32
|  +--ro num-req-rcvd-cancel-sent?    yang:counter32
|  +--ro num-req-rcvd-error-sent?     yang:counter32
|  +--ro num-req-rcvd-cancel-rcvd?    yang:counter32
|  +--ro num-rep-rcvd-unknown?        yang:counter32
|  +--ro num-req-rcvd-unknown?        yang:counter32
|  +--ro svec {svec}?
|  |  +--ro num-svec-sent?       yang:counter32
|  |  +--ro num-svec-req-sent?   yang:counter32
|  |  +--ro num-svec-rcvd?       yang:counter32
|  |  +--ro num-svec-req-rcvd?   yang:counter32
|  +--ro stateful {stateful}?
```

```
|  |  +--ro num-pcrpt-sent?            yang:counter32
|  |  +--ro num-pcrpt-rcvd?            yang:counter32
|  |  +--ro num-pcupd-sent?            yang:counter32
|  |  +--ro num-pcupd-rcvd?            yang:counter32
|  |  +--ro num-rpt-sent?              yang:counter32
|  |  +--ro num-rpt-rcvd?              yang:counter32
|  |  +--ro num-rpt-rcvd-error-sent?   yang:counter32
|  |  +--ro num-upd-sent?              yang:counter32
|  |  +--ro num-upd-rcvd?              yang:counter32
|  |  +--ro num-upd-rcvd-unknown?      yang:counter32
|  |  +--ro num-upd-rcvd-undelegated?  yang:counter32
|  |  +--ro num-upd-rcvd-error-sent?   yang:counter32
|  |  +--ro initiation {pce-initiated}?
|  |     +--ro num-pcinitiate-sent?           yang:counter32
|  |     +--ro num-pcinitiate-rcvd?           yang:counter32
|  |     +--ro num-initiate-sent?             yang:counter32
|  |     +--ro num-initiate-rcvd?             yang:counter32
|  |     +--ro num-initiate-rcvd-error-sent?  yang:counter32
|  +--ro num-req-sent-closed?        yang:counter32
|  +--ro num-req-rcvd-closed?        yang:counter32
+--ro sessions
   +--ro session* [initiator]
      +--ro initiator               pcep-initiator
      +--ro state-last-change?      yang:timestamp
      +--ro state?                  pcep-sess-state
      +--ro session-creation?       yang:timestamp
      +--ro connect-retry?          yang:counter32
      +--ro local-id?               uint32
      +--ro remote-id?              uint32
      +--ro keepalive-timer?        uint32
      +--ro peer-keepalive-timer?   uint32
      +--ro dead-timer?             uint32
      +--ro peer-dead-timer?        uint32
      +--ro ka-hold-time-rem?       uint32
      +--ro overloaded?             boolean
      +--ro overload-time?          uint32
      +--ro peer-overloaded?        boolean
      +--ro peer-overload-time?     uint32
      +--ro lspdb-sync?             sync-state {stateful}?
      +--ro discontinuity-time?     yang:timestamp
      +--ro pcep-stats
         +--ro avg-rsp-time?              uint32
         +--ro lwm-rsp-time?              uint32
         +--ro hwm-rsp-time?              uint32
         +--ro num-pcreq-sent?            yang:counter32
         +--ro num-pcreq-rcvd?            yang:counter32
         +--ro num-pcrep-sent?            yang:counter32
         +--ro num-pcrep-rcvd?            yang:counter32
```

```
                        +--ro num-pcerr-sent?           yang:counter32
                        +--ro num-pcerr-rcvd?           yang:counter32
                        +--ro num-pcntf-sent?           yang:counter32
                        +--ro num-pcntf-rcvd?           yang:counter32
                        +--ro num-keepalive-sent?       yang:counter32
                        +--ro num-keepalive-rcvd?       yang:counter32
                        +--ro num-unknown-rcvd?         yang:counter32
                        +--ro num-corrupt-rcvd?         yang:counter32
                        +--ro num-req-sent?             yang:counter32
                        +--ro num-req-sent-pend-rep?    yang:counter32
                        +--ro num-req-sent-ero-rcvd?    yang:counter32
                        +--ro num-req-sent-nopath-rcvd?   yang:counter32
                        +--ro num-req-sent-cancel-rcvd?   yang:counter32
                        +--ro num-req-sent-error-rcvd?    yang:counter32
                        +--ro num-req-sent-timeout?     yang:counter32
                        +--ro num-req-sent-cancel-sent?   yang:counter32
                        +--ro num-req-rcvd?             yang:counter32
                        +--ro num-req-rcvd-pend-rep?    yang:counter32
                        +--ro num-req-rcvd-ero-sent?    yang:counter32
                        +--ro num-req-rcvd-nopath-sent?   yang:counter32
                        +--ro num-req-rcvd-cancel-sent?   yang:counter32
                        +--ro num-req-rcvd-error-sent?    yang:counter32
                        +--ro num-req-rcvd-cancel-rcvd?   yang:counter32
                        +--ro num-rep-rcvd-unknown?     yang:counter32
                        +--ro num-req-rcvd-unknown?     yang:counter32
                        +--ro svec {svec}?
                        |  +--ro num-svec-sent?       yang:counter32
                        |  +--ro num-svec-req-sent?   yang:counter32
                        |  +--ro num-svec-rcvd?       yang:counter32
                        |  +--ro num-svec-req-rcvd?   yang:counter32
                        +--ro stateful {stateful}?
                           +--ro num-pcrpt-sent?            yang:counter32
                           +--ro num-pcrpt-rcvd?            yang:counter32
                           +--ro num-pcupd-sent?            yang:counter32
                           +--ro num-pcupd-rcvd?            yang:counter32
                           +--ro num-rpt-sent?              yang:counter32
                           +--ro num-rpt-rcvd?              yang:counter32
                           +--ro num-rpt-rcvd-error-sent?   yang:counter32
                           +--ro num-upd-sent?              yang:counter32
                           +--ro num-upd-rcvd?              yang:counter32
                           +--ro num-upd-rcvd-unknown?      yang:counter32
                           +--ro num-upd-rcvd-undelegated?  yang:counter32
                           +--ro num-upd-rcvd-error-sent?   yang:counter32
                           +--ro initiation {pce-initiated}?
                              +--ro num-pcinitiate-sent?           yang:counter
32
                              +--ro num-pcinitiate-rcvd?           yang:counter
32
                              +--ro num-initiate-sent?             yang:counter
32
                              +--ro num-initiate-rcvd?             yang:counter
32
```

```
                              +--ro num-initiate-rcvd-error-sent?   yang:counter
32
notifications:
   +---n pcep-session-up
   |  +--ro peer-addr?           -> /pcep-state/entity/peers/peer/addr
   |  +--ro session-initiator?   -> /pcep-state/entity/peers/peer/sessions/sessi
on/initiator
   |  +--ro state-last-change?   yang:timestamp
   |  +--ro state?               pcep-sess-state
   +---n pcep-session-down
   |  +--ro peer-addr?           -> /pcep-state/entity/peers/peer/addr
   |  +--ro session-initiator?   pcep-initiator
   |  +--ro state-last-change?   yang:timestamp
   |  +--ro state?               pcep-sess-state
   +---n pcep-session-local-overload
   |  +--ro peer-addr?           -> /pcep-state/entity/peers/peer/addr
   |  +--ro session-initiator?   -> /pcep-state/entity/peers/peer/sessions/sessi
on/initiator
   |  +--ro overloaded?          boolean
   |  +--ro overload-time?       uint32
   +---n pcep-session-local-overload-clear
   |  +--ro peer-addr?    -> /pcep-state/entity/peers/peer/addr
   |  +--ro overloaded?   boolean
   +---n pcep-session-peer-overload
   |  +--ro peer-addr?            -> /pcep-state/entity/peers/peer/addr
   |  +--ro session-initiator?    -> /pcep-state/entity/peers/peer/sessions/sess
ion/initiator
   |  +--ro peer-overloaded?      boolean
   |  +--ro peer-overload-time?   uint32
   +---n pcep-session-peer-overload-clear
      +--ro peer-addr?         -> /pcep-state/entity/peers/peer/addr
      +--ro peer-overloaded?   boolean
```

5.1. The Entity

The PCEP yang module may contain status information for the local PCEP entity.

The entity has an IP address (using ietf-inet-types [RFC6991]) and a "role" leaf (the local entity PCEP role) as mandatory.

Note that, the PCEP MIB module [RFC7420] uses an entity list and a system generated entity index as a primary index to the read only entity table. If the device implements the PCEP MIB, the "index" leaf MUST contain the value of the corresponding pcePcepEntityIndex and only one entity is assumed.

5.2. The Peer Lists

The peer list contains peer(s) that the local PCEP entity knows about. A PCEP speaker is identified by its IP address. If there is a PCEP speaker in the network that uses multiple IP addresses then it looks like multiple distinct peers to the other PCEP speakers in the network.

Since PCEP sessions can be ephemeral, the peer list tracks a peer even when no PCEP session currently exists to that peer. The statistics contained are an aggregate of the statistics for all successive sessions to that peer.

To limit the quantity of information that is stored, an implementation MAY choose to discard this information if and only if no PCEP session exists to the corresponding peer.

The data model for PCEP peer presented in this document uses a flat list of peers. Each peer in the list is identified by its IP address (addr-type, addr).

There is one list for static peer configuration ("/pcep/entity/peers"), and a separate list for the operational state of all peers (i.e. static as well as discovered)("/pcep-state/entity/peers"). The former is used to enable remote PCE configuration at PCC (or PCE) while the latter has the operational state of these peers as well as the remote PCE peer which were discovered and PCC peers that have initiated session.

5.3. The Session Lists

The session list contains PCEP session that the PCEP entity (PCE or PCC) is currently participating in. The statistics in session are semantically different from those in peer since the former applies to the current session only, whereas the latter is the aggregate for all sessions that have existed to that peer.

Although [RFC5440] forbids more than one active PCEP session between a given pair of PCEP entities at any given time, there is a window during session establishment where two sessions may exist for a given pair, one representing a session initiated by the local PCEP entity and the other representing a session initiated by the peer. If either of these sessions reaches active state first, then the other is discarded.

The data model for PCEP session presented in this document uses a flat list of sessions. Each session in the list is identified by its initiator. This index allows two sessions to exist transiently for a given peer, as discussed above.

There is only one list for the operational state of all sessions ("/pcep-state/entity/peers/peer/sessions/session").

5.4. Notifications

This YANG model defines a list of notifications to inform client of important events detected during the protocol operation. The notifications defined cover the PCEP MIB notifications.

6. Advanced PCE Features

This document contains a specification of the base PCEP YANG module, "ietf-pcep" which provides the basic PCEP [RFC5440] data model.

This document further handles advanced PCE features like -

- o Capability and Scope
- o Domain information (local/neighbour)
- o Path-Key
- o OF
- o GCO
- o P2MP
- o GMPLS
- o Inter-Layer
- o Stateful PCE
- o Segement Routing
- o Authentication including PCEPS (TLS)

[Editor’s Note - Some of them would be added in a future revision.]

6.1. Stateful PCE’s LSP-DB

In the operational state of PCEP which supports stateful PCE mode, the list of LSP state are maintained in LSP-DB. The key is the PLSP-ID and the PCC IP address.

The PCEP data model contains the operational state of LSPs (/pcep-state/entity/lsp-db/lsp/) with PCEP specific attributes. The generic TE attributes of the LSP are defined in [I-D.ietf-teas-yang-te]. A reference to LSP state in TE model is maintained.

7. Open Issues and Next Step

This section is added so that open issues can be tracked. This section would be removed when the document is ready for publication.

7.1. The PCE-Initiated LSP

The TE Model at [I-D.ietf-teas-yang-te] should support creationg of tunnels at the controller (PCE) and marking them as PCE-Initiated. The LSP-DB in the PCEP Yang (/pcep-state/entity/lsp-db/lsp/initiation) also marks the LSPs which are PCE-initiated.

7.2. PCEP over TLS (PCEPS)

A future version of this document would add TLS related configurations.

8. PCEP YANG Module

RFC Ed.: In this section, replace all occurrences of 'XXXX' with the actual RFC number and all occurrences of the revision date below with the date of RFC publication (and remove this note).

```
<CODE BEGINS> file "ietf-pcep@2016-07-07.yang"
module ietf-pcep {
    namespace "urn:ietf:params:xml:ns:yang:ietf-pcep";
    prefix pcep;

    import ietf-inet-types {
        prefix "inet";
    }

    import ietf-yang-types {
        prefix "yang";
    }

    import ietf-te {
        prefix "te";
    }

    import ietf-key-chain {
       prefix "key-chain";
```

```
    }


    organization
        "IETF PCE (Path Computation Element) Working Group";

    contact
        "WG Web:   <http://tools.ietf.org/wg/pce/>
         WG List:  <mailto:pce@ietf.org>
         WG Chair: JP Vasseur
                   <mailto:jpv@cisco.com>
         WG Chair: Julien Meuric
                   <mailto:julien.meuric@orange.com>
         WG Chair: Jonathan Hardwick
                   <mailto:Jonathan.Hardwick@metaswitch.com>
         Editor:   Dhruv Dhody
                   <mailto:dhruv.ietf@gmail.com>";

    description
        "The YANG module defines a generic configuration and
         operational model for PCEP common across all of the
         vendor implementations.";


    revision 2016-07-07 {
        description "Initial revision.";
        reference
            "RFC XXXX:  A YANG Data Model for Path Computation
                        Element Communications Protocol
                        (PCEP)";
    }

    /*
     * Identities
     */

    identity pcep {
        description "Identity for the PCEP protocol.";
    }

    /*
     * Typedefs
     */
    typedef pcep-role {
        type enumeration {
            enum unknown {
                value "0";
                description
```

```
                "An unknown role";
            }
            enum pcc {
                value "1";
                description
                "The role of a Path Computation Client";
            }
            enum pce {
                value "2";
                description
                "The role of Path Computation Element";
            }
            enum pcc-and-pce {
                value "3";
                description
                "The role of both Path Computation Client and
                 Path Computation Element";
            }
        }

        description
            "The role of a PCEP speaker.
             Takes one of the following values
             - unknown(0): the role is not known.
             - pcc(1): the role is of a Path Computation
               Client (PCC).
             - pce(2): the role is of a Path Computation
               Server (PCE).
             - pccAndPce(3): the role is of both a PCC and
               a PCE.";

    }

    typedef pcep-admin-status {
        type enumeration {
                enum admin-status-up {
                value "1";
                description
                "Admin Status is Up";
            }
            enum admin-status-down {
                value "2";
                description
                "Admin Status is Down";
            }
        }

        description
```

```
        "The Admin Status of the PCEP entity.
         Takes one of the following values
             - admin-status-up(1): Admin Status is Up.
             - admin-status-down(2): Admin Status is Down";
    }

    typedef pcep-oper-status {
        type enumeration {
            enum oper-status-up {
                value "1";
                description
                "The PCEP entity is active";
            }
            enum oper-status-down {
                value "2";
                description
                "The PCEP entity is inactive";
            }
            enum oper-status-going-up {
                value "3";
                description
                "The PCEP entity is activating";
            }
            enum oper-status-going-down {
                value "4";
                description
                "The PCEP entity is deactivating";
            }
            enum oper-status-failed {
                value "5";
                description
                "The PCEP entity has failed and will recover
                 when possible.";
            }
            enum oper-status-failed-perm {
                value "6";
                description
                "The PCEP entity has failed and will not recover
                 without operator intervention";
            }
        }
        description
        "The operational status of the PCEP entity.
         Takes one of the following values
             - oper-status-up(1): Active
             - oper-status-down(2): Inactive
             - oper-status-going-up(3): Activating
             - oper-status-going-down(4): Deactivating
```

```
             - oper-status-failed(5): Failed
             - oper-status-failed-perm(6): Failed Permanantly";
    }

    typedef pcep-initiator {
        type enumeration {
            enum local {
                value "1";
                description
                "The local PCEP entity initiated the session";
            }

            enum remote {
                value "2";
                description
                "The remote PCEP peer initiated the session";
            }
        }
        description
        "The initiator of the session, that is, whether the TCP
         connection was initiated by the local PCEP entity or
         the remote peer.
         Takes one of the following values
             - local(1): Initiated locally
             - remote(2): Initiated remotely";
    }

    typedef pcep-sess-state {
        type enumeration {
            enum tcp-pending {
                value "1";
                description
                    "The tcp-pending state of PCEP session.";
            }

            enum open-wait {
                value "2";
                description
                    "The open-wait state of PCEP session.";
            }

            enum keep-wait {
                value "3";
                description
                    "The keep-wait state of PCEP session.";
            }

            enum session-up {
```

```
                value "4";
                description
                    "The session-up state of PCEP session.";
            }
        }
        description
            "The current state of the session.
             The set of possible states excludes the idle state
             since entries do not exist in the idle state.
             Takes one of the following values
                - tcp-pending(1): PCEP TCP Pending state
                - open-wait(2): PCEP Open Wait state
                - keep-wait(3): PCEP Keep Wait state
                - session-up(4): PCEP Session Up state";
    }

    typedef domain-type {
        type enumeration {
            enum ospf-area {
                value "1";
                description
                    "The OSPF area.";
            }
            enum isis-area {
                value "2";
                description
                    "The IS-IS area.";
            }
            enum as {
                value "3";
                description
                    "The Autonomous System (AS).";
            }
        }
        description
            "The PCE Domain Type";
    }

    typedef domain-ospf-area {
        type union {
            type uint32;
            type yang:dotted-quad;
       }
       description
            "OSPF Area ID.";
    }

    typedef domain-isis-area {
```

```
        type string {
            pattern '[0-9A-Fa-f]{2}\.([0-9A-Fa-f]{4}\.){0,3}';
        }
        description
            "IS-IS Area ID.";
    }

    typedef domain-as {
        type uint32;
        description
            "Autonomous System number.";

    }

    typedef domain {
        type union {
            type domain-ospf-area;
            type domain-isis-area;
            type domain-as;
        }
        description
            "The Domain Information";
    }

    typedef operational-state {
        type enumeration {
            enum down {
                value "0";
                description
                    "not active.";
            }
            enum up {
                value "1";
                description
                    "signalled.";
            }
            enum active {
                value "2";
                description
                    "up and carrying traffic.";
            }
            enum going-down {
                value "3";
                description
                    "LSP is being torn down, resources are
                     being released.";
            }
            enum going-up {
```

```
                value "4";
                description
                    "LSP is being signalled.";
            }
        }
        description
            "The operational status of the LSP";
    }

    typedef lsp-error {
        type enumeration {
            enum no-error {
                value "0";
                description
                    "No error, LSP is fine.";
            }
            enum unknown {
                value "1";
                description
                    "Unknown reason.";
            }
            enum limit {
                value "2";
                description
                    "Limit reached for PCE-controlled LSPs.";
            }
            enum pending {
                value "3";
                description
                    "Too many pending LSP update requests.";
            }
            enum unacceptable {
                value "4";
                description
                    "Unacceptable parameters.";
            }
            enum internal {
                value "5";
                description
                    "Internal error.";
            }
            enum admin {
                value "6";
                description
                    "LSP administratively brought down.";
            }
            enum preempted {
                value "7";
```

```
                description
                    "LSP preempted.";
            }
            enum rsvp {
                value "8";
                description
                    "RSVP signaling error.";
            }
        }
        description
            "The LSP Error Codes.";
    }

    typedef sync-state {
        type enumeration {
            enum pending {
                value "0";
                description
                    "The state synchronization
                     has not started.";
            }
            enum ongoing {
                value "1";
                description
                    "The state synchronization
                     is ongoing.";
            }
            enum finished {
                value "2";
                description
                    "The state synchronization
                     is finished.";
            }
        }
        description
            "The LSP-DB state synchronization operational status.";
    }

        typedef pst{
                type enumeration{
                        enum rsvp-te{
                                value "0";
                                description
                                        "RSVP-TE signaling protocol";
                        }
                        enum sr{
                                value "1";
                                description
```

```
                                        "Segment Routing Traffic Engineering";
                        }
                }
                description
                        "The Path Setup Type";
        }

    typedef assoc-type{
        type enumeration{
            enum protection{
                value "1";
                description
                    "Path Protection Association Type";
            }
        }
        description
            "The PCEP Association Type";
    }

    /*
     * Features
     */

    feature svec {
        description
            "Support synchronized path computation.";
    }

    feature gmpls {
        description
            "Support GMPLS.";
    }

    feature obj-fn {
        description
            "Support OF as per RFC 5541.";
    }

    feature gco {
        description
            "Support GCO as per RFC 5557.";
    }

    feature pathkey {
        description
            "Support pathkey as per RFC 5520.";
    }
```

```
    feature p2mp {
        description
            "Support P2MP as per RFC 6006.";
    }

    feature stateful {
        description
            "Support stateful PCE.";
    }

    feature pce-initiated {
        description
            "Support PCE-Initiated LSP.";
    }

    feature tls {
        description
            "Support PCEP over TLS.";
    }

    feature sr {
        description
            "Support Segement Routing for PCE.";
    }

    /*
     * Groupings
     */



    grouping pcep-entity-info{
        description
            "This grouping defines the attributes for PCEP entity.";
        leaf connect-timer {
            type uint32 {
                range "1..65535";
            }
            units "seconds";
            default 60;
            description
                "The time in seconds that the PCEP entity will wait
                 to establish a TCP connection with a peer.  If a
                 TCP connection is not established within this time
                 then PCEP aborts the session setup attempt.";
            reference
                "RFC 5440: Path Computation Element (PCE)
                           Communication Protocol (PCEP)";
```

```
        }

        leaf connect-max-retry {
            type uint32;
            default 5;
            description
                "The maximum number of times the system tries to
                 establish a TCP connection to a peer before the
                 session with the peer transitions to the idle
                 state.";
            reference
                "RFC 5440: Path Computation Element (PCE)
                           Communication Protocol (PCEP)";
        }

        leaf init-backoff-timer {
            type uint32 {
                range "1..65535";
            }
            units "seconds";
            description
              "The initial back-off time in seconds for retrying
                a failed session setup attempt to a peer.
                The back-off time increases for each failed
                session setup attempt, until a maximum back-off
                time is reached.  The maximum back-off time is
                max-backoff-timer.";
        }

        leaf max-backoff-timer {
            type uint32;
            units "seconds";
            description
              "The maximum back-off time in seconds for retrying
                a failed session setup attempt to a peer.
                The back-off time increases for each failed session
                setup attempt, until this maximum value is reached.
                Session setup attempts then repeat periodically
                without any further increase in back-off time.";
        }

        leaf open-wait-timer {
            type uint32 {
                range "1..65535";
            }
            units "seconds";
            default 60;
            description
```

```
               "The time in seconds that the PCEP entity will wait
                to receive an Open message from a peer after the
                TCP connection has come up.
                If no Open message is received within this time then
                PCEP terminates the TCP connection and deletes the
                associated sessions.";
            reference
                "RFC 5440: Path Computation Element (PCE)
                           Communication Protocol (PCEP)";
        }

        leaf keep-wait-timer {
            type uint32 {
                range "1..65535";
            }
            units "seconds";
            default 60;
            description
                "The time in seconds that the PCEP entity will wait
                 to receive a Keepalive or PCErr message from a peer
                 during session initialization after receiving an
                 Open message.  If no Keepalive or PCErr message is
                 received within this time then PCEP terminates the
                 TCP connection and deletes the associated
                 sessions.";
            reference
                "RFC 5440: Path Computation Element (PCE)
                           Communication Protocol (PCEP)";
        }

        leaf keep-alive-timer {
            type uint32 {
                range "0..255";
            }
            units "seconds";
            default 30;
            description
                "The keep alive transmission timer that this PCEP
                 entity will propose in the initial OPEN message of
                 each session it is involved in.  This is the
                 maximum time between two consecutive messages sent
                 to a peer. Zero means that the PCEP entity prefers
                 not to send Keepalives at all.
                 Note that the actual Keepalive transmission
                 intervals, in either direction of an active PCEP
                 session, are determined by negotiation between the
                 peers as specified by RFC 5440, and so may differ
                 from this configured value.";
```

```
            reference
                "RFC 5440: Path Computation Element (PCE)
                           Communication Protocol (PCEP)";
        }

        leaf dead-timer {
            type uint32 {
                range "0..255";
            }
            units "seconds";
            must ". >= ../keep-alive-timer" {
                error-message "The dead timer must be "
                       + "larger than the keep alive timer";
                description
                    "This value MUST be greater than
                     keep-alive-timer.";

            }
            default 120;
            description
                "The dead timer that this PCEP entity will propose
                 in the initial OPEN message of each session it is
                 involved in. This is the time after which a peer
                 should declare a session down if it does not
                 receive any PCEP messages. Zero suggests that the
                 peer does not run a dead timer at all." ;
            reference
                "RFC 5440: Path Computation Element (PCE)
                           Communication Protocol (PCEP)";
        }

        leaf allow-negotiation{
            type boolean;
            description
                "Whether the PCEP entity will permit negotiation of
                 session parameters.";
        }

        leaf max-keep-alive-timer{
            type uint32 {
                range "0..255";
            }
            units "seconds";
            description
                "In PCEP session parameter negotiation in seconds,
                 the maximum value that this PCEP entity will
                 accept from a peer for the interval between
                 Keepalive transmissions. Zero means that the PCEP
```

```
                  entity will allow no Keepalive transmission at
                  all." ;
        }

        leaf max-dead-timer{
            type uint32 {
                range "0..255";
            }
            units "seconds";
            description
                "In PCEP session parameter negotiation in seconds,
                 the maximum value that this PCEP entity will accept
                 from a peer for the Dead timer.  Zero means that
                 the PCEP entity will allow not running a Dead
                 timer.";
        }

        leaf min-keep-alive-timer{
            type uint32 {
                range "0..255";
            }
            units "seconds";
            description
                "In PCEP session parameter negotiation in seconds,
                 the minimum value that this PCEP entity will
                 accept for the interval between Keepalive
                 transmissions. Zero means that the PCEP entity
                 insists on no Keepalive transmission at all.";
        }

        leaf min-dead-timer{
            type uint32 {
                range "0..255";
            }
            units "seconds";
            description
                 "In PCEP session parameter negotiation in
                  seconds, the minimum value that this PCEP entity
                  will accept for the Dead timer.  Zero means that
                  the PCEP entity insists on not running a Dead
                  timer.";
        }

        leaf sync-timer{
            if-feature svec;
            type uint32 {
                range "0..65535";
            }
```

```
            units "seconds";
            default 60;
            description
                "The value of SyncTimer in seconds is used in the
                 case of synchronized path computation request
                 using the SVEC object. Consider the case where a
                 PCReq message is received by a PCE that contains
                 the SVEC object referring to M synchronized path
                 computation requests.  If after the expiration of
                 the SyncTimer all the M path computation requests
                 have not been, received a protocol error is
                 triggered and the PCE MUST cancel the whole set
                 of path computation  requests.
                 The aim of the SyncTimer is to avoid the storage
                 of unused synchronized requests should one of
                 them get lost for some reasons (for example, a
                 misbehaving PCC).
                 Zero means that the PCEP entity does not use the
                 SyncTimer.";
            reference
                "RFC 5440: Path Computation Element (PCE)
                           Communication Protocol (PCEP)";
        }


        leaf request-timer{
            type uint32 {
                range "1..65535";
            }
            units "seconds";
            description
                "The maximum time that the PCEP entity will wait
                 for a response to a PCReq message.";
        }

        leaf max-sessions{
            type uint32;
            description
               "Maximum number of sessions involving this PCEP
                entity that can exist at any time.";
        }

        leaf max-unknown-reqs{
            type uint32;
            default 5;
            description
              "The maximum number of unrecognized requests and
               replies that any session on this PCEP entity is
```

```
               willing to accept per minute before terminating
               the session.
               A PCRep message contains an unrecognized reply
               if it contains an RP object whose request ID
               does not correspond to any in-progress request
               sent by this PCEP entity.
               A PCReq message contains an unrecognized request
               if it contains an RP object whose request ID is
               zero.";
            reference
                "RFC 5440: Path Computation Element (PCE)
                           Communication Protocol (PCEP)";
        }

        leaf max-unknown-msgs{
            type uint32;
            default 5;
            description
             "The maximum number of unknown messages that any
              session on this PCEP entity is willing to accept
              per minute before terminating the session.";
            reference
                "RFC 5440: Path Computation Element (PCE)
                           Communication Protocol (PCEP)";
        }

    }//pcep-entity-info

    grouping  pce-scope{
        description
            "This grouping defines PCE path computation scope
             information which maybe relevant to PCE selection.
             This information corresponds to PCE auto-discovery
             information.";
        reference
            "RFC 5088: OSPF Protocol Extensions for Path
                       Computation Element (PCE)
                       Discovery
             RFC 5089: IS-IS Protocol Extensions for Path
                       Computation Element (PCE)
                       Discovery";
        leaf intra-area-scope{
            type boolean;
            default true;
            description
                "PCE can compute intra-area paths.";
        }
        leaf intra-area-pref{
```

```
            type uint8{
                range "0..7";
            }
            description
              "The PCE’s preference for intra-area TE LSP
              computation.";
        }
        leaf inter-area-scope{
            type boolean;
            default false;
            description
                "PCE can compute inter-area paths.";
        }
        leaf inter-area-scope-default{
            type boolean;
            default false;
            description
                "PCE can act as a default PCE for inter-area
                 path computation.";
        }
        leaf inter-area-pref{
            type uint8{
                range "0..7";
            }
            description
              "The PCE’s preference for inter-area TE LSP
              computation.";
        }
        leaf inter-as-scope{
            type boolean;
            default false;
            description
                "PCE can compute inter-AS paths.";
        }
        leaf inter-as-scope-default{
            type boolean;
            default false;
            description
                "PCE can act as a default PCE for inter-AS
                 path computation.";
        }
        leaf inter-as-pref{
            type uint8{
                range "0..7";
            }
            description
              "The PCE’s preference for inter-AS TE LSP
              computation.";
```

```
        }
        leaf inter-layer-scope{
            type boolean;
            default false;
            description
                "PCE can compute inter-layer paths.";
        }
        leaf inter-layer-pref{
            type uint8{
                range "0..7";
            }
            description
              "The PCE’s preference for inter-layer TE LSP
              computation.";
        }
    }//pce-scope

    grouping  domain{
        description
            "This grouping specifies a Domain where the
             PCEP speaker has topology visibility.";
        leaf domain-type{
            type domain-type;
            description
              "The domain type.";
        }
        leaf domain{
            type domain;
            description
              "The domain Information.";
        }
    }//domain

    grouping capability{
        description
            "This grouping specifies a capability
             information of local PCEP entity. This maybe
             relevant to PCE selection as well. This
             information corresponds to PCE auto-discovery
             information.";
        reference
            "RFC 5088: OSPF Protocol Extensions for Path
                       Computation Element (PCE)
                       Discovery
             RFC 5089: IS-IS Protocol Extensions for Path
                       Computation Element (PCE)
                       Discovery";
        leaf gmpls{
```

```
            if-feature gmpls;
            type boolean;
            description
              "Path computation with GMPLS link
               constraints.";
        }
        leaf bi-dir{
            type boolean;
            description
              "Bidirectional path computation.";
        }
        leaf diverse{
            type boolean;
            description
              "Diverse path computation.";
        }
        leaf load-balance{
            type boolean;
            description
              "Load-balanced path computation.";
        }
        leaf synchronize{
            if-feature svec;
            type boolean;
            description
              "Synchronized paths computation.";
        }
        leaf objective-function{
            if-feature obj-fn;
            type boolean;
            description
              "Support for multiple objective functions.";
        }
        leaf add-path-constraint{
            type boolean;
            description
              "Support for additive path constraints (max
               hop count, etc.).";
        }
        leaf prioritization{
            type boolean;
            description
              "Support for request prioritization.";
        }
        leaf multi-request{
            type boolean;
            description
              "Support for multiple requests per message.";
```

```
        }
        leaf gco{
            if-feature gco;
            type boolean;
            description
              "Support for Global Concurrent Optimization
               (GCO).";
        }
        leaf p2mp{
            if-feature p2mp;
            type boolean;
            description
              "Support for P2MP path computation.";
        }

        container stateful{
            if-feature stateful;
            description
                "If stateful PCE feature is present";
            leaf enabled{
                type boolean;
                description
                    "Enabled or Disabled";
            }
            leaf active{
                type boolean;
                description
                  "Support for active stateful PCE.";
            }
            leaf pce-initiated{
                if-feature pce-initiated;
                type boolean;
                description
                  "Support for PCE-initiated LSP.";
            }
        }
                container sr{
                        if-feature sr;
                        description
                                "If segment routing is supported";
                        leaf enabled{
                type boolean;
                description
                    "Enabled or Disabled";
                        }
                        leaf msd{ /*should be in MPLS yang model (?)*/
                                type uint8;
                must "((../../role == 'pcc')" +
```

```
                     " or " +
                    "(../../role == 'pcc-and-pce')))"
                {
                    error-message
                        "The PCEP entity must be PCC";
                    description
                        "When PCEP entity is PCC for
                        MSD to be applicable";
                }
                                description
                                        "Maximum SID Depth";
                        }
                }
    }//capability

    grouping  info{
        description
            "This grouping specifies all information which
             maybe relevant to both PCC and PCE.
             This information corresponds to PCE auto-discovery
             information.";
        container domain{
            description
                "The local domain for the PCEP entity";
            list domain{
                key "domain-type domain";
                description
                    "The local domain.";
                uses domain{
                    description
                        "The local domain for the PCEP entity.";
                }
            }
        }
        container capability{
            description
                "The PCEP entity capability";
            uses capability{
                description
                    "The PCEP entity supported
                    capabilities.";
            }
        }
    }//info

    grouping  pce-info{
        description
            "This grouping specifies all PCE information
```

```
             which maybe relevant to the PCE selection.
             This information corresponds to PCE auto-discovery
             information.";
        container scope{
            description
                "The path computation scope";
            uses pce-scope;
        }

        container neigh-domains{
            description
                "The list of neighbour PCE-Domain
                 toward which a PCE can compute
                 paths";
            list domain{
                key "domain-type domain";

                description
                    "The neighbour domain.";
                uses domain{
                    description
                        "The PCE neighbour domain.";
                }
            }
        }
    }//pce-info

    grouping  pcep-stats{
        description
            "This grouping defines statistics for PCEP. It is used
             for both peer and current session.";
        leaf avg-rsp-time{
            type uint32;
            units "milliseconds";
            must "(/pcep-state/entity/peers/peer/role != 'pcc'" +
                 " or " +
                 "(/pcep-state/entity/peers/peer/role = 'pcc'" +
                 " and avg-rsp-time = 0))" {
                error-message
                    "Invalid average response time";
                description
                    "If role is pcc then this leaf is meaningless
                     and is set to zero.";
            }
            description
              "The average response time.
               If an average response time has not been
               calculated then this leaf has the value zero.";
```

```
        }

        leaf lwm-rsp-time{
            type uint32;
            units "milliseconds";
            must "(/pcep-state/entity/peers/peer/role != 'pcc'" +
                 " or " +
                 "(/pcep-state/entity/peers/peer/role = 'pcc'" +
                 " and lwm-rsp-time = 0))" {
                error-message
                    "Invalid smallest (low-water mark)
                     response time";
                description
                    "If role is pcc then this leaf is meaningless
                     and is set to zero.";
            }
            description
             "The smallest (low-water mark) response time seen.
              If no responses have been received then this
              leaf has the value zero.";
        }

        leaf hwm-rsp-time{
            type uint32;
            units "milliseconds";
            must "(/pcep-state/entity/peers/peer/role != 'pcc'" +
                 " or " +
                 "(/pcep-state/entity/peers/peer/role = 'pcc'" +
                 " and hwm-rsp-time = 0))" {
                error-message
                    "Invalid greatest (high-water mark)
                     response time seen";
                description
                    "If role is pcc then this field is
                     meaningless and is set to zero.";
            }
            description
             "The greatest (high-water mark) response time seen.
              If no responses have been received then this object
              has the value zero.";
        }

        leaf num-pcreq-sent{
            type yang:counter32;
            description
             "The number of PCReq messages sent.";
        }
```

```
        leaf num-pcreq-rcvd{
            type yang:counter32;
            description
              "The number of PCReq messages received.";
        }

        leaf num-pcrep-sent{
            type yang:counter32;
            description
              "The number of PCRep messages sent.";
        }

        leaf num-pcrep-rcvd{
            type yang:counter32;
            description
              "The number of PCRep messages received.";
        }

        leaf num-pcerr-sent{
            type yang:counter32;
            description
              "The number of PCErr messages sent.";
        }

        leaf num-pcerr-rcvd{
            type yang:counter32;
            description
              "The number of PCErr messages received.";
        }

        leaf num-pcntf-sent{
            type yang:counter32;
            description
              "The number of PCNtf messages sent.";
        }

        leaf num-pcntf-rcvd{
            type yang:counter32;
            description
              "The number of PCNtf messages received.";
        }

        leaf num-keepalive-sent{
            type yang:counter32;
            description
              "The number of Keepalive messages sent.";
        }
```

```
        leaf num-keepalive-rcvd{
            type yang:counter32;
            description
              "The number of Keepalive messages received.";
        }

        leaf num-unknown-rcvd{
            type yang:counter32;
            description
              "The number of unknown messages received.";
        }

        leaf num-corrupt-rcvd{
            type yang:counter32;
            description
              "The number of corrupted PCEP message received.";
        }

        leaf num-req-sent{
            type yang:counter32;
            description
              "The number of requests sent.  A request corresponds
               1:1 with an RP object in a PCReq message. This might
               be greater than num-pcreq-sent because multiple
               requests can be batched into a single PCReq
               message.";
        }

        leaf num-req-sent-pend-rep{
            type yang:counter32;
            description
              "The number of requests that have been sent for
               which a response is still pending.";
        }

        leaf num-req-sent-ero-rcvd{
            type yang:counter32;
            description
               "The number of requests that have been sent for
                which a response with an ERO object was received.
                Such responses indicate that a path was
                successfully computed by the peer.";
        }

        leaf num-req-sent-nopath-rcvd{
            type yang:counter32;
            description
              "The number of requests that have been sent for
```

```
               which a response with a NO-PATH object was
               received. Such responses indicate that the peer
               could not  find a path to satisfy the
               request.";
        }

        leaf num-req-sent-cancel-rcvd{
            type yang:counter32;
            description
              "The number of requests that were cancelled with
               a PCNtf message.
               This might be different than num-pcntf-rcvd because
               not all PCNtf messages are used to cancel requests,
               and a single PCNtf message can cancel multiple
               requests.";
        }

        leaf num-req-sent-error-rcvd{
            type yang:counter32;
            description
             "The number of requests that were rejected with a
              PCErr message.
              This might be different than num-pcerr-rcvd because
              not all PCErr messages are used to reject requests,
              and a single PCErr message can reject multiple
              requests.";
        }

        leaf num-req-sent-timeout{
            type yang:counter32;
            description
             "The number of requests that have been sent to a peer
             and have been abandoned because the peer has taken too
             long to respond to them.";
        }

        leaf num-req-sent-cancel-sent{
            type yang:counter32;
            description
             "The number of requests that were sent to the peer and
              explicitly cancelled by the local PCEP entity sending
              a PCNtf.";
        }

        leaf num-req-rcvd{
            type yang:counter32;
            description
             "The number of requests received.  A request
```

```
              corresponds 1:1 with an RP object in a PCReq
              message.
              This might be greater than num-pcreq-rcvd because
              multiple requests can be batched into a single
              PCReq message.";
        }

        leaf num-req-rcvd-pend-rep{
            type yang:counter32;
            description
             "The number of requests that have been received for
              which a response is still pending.";
        }

        leaf num-req-rcvd-ero-sent{
            type yang:counter32;
            description
             "The number of requests that have been received for
              which a response with an ERO object was sent.  Such
              responses indicate that a path was successfully
              computed by the local PCEP entity.";
        }

        leaf num-req-rcvd-nopath-sent{
            type yang:counter32;
            description
             "The number of requests that have been received for
              which a response with a NO-PATH object was sent. Such
              responses indicate that the local PCEP entity could
              not find a path to satisfy the request.";
        }

        leaf num-req-rcvd-cancel-sent{
            type yang:counter32;
            description
             "The number of requests received that were cancelled
              by the local PCEP entity sending a PCNtf message.
              This might be different than num-pcntf-sent because
              not all PCNtf messages are used to cancel requests,
              and a single PCNtf message can cancel multiple
              requests.";
        }

        leaf num-req-rcvd-error-sent{
            type yang:counter32;
            description
             "The number of requests received that were cancelled
              by the local PCEP entity sending a PCErr message.
```

```
              This might be different than num-pcerr-sent because
              not all PCErr messages are used to cancel requests,
              and a single PCErr message can cancel multiple
              requests.";
        }

        leaf  num-req-rcvd-cancel-rcvd{
            type yang:counter32;
            description
             "The number of requests that were received from the
              peer and explicitly cancelled by the peer sending
              a PCNtf.";
        }

        leaf  num-rep-rcvd-unknown{
            type yang:counter32;
            description
              "The number of responses to unknown requests
               received. A response to an unknown request is a
               response whose RP object does not contain the
               request ID of any request that is currently
               outstanding on the session.";
        }

        leaf  num-req-rcvd-unknown{
            type yang:counter32;
            description
              "The number of unknown requests that have been
               received. An unknown request is a request
               whose RP object contains a request ID of
               zero.";
        }

        container svec{
            if-feature svec;
            description
                "If synchronized path computation is supported";
            leaf num-svec-sent{
                type yang:counter32;
                description
                  "The number of SVEC objects sent in PCReq messages.
                   An SVEC object represents a set of synchronized
                   requests.";
            }

            leaf num-svec-req-sent{
                type yang:counter32;
                description
```

```
                  "The number of requests sent that appeared in one
                   or more SVEC objects.";
            }

            leaf num-svec-rcvd{
                type yang:counter32;
                description
                 "The number of SVEC objects received in PCReq
                  messages. An SVEC object represents a set of
                  synchronized requests.";
            }

            leaf num-svec-req-rcvd{
                type yang:counter32;
                description
                  "The number of requests received that appeared
                   in one or more SVEC objects.";
            }
        }
        container stateful{
            if-feature stateful;
            description
                "Stateful PCE related statistics";
            leaf num-pcrpt-sent{
                type yang:counter32;
                description
                 "The number of PCRpt messages sent.";
            }

            leaf num-pcrpt-rcvd{
                type yang:counter32;
                description
                  "The number of PCRpt messages received.";
            }

            leaf num-pcupd-sent{
                type yang:counter32;
                description
                 "The number of PCUpd messages sent.";
            }

            leaf num-pcupd-rcvd{
                type yang:counter32;
                description
                  "The number of PCUpd messages received.";
            }

            leaf num-rpt-sent{
```

```
                type yang:counter32;
                description
                  "The number of LSP Reports sent.  A LSP report
                   corresponds 1:1 with an LSP object in a PCRpt
                   message. This might be greater than
                   num-pcrpt-sent because multiple reports can
                   be batched into a single PCRpt message.";
            }

            leaf num-rpt-rcvd{
                type yang:counter32;
                description
                 "The number of LSP Reports received.  A LSP report
                  corresponds 1:1 with an LSP object in a PCRpt
                  message.
                  This might be greater than num-pcrpt-rcvd because
                  multiple reports can be batched into a single
                  PCRpt message.";
            }

            leaf  num-rpt-rcvd-error-sent{
                type yang:counter32;
                description
                  "The number of reports of LSPs received that were
                  responded by the local PCEP entity by sending a
                  PCErr message.";
            }

            leaf num-upd-sent{
                type yang:counter32;
                description
                  "The number of LSP updates sent.  A LSP update
                   corresponds 1:1 with an LSP object in a PCUpd
                   message. This might be greater than
                   num-pcupd-sent because multiple updates can
                   be batched into a single PCUpd message.";
            }

            leaf num-upd-rcvd{
                type yang:counter32;
                description
                 "The number of LSP Updates received.  A LSP update
                  corresponds 1:1 with an LSP object in a PCUpd
                  message.
                  This might be greater than num-pcupd-rcvd because
                  multiple updates can be batched into a single
                  PCUpd message.";
            }
```

```
            leaf  num-upd-rcvd-unknown{
                type yang:counter32;
                description
                  "The number of updates to unknown LSPs
                   received. An update to an unknown LSP is a
                   update whose LSP object does not contain the
                   PLSP-ID of any LSP that is currently
                   present.";
            }

            leaf  num-upd-rcvd-undelegated{
                type yang:counter32;
                description
                  "The number of updates to not delegated LSPs
                   received. An update to an undelegated LSP is a
                   update whose LSP object does not contain the
                   PLSP-ID of any LSP that is currently
                   delegated to current PCEP session.";
            }

            leaf  num-upd-rcvd-error-sent{
                type yang:counter32;
                description
                  "The number of updates to LSPs received that were
                  responded by the local PCEP entity by sending a
                  PCErr message.";
            }
            container initiation {
                if-feature pce-initiated;
                description
                    "PCE-Initiated related statistics";
                leaf num-pcinitiate-sent{
                    type yang:counter32;
                    description
                     "The number of PCInitiate messages sent.";
                }

                leaf num-pcinitiate-rcvd{
                    type yang:counter32;
                    description
                      "The number of PCInitiate messages received.";
                }

                leaf num-initiate-sent{
                    type yang:counter32;
                    description
                      "The number of LSP Initiation sent via PCE.
                       A LSP initiation corresponds 1:1 with an LSP
```

```
                          object in a PCInitiate message. This might be
                          greater than num-pcinitiate-sent because
                          multiple initiations can be batched into a
                          single PCInitiate message.";
                }

                leaf num-initiate-rcvd{
                    type yang:counter32;
                    description
                      "The number of LSP Initiation received from
                       PCE.  A LSP initiation corresponds 1:1 with
                       an LSP object in a PCInitiate message. This
                       might be greater than num-pcinitiate-rcvd
                       because multiple initiations can be batched
                       into a single PCInitiate message.";
                }

                leaf  num-initiate-rcvd-error-sent{
                    type yang:counter32;
                    description
                      "The number of initiations of LSPs received
                       that were responded by the local PCEP entity
                       by sending a PCErr message.";
                }
            }
        }
    }//pcep-stats

    grouping lsp-state{
        description
            "This grouping defines the attributes for LSP in LSP-DB.
             These are the attributes specifically from the PCEP
             perspective";
        leaf plsp-id{
            type uint32{
                range "1..1048575";
            }
            description
                "A PCEP-specific identifier for the LSP.  A PCC
                 creates a unique PLSP-ID for each LSP that is
                 constant for the lifetime of a PCEP session.
                 PLSP-ID is 20 bits with 0 and 0xFFFFF are
                 reserved";
        }
        leaf pcc-id{
            type inet:ip-address;
            description
                "The local internet address of the PCC, that
```

```
                 generated the PLSP-ID.";
        }

        container lsp-ref{
            description
                "reference to ietf-te lsp state";

            leaf source {
                type leafref {
                    path "/te:te/te:lsps-state/te:lsp/te:source";
                }
                description
                  "Tunnel sender address extracted from
                  SENDER_TEMPLATE  object";
                reference "RFC3209";
            }
            leaf destination {
                type leafref {
                    path "/te:te/te:lsps-state/te:lsp/te:"
                         + "destination";
                }
                description
                    "Tunnel endpoint address extracted from
                    SESSION object";
                reference "RFC3209";
            }
            leaf tunnel-id {
                type leafref {
                    path "/te:te/te:lsps-state/te:lsp/te:tunnel-id";
                }
                description
                    "Tunnel identifier used in the SESSION
                    that remains constant over the life
                    of the tunnel.";
                reference "RFC3209";
            }
            leaf lsp-id {
                type leafref {
                    path "/te:te/te:lsps-state/te:lsp/te:lsp-id";
                }
                description
                    "Identifier used in the SENDER_TEMPLATE
                    and the FILTER_SPEC that can be changed
                    to allow a sender to share resources with
                    itself.";
                reference "RFC3209";
            }
            leaf extended-tunnel-id {
```

```
                type leafref {
                    path "/te:te/te:lsps-state/te:lsp/te:"
                    + "extended-tunnel-id";
                }
                description
                    "Extended Tunnel ID of the LSP.";
                reference "RFC3209";
            }
            leaf type {
                type leafref {
                    path "/te:te/te:lsps-state/te:lsp/te:type";
                }
                description "LSP type P2P or P2MP";
            }
        }

        leaf admin-state{
            type boolean;
            description
                "The desired operational state";
        }
        leaf operational-state{
            type operational-state;
            description
                "The operational status of the LSP";
        }
        container delegated{
            description
                "The delegation related parameters";
            leaf enabled{
                type boolean;
                description
                    "LSP is delegated or not";
            }
            leaf pce{
                type leafref {
                    path "/pcep-state/entity/peers/peer/addr";
                }
                must "((../enabled == true)" +
                     " and " +
                     "((../../role == 'pcc')" +
                     " or " +
                    "(../../role == 'pcc-and-pce')))"
                {
                    error-message
                        "The PCEP entity must be PCC
                         and the LSP be delegated";
                    description
```

```
                        "When PCEP entity is PCC for
                        delegated LSP";
                }
                description
                    "The reference to the PCE peer to
                    which LSP is delegated";
            }
            leaf srp-id{
                type uint32;
                description
                    "The last SRP-ID-number associated with this
                    LSP.";
            }
        }
        container initiation {
            if-feature pce-initiated;
            description
                "The PCE initiation related parameters";
            leaf enabled{
                type boolean;
                description
                    "LSP is PCE-initiated or not";
            }
            leaf pce{
                type leafref {
                    path "/pcep-state/entity/peers/peer/addr";
                }
                must "(../enabled == true)"
                {
                    error-message
                        "The LSP must be PCE-Initiated";
                    description
                        "When the LSP must be PCE-Initiated";
                }
                description
                    "The reference to the PCE
                    that initiated this LSP";
            }
        }
        leaf symbolic-path-name{
            type string;
            description
                "The symbolic path name associated with the LSP.";
        }
        leaf last-error{
            type lsp-error;
            description
                "The last error for the LSP.";
```

```
        }
                leaf pst{
                        type pst;
                        default "rsvp-te";
                        description
                                "The Path Setup Type";

                }

    }//lsp-state

    grouping notification-instance-hdr {
        description
            "This group describes common instance specific data
             for notifications.";

        leaf peer-addr {
            type leafref {
                path "/pcep-state/entity/peers/peer/addr";
            }
            description
                "Reference to peer address";
        }

    }// notification-instance-hdr

    grouping notification-session-hdr {
        description
        "This group describes common session instance specific
         data for notifications.";

        leaf session-initiator {
            type leafref {
            path "/pcep-state/entity/peers/peer/sessions/" +
                 "session/initiator";
            }
            description
                "Reference to pcep session initiator leaf";
        }
    }// notification-session-hdr

    grouping stateful-pce-parameter {
        description
        "This group describes stateful PCE specific
         parameters.";
        leaf state-timeout{
            type uint32;
            units "seconds";
```

```
            description
                "When a PCEP session is terminated, a PCC
                 waits for this time period before flushing
                 LSP state associated with that PCEP session
                 and reverting to operator-defined default
                 parameters or behaviours.";
        }
        leaf redelegation-timeout{
            type uint32;
            units "seconds";
            must "((../role == 'pcc')" +
                 " or " +
                 "(../role == 'pcc-and-pce'))"
            {
                error-message "The PCEP entity must be PCC";
                description
                    "When PCEP entity is PCC";
            }
            description
                "When a PCEP session is terminated, a PCC
                 waits for this time period before revoking
                 LSP delegation to a PCE and attempting to
                 redelegate LSPs associated with the
                 terminated PCEP session to an alternate
                 PCE.";
        }
        leaf rpt-non-pcep-lsp{
            type boolean;
            must "((../role == 'pcc')" +
                 " or " +
                 "(../role == 'pcc-and-pce'))"
            {
                error-message "The PCEP entity must be PCC";
                description
                    "When PCEP entity is PCC";
            }
            description
                "If set, a PCC reports LSPs that are not
                controlled by any PCE (for example, LSPs
                that are statically configured at the
                PCC). ";
        }

    }

    grouping authentication {
        description "Authentication Information";
        choice auth-type-selection {
```

```
            description
                "Options for expressing authentication setting.";
            case auth-key-chain {
                leaf key-chain {
                    type key-chain:key-chain-ref;
                    description
                        "key-chain name.";
                }
            }
            case auth-key {
                leaf key {
                    type string;
                description
                    "Key string in ASCII format.";
                }
                container crypto-algorithm {
                    uses key-chain:crypto-algorithm-types;
                        description
                            "Cryptographic algorithm associated
                             with key.";
                }
            }
            case auth-tls {
                if-feature tls;
                container tls {
                    description
                        "TLS related information - TBD";
                }
            }
        }
    }

    grouping association {
        description
            "Generic Association parameters";
        leaf type {
            type "assoc-type";
            description
                "The PCEP association type";
        }
        leaf id {
            type uint16;
            description
                "PCEP Association ID";
        }
        leaf source {
          type inet:ip-address;
          description
```

```
                "PCEP Association Source.";
        }
        leaf global-source {
          type uint32;
          description
                "PCEP Association Global
                Source.";
        }
        leaf extended-id{
            type string;
            description
                "Additional information to
                support unique identification.";
        }
    }
    grouping association-ref {
        description
            "Generic Association parameters";
        leaf id {
            type leafref {
                path "/pcep-state/entity/lsp-db/"
                + "association-list/id";
            }
            description
                "PCEP Association ID";
        }
        leaf source {
            type leafref {
                path "/pcep-state/entity/lsp-db/"
                + "association-list/source";
            }
          description
                "PCEP Association Source.";
        }
        leaf global-source {
            type leafref {
                path "/pcep-state/entity/lsp-db/"
                + "association-list/global-source";
            }
          description
                "PCEP Association Global
                Source.";
        }
        leaf extended-id{
            type leafref {
                path "/pcep-state/entity/lsp-db/"
                + "association-list/extended-id";
            }
```

```
            description
                "Additional information to
                support unique identification.";
        }
    }
    /*
     * Configuration data nodes
     */
    container pcep{

        presence
            "The PCEP is enabled";

            description
            "Parameters for list of configured PCEP entities
             on the device.";

        container entity {

            description
                "The configured PCEP entity on the device.";

            leaf addr {
                type inet:ip-address;
                mandatory true;
                description
                    "The local Internet address of this PCEP
                    entity.
                    If operating as a PCE server, the PCEP
                    entity listens on this address.
                    If operating as a PCC, the PCEP entity
                    binds outgoing TCP connections to this
                    address.
                    It is possible for the PCEP entity to
                    operate both as a PCC and a PCE Server, in
                    which case it uses this address both to
                    listen for incoming TCP connections and to
                    bind outgoing TCP connections.";
            }

            leaf enabled {
                type boolean;
                default true;
                description
                    "The administrative status of this PCEP
                     Entity.";
            }
```

```
            leaf role {
                type pcep-role;
                mandatory true;
                description
                    "The role that this entity can play.
                     Takes one of the following values.
                     - unknown(0): this PCEP Entity role is not
                       known.
                     - pcc(1): this PCEP Entity is a PCC.
                     - pce(2): this PCEP Entity is a PCE.
                     - pcc-and-pce(3): this PCEP Entity is both
                       a PCC and a PCE.";
            }

            leaf description {
                type string;
                description
                    "Description of the PCEP entity configured
                     by the user";
            }

            uses info {
                description
                    "Local PCEP entity information";
            }


            container pce-info {
                must "((../role == ’pce’)" +
                     " or " +
                     "(../role == ’pcc-and-pce’))"
                {
                    error-message "The PCEP entity must be PCE";
                    description
                        "When PCEP entity is PCE";
                }
                uses pce-info {
                    description
                        "Local PCE information";
                }
                                uses authentication {
                                        description
                                                "Local PCE authentication inform
ation";
                        }


                description
                    "The Local PCE Entity PCE information";
```

```
            }

            uses pcep-entity-info {
                description
                    "The configuration related to the PCEP
                     entity.";
            }



            leaf pcep-notification-max-rate {
                type uint32;
                mandatory true;
                description
                    "This variable indicates the maximum number of
                     notifications issued per second. If events occur
                     more rapidly, the implementation may simply fail
                     to emit these notifications during that period,
                     or may queue them until an appropriate time. A
                     value of 0 means no notifications are emitted
                     and all should be discarded (that is, not
                     queued).";
            }

            container stateful-parameter{
                if-feature stateful;
                must "(../info/capability/stateful/active == true)"
                {
                    error-message
                        "The Active Stateful PCE must be enabled";
                    description
                        "When PCEP entity is active stateful
                         enabled";
                }
                uses stateful-pce-parameter;

                description
                    "The configured stateful parameters";
            }


            container peers{
                must "((../role == 'pcc')" +
                     " or " +
                     "(../role == 'pcc-and-pce'))"
                {
                    error-message
                        "The PCEP entity must be PCC";
```

```
                description
                    "When PCEP entity is PCC, as remote
                     PCE peers are configured.";
            }
            description
                "The list of configured peers for the
                 entity (remote PCE)";
            list peer{
                key "addr";

                description
                    "The peer configured for the entity.
                     (remote PCE)";

                leaf addr {
                    type inet:ip-address;
                    description
                        "The local Internet address of this
                         PCEP peer.";
                }

                leaf description {
                    type string;
                    description
                        "Description of the PCEP peer
                         configured by the user";
                }
                uses info {
                    description
                        "PCE Peer information";
                }
                uses pce-info {
                    description
                        "PCE Peer information";
                }

                leaf delegation-pref{
                    if-feature stateful;
                    type uint8{
                        range "0..7";
                    }
                    must "(../../info/capability/stateful/active"
                         + "== true)"
                    {
                        error-message
                            "The Active Stateful PCE must be
                             enabled";
                        description
```

```
                                "When PCEP entity is active stateful
                                 enabled";
                        }
                        description
                            "The PCE peer delegation preference.";
                    }
                    uses authentication {
                        description
                            "PCE Peer authentication";
                    }
                }//peer
            }//peers
        }//entity
    }//pcep

    /*
     * Operational data nodes
     */

    container pcep-state{
        config false;
        description
            "The list of operational PCEP entities on the
             device.";

        container entity{
            description
                "The operational PCEP entity on the device.";

            leaf addr {
                type inet:ip-address;
                description
                    "The local Internet address of this PCEP
                    entity.
                    If operating as a PCE server, the PCEP
                    entity listens on this address.
                    If operating as a PCC, the PCEP entity
                    binds outgoing TCP connections to this
                    address.
                    It is possible for the PCEP entity to
                    operate both as a PCC and a PCE Server, in
                    which case it uses this address both to
                    listen for incoming TCP connections and to
                    bind outgoing TCP connections.";
            }

            leaf index{
                type uint32;
```

```
                description
                    "The index of the operational PECP
                     entity";
            }


            leaf admin-status {
                type pcep-admin-status;
                description
                    "The administrative status of this PCEP Entity.
                     This is the desired operational status as
                     currently set by an operator or by default in
                     the implementation.  The value of enabled
                     represents the current status of an attempt
                     to reach this desired status.";
            }

            leaf oper-status {
                type pcep-admin-status;
                description
                   "The operational status of the PCEP entity.
                    Takes one of the following values.
                    - oper-status-up(1): the PCEP entity is
                      active.
                    - oper-status-down(2): the PCEP entity is
                      inactive.
                    - oper-status-going-up(3): the PCEP entity is
                      activating.
                    - oper-status-going-down(4): the PCEP entity is
                      deactivating.
                    - oper-status-failed(5): the PCEP entity has
                      failed and will recover when possible.
                    - oper-status-failed-perm(6): the PCEP entity
                      has failed and will not recover without
                      operator intervention.";
            }

            leaf role {
                type pcep-role;
                description
                    "The role that this entity can play.
                     Takes one of the following values.
                     - unknown(0): this PCEP entity role is
                       not known.
                     - pcc(1): this PCEP entity is a PCC.
                     - pce(2): this PCEP entity is a PCE.
                     - pcc-and-pce(3): this PCEP entity is
                       both a PCC and a PCE.";
```

```
            }

            uses info {
                description
                    "Local PCEP entity information";
            }

            container pce-info {
                when "((../role == 'pce')" +
                     " or " +
                     "(../role == 'pcc-and-pce'))"
                {
                    description
                        "When PCEP entity is PCE";
                }
                uses pce-info {
                    description
                        "Local PCE information";
                }
                                uses authentication {
                                        description
                                                "Local PCE authentication inform
ation";
                        }
                description
                    "The Local PCE Entity PCE information";
            }

            uses pcep-entity-info{
                description
                    "The operational information related to the
                     PCEP entity.";
            }

            container stateful-parameter{
                if-feature stateful;
                must "(../info/capability/stateful/active == true)"
                {
                    error-message
                        "The Active Stateful PCE must be enabled";
                    description
                        "When PCEP entity is active stateful
                         enabled";
                }
                uses stateful-pce-parameter;

                description
                    "The operational stateful parameters";
            }
```

```
            container lsp-db{
                if-feature stateful;
                description
                    "The LSP-DB";
                list association-list {
                    key "id source global-source extended-id";
                    description
                        "List of all PCEP associations";
                    uses association {
                        description
                            "The Association attributes";
                    }
                    list lsp {
                        key "plsp-id pcc-id";
                        description
                            "List of all LSP in this association";
                        leaf plsp-id {
                            type leafref {
                                path "/pcep-state/entity/lsp-db/"
                                + "lsp/plsp-id";
                            }
                            description
                                "Reference to PLSP-ID in LSP-DB";
                        }
                        leaf pcc-id {
                            type leafref {
                                path "/pcep-state/entity/lsp-db/"
                                + "lsp/pcc-id";
                            }
                            description
                                "Reference to PCC-ID in LSP-DB";
                        }
                    }
                }
                list lsp{
                    key "plsp-id pcc-id";
                    description
                        "List of all LSPs in LSP-DB";
                    uses lsp-state{
                        description
                            "The PCEP specific attributes for
                             LSP-DB.";
                    }
                    list association-list {
                        key "id source global-source extended-id";
                        description
                            "List of all PCEP associations";
                        uses association-ref {
```

```
                        description
                            "Reference to the Association
                             attributes";
                    }
                }

            }
        }
        container peers{
            description
                    "The list of peers for the entity";

            list peer{
                key "addr";

                description
                    "The peer for the entity.";

                leaf addr {
                    type inet:ip-address;
                    description
                        "The local Internet address of this PCEP
                         peer.";
                }

                leaf role {
                    type pcep-role;
                    description
                        "The role of the PCEP Peer.
                         Takes one of the following values.
                         - unknown(0): this PCEP peer role
                           is not known.
                         - pcc(1): this PCEP peer is a PCC.
                         - pce(2): this PCEP peer is a PCE.
                         - pcc-and-pce(3): this PCEP peer
                           is both a PCC and a PCE.";
                }


                uses info {
                    description
                        "PCEP peer information";
                }


                container pce-info {
                    when "((../role == 'pce')" +
                    " or " +
```

```
                        "(../role == 'pcc-and-pce'))"
                    {
                        description
                            "When PCEP entity is PCE";
                    }
                    uses pce-info {
                        description
                            "PCE Peer information";
                    }
                description
                    "The PCE Peer information";
                }

                leaf delegation-pref{
                    if-feature stateful;
                    type uint8{
                        range "0..7";
                    }
                    must "((../../role == 'pcc')" +
                         " or " +
                         "(../../role == 'pcc-and-pce'))"
                    {
                        error-message
                            "The PCEP entity must be PCC";
                        description
                            "When PCEP entity is PCC";
                    }
                    must "(../../info/capability/stateful/active"
                         + " == true)"
                    {
                        error-message
                            "The Active Stateful PCE must be
                             enabled";
                        description
                            "When PCEP entity is active stateful
                             enabled";
                    }
                    description
                        "The PCE peer delegation preference.";
                }

                uses authentication {
                    description
                        "PCE Peer authentication";
                }

                leaf discontinuity-time {
                    type yang:timestamp;
```

```
                    description
                        "The timestamp of the time when the
                         information and statistics were
                         last reset.";
                }

                leaf initiate-session {
                    type boolean;
                    description
                        "Indicates whether the local PCEP
                         entity initiates sessions to this peer,
                         or waits for the peer to initiate a
                         session.";
                }

                leaf session-exists{
                    type boolean;
                    description
                        "Indicates whether a session with
                         this peer currently exists.";
                }

                leaf num-sess-setup-ok{
                    type yang:counter32;
                    description
                        "The number of PCEP sessions successfully
                         successfully established with the peer,
                         including any current session.  This
                         counter is incremented each time a
                         session with this peer is successfully
                         established.";
                }

                leaf num-sess-setup-fail{
                    type yang:counter32;
                    description
                       "The number of PCEP sessions with the peer
                        that have been attempted but failed
                        before being fully established. This
                        counter is incremented each time a
                        session retry to this peer fails.";
                }

                leaf session-up-time{
                    type yang:timestamp;
                    must "(../num-sess-setup-ok != 0  or " +
                        "(../num-sess-setup-ok = 0  and "  +
                        "session-up-time = 0))" {
```

```
                        error-message
                            "Invalid Session Up timestamp";
                        description
                            "If num-sess-setup-ok is zero,
                             then this leaf contains zero.";
                    }
                description
                   "The timestamp value of the last time a
                    session with this peer was successfully
                    established.";
            }

            leaf session-fail-time{
                type yang:timestamp;
                must "(../num-sess-setup-fail != 0  or " +
                    "(../num-sess-setup-fail = 0  and "  +
                    "session-fail-time = 0))" {
                        error-message
                            "Invalid Session Fail timestamp";
                        description
                            "If num-sess-setup-fail is zero,
                             then this leaf contains zero.";
                    }
                description
                  "The timestamp value of the last time a
                   session with this peer failed to be
                   established.";
            }

            leaf session-fail-up-time{
                type yang:timestamp;
                must "(../num-sess-setup-ok != 0  or "     +
                    "(../num-sess-setup-ok = 0  and "     +
                    "session-fail-up-time = 0))" {
                        error-message
                            "Invalid Session Fail from
                             Up timestamp";
                        description
                            "If num-sess-setup-ok is zero,
                             then this leaf contains zero.";
                    }
                description
                  "The timestamp value of the last time a
                   session with this peer failed from
                   active.";
            }

            container pcep-stats {
```

```
                        description
                            "The container for all statistics at peer
                             level.";
                        uses pcep-stats{
                            description
                                "Since PCEP sessions can be
                                ephemeral, the peer statistics tracks
                                a peer even when no PCEP session
                                currently exists to that peer. The
                                statistics contained are an aggregate
                                of the statistics for all successive
                                sessions to that peer.";
                        }

                        leaf num-req-sent-closed{
                            type yang:counter32;
                            description
                                "The number of requests that were
                                 sent to the peer and implicitly
                                 cancelled when the session they were
                                 sent over was closed.";
                        }

                        leaf  num-req-rcvd-closed{
                            type yang:counter32;
                            description
                                "The number of requests that were
                                 received from the peer and
                                 implicitly cancelled when the
                                 session they were received over
                                 was closed.";
                        }
                    }//pcep-stats


                    container sessions {
                        description
                            "This entry represents a single PCEP
                             session in which the local PCEP entity
                             participates.
                             This entry exists only if the
                             corresponding PCEP session has been
                             initialized by some event, such as
                             manual user configuration, auto-
                             discovery of a peer, or an incoming
                             TCP connection.";
```

```
                        list session {
                            key "initiator";

                            description
                                "The list of sessions, note that
                                 for a time being two sessions
                                 may exist for a peer";

                            leaf initiator {
                                type pcep-initiator;
                                description
                                    "The initiator of the session,
                                     that is, whether the TCP
                                     connection was initiated by
                                     the local PCEP entity or the
                                     peer.
                                     There is a window during
                                     session initialization where
                                     two sessions can exist between
                                     a pair of PCEP speakers, each
                                     initiated by one of the
                                     speakers. One of these
                                     sessions is always discarded
                                     before it leaves OpenWait state.
                                     However, before it is discarded,
                                     two sessions to the given peer
                                     appear transiently in this MIB
                                     module. The sessions are
                                     distinguished by who initiated
                                     them, and so this field is the
                                     key.";
                            }

                            leaf state-last-change {
                                type yang:timestamp;
                                description
                                    "The timestamp value at the
                                     time this session entered its
                                     current state as denoted by
                                     the state leaf.";
                            }

                            leaf state {
                                type pcep-sess-state;
                                description
                                    "The current state of the
                                     session.
                                     The set of possible states
```

```
                             excludes the idle state since
                             entries do not exist in the
                             idle state.";
                    }

                    leaf session-creation {
                        type yang:timestamp;
                        description
                            "The timestamp value at the
                             time this session was
                             created.";
                    }

                    leaf connect-retry {
                        type yang:counter32;
                        description
                             "The number of times that the
                              local PCEP entity has
                              attempted to establish a TCP
                              connection for this session
                              without success. The PCEP
                              entity gives up when this
                              reaches connect-max-retry.";
                    }

                    leaf local-id {
                        type uint32 {
                          range "0..255";
                        }
                        description
                             "The value of the PCEP session
                              ID used by the local PCEP
                              entity in the Open message
                              for this session.
                              If state is tcp-pending then
                              this is the session ID that
                              will be used in the Open
                              message. Otherwise, this is
                              the session ID that was sent
                              in the Open message.";
                    }

                    leaf remote-id {
                        type uint32 {
                            range "0..255";
                        }
                        must "((../state != 'tcp-pending'" +
                             "and " +
```

```
                             "../state != 'open-wait' )" +
                             "or " +
                             "((../state = 'tcp-pending'" +
                             " or " +
                             "../state = 'open-wait' )" +
                             "and remote-id = 0))" {
                                error-message
                                    "Invalid remote-id";
                                description
                                     "If state is tcp-pending
                                      or open-wait then this
                                      leaf is not used and
                                      MUST be set to zero.";
                            }
                        description
                             "The value of the PCEP session
                              ID used by the peer in its
                              Open message for this
                              session.";
                    }

                    leaf keepalive-timer {
                        type uint32 {
                          range "0..255";
                        }
                        units "seconds";
                        must "(../state = 'session-up'" +
                             "or " +
                             "(../state != 'session-up'" +
                             "and keepalive-timer = 0))" {
                                error-message
                                    "Invalid keepalive
                                     timer";
                                description
                                    "This field is used if
                                     and only if state is
                                     session-up. Otherwise,
                                     it is not used and
                                     MUST be set to
                                     zero.";
                            }
                        description
                             "The agreed maximum interval at
                              which the local PCEP entity
                              transmits PCEP messages on this
                              PCEP session.  Zero means that
                              the local PCEP entity never
                              sends Keepalives on this
```

```
                          session.";
                }

                leaf peer-keepalive-timer {
                    type uint32 {
                      range "0..255";
                    }
                    units "seconds";
                    must "(../state = 'session-up'" +
                         "or "   +
                         "(../state != 'session-up'" +
                         "and " +
                         "peer-keepalive-timer = 0))" {
                            error-message
                                "Invalid Peer keepalive
                                 timer";
                            description
                                "This field is used if
                                 and only if state is
                                 session-up. Otherwise,
                                 it is not used and MUST
                                 be set to zero.";
                        }
                    description
                         "The agreed maximum interval at
                          which the peer transmits PCEP
                          messages on this PCEP session.
                          Zero means that the peer never
                          sends Keepalives on this
                          session.";
                }

                leaf dead-timer {
                    type uint32 {
                      range "0..255";
                    }
                    units "seconds";
                    description
                         "The dead timer interval for
                          this PCEP session.";
                }

                leaf peer-dead-timer {
                    type uint32 {
                      range "0..255";
                    }
                    units "seconds";
                    must "((../state != 'tcp-pending'" +
```

```
                                  "and " +
                                  "../state != 'open-wait' )" +
                                  "or " +
                                  "((../state = 'tcp-pending'" +
                                  " or " +
                                  "../state = 'open-wait' )" +
                                  "and " +
                                  "peer-dead-timer = 0))" {
                                     error-message
                                         "Invalid Peer Dead
                                          timer";
                                     description
                                         "If state is tcp-
                                          pending or open-wait
                                          then this leaf is not
                                          used and MUST be set to
                                          zero.";
                                 }
                             description
                                  "The peer's dead-timer interval
                                   for this PCEP session.";
                         }

                         leaf ka-hold-time-rem {
                             type uint32 {
                               range "0..255";
                             }
                             units "seconds";
                             must "((../state != 'tcp-pending'" +
                                  "and " +
                                  "../state != 'open-wait' ) " +
                                  "or " +
                                  "((../state = 'tcp-pending'" +
                                  "or " +
                                  "../state = 'open-wait' )" +
                                  "and " +
                                  "ka-hold-time-rem = 0))" {
                                     error-message
                                         "Invalid Keepalive hold
                                          time remaining";
                                     description
                                         "If state is tcp-pending
                                          or open-wait then this
                                          field is not used and
                                          MUST be set to zero.";
                                 }
                             description
                                  "The keep alive hold time
```

```
                          remaining for this session.";
                }

                leaf overloaded {
                    type boolean;
                    description
                         "If the local PCEP entity has
                          informed the peer that it is
                          currently overloaded, then this
                          is set to true.  Otherwise, it
                          is set to false.";
                }

                leaf overload-time {
                    type uint32;
                    units "seconds";
                    must "(../overloaded = true or" +
                        "(../overloaded != true and" +
                        " overload-time = 0))" {
                            error-message
                                "Invalid overload-time";
                            description
                                "This field is only used
                                 if overloaded is set to
                                 true. Otherwise, it is
                                 not used and MUST be set
                                 to zero.";
                        }
                    description
                         "The interval of time that is
                          remaining until the local PCEP
                          entity will cease to be
                          overloaded on this session.";
                }

                leaf peer-overloaded {
                    type boolean;
                    description
                         "If the peer has informed the
                          local PCEP entity that it is
                          currently overloaded, then this
                          is set to true. Otherwise, it
                          is set to false.";
                }

                leaf peer-overload-time {
                    type uint32;
                    units "seconds";
```

```
                                must "(../peer-overloaded = true" +
                                     " or " +
                                     "(../peer-overloaded != true" +
                                     " and " +
                                     "peer-overload-time = 0))" {
                                        error-message
                                            "Invalid peer overload
                                             time";
                                        description
                                            "This field is only used
                                             if peer-overloaded is
                                             set to true. Otherwise,
                                             it is not used and MUST
                                             be set to zero.";
                                    }
                                description
                                     "The interval of time that is
                                      remaining until the peer will
                                      cease to be overloaded.  If it
                                      is not known how long the peer
                                      will stay in overloaded state,
                                      this leaf is set to zero.";
                            }
                            leaf lspdb-sync {
                                if-feature stateful;
                                type sync-state;
                                description
                                    "The LSP-DB state synchronization
                                    status.";
                            }
                            leaf discontinuity-time {
                                type yang:timestamp;
                                description
                                     "The timestamp value of the time
                                      when the statistics were last
                                      reset.";
                            }

                            container pcep-stats {
                                description
                                    "The container for all statistics
                                     at session level.";
                                uses pcep-stats{
                                    description
                                        "The statistics contained are
                                         for the current sessions to
                                         that peer. These are lost
                                         when the session goes down.
```

```
                                          ";
                                 }
                             }//pcep-stats

                         } // session
                     } // sessions
                 }//peer
             }//peers
         }//entity
     }//pcep-state

     /*
      * Notifications
      */
     notification pcep-session-up {
         description
             "This notification is sent when the value of
              '/pcep/pcep-state/peers/peer/sessions/session/state'
              enters the 'session-up' state.";

         uses notification-instance-hdr;

         uses notification-session-hdr;

         leaf state-last-change {
             type yang:timestamp;
             description
                 "The timestamp value at the time this session entered
                 its current state as denoted by the state leaf.";
         }

         leaf state {
             type pcep-sess-state;
             description
                 "The current state of the session.
                  The set of possible states excludes the idle state
                  since entries do not exist in the idle state.";
         }
     } //notification

     notification pcep-session-down {
         description
             "This notification is sent when the value of
              '/pcep/pcep-state/peers/peer/sessions/session/state'
              leaves the 'session-up' state.";

         uses notification-instance-hdr;
```

```
        leaf session-initiator {
            type pcep-initiator;
            description
                "The initiator of the session.";
        }

        leaf state-last-change {
            type yang:timestamp;
            description
                "The timestamp value at the time this session entered
                its current state as denoted by the state leaf.";
        }

        leaf state {
            type pcep-sess-state;
            description
                "The current state of the session.
                 The set of possible states excludes the idle state
                 since entries do not exist in the idle state.";
        }
    } //notification

    notification pcep-session-local-overload {
        description
            "This notification is sent when the local PCEP entity
            enters overload state for a peer.";

        uses notification-instance-hdr;

        uses notification-session-hdr;

        leaf overloaded {
            type boolean;
            description
                 "If the local PCEP entity has informed the peer that
                 it is currently overloaded, then this is set to
                 true. Otherwise, it is set to false.";
        }

        leaf overload-time {
            type uint32;
            units "seconds";
            must "(../overloaded = true or "    +
                "(../overloaded != true  and "  +
                "overload-time = 0))" {
                    error-message
                        "Invalid overload-time";
                    description
```

```
                        "This field is only used if overloaded is
                         set to true. Otherwise, it is not used
                         and MUST be set to zero.";
                }
            description
                 "The interval of time that is remaining until the
                 local PCEP entity will cease to be overloaded on
                 this session.";
        }
    } //notification

    notification pcep-session-local-overload-clear {
        description
            "This notification is sent when the local PCEP entity
            leaves overload state for a peer.";

        uses notification-instance-hdr;

        leaf overloaded {
            type boolean;
            description
                 "If the local PCEP entity has informed the peer
                 that it is currently overloaded, then this is set
                 to true.  Otherwise, it is set to false.";
        }
    } //notification

    notification pcep-session-peer-overload {
        description
            "This notification is sent when a peer enters overload
            state.";

        uses notification-instance-hdr;

        uses notification-session-hdr;

        leaf peer-overloaded {
            type boolean;
            description
                "If the peer has informed the local PCEP entity that
                it is currently overloaded, then this is set to true.
                Otherwise, it is set to false.";
        }

        leaf peer-overload-time {
            type uint32;
            units "seconds";
            must "(../peer-overloaded = true or "    +
```

```
                "(../peer-overloaded != true  and "  +
                "peer-overload-time = 0))" {
                    error-message
                        "Invalid peer-overload-time";
                    description
                        "This field is only used if
                         peer-overloaded is set to true.
                         Otherwise, it is not used and MUST
                         be set to zero.";
                }
            description
                "The interval of time that is remaining until the
                peer will cease to be overloaded.  If it is not known
                how long the peer will stay in overloaded state, this
                leaf is set to zero.";
        }
    } //notification

    notification pcep-session-peer-overload-clear {
        description
            "This notification is sent when a peer leaves overload
            state.";

        uses notification-instance-hdr;

        leaf peer-overloaded {
            type boolean;
            description
                "If the peer has informed the local PCEP entity that
                it is currently overloaded, then this is set to true.
                Otherwise, it is set to false.";
        }
    } //notification
}//module

<CODE ENDS>
```

9. Security Considerations

The YANG module defined in this memo is designed to be accessed via the NETCONF protocol [RFC6241]. The lowest NETCONF layer is the secure transport layer and the mandatory-to-implement secure transport is SSH [RFC6242]. The NETCONF access control model [RFC6536] provides the means to restrict access for particular NETCONF users to a pre-configured subset of all available NETCONF protocol operations and content.

There are a number of data nodes defined in the YANG module which are writable/creatable/deletable (i.e., config true, which is the default). These data nodes may be considered sensitive or vulnerable in some network environments. Write operations (e.g., <edit-config>) to these data nodes without proper protection can have a negative effect on network operations.

TBD: List specific Subtrees and data nodes and their sensitivity/ vulnerability.

10. Manageability Considerations

10.1. Control of Function and Policy

10.2. Information and Data Models

10.3. Liveness Detection and Monitoring

10.4. Verify Correct Operations

10.5. Requirements On Other Protocols

10.6. Impact On Network Operations

11. IANA Considerations

This document registers a URI in the "IETF XML Registry" [RFC3688]. Following the format in RFC 3688, the following registration has been made.

URI: urn:ietf:params:xml:ns:yang:ietf-pcep

Registrant Contact: The PCE WG of the IETF.

XML: N/A; the requested URI is an XML namespace.

This document registers a YANG module in the "YANG Module Names" registry [RFC6020].

```
Name:         ietf-pcep
Namespace:    urn:ietf:params:xml:ns:yang:ietf-pcep
Prefix:       pcep
Reference:    This I-D
```

12. Acknowledgements

The initial document is based on the PCEP MIB [RFC7420]. Further this document structure is based on Routing Yang Module [I-D.ietf-netmod-routing-cfg]. We would like to thank the authors of aforementioned documents.

13. References

13.1. 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>.

[RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, DOI 10.17487/RFC3688, January 2004, <http://www.rfc-editor.org/info/rfc3688>.

[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>.

[RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, DOI 10.17487/RFC6020, October 2010, <http://www.rfc-editor.org/info/rfc6020>.

[RFC6991] Schoenwaelder, J., Ed., "Common YANG Data Types", RFC 6991, DOI 10.17487/RFC6991, July 2013, <http://www.rfc-editor.org/info/rfc6991>.

[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-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-lsp-setup-type]
Sivabalan, S., Medved, J., Minei, I., Crabbe, E., Varga, R., Tantsura, J., and J. Hardwick, "Conveying path setup type in PCEP messages", draft-ietf-pce-lsp-setup-type-03 (work in progress), June 2015.

[I-D.ietf-pce-segment-routing]
Sivabalan, S., Medved, J., Filsfils, C., Crabbe, E., Lopez, V., Tantsura, J., Henderickx, W., and J. Hardwick, "PCEP Extensions for Segment Routing", draft-ietf-pce-segment-routing-07 (work in progress), March 2016.

13.2. 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>.

[RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <http://www.rfc-editor.org/info/rfc6241>.

[RFC6242] Wasserman, M., "Using the NETCONF Protocol over Secure Shell (SSH)", RFC 6242, DOI 10.17487/RFC6242, June 2011, <http://www.rfc-editor.org/info/rfc6242>.

[RFC6536] Bierman, A. and M. Bjorklund, "Network Configuration Protocol (NETCONF) Access Control Model", RFC 6536, DOI 10.17487/RFC6536, March 2012, <http://www.rfc-editor.org/info/rfc6536>.

[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, <http://www.rfc-editor.org/info/rfc7420>.

[I-D.ietf-netmod-routing-cfg]
Lhotka, L. and A. Lindem, "A YANG Data Model for Routing Management", draft-ietf-netmod-routing-cfg-22 (work in progress), July 2016.

[I-D.ietf-netmod-rfc6087bis]
Bierman, A., "Guidelines for Authors and Reviewers of YANG Data Model Documents", draft-ietf-netmod-rfc6087bis-06 (work in progress), March 2016.

[I-D.ietf-teas-yang-te]
Saad, T., Gandhi, R., Liu, X., Beeram, V., Shah, H., Chen, X., Jones, R., and B. Wen, "A YANG Data Model for Traffic Engineering Tunnels and Interfaces", draft-ietf-teas-yang-te-03 (work in progress), March 2016.

Appendix A. Contributor Addresses

Rohit Pobbathi
Huawei Technologies
Divyashree Techno Park, Whitefield
Bangalore, Karnataka 560066
India

EMail: rohit.pobbathi@huawei.com

Vinod KumarS
Huawei Technologies
Divyashree Techno Park, Whitefield
Bangalore, Karnataka 560066
India

EMail: vinods.kumar@huawei.com

Zafar Ali
Cisco Systems
Canada

EMail: zali@cisco.com

Xufeng Liu
Ericsson
1595 Spring Hill Road, Suite 500
Vienna, VA 22182
USA

EMail: xufeng.liu@ericsson.com

Young Lee
Huawei Technologies
5340 Legacy Drive, Building 3
Plano, TX 75023, USA

Phone: (469) 277-5838
EMail: leeyoung@huawei.com

Udayasree Palle
Huawei Technologies
Divyashree Techno Park, Whitefield
Bangalore, Karnataka 560066
India

EMail: udayasree.palle@huawei.com

Xian Zhang
Huawei Technologies
Bantian, Longgang District
Shenzhen 518129
P.R.China

EMail: zhang.xian@huawei.com

Avantika
Huawei Technologies
Divyashree Techno Park, Whitefield
Bangalore, Karnataka 560066
India

EMail: avantika.sushilkumar@huawei.com

Authors' Addresses

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

EMail: dhruv.ietf@gmail.com

Jonathan Hardwick
Metaswitch
100 Church Street
Enfield EN2 6BQ
UK

EMail: jonathan.hardwick@metaswitch.com

Vishnu Pavan Beeram
Juniper Networks
USA

EMail: vbeeram@juniper.net

Jeff Tantsura
USA

EMail: jefftant@gmail.com

Pce Working Group
Internet-Draft
Intended status: Standards Track
Expires: April 30, 2015

X. Xu
J. You
Huawei
S. Sivabalan
Cisco Systems
H. Shah
Ciena
L. Contreras
Telefonica I+D
October 27, 2014

PCEP Extensions for SFC in SPRING Networks
draft-xu-pce-sr-sfc-02

Abstract

[I-D.xu-spring-pce-based-sfc-arch] describes a PCE-based SFC architecture in which the PCE is used to compute service function paths in SPRING networks. Based on the above architecture, this document describes extensions to the Path Computation Element Protocol (PCEP) that allow a PCE to compute and instantiate service function paths in SPRING networks. The extensions specified in this document are applicable to both the stateless PCE model and the stateful PCE model.

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 April 30, 2015.

Copyright Notice

Copyright (c) 2014 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

Service Function Chaining [I-D.ietf-sfc-architecture] provides a flexible way to construct services. When applying a particular Service Function Chain (SFC) to the traffic classified by the Classifier, the traffic needs to be steered through an ordered set of Service Functions (SF) in the network. This ordered set of SFs in the network, referred to as a Service Function Path (SFP), is an instantiation of the SFC in the network. For example, as shown in Figure 1, an SFP corresponding to the SFC of {SF1, SF3} can be expressed as {SFF1, SF1, SFF2, SF3}.

```
         +-------+
      +--+  PCE  |
      |  +-------+
      |
      |
      |
      |   +-------------------------------------------------+
      |   |                                     SR Netowrks |
      |   |         +-----+   +-----+                       |
      |   |         | SF1 |   | SF2 |                       |
      |   |         +--+--+   +--+--+                       |
      |   |            |         |                          |
      |   |         ^  |         |                          |
      |   |      (2)|  +---+ +---+                          |
      |   |         +--+   | |                              |
     ++---------+      |   | |          +--------------+    |
     |    +----+|      V   | |          |   +-----+    |    |
     |    |PCC || (1)  +---+-+----+ (3) |   | SF3 |    |    |
 --> |SFC +----+|----> |   SFF1   |---->|   +-----+    |---->
 ----+Classifier+------+          +-----+    SFF2      +--------
     +----------+      +----------+     +--------------+    |
          |                                                 |
          +-------------------------------------------------+
```

Figure 1: PCE-based Service Function Chaining in SR Network

[I-D.xu-spring-pce-based-sfc-arch] describes a PCE-based SFC architecture in which the PCE is used to compute an SFP (i.e., instantiate a service function chain) in SPRING networks (a.k.a., Segment Routing networks or SR networks in short). This document describes extensions to the PCEP on basis of that architecture. The extensions specified in this document are applicable to both the stateless PCE model defined in [RFC5440] and the stateful PCE model defined in [I-D.ietf-pce-stateful-pce].

2. Terminology

This section contains definitions for terms used frequently throughout this document. However, many additional definitions can be found in [RFC5440], [I-D.sivabalan-pce-segment-routing] and [I-D.xu-spring-pce-based-sfc-arch].

PCC: Path Computation Client

PCE: Path Computation Element

PCEP: Path Computation Element Protocol

ERO: Explicit Route Object

Service Function Chain (SFC): A service function chain defines a set of abstract service functions and ordering constraints that must be applied to packets and/or frames selected as a result of classification.

SF Identifier (SF ID): A unique identifier that represents a service function within an SFC-enabled domain.

Service Function Forwarder (SFF): A service function forwarder is responsible for delivering traffic received from the network to one or more connected service functions according to information carried in the SFC encapsulation, as well as handling traffic coming back from the SF.

Service Function Path (SFP): The SFP provides a level of indirection between the fully abstract notion of service chain as a sequence of abstract service functions to be delivered, and the fully specified notion of exactly which SFF/SFs the packet will visit when it actually traverses the network. Specifically, it is an ordered list of SFFs and SF IDs.

Compact SFP: An ordered list of SFFs.

SID: Segment Identifier

Service Function SID : A locally unique SID indicating a particular service function on an SFF.

SR: Segment Routing

SR-specific SFP: An ordered list of node SIDs (representing SFFs) and Service Function SIDs.

Compact SR-specific SFP: An ordered list of node SIDs (representing SFFs).

3. Overview of PCEP Extensions for SFC in SR Networks

As discussed in [I-D.xu-spring-pce-based-sfc-arch], the PCC provides an ordered list of SF IDs to the PCE and indicates to the PCE that what type SFs and paths are requested (e.g., an SFP, or a compact SFP, or an SR-specific SFP, or a compact SR-specific SFP) through the path computation request message, and then the PCE responds with a corresponding path through the path computation response message. This specification is applicable to both stateful and stateless PCEs.

4. PCEP Message Extensions for SR-based SFC

4.1. PCReq Message

This document does not specify any changes to the PCReq message format. This document requires the PATH-SETUP-TYPE TLV [I-D.sivabalan-pce-lsp-setup-type] to be carried in the RP Object in order for a PCC to request a particular type of path. Four new Path Setup Types need to be defined for SR-based SFC, or SR-SFC in short (Section 5.2). This document also requires the Include Route Object (IRO) to be carried in the PCReq message in order for a PCC to specify SFC. A new IRO sub-object type needs to be defined for SF (Section 5.3).

4.2. PCRep Message

This document defines the format of the PCRep message carrying an SFP. The message is sent by a PCE to a PCC in response to a previously received PCReq message, where the PCC requested an SFP. The format of the SFC-specific PCRep message is as follows:

```
<PCRep Message>::=<Common Header>
                  <response-list>

Where:

 <response-list>::=<response>[<response-list>]

 <response>::=<RP>
             [<NO-PATH>]
             [<path-list>]


 Where:

  <path-list>::=<SR-SFC-ERO>[<path-list>]
```

The RP and NO-PATH Objects are defined in [RFC5440]. The <SR-SFC-ERO> object contains the SFP and is defined in Section 5.4.

4.3. PCUpd Message

This document defines the format of the PCUpd message carrying an SFP update. The message is sent forwardly by a PCE to a PCC to update an previously computed SFP.

The format of the PCUpd message is as follows:

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

Where:

 <udpate-request-list>::=<udpate-request>[<udpate-request-list>]

 <udpate-request>::=<SRP><path-list>


 Where:

  <path-list>::=<SR-SFC-ERO>[<path-list>]
```

4.4. PCRpt Message

PCPRpt message sent from a PCC to PCE as a respond to a PCUpd message or in an unsolicited manner (e.g., during state synchronization).

The format of the PCUpd message is as follows:

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

              Where:

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

               <state-report>::=[<SRP>]<path-list>


               Where:

                <path-list>::=<SR-SFC-ERO>[<path-list>]
```

5. Object Formats

5.1. OPEN Object

This document defines a new optional TLV for use in the OPEN Object.

5.1.1. SR-SFC PCE Capability TLV

The SR-SFC-PCE-CAPABILITY TLV is an optional TLV for use in the OPEN Object to negotiate SR-SFC capability on the PCEP session. The format of the SR-SFC-PCE-CAPABILITY TLV is shown in the following Figure 2:

```
     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              |       Length=4                |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |          Reserved             | Flags         |      MSD      |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 2: SR-SFC-PCE-CAPABILITY TLV format

The code point for the TLV type is to be defined by IANA. The TLV length is 4 octets. The 32-bit value is formatted as follows. The "Maximum SID Depth" (1 octet) field (MSD) specifies the maximum number of SIDs that a PCC is capable of imposing on a packet. The "Flags" (1 octet) and "Reserved" (2 octets) fields are currently unused, and MUST be set to zero and ignored on receipt.

5.1.1.1. Negotiating SR-SFC Capability

The SR-SFC capability TLV is contained in the OPEN object. By including the TLV in the OPEN message to a PCE, a PCC indicates its support for SFPs. By including the TLV in the OPEN message to a PCC, a PCE indicates that it is capable of computing SFPs.

5.2. RP/SRP Object

In order to setup an SFP, the RP or SRP object MUST carry a PATH-SETUP-TYPE TLV specified in [I-D.sivabalan-pce-lsp-setup-type]. This document defines four new Path Setup Types (PST) for SR-SFC as follows:

PST = 2: The path is an SFP.

PST = 3: The path is a compact SFP.

PST = 4: The path is an SR-specific SFP.

PST = 5: The path is a compact SR-specific SFP.

5.3. Include Route Object

The IRO (Include Route Object) MUST be carried within PCReq messages to indicate a particular SFC. Furthermore, the IRO MAY be carried in PCRep messages. When carried within a PCRep message with the NO-PATH object, the IRO indicates the set of service functions that cause the PCE to fail to find a path.

This document defines a new sub-object type for the SR-SFC as follows:

Type	Sub-object
5	Service Function ID

5.4. SR-SFC-ERO Object

Generally speaking, an SR-SFC-ERO object consists of one or more ERO subobjects described in the following sub-sections to represent a particular type of service function path. In the ERO subobject, each SID is associated with an identifier that represents either a service node or a service function. This identifier is referred to as the 'Node or Service Identifier' (NSI). As described later, an NSI can be represented in various formats (e.g., IPv4 address, IPv6 address, SF identifier, etc). Specifically, in the SFP case, the NSI of every ERO subobject contained in the SR-SFC-ERO object represents a service node or a service function while the SID of each ERO subobject is set to null. In the compact SFP case, the NSI of every ERO subobject contained in the SR-SFC-ERO object only represents an SFF meanwhile the SID of every ERO subobject is set to null. In the SR-specific SFP, the NSI of every ERO subobject contained in the SR-SFC-ERO object represents an SFF or a service function while the SID of every ERO subject MUST NOT be null. In the compact SR-specific SFP, the NSI of every ERO subobject contained in the SR-SFC-ERO object represents an SFF meanwhile the SID of every ERO subobject MUST NOT be null.

5.4.1. SR-SFC-ERO Subobject

An SR-SFC-ERO subobject (as shown in Figure 3) consists of a 32-bit header followed by the SID and the NSI associated with the SID. The SID is a 32-bit or 128 bit number. The size of the NSI depends on its respective type, as described in the following sub-sections.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|L|    Type     |    Length     |  NSIT   |  Flags    |P|F|S|C|M|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
//         SID (variable:4 or 16 octets)                       //
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
//              NSI (variable)                                 //
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 3: SR-SFC-ERO Subobject Format

The fields in the ERO Subobject are as follows:

’L’ Flag: indicates whether the subobject represents a loose-hop in the explicit route [RFC3209]. If this flag is unset, a PCC MUST not overwrite the SID value present in the SR-SFC-ERO subobject. Otherwise, a PCC MAY expand or replace one or more SID value(s) in the received SR-SFC-ERO based on its local policy.

Type: is the type of the SR-SFC-ERO Subobject. This document defines the SR-SFC-ERO Subobject type. A new code point will be requested for the SR-SFC-ERO Subobject from IANA.

Length: contains the total length of the subobject in octets, including the L, Type and Length fields. Length MUST be at least 4, and MUST be a multiple of 4.

NSI Type (NSIT): indicates the type of NSI associated with the SID. The NSI-Type values are described later in this document.

Flags: is used to carry any additional information pertaining to SID. Currently, the following flag bits are defined:

M: When this bit is set, the SID value represents an MPLS label stack entry as specified in [RFC5462], where only the label value is specified by the PCE. Other fields (TC, S, and TTL) fields MUST be considered invalid, and PCC MUST set these fields according to its local policy and MPLS forwarding rules.

C: When this bit as well as the M bit are set, then the SID value represents an MPLS label stack entry as specified in [RFC5462], where all the entry's fields (Label, TC, S, and TTL) are specified by the PCE. However, a PCC MAY choose to override TC, S, and TTL values according its local policy and MPLS forwarding rules.

S: When this bit is set, the SID value in the subobject body is null. In this case, the PCC is responsible for choosing the SID value, e.g., by looking up its Traffic Engineering Database (TED) using node/service identifier in the subobject body.

F: When this bit is set, the NSI value in the subobject body is null.

P: When this bit is set, the SID value represents an IPv6 address.

SID: is the 4-octect or 16-octect Segment Identifier

NSI: contains the NSI associated with the SID. Depending on the value of NSIT, the NSI can have different format as described in the following sub-section.

5.4.2. NSI Associated with SID

This document defines the following NSIs:

'IPv4 Node ID': is specified as an IPv4 address. In this case, NSIT and Length are 1 and 12 respectively.

'IPv6 Node ID': is specified as an IPv6 address. In this case, NSIT and Length are 2 and 24 respectively.

'Service Function ID': is specified as an SF ID. In this case, NSIT and Length are TBD.

5.4.3. SR-SFC-ERO Processing

TBD.

6. IANA Considerations

6.1. PCEP Objects

IANA is requested to allocate an ERO subobject type (recommended value= 6) for the SR-SFC-ERO subobject.

6.2. PCEP-Error Object

TBD.

6.3. PCEP TLV Type Indicators

This document defines the following new PCEP TLVs:

Value	Meaning	Reference
27	SR-SFC-PCE-CAPABILITY	This document

6.4. New Path Setup Type

This document defines a new setup type for the PATH-SETUP-TYPE TLV as follows:

Value	Description	Reference
2	The path is an SFP.	This document
3	The path is a compact SFP.	This document
4	The path is an SR-specific SFP.	This document
5	The path is a compact SR-specific SFP.	This document

6.5. New IRO Sub-object Type

This document defines a new IRO sub-object type for the SFC as follows:

Type	Sub-object
5	Service Function ID

7. Security considerations

This document does not introduce any new security considerations.

8. Acknowledgement

TBD.

9. References

9.1. Normative References

[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-09 (work in progress), June 2014.

[I-D.ietf-sfc-architecture]
Halpern, J. and C. Pignataro, "Service Function Chaining (SFC) Architecture", draft-ietf-sfc-architecture-02 (work in progress), September 2014.

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

[RFC3209] Awduche, D., Berger, L., Gan, D., Li, T., Srinivasan, V., and G. Swallow, "RSVP-TE: Extensions to RSVP for LSP Tunnels", RFC 3209, December 2001.

[RFC5440] Vasseur, JP. and JL. Le Roux, "Path Computation Element (PCE) Communication Protocol (PCEP)", RFC 5440, March 2009.

[RFC5462] Andersson, L. and R. Asati, "Multiprotocol Label Switching (MPLS) Label Stack Entry: "EXP" Field Renamed to "Traffic Class" Field", RFC 5462, February 2009.

9.2. Informative References

[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.

[I-D.sivabalan-pce-segment-routing]
Sivabalan, S., Medved, J., Filsfils, C., Crabbe, E., Raszuk, R., Lopez, V., and J. Tantsura, "PCEP Extensions for Segment Routing", draft-sivabalan-pce-segment-routing-03 (work in progress), July 2014.

[I-D.xu-spring-pce-based-sfc-arch]
Xu, X., You, J., Shah, H., and L. Contreras, "PCE-based SFC Architecture in SR Networks", draft-xu-spring-pce-based-sfc-arch-01 (work in progress), June 2014.

Authors' Addresses

Xiaohu Xu
Huawei

Email: xuxiaohu@huawei.com

Jianjie You
Huawei
101 Software Avenue, Yuhuatai District
Nanjing, 210012
China

Email: youjianjie@huawei.com

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

Email: msiva@cisco.com

Himanshu Shah
Ciena
3939 North First Street
San Jose, CA 95134
USA

Email: hshah@ciena.com

Luis M. Contreras
Telefonica I+D
Ronda de la Comunicacion, s/n
Sur-3 building, 3rd floor
Madrid, 28050
Spain

Email: luismiguel.contrerasmurillo@telefonica.com
URI: http://people.tid.es/LuisM.Contreras/

PCE Working Group
Internet-Draft
Intended status: Standards Track
Expires: January 4, 2018

X. Xu
Huawei
J. You

S. Sivabalan
Cisco
H. Shah
Ciena
L. Contreras
Telefonica I+D
D. Bernier
Bell Canada
S. Ma
Juniper
July 3, 2017

PCEP Extensions for Unifed Source Routing-based SFC
draft-xu-pce-sr-sfc-05

Abstract

MPLS-SPRING (a.k.a., MPLS Segment Routing) could be leveraged to realize a unified source routing mechanism across MPLS, IPv4 and IPv6 data planes by using a unified source routing instruction set while preserving backward compatibility with MPLS-SPRING. More specifically, the source routing instruction set information contained in a source routed packet could be uniformly encoded as an MPLS label stack no matter the underlay is IPv4, IPv6 or MPLS. The unified source routing mechanism could be leveraged to realize a transport-independent service function chaining by encoding the service function path information or service function chain information as an MPLS label stack. This document describes extensions to the Path Computation Element Protocol (PCEP) that allow a PCE to compute and instantiate service function paths in the unifed source routing based service function chaining context. The extensions specified in this document are applicable to both the stateless PCE model and the stateful PCE model.

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 January 4, 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

Service Function Chaining [RFC7665] provides a flexible way to construct services. When applying a particular Service Function Chain (SFC) to the traffic classified by the Classifier, the traffic needs to be steered through an ordered set of Service Function Forwarders (SFF) and Service Functions (SF) in the network. This ordered set of SFFs and SFs in the network, referred to as a Service Function Path (SFP), is an instantiation of the SFC in the network. For example, as shown in Figure 1, an SFP corresponding to the SFC of {SF1, SF3} can be expressed as {SFF1, SF1, SFF2, SF3}.

```
            +-------+
         +--+  PCE  |
         |  +-------+
         |
         |
         |
         |   +-------------------------------------------------+
         |   |                                MPLS-SR Netowrks |
         |   |         +-----+   +-----+                       |
         |   |         | SF1 |   | SF2 |                       |
         |   |         +--+--+   +--+--+                       |
         |   |            |         |                          |
         |   |         ^  |         |                          |
         |   |      (2)|  +---+ +---+                          |
         |   |         +--+   | |                              |
        ++---------+      |   | |          +--------------+    |
        |    +----+|      V   | |          |   +-----+    |    |
        |    |PCC || (1)  +---+-+----+ (3) |   | SF3 |    |    |
    --> |SFC +----+|----> |   SFF1   |---->|   +-----+    |---->
    ----+Classifier+------+          +-----+    SFF2      +---------
        +----------+      +----------+     +--------------+    |
             |                                                 |
             +-------------------------------------------------+
```

Figure 1: PCE-based Service Function Chaining in MPLS-SR Network

MPLS-SPRING (a.k.a., MPLS Segment Routing) could be leveraged to realize a unified source routing mechanism across MPLS, IPv4 and IPv6 data planes by using a unified source routing instruction set while preserving backward compatibility with MPLS-SPRING as descried in [I-D.xu-mpls-unified-source-routing-instruction]. More specifically, the source routing instruction set information contained in a source routed packet could be uniformly encoded as an MPLS label stack no matter the underlay is IPv4, IPv6 or MPLS. The unified source routing mechanism in turn could be leveraged to realize a transport-independent service function chaining by encoding the service function path information or service function chain information as an MPLS label stack as described in [I-D.xu-mpls-service-chaining].

This document describes extensions to the Path Computation Element Protocol (PCEP) that allow a PCE to compute and instantiate service function paths in the MPLS source routing based service function chaining context. More specifically, the PCC provides an ordered list of SF IDs to the PCE and indicates to the PCE that what type SFs and paths are requested (e.g., an SFP, or a compact SFP, or an SR-specific SFP, or a compact SR-specific SFP) through the path computation request message, and then the PCE responds with a corresponding path through the path computation response message.

The extensions specified in this document are applicable to both the stateless PCE model [RFC5440] and the stateful PCE model [I-D.ietf-pce-stateful-pce].

2. Terminology

This memo makes use of the terms defined in [RFC5440], [I-D.ietf-pce-segment-routing] and [I-D.xu-mpls-service-chaining]. In addition, this memo defines the following two additional terms:

Compact SFP: An ordered list of SFFs.

SR-specific SFP: An ordered list of node SIDs (representing SFFs) and Service Function SIDs.

Compact SR-specific SFP: An ordered list of node SIDs (representing SFFs).

3. PCEP Message Extensions for MPLS Source Routing-based SFC

3.1. PCReq Message

This document does not specify any changes to the PCReq message format. This document requires the PATH-SETUP-TYPE TLV [I-D.ietf-pce-lsp-setup-type] to be carried in the RP Object in order for a PCC to request a particular type of path. Four new Path Setup Types need to be defined for MPLS source routing-based SFC (see Section 4.2). This document also requires the Include Route Object (IRO) to be carried in the PCReq message in order for a PCC to specify an SFC. A new IRO sub-object type needs to be defined for SF (see Section 4.3).

3.2. PCRep Message

This document defines the format of the PCRep message carrying an SFP. The message is sent by a PCE to a PCC in response to a previously received PCReq message, where the PCC requested an SFP. The format of the SFC-specific PCRep message is defined as follows:

```
<PCRep Message>::=<Common Header>
                  <response-list>
   Where:
    <response-list>::=<response>[<response-list>]
    <response>::=<RP>
                 [<NO-PATH>]
                 [<path-list>]
    Where:
     <path-list>::=<SR-SFC-ERO>[<path-list>]
```

The RP and NO-PATH Objects are defined in [RFC5440]. The <SR-SFC-ERO> object contains an SFP and is defined in Section 4.4.

3.3. PCUpd Message

This document defines the format of the PCUpd message carrying an SFP update. The message is sent forwardly by a PCE to a PCC to update an previously computed SFP.

The format of the PCUpd message is defined as follows:

```
<PCUpd Message>::=<Common Header>
                  <udpate-request-list>
Where:
 <udpate-request-list>::=<udpate-request>[<udpate-request-list>]
 <udpate-request>::=<SRP><path-list>
 Where:
  <path-list>::=<SR-SFC-ERO>[<path-list>]
```

3.4. PCRpt Message

PCPRpt message sent from a PCC to PCE as a respond to a PCUpd message or in an unsolicited manner (e.g., during state synchronization).

The format of the PCUpd message is defined as follows:

```
<PCUpd Message>::=<Common Header>
                  <state-report-list>
Where:
 <state-report-list>::=<state-report>[<state-report-list>]
 <state-report>::=[<SRP>]<path-list>
 Where:
  <path-list>::=<SR-SFC-ERO>[<path-list>]
```

4. Object Formats

4.1. OPEN Object

This document defines a new optional TLV for use in the OPEN Object.

4.1.1. SR-SFC PCE Capability TLV

The SR-SFC-PCE-CAPABILITY TLV is an optional TLV for use in the OPEN Object to negotiate SR-SFC capability on the PCEP session. The format of the SR-SFC-PCE-CAPABILITY TLV is shown in the following Figure 2:

```
 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              |       Length=4                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Reserved             | Flags         |      MSD      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 2: SR-SFC-PCE-CAPABILITY TLV Format

The code point for the TLV type is to be defined by IANA. The TLV length is 4 octets. The 32-bit value is formatted as follows. The "Maximum SID Depth" (1 octet) field (MSD) specifies the maximum number of SIDs that a PCC is capable of imposing on a packet. The "Flags" (1 octet) and "Reserved" (2 octets) fields are currently unused, and MUST be set to zero and ignored on receipt.

4.1.1.1. Negotiating SR-SFC Capability

The SR-SFC capability TLV is contained in the OPEN object. By including the TLV in the OPEN message to a PCE, a PCC indicates its support for SFPs. By including the TLV in the OPEN message to a PCC, a PCE indicates that it is capable of computing SFPs.

4.2. RP/SRP Object

In order to setup an SFP, the RP or SRP object MUST carry a PATH-SETUP-TYPE TLV specified in [I-D.ietf-pce-lsp-setup-type]. This document defines four new Path Setup Types (PST) for SR-SFC as follows:

PST = 2: The path is an SFP.

PST = 3: The path is a compact SFP.

PST = 4: The path is an SR-specific SFP.

PST = 5: The path is a compact SR-specific SFP.

4.3. Include Route Object

The IRO (Include Route Object) MUST be carried within PCReq messages to indicate a particular SFC. Furthermore, the IRO MAY be carried in PCRep messages. When carried within a PCRep message with the NO-PATH object, the IRO indicates the set of service functions that cause the PCE to fail to find a path. This document defines a new sub-object type for the SR-SFC as follows:

```
   Type       Sub-object

   5          Service Function ID
```

4.4. SR-SFC-ERO Object

Generally speaking, an SR-SFC-ERO object consists of one or more ERO subobjects described in the following sub-sections to represent a particular type of service function path. In the ERO subobject, each SID is associated with an identifier that represents either an SFF or an SF. This identifier is referred to as the 'Node or Service Identifier' (NSI). As described later, an NSI can be represented in various formats (e.g., IPv4 address, IPv6 address, SF identifier, etc). Specifically, in the SFP case, the NSI of every ERO subobject contained in the SR-SFC-ERO object represents an SFF or an SF while the SID of each ERO subobject is set to null. In the compact SFP case, the NSI of every ERO subobject contained in the SR-SFC-ERO object only represents an SFF meanwhile the SID of every ERO subobject is set to null. In the SR-specific SFP, the NSI of every ERO subobject contained in the SR-SFC-ERO object represents an SFF or an SF while the SID of every ERO subject MUST NOT be null. In the compact SR-specific SFP, the NSI of every ERO subobject contained in the SR-SFC-ERO object represents an SFF meanwhile the SID of every ERO subobject MUST NOT be null.

4.4.1. SR-SFC-ERO Subobject

An SR-SFC-ERO subobject (as shown in Figure 3) consists of a 32-bit header followed by the SID and the NSI associated with the SID. The SID is a 32-bit or 128 bit number. The size of the NSI depends on its respective type, as described in the following sub-sections.

```
    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
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |L|    Type     |    Length     |  NSIT   |  Flags    |P|F|S|C|M|
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   //         SID (variable:4 or 16 octets)                       //
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   //              NSI (variable)                                 //
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 3: SR-SFC-ERO Subobject Format

'L' Flag: indicates whether the subobject represents a loose-hop in the explicit route [RFC3209]. If this flag is unset, a PCC MUST not overwrite the SID value present in the SR-SFC-ERO subobject. Otherwise, a PCC MAY expand or replace one or more SID value(s) in the received SR-SFC-ERO based on its local policy.

Type: is the type of the SR-SFC-ERO Subobject. This document defines the SR-SFC-ERO Subobject type. A new code point will be requested for the SR-SFC-ERO Subobject from IANA.

Length: contains the total length of the subobject in octets, including the L, Type and Length fields. Length MUST be at least 4, and MUST be a multiple of 4.

NSI Type (NSIT): indicates the type of NSI associated with the SID. The NSI-Type values are described later in this document.

Flags: is used to carry any additional information pertaining to SID. Currently, the following flag bits are defined:

M: When this bit is set, the SID value represents an MPLS label stack entry as specified in [RFC5462], where only the label value is specified by the PCE. Other fields (TC, S, and TTL) fields MUST be considered invalid, and PCC MUST set these fields according to its local policy and MPLS forwarding rules.

C: When this bit as well as the M bit are set, then the SID value represents an MPLS label stack entry as specified in [RFC5462], where all the entry's fields (Label, TC, S, and TTL) are specified by the PCE. However, a PCC MAY choose to override TC, S, and TTL values according its local policy and MPLS forwarding rules.

S: When this bit is set, the SID value in the subobject body is null. In this case, the PCC is responsible for choosing the SID value, e.g., by looking up its Traffic Engineering Database (TED) using node/service identifier in the subobject body.

F: When this bit is set, the NSI value in the subobject body is null.// When will the NSI value is null?

P: When this bit is set, the SID value represents an IPv6 address.

SID: is the 4-octect or 16-octect Segment Identifier.

NSI: contains the NSI associated with the SID. Depending on the value of NSIT, the NSI can have different format as described in the following sub-section.

4.4.2. NSI Associated with SID

This document defines the following NSIs:

'IPv4 Node ID': is specified as an IPv4 address. In this case, NSIT and Length are 1 and 12 respectively.

'IPv6 Node ID': is specified as an IPv6 address. In this case, NSIT and Length are 2 and 24 respectively.

'Service Function ID': is specified as an SF ID. In this case, NSIT and Length are TBD.

4.4.3. SR-SFC-ERO Processing

TBD

5. Acknowledgements

TBD.

6. IANA Considerations

6.1. PCEP Objects

IANA is requested to allocate an ERO subobject type (recommended value= 6) for the SR-SFC-ERO subobject.

6.2. PCEP-Error Object

TBD

6.3. PCEP TLV Type Indicators

This document defines the following new PCEP TLV:

Value	Meaning	Reference
27	SR-SFC-PCE-CAPABILITY	This document

6.4. New Path Setup Type

This document defines the following four new setup types for the PATH-SETUP-TYPE TLV:

Value	Description	Reference
2	The path is an SFP.	This document
3	The path is a compact SFP.	This document
4	The path is an SR-specific SFP.	This document
5	The path is a compact SR-specific SFP.	This document

6.5. New IRO Sub-object Type

This document defines a new IRO sub-object type for SFC as follows:

Type	Sub-object
5	Service Function ID

7. Security Considerations

This document does not introduce any new security considerations.

8. References

8.1. Normative References

[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.

[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>.

8.2. Informative References

[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-04 (work in progress), April 2017.

[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-spring-segment-routing-mpls]
Filsfils, C., Previdi, S., Bashandy, A., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing with MPLS data plane", draft-ietf-spring-segment-routing-mpls-10 (work in progress), June 2017.

[I-D.xu-mpls-service-chaining]
Xu, X., Bryant, S., Assarpour, H., Shah, H., Contreras, L., daniel.bernier@bell.ca, d., jefftant@gmail.com, j., Ma, S., and M. Vigoureux, "Service Chaining using Unified Source Routing Instructions", draft-xu-mpls-service-chaining-03 (work in progress), June 2017.

[I-D.xu-mpls-unified-source-routing-instruction]
Xu, X., Bryant, S., Raszuk, R., Chunduri, U., Contreras, L., Jalil, L., Assarpour, H., Velde, G., Tantsura, J., and S. Ma, "Unified Source Routing Instruction using MPLS Label Stack", draft-xu-mpls-unified-source-routing-instruction-02 (work in progress), June 2017.

[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>.

Authors' Addresses

Xiaohu Xu
Huawei

Email: xuxiaohu@huawei.com

JIanjie You

Email: jianjie.you@gmail.com

Siva Sivabalan
Cisco

Email: msiva@cisco.com

Himanshu Shah
Ciena

Email: hshah@ciena.com

Luis M. Contreras
Telefonica I+D
Ronda de la Comunicacion, s/n
Sur-3 building, 3rd floor
Madrid, 28050
Spain

Email: luismiguel.contrerasmurillo@telefonica.com
URI: http://people.tid.es/LuisM.Contreras/

Daniel Bernier
Bell Canada

Email: daniel.bernier@bell.ca

Shaowen Ma
Juniper

Email: mashaowen@gmail.com

PCE Working Group
Internet Draft
Category: Standards track

Xian Zhang
Haomian Zheng
Huawei
Oscar Gonzales de Dios
Victor Lopez
Telefonica I+D

Expires: August 27, 2015
February 27, 2015

Extensions to Path Computation Element Protocol (PCEP) to Support Resource Sharing-based Path Computation

draft-zhang-pce-resource-sharing-03.txt

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/1id-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 August 27, 2015.

Copyright Notice

Copyright (c) 2014 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.

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].

Abstract

Resource sharing in a network means two or more Label Switched Paths (LSPs) use common piece(s) of resource along their paths. This can help save network resource and useful in scenarios such as LSP recovery or when two LSPs do not need to be active at the same time. A Path Computation Element (PCE) is a centralized entity, responsible for path computation. Given this feature and its access to the network resource information and possibly active LSPs information, it can be used to support resource-sharing-based path computation with better efficiency.

This document extends the Path Computation Element Protocol (PCEP) in order to support resource sharing-based path computation.

Table of Contents

1. Introduction and Motivation

A Path Computation Element (PCE) provides an alternative way for providing path computation function, and it is especially useful in the scenarios where complex constraints and/or a demanding amount of computation resource are required [RFC4655]. The development of PCE standardization has evolved from stateless to stateful. A stateful PCE has access to the LSP database information of the network(s) it serves as a computation engine [Stateful-PCE]. Unless specified, this document assumes a PCE mentioned is a stateful PCE (either passive or active).

Resource sharing denotes that two or more Label Switched Paths (LSPs) share common piece(s) of resource, (such as a common time slot of a link in an Optical Transport Network (OTN)). This is usually useful in the scenario where only one LSP is active and the benefit herein is to save network resources. A simple example of this is dynamically calculating a LSP for an existing LSP undergoing a link failure. Note that the resource sharing can be worked out using a statelss PCE, but the mechanism may be complex and is out the scope of this draft.

This document considers the following requirement: resource sharing with one or multiple existing LSPs.

In a single domain, this is a common requirement in the recovery cases especially in order to increase traffic resilience against failure while reducing the amount of network resource used for recovery purpose [RFC4428].

The current protocol supporting the communication between a PCE and a Path Computation Client (PCC), i.e. PCE Protocol (PCEP), allows for re-optimization of an existing LSP [RFC5440]. This is achieved by setting R bit in the Request Parameter (RP) object, together with some additional information if applicable, in the Path Computation Request (PCReq) message sent from a PCC to the PCE. To support this type of resource sharing, a PCC needs to ask a PCE to compute a new path with the constraints of sharing resource with one or multiple existing LSPs. It is worth noting the 'resource sharing' in this draft not only means one LSP re-using the same link(s) of another LSP, but also the same slice of bandwidth. This may occur when an LSP is required for re-routing, or online re-optimization. Current PCEP specifications do not provide such function.

As mentioned in [stateful-PCE], the PLSP-ID is unique during a PCEP session between PCC and PCE. Such identification is helpful in supporting the above resource sharing requirement for standardization of stateful PCEs. With a unique identifier, the configuration of PCCs is greatly simplified. Instead of determining all the resources to be shared, the PCC could request resource sharing directly from PCE.

The resource sharing can also be required in an inter-layer PCEP session. This is similar to the previous requirement. However, it is more complex and therefore deserves a more detailed explanation here.

In a multi-layer network, Label Switched Paths (LSPs) in a lower layer are used to carry higher-layer LSPs across the lower-layer network [RFC5623]. Therefore, the resource sharing constraints in the higher layer might actually relate to the resource sharing in the lower layer. Thus, it is useful to consider how this can be achieved and whether additional extensions are needed using the models defined in [RFC5623].

In the next sections, use cases are provided to show what information needs to be exchanged to fulfill these requirements. This memo then provides extensions to PCEP to enable this function.

2. Motivation

2.1. Single PCE Use Case

Figure 1 shows a single domain network with a stateful PCE. Assume a working LSP (N1-N2-N3) exists in the network, when there is failure on the link N2-N3, it is desired to set up a restoration path for this working LSP. Suppose N1 serves as the PCC and sends a request to the stateful PCE for such an LSP. Before sending the request, N1 may need to check what policy should be applied for the path computation. For example, it might value resource sharing and prefer to share as much resource with the working LSP as possible and specify this policy in the PCReq message. If resources are shared between the old and new LSPs, there will be some ’interruption’ when the traffic is switched from the old LSP to the new LSP.

On the other hand, in some scenarios there are different policies, for example the LSP should be restored without any interruption with best effort. An example can be found in Fig. 1 without failure on N2-N3 link, instead, an online re-optimization is needed for the working LSP (N1-N2-N3) from the stateful PCE. In such cases, the best choice is to set up a backup LSP for the working LSP with totally separate routing (for example N1-N5-N4-N3), and move the traffic to that backup LSP. After that the working LSP can be torn down, which will not result in any interruption during the optimization procedure. This can actually be implemented with existing PCEP mechanism. However, if there is no such separate path, existing PCEP will reply error. A secondary option for this case is to set up an LSP and complete such re-optimization with resource sharing, even if some interruption introduced. Given the resource from the LSP to be interrupted, there may be some solutions instead of Path Compute error due to the lack of resource.

A simple illustration is provided below:

```
              +--------------+
              |              |
              | Stateful PCE |
              |              |
              +--------------+



         +------+          +------+          +------+
         |  N1  +----------+  N2  +-----X----+  N3  |
         +--+---+          +---+--+          +---+--+
            |                  |                 |
            |                  +---------+       |
            |                            |       |
            |     +------+          +------+     |
            +-----+  N5  +----------+  N4  +-----+
                  +------+          +------+
```

Figure 1: A Single Domain Example

Available recovery paths computed by the stateful PCE:

LSP1: N1-N2-N4-N3
LSP2: N1-N5-N4-N3

If resource sharing is preferred, the stateful PCE will reply with LSP1 information. Instead, if PCC prefer to have less interruption, PCE will reply with LSP2 information.

Another piece of information that needs to be conveyed to the PCE is the information about the working path LSP. Note this simple use case assumes end-to-end recovery. But in order to be applicable to use cases such as shared mesh protection purpose, where the head-end or tail-end nodes may be different, this information is necessary in the message exchange between PCCs and PCEs, so that the stateful PCE knows which LSP the path computation request wants to share the resource.

Besides, parameter changes during the resource sharing computation also need to be considered. For example, the bandwidth of the request LSP may be different with the existing LSP, while resource sharing is still preferred by the PCC. PCE should consider the sharing request together with the policy and available resource(s) in the network. Details can be found in Section 3.3.

2.2. Multiple PCEs Use Case

Figure 2 shows a two-layer network example, with each layer managed by a PCE. As Discussed in Section 3 of [RFC5623], there are three models for inter-layer path computation. They are single PCE computation, multiple PCE with inter-PCE communication and multiple PCE without inter-PCE communication, respectively. For the single PCE computation, the process would be similar to that of the use case in Section 2.1. Thus, this model is not discussed further.

```
                                                               -----
                             .................................| LSR |
                           .:                                 | H5  |
                         .:                                   /-----
                       .:                                    /   |
       -----    -----.:                       -----    -----/    |
      | LSR |--| LSR |.......................| LSR |--| LSR |   /
      | H1  |  | H2  |                       | H3  |  | H4  |  /
       -----    -----\                       /-----    -----  /
                      \                     /                /
                       \                   /                /
                        \                 /                /
                         \               /                /
                          \-----   -----/                /
                          | LSR |-| LSR |               /
                          | L1  | | L2  |              /
                           -----   -----\             /
                             |           \           /
                             |            \         /
                             |             \       /
                           -----            \-----/
                          | LSR |-----------| LSR |
                          | L3  |           | L4  |
                           -----             -----
```

Figure 2: A Two-layer Network Example

An inter-layer path computation example is shown in Fig. 2, assume a LSP (LSP1: H2-H3) has been established already, visible as H2-H3 from view of higher-layer PCE and H2-L1-L2-H3 from the global view (or from the view of lower-layer PCE). A new request comes at H2 to establish a new LSP (LSP2: from H2 to H5), given the constraint it can share resource with LSP1. This requirement is possible if only one of the LSPs needs to be active and resource sharing is the target.

If multiple PCE with inter-PCE communication model is employed, the path computation request sent by H2 to higher-layer PCE will be forwarded to lower-layer PCE since there is no resource readily available in the higher layer. So it leaves the lower-layer PCE to compute a path in the lower layer in order to support the higher layer request. In this case, lower-layer PCE is required to compute a path between H2 and H5 under the constraint that it can share the resource with that of the LSP1. At this moment the lower-layer PCE has the knowledge on the explicit routing that LSP1 go through (H2-L1-L2-H3). So when lower-layer PCE computes the path for LSP2, it can consider the resource used by LSP1 as available with higher priority. For example, lower-layer PCE may choose H2-L1-L2-L4-H5 as the computation result. On the other hand, if the path computation policy is to have a separate path with LSP1, the lower-layer PCE may choose H2-L1-L3-L4-H5.

During this procedure higher-layer PCE can only use LSP1 information (such as its five-tuple LSP information) as the information, an issue to solve is how lower-layer PCE can resolve this information to the actual resource usage in its own layer, i.e. lower layer. This could be solved by edge LSR L1 reporting this higher-lower layer LSP correlation to the lower-layer PCE as part of the LSP information during the LSP state synchronization process. If needed, it can be later updated when there is a change in this information. Alternatively, the lower-layer PCE can get this information from other sources, such as network management system, where this information should be stored.

If multiple PCE without inter-PCE communication model is employed, the path computation request in the lower layer will be initiated the border LSR node, i.e., L1. The process would be similar to that of the previous scenario. A point worth noting is that the border LSR node may be able to resolve the higher layer LSP information itself, such as mapping it to the corresponding LSP in the lower layer, in this way lower-layer PCE does not need to perform this function. Otherwise, the mapping method mentioned above can still be used.

3. Extensions to PCEP

This section provides PCEP extensions. Currently the text focuses only on passive stateful PCE and corresponding PCReq. But if active stateful PCE delegation is used, we would like to convey the same information via RSO in PCRpt. In the passive stateful PCE architecture, a PCC is allowed to specify resource sharing when sending a PCReq message. It also details the processing rule and error codes needed.

3.1. Resource Sharing Object

The PCEP Resource Sharing Object (RSO) is optional. It MAY be carried within a PCRep message from the network element (or other PCCs) so as to indicate the desired resource sharing requirements to be applied by the stateful PCE during path computation.

The RSO object format is compliant with the PCEP object format defined in [RFC5440].

The RSO Object-Class is TBA.

The RSO Object-type is 1.

The format of the RSO object body is:

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| RSO Flags                    |R|D|       Reserved              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
~                          Optional TLVs                        ~
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 3: RSO Object Format

RSO flags (16 bits): the objective of the resource sharing. Currently, the following objectives are defined:

D (1 bit): sharing as little as possible.

R (1 bit): sharing as much as possible

It is possible that multiple computation results satisfy the request. Among these results, D set to 1 will select the most separate one, while R set to 1 will select the one sharing most resources. Both D and R set to 0 don't specify any constraint and will result in a random selection among these results. The combination of D=1 and R=1 is not allowed.

Reserved (2 bytes): This field MUST be set to zero on transmission and MUST be ignored on receipt.

Optional TLVs may be needed to indicate the LSP(s) with which the resource is shared. The LSP Info TLV, include the IPv4-LSP-IDENTIFIERS TLV and IPv6-LSP IDENTIFIERS TLV, are defined in the same way as in [stateful-pce].

3.2. Processing Rules

To request a path allowing sharing resource with one or multiple existing LSPs, a PCC includes a RSO object in the PCReq message.

On receipt of a PCReq message with a RSO object, a stateful PCE MUST proceed as follows:

- If the RSO object is unknown/unsupported, the PCE will follow procedures defined in [RFC5440]. That is, the PCE sends a PCErr message with error type 3 or 4 (Unknown / Not supported object) and error value 1 or 2 (unknown / unsupported object class / object type), and the related path computation request is discarded.

- If TLV(s) present in the RSO object are unknown/unsupported and the P bit is set, the PCE MUST send a PCErr message with error type 3 or 4 (Unknown / Not supported object) and error value 4 (Unrecognized/Unsupported parameter), and the related path computation request MUST be discarded as defined in [RFC5440].

- If the resource sharing information is extracted correctly, the PCE MUST apply the requested resource sharing requirement.

The procedure of setting R and/or D bit follows the rules defined in Section 3.1. The RSO flags may be locally configured on the requesting nodes via external entities, such as a network management system or the entity that impose the resource sharing requirement.

3.3. Carrying RSO in a PCEP Message

The RSO is applied to an individual path computation request and the format of the PCReq message is updated as follows:

```
<PCReq Message> ::= <Common Header>

                    [<svec-list>]

                    <request-list>

where:

     <svec-list> ::= <SVEC>

                     [<OF>]

                     [<metric-list>]

                     [<svec-list>]


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


    <request> ::= <RP>

                  <END-POINTS>

                  [<LSPA>]

                  [<BANDWIDTH>]

                  [<metric-list>]

                  [<OF>]

                  [<RRO>[<BANDWIDTH>]]

                  [<IRO>]

            [<RSO>]

                  [<LOAD-BALANCING>]
```

and where:

<metric-list> ::= <METRIC>[<metric-list>]

4. Security Considerations

Security of PCEP is discussed in [RFC5440] and [RFC6952]. The extensions in this document do not change the fundamentals of security for PCEP.

However, the introduction of the RSO provides a vector that may be used to probe for information from a network. For example, a PCC that wants to discover the path of an LSP with which it is not involved, can issue a PCReq with an RSO and may be able to get back quite a lot of information about the path of the LSP through issuing multiple such requests for different endpoints and analyzing the received results. To protect against this, a PCE should be configured with access and authorization controls such that only authorized PCCs (for example, those within the network) can make computation requests, only specifically authorized PCCs can make requests using the RSO, and resource sharing requests relating to specific LSPs are further limited to a select few PCCs. How such access controls and authorization is managed is outside the scope of this document, but it will at the least include Access Control Lists.

Furthermore, a PCC must be aware that setting up an LSP that shares resources with another LSP may be a way of attacking the other LSP, for example by depriving it of the resources it needs to operate correctly. Thus it is important that, both in PCEP and the associated signaling protocols, only authorized resource sharing is allowed.

5. IANA Considerations

5.1. New Object Type

IANA manages the PCEP Objects code point registry (see [RFC5440]). This is maintained as the "PCEP Objects" sub-registry of the "Path Computation Element Protocol (PCEP) Numbers" registry.

This document defines a new PCEP object, the RSO object, to be carried in PCReq messages. IANA is requested to make the following allocation in the "PCEP Objects" sub-registry:

Object Class	Name	Object Type	Name	Reference

TBA	RSO	Resource Sharing	[this document]

5.2 RSO flags field

IANA is requested to create and maintain a new sub-registry named "RSO flags". The following flags are defined in this document:

Bit	Flag name	Description	Reference
0	D	sharing as little as possible	[this document]
1	R	sharing as much as possible	[this document]

6. References

6.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to indicate requirements levels", RFC 2119, March 1997.

[RFC4655] Farrel, A., Vasseur, J.-P., and Ash, J., "A Path Computation Element (PCE)-Based Architecture", RFC 4655, August 2006.

[RFC5440] Vasseur, J.-P., and Le Roux, JL., "Path Computation Element (PCE) Communication Protocol (PCEP)", RFC 5440, March 2009.

[Stateful-PCE] Crabbe, E., Medved, J., Minei, I., and R. Varga, "PCEP Extensions for Stateful PCE", draft-ietf-pce-stateful-pce-10 (work in progress), October 2014.

6.2. Informative References

[RFC4428] Papadimitriou, D., Mannie., E., "Analysis of Generalized Multi-Protocol Label Switching (GMPLS)-based Recovery Mechanisms (including Protection and Restoration)", RFC4428, March 2006.

[RFC5623] Oki., E., Takeda, T., Le Roux, JL., Farrel, A., "Framework for PCE-Based Inter-Layer MPLS and GMPLS Traffic Engineering", RFC5623, September 2009.

[RFC6952] Jethanandani, M., Patel, K., Zheng, L., "Analysis of BGP, LDP, PCEP, and MSDP Issues According to the Keying and Authentication for Routing Protocols (KARP) Design Guide", RFC6952, May 2013.

7. Authors' Addresses

Xian Zhang
Huawei Technologies

Email: zhang.xian@huawei.com

Haomian Zheng
Huawei Technologies

Email: zhenghaomian@huawei.com

Oscar Gonzalez de Dios
Telefonica I+D
Don Ramon de la Cruz 82-84
Madrid 28045
Spain
EMail: ogondio@tid.es
Victor Lopez
Telefonica I+D
Don Ramon de la Cruz 82-84
Madrid 28045
Spain
EMail: vlopez@tid.es

Contributor's Address :

Dhruv Dhody
Huawei Technologies

Email: dhruv.dhody@huawei.com

Igor Bryskin

ADVA Optical

Email: IBryskin@advaoptical.com

PCE Working Group   Xian Zhang
Internet Draft   Haomian Zheng
Category: Standards track   Huawei Technologies
Oscar Gonzales de Dios
Victor Lopez
Telefonica I+D
Yunbin Xu
CAICT

Expires: October 21, 2020   April 21, 2020

Extensions to the Path Computation Element Protocol (PCEP) to Support Resource Sharing-based Path Computation

draft-zhang-pce-resource-sharing-12

Abstract

Resource sharing in a network means two or more Label Switched Paths (LSPs) use common pieces of resource along their paths. This can help save network resources and is useful in scenarios such as LSP recovery or when two LSPs do not need to be active at the same time. A Path Computation Element (PCE) is responsible for path computation with such requirement.

Existing extensions to the Path Computation Element Protocol (PCEP) allow one path computation request for an LSP to be associated with other (existing) LSPs through the use of the PCEP Association Object.

This document extends PCEP in order to support resource-sharing-based path computation as another use of the Association Object to enable better efficiency in the computation and in the resultant paths and network resource usage.

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/1id-abstracts.txt.

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

This Internet-Draft will expire on October 21, 2020.

Copyright Notice

Copyright (c) 2020 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 and Motivation

A Path Computation Element (PCE) is a way to provide path computation function, and it is especially useful in the scenarios where complex constraints and/or a demanding amount of computation resource are required [RFC4655]. The development of PCE standardization has evolved from stateless to stateful. A stateful PCE has access to the LSP database information of the networks it serves as a computation engine [RFC8231]. Unless specified, this document assumes a PCE mentioned is a stateful PCE.

Resource sharing denotes that two or more Label Switched Paths (LSPs) share common pieces of resource, (such as a common time slot of a link in an Optical Transport Network (OTN)). This is usually useful in the scenario where only one of the LSPs is active and the benefit is to save network resources. A simple example of this is dynamically calculating a recovery LSP for an existing LSP undergoing a link failure. Note that resource sharing can be worked out using a stateless PCE, but the mechanism may be complex and is out the scope of this document.

This document considers the requirement that a new LSP may request for resource sharing with one or multiple existing LSPs. Furthermore, if there is resource sharing between a new LSP and existing an LSP, the two LSPs cannot be used to carry traffic simultaneously, the new LSP will take over the traffic from the existing LSP.

In a single domain, this is a common requirement in the recovery cases especially in order to increase traffic resilience against failure while reducing the amount of network resource used for recovery purposes [RFC4428].

The current protocol supporting the communication between a PCE and a Path Computation Client (PCC), i.e. PCE Protocol (PCEP), allows for re-optimization of an existing LSP [RFC5440]. This is achieved by setting the R bit in the Request Parameter (RP) object, together with some additional information if applicable, in the Path Computation Request (PCReq) message sent from a PCC to the PCE. To support this type of resource sharing, a PCC needs to ask a PCE to compute a new path with the constraints of sharing resource with one or multiple existing LSPs. It is worth noting the "resource sharing" in this draft not only means one LSP re-using the same links of another LSP, but also the same slice of bandwidth in the network. This may occur when an LSP is required for re-routing, or online re-optimization. Current PCEP specifications do not provide such function. More specifically, this document describes the resource sharing issue during the procedure when a new LSP is required to replace an existing LSP for use together with Make-before-break (MBB) described in [RFC3209].

As mentioned in [RFC8231], the PLSP-ID provides a unique identifier for an LSP during a PCEP session between PCC and PCE. Such identification is helpful in supporting the resource sharing requirement for stateful PCEs because it greatly simplifies the operation of a PCC. Instead of the PCC determining all the resources to be shared, the PCC can request that the PCE share the resources of a specific LSP: the stateful PCE is able to determine those resource itself.

Resource sharing can also be required in an inter-layer PCEP session. This is similar to the previous requirement. However, it is more complex and therefore deserves a more detailed explanation here.

In a multi-layer network, LSPs in a lower layer are used to carry higher-layer LSPs across the lower-layer network [RFC5623]. Therefore, the resource sharing constraints in the higher layer might actually relate to resource sharing in the lower layer. Thus, it is useful to consider how this can be achieved and whether additional extensions are needed using the models defined in [RFC5623].

In the next sections, use cases are provided to show what information needs to be exchanged to fulfill these requirements. This memo then provides extensions to PCEP to enable this function.

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. Motivation

2.1. Single Domain Use Case

There are two potential cases that request resource to be shared: restoration and re-optimization. Figure 1 shows a single domain network with a stateful PCE, and is used as an example for the resource sharing application.

```
                +---------------+
               |              |
                | Stateful PCE |
                |              |
                +---------------+



           +------+          +------+          +------+
           |  N1  +----------+  N2  +-----X---+  N3  |
           +--+---+          +---+--+          +---+--+
              |                  |                 |
              |                  +---------+       |
              |                            |       |
              |     +------+          +------+     |
              +-----+  N5  +----------+  N4  +-----+
                    +------+          +------+
```

Figure 1: A Single Domain Example

LSP0 (existing): N1-N2-N3
LSP1 (restoration): N1-N2-N4-N3
LSP2 (re-optimization): N1-N5-N4-N3

For the failure restoration, we can assume a working LSP (LSP0) exists in the network. When there is failure on the link N2-N3, it is desired to set up a restoration path for this working LSP. Suppose N1 serves as the PCC and sends a request to the stateful PCE for such an LSP. Before sending the request, N1 may need to check what policy should be applied for the restoration. For example, it might value resource sharing and prefer to share as much resource with the working LSP as possible and specify this policy in the PCReq message. Given such policy, a probable outcome from the path computation would be LSP1, which shares the link 'N1-N2' with the existing LSP.

Re-optimization does not usually result from a specific failure in the network, but takes place on a stable network when more optimal paths may have become available. Thus switching from the existing LSP to the new LSP happens with live traffic. An example can be found in Figure 1 without failure on the link N2-N3. Instead, an online re-optimization is needed for the working LSP (LSP0) from the stateful PCE. In such cases, the best choice is to set up a backup LSP for the working LSP with totally separate routing (for example, LSP2), and move the traffic to that backup LSP. After that, the working LSP can be torn down, which will not result in any interruption during the optimization procedure. This can actually be implemented with existing PCEP mechanisms. However, if there is no such separate path, existing PCEP mechanisms will return an error. A secondary option for this case is to set up an LSP and complete re-optimization with resource sharing, even if some interruption is introduced.

In the example from Figure 1 it is assumed that the restored LSP or re-optimized LSP have the same source and destination nodes. But in some applications there is no restriction for this assumption, i.e., after an LSP is failed, it can be restored as a new LSP with different source/destination.

In the use cases above it is also assumed that the characteristics of the restored LSP or re-optimized LSP are unchanged. However, it is possible to have parameter changes during the resource sharing computation. For example, the bandwidth of the request LSP may be different from the existing LSP, while resource sharing is still preferred by the PCC. The PCE should consider the sharing request together with the policy and available resources in the network. Details can be found in Section 3.3.

Conversely to resource sharing, it may also be required to apply a disjoint constraint for the path computation. [ietf-pce-association-diversity] discusses the solution under such a scenario, which is a companion work to this document.

2.2. Multiple Layers/Domains Use Case

As Discussed in Section 3 of [RFC5623], there are three models for inter-layer path computation. They are single PCE computation, multiple PCE with inter-PCE communication, and multiple PCE without inter-PCE communication. For the single PCE computation, the process would be similar to that of the use case in Section 2.1.

An inter-layer path computation example is shown in Figure 2. Assume an LSP (LSP1: H2-H3) has been established already, visible as H2-H3 from the view of the higher-layer PCE, and as H2-L1-L2-H3 from the global view (or from the view of the lower-layer PCE). A new request is received by H2 to establish a new LSP (LSP2: from H2 to H5), given the constraint that it can share resources with LSP1. This requirement is possible if only one of the LSPs needs to be active and resource sharing is the target.

```
                                                   -----
                             ..................................| LSR |
                           .:                                  | H5  |
                         .:                                   /-----
                       .:                                    /   |
       -----    -----.:                       -----    -----/    |
      | LSR |--| LSR |.......................| LSR |--| LSR |   /
      | H1  |  | H2  |                       | H3  |  | H4  |  /
       -----    -----\                       /-----    -----  /
                      \                     /                /
                       \                   /                /
                        \                 /                /
                         \               /                /
                          \-----    -----/               /
                          | LSR |-| LSR |               /
                          | L1  | | L2  |              /
                           -----   -----\             /
                             |           \           /
                             |            \         /
                             |             \       /
                           -----            \-----/
                          | LSR |-----------| LSR |
                          | L3  |           | L4  |
                           -----             -----
```

Figure 2: A Two-layer Network Example

If the model of multiple PCEs with inter-PCE communication is employed, the path computation request sent by H2 to higher-layer PCE will be forwarded to lower-layer PCE since there is no resource readily available in the higher layer. So it leaves the lower-layer PCE to compute a path in the lower layer in order to support the higher layer request. In this case, the lower-layer PCE is required to compute a path between H2 and H5 under the constraint that it can share the resource with that of LSP1. At this moment the lower-layer PCE has knowledge of the explicit route of LSP1 (H2-L1-L2-H3), and therefore can map the lower layer LSP with the higher-layer one. So when the lower-layer PCE computes the path for LSP2, it can consider the resource used by LSP1 as available with higher priority. For example, the lower-layer PCE may choose H2-L1-L2-L4-H5 as the computation result. On the other hand, if the path computation policy is to have a separate path with LSP1, the lower-layer PCE may choose H2-L1-L3-L4-H5.

During this procedure the higher-layer PCE can only use information about LSP1 (such as its five-tuple LSP information). An issue to solve is how the lower-layer PCE can resolve this information to the actual resource usage in its own layer, i.e. the lower layer. This could be solved by the edge LSR (L1) reporting this higher-lower LSP correlation to the lower-layer PCE as part of the LSP information during the LSP state synchronization process. If needed, it can be updated later when there is a change in this information. Alternatively, the lower-layer PCE can get this information from other sources, such as a network management system, where this information should be stored.

If the model of multiple PCEs without inter-PCE communication is employed, the path computation request in the lower layer will be initiated by the border LSR node, i.e., L1. The process would be similar to that of the previous scenario. A point worth noting is that the border LSR node may be able to resolve the higher layer LSP information itself, such as by mapping it to the corresponding LSP in the lower layer, in this way the lower-layer PCE does not need to perform this function. Otherwise, the mapping method mentioned above can still be used.

2.3. Bulk Path Computation Use Case

There is a potential need for resource sharing during bulk path computation, especially the processing of the "sticky resources" in [RFC7399]. It would be useful to specify the resources that can be shared among different paths, i.e., the bandwidth information.

Considering the H-PCE architecture in [RFC8751], when the parent PCE asks for a single path across a few domains, such a request may become a bulk path computation to a certain child PCE. Figure 3 shows an example of 3 domains. The parent PCE will select one of these path for establishment.

```
                +-------+
               /| P-PCE |\
              / +---+---+ \
             /      |      \
            /       |       \
```

```
                         /         |       \
                        /          |        \
                       /           |         \
                      /            |          \
               +-----/+        +---+---+      +\------+
               |C-PCE1|        |C-PCE2 |      |C-PCE3 |
               +------+        +-------+      +-------+
                /                  |                  \
    ---------------      -----------------------       -------------
   /                \   /                       \    /              \
   | +---+     +---+ |  |  +---+   +---+   +---+ |  | +---+    +---+ |
   | | A +-----+ B +-+--+--+ D +---+ E +---+ H +-+--+-+ J +----+ L | |
   | +-\-+     +---+ |  |  +---+   +---+   +--\+ |  | +---+    +-/-+ |
   |     \           |  |          /           \ |  |           /    |
   |      \          |  |         /             \|  |          /     |
   |        \  +---+ |  |  +---+ /               |\\|    +---+/      |
   |          \+ C +-+--+--+ G +/                |  |----| K |       |
   \           +---+/   \  +---+                /    \   +---+      /
    ---------------      -----------------------      --------------
```

Figure 3: Bulk Request example with Hierarchical PCEs

A 3-domain example is shown in Figure 3, with the hierarchical PCE architecture. In this example nodes A/B/C belong to domain 1, nodes D/E/G/H belong to domain 2, and nodes J/K/L belong to domain 3. Inter-domain links are B-D/C-G between domains 1 and 2, and H-J/H-K between domains 2 and 3. Given a path computation request from A to L, a bulk request from P-PCE would be helpful to understand whether it is possible to have different combinations on the inter-domain links. However, the resources on some specific links become 'sticky' and have to be indicated as 'sharing allowed' to avoid unnecessary resource competition. For example, both the route A-B-D-E-H-J-L and A-C-G-E-H-K-L are qualified, but these routes are competing for the resource on the link E-H and cannot be established simultaneously, so there must be one route failed to be reported to P-PCE. Given the indication of allowing resource sharing on the link E-H, both of these routes can be reported for P-PCE's decision, and there will not be any competition as the P-PCE understands that only one path needs to be set up.

3. Extensions to PCEP

3.1. Association Group and Type

According to the definition in [RFC8697], the association group is used to associate multiple LSPs into one group for further path computation considerations, such as disjointness and resource sharing. An association ID will be used to identify the resource sharing group. An association type that described disjointness has been defined in [ietf-pce-association-diversity]. In this document, a new association type is defined as follows:

Association type = TBD1 ("Sharing Association Type").

A sharing group should have multiple LSPs. The number of LSPs and the criteria for how LSPs share among each other are dependent on local policy.

3.2. Resource Sharing TLV

The PCEP Resource Sharing group MAY carry the following TLV. It MAY be carried within a PCReq message from the network element (or other PCCs) so as to indicate the desired resource sharing requirements to be applied by the stateful PCE during path computation.

```
 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 = [TBD2]         |            Length             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                   Flags                               |B|S|N|L|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         Optional TLVs                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The following flags are defined:

* L (Link share) bit: when set, this flag indicates that the PCE should prioritize the links shared by existing LSPs within the sharing group for path computation.

* N (Node share) bit: when set, this flag indicates that the PCE should prioritize the nodes shared by existing LSPs within the sharing group for path computation.

* S (SRLG share) bit: when set, this flag indicates that the PCE should set the SRLG (Shared Risk Link Group) of the computed LSP to the same as existing LSPs within the sharing group for path computation.

* B (Bandwidth share) bit: when set, this flag indicates that the PCE should prioritize the bandwidth to be shared by LSPs within the sharing group for bulk path computation.

It is worth noting that there can be multiple flags set which may conflict with each. In this scenario, the result for path computation will be dependent on the policy of PCE.

Optional TLVs may be needed to indicate the LSPs with which the resource is shared. If multiple LSPs are required, the PCE may need to consider different sharing policies, which is implementation dependent and may result in a different computing result. The selection policy among multiple computation result is out of the scope of this document.

3.3. Processing Rules

To request a path allowing resource sharing with one or multiple existing LSPs, a PCC includes a Resource Sharing TLV in the Association Group Object in any kind of path computation request message, such as the PCReq, PCUpd, or PCInitiate messages specified in [RFC8231] and [RFC8281].

On receipt of a PCEP message with a Resource Sharing TLV, a stateful PCE MUST proceed as follows:

- If the Resource Sharing TLV is unknown/unsupported, the PCE will follow procedures defined in [RFC5440]. That is, the PCE sends a PCErr message with error type 26 (Association Error) and error value 6 (Association Information Mismatch), and the related path computation request is discarded.

- If the Resource Sharing TLV is extracted correctly, the PCE MUST apply the requested resource sharing requirement.

The procedure of setting flags follows the rules defined in Section 3.1. The flags in the Resource Sharing TLV may be locally configured on the requesting nodes via external entities, such as a network management system or the entity that imposes the resource sharing requirement.

It is worth noting that the Resource Sharing TLV can be used together with other path indication objects like the IRO/XRO, with different objectives. The first difference is, the use of the Resource Sharing TLV is to set up an alternative path, instead a new path. It is also dependent on the knowledge held be the PCC, e.g., if the PCC has full knowledge of the path information and has a strong preference on the route, it may send the request message with an IRO to specify the route. On the other hand, if the PCC does not know how the path should go but just wants to set up a new LSP to replace the old one, it may use the Resource Sharing TLV instead of an IRO. The second difference is that the Resource Sharing TLV is a loose requirement. For example, if the constraint specified in an IRO/XRO in an A-Z path computation request cannot be satisfied, the reply message from PCE to PCC would be unsuccessful. However it is still possible to have a path from the A-Z. If the target node/link/SRLG/Bandwidth is set in the Resource Sharing TLV rather than an IRO, the PCE may feedback a path from A-Z that does not share the target specified in the Resource Sharing TLV.

4. Implementation Status

[Note to the RFC Editor - remove this section before publication, as well as remove the reference to [RFC7942].

Currently the authors are not aware of any implementations.

5. Manageability Considerations

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

5.1. Control of Function and Policy

A PCE or PCC implementation MUST allow operator-configured associations and SHOULD allow setting of the resource sharing TLV (Section 3.4) as described in this document.

5.2. Information and Data Models

An implementation SHOULD allow the operator to view the resource sharing configured or created dynamically. Further implementation SHOULD allow to view resource sharing associations reported by each peer, and the current set of LSPs in the association. The PCEP YANG module [ietf-pce-pcep-yang] includes association groups information.

5.3. Liveness Detection and Monitoring

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

5.4. Verify Correct Operations

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

5.5. Requirements on Other Protocols

Mechanisms defined in this document do not imply any new requirements on other protocols. The configuration on local policy may be accomplished by other protocols, such as Netconf.

5.6. Impact on Network Operations

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

6. Security Considerations

Security of PCEP is discussed in [RFC5440] and [RFC6952]. The extensions in this document do not change the fundamentals of security for PCEP.

However, the introduction of the Resource Sharing TLV in the Association Group Object provides a vector that may be used to probe for information from a network. For example, a PCC that wants to discover the path of an LSP with which it is not involved can issue a request message with a Resource Sharing TLV and may be able to get back quite a lot of information about the path of the LSP through issuing multiple such requests for different endpoints and analyzing the received results. To protect against this, a PCE SHOULD be configured with access and authorization controls such that only authorized PCCs (for example, those within the network) can make computation requests, only specifically authorized PCCs can make requests for resource sharing, and such requests relating to specific LSPs are further limited to a select few PCCs. How such access controls and authorization is managed is outside the scope of this document, but it will at the least include Access Control Lists.

Furthermore, a PCC must be aware that setting up an LSP that shares resources with another LSP may be a way of attacking the other LSP, for example by depriving it of the resources it needs to operate correctly. Thus it is important that, both in PCEP and the associated signaling protocols, only authorized resource sharing is allowed.

7. IANA Considerations

7.1. Association Object Type Indicators

IANA maintains a registry called the "Path Computation Element Protocol (PCEP) Numbers" registry with a subregistry called the "Association Type Field" subregistry. IANA is requested to make an assignment from that subregistry as follows:

Object Class	Name	Object Type	Reference
TBD1	Sharing-group	Association Type	[this document]

7.2. PCEP TLV Definitions

This document defines the following TLVs to support the resource sharing scenario:

Value	Name	Reference
TBD2	Resource-sharing TLV	[this document]

IANA is requested to allocate the following bit numbers in the flag spaces of Resource-sharing TLV:

Bit	Flag name	Reference
31	Link Share	[this document]
30	Node Share	[this document]
29	SRLG Share	[this document]
28	Bandwidth Share	[this document]

8. References

8.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to indicate requirements levels", RFC 2119, 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, December 2001. <https://www.rfc-editor.org/info/rfc3209>.

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

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

[RFC8231] Crabbe, E., Medved, J., Minei, I., and R. Varga, "PCEP Extensions for Stateful PCE", RFC8231, June 2017. <https://www.rfc-editor.org/info/rfc8231>.

[RFC8281] Crabbe, E., Minei, I., Sivabalan, S., and R. Varga, "PCEP Extensions for PCE-initiated LSP Setup in a Stateful PCE Model", RFC 8281, October 2017. <https://www.rfc-editor.org/info/rfc8281>.

[RFC8697] Minei, I., Crabbe E., Sivabalan S., Ananthakrishnan H., Dhody D., Tanaka Y., "PCEP Extensions for Establishing Relationships Between Sets of LSPs", RFC8697, January 2020. <https://www.rfc-editor.org/info/rfc8697>.

[ietf-pce-association-diversity] Litkowski, S., Sivabalan, S., Barth, C., Dhody, D., "Path Computation Element communication Protocol extension for signaling LSP diversity constraint", work in progress.

8.2. Informative References

[RFC4428] Papadimitriou, D., Mannie., E., "Analysis of Generalized Multi-Protocol Label Switching (GMPLS)-based Recovery Mechanisms (including Protection and Restoration)", RFC4428, March 2006. <https://www.rfc-editor.org/info/rfc4428>.

[RFC4655] Farrel, A., Vasseur, J.-P., and Ash, J., "A Path Computation Element (PCE)-Based Architecture", RFC 4655, August 2006. <https://www.rfc-editor.org/info/rfc4655>.

[RFC5623] Oki., E., Takeda, T., Le Roux, JL., Farrel, A., "Framework for PCE-Based Inter-Layer MPLS and GMPLS Traffic Engineering", RFC5623, September 2009. <https://www.rfc-editor.org/info/rfc5623>.

[RFC6952] Jethanandani, M., Patel, K., Zheng, L., "Analysis of BGP, LDP, PCEP, and MSDP Issues According to the Keying and Authentication for Routing Protocols (KARP) Design Guide", RFC6952, May 2013. <https://www.rfc-editor.org/info/rfc6952>.

[RFC7399] Farrel, A., King, D., "Unanswered Questions in the Path Computation Element Architecture", RFC7399, October 2014. <https://www.rfc-editor.org/info/rfc7399>.

[RFC7942] Sheffer, Y., Farrel, A., "Improving Awareness of Running Code: The Implementation Status Section", RFC7942, July 2016. <https://www.rfc-editor.org/info/rfc7942>.

[RFC8751] Dhody, D., Lee, Y., Ceccarelli, D., Shin, J., King, D., Gonzalez de Dios, O., "Hierarchical Stateful Path Computation Element (PCE)", RFC8751, March 2020. <https://www.rfc-editor.org/info/rfc8751>.

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

9. Acknowledgements

The authors would like to thank Adrian Farrel for his review and valuable comments.

10. Contributor's Address

Dhruv Dhody
Huawei Technologies
Email: dhruv.dhody@huawei.com

Igor Bryskin
Huawei Technologies

Email: Igor.Bryskin@huawei.com

11. Authors' Addresses

Xian Zhang
Huawei Technologies
Email: zhang.xian@huawei.com

Haomian Zheng
Huawei Technologies
Email: zhenghaomian@huawei.com

Oscar Gonzalez de Dios
Telefonica I+D/gCTIO
Distrito Telefonica
E-28050 Madrid, Spain
EMail: oscar.gonzalezdedios@telefonica.com

Victor Lopez
Telefonica I+D/gCTIO
Distrito Telefonica
E-28050 Madrid, Spain
EMail: victor.lopezalvarez@telefonica.com

Yunbin Xu
CAICT
xuyunbin@caict.ac.cn

PCE Working Group
Internet-Draft
Intended status: Experimental
Expires: September 3, 2015

Q. Zhao
K. Zhao
Z. Li
D. Dhody
U. Palle
Huawei Technologies
B. Zhang
Telus Ltd.

March 2, 2015

PCEP Procedures and Protocol Extensions for Using PCE as a Central Controller (PCECC) of LSPs
draft-zhao-pce-pcep-extension-for-pce-controller-01

Abstract

In certain networks deployment scenarios, service providers would like to keep all the existing MPLS functionalities in both MPLS and GMPLS while removing the complexity of existing signaling protocols such as LDP and RSVP-TE. In [I-D.zhao-pce-central-controller-user-cases], we propose to use the PCE [RFC5440] as a central controller (PCECC) so that LSP can be calculated/ signaled/initiated and label forwarding entries are downloaded through a centralized PCE server to each network devices along the LSP 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 and user cases where LSPs are calculated/setup/initiated and label forwarding entries are downloaded through extending the existing PCE architectures and PCEP.

This document also discuss the role of PCECC in Segment Routing (SR).

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 3, 2015.

Copyright Notice

Copyright (c) 2015 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

In certain network deployment scenarios, service providers would like to have the ability to dynamically adapt to a wide range of customer's requests for the sake of flexible network service delivery, Software Defined Networks(SDN) has provides additional flexibility in how the network is operated compared to the traditional network.

The existing networking ecosystem has become awfully complex and highly demanding in terms of robustness, performance, scalability, flexibility, agility, etc. By migrating to the SDN enabled network from the existing network, service providers and network operators must have a solution which they can evolve easily from the existing network into the SDN enabled network while keeping the network services remain scalable, guarantee robustness and availability etc.

Taking the smooth transition between traditional network and the new SDN enabled network into account, especially from a cost impact assessment perspective, using the existing PCE components from the current network to function as the central controller of the SDN network is one choice, which not only achieves the goal of having a centralized controller, but also leverages the existing PCE network components.

The Path Computation Element communication Protocol (PCEP) provides mechanisms for Path Computation Elements (PCEs) to perform route computations in response to Path Computation Clients (PCCs) requests. PCEP Extensions for PCE-initiated LSP Setup in a Stateful PCE Model [I-D.ietf-pce-stateful-pce] describes a set of extensions to PCEP to enable active control of MPLS-TE and GMPLS tunnels.

[I-D.ietf-pce-pce-initiated-lsp] describes the setup and teardown of PCE-initiated LSPs under the active stateful PCE model, without the need for local configuration on the PCC, thus allowing for a dynamic MPLS network that is centrally controlled and deployed.

[I-D.ietf-pce-remote-initiated-gmpls-lsp] complements [I-D.ietf-pce-pce-initiated-lsp] by addressing the requirements for remote-initiated GMPLS LSPs.

Segment Routing (SR) technology leverages the source routing and tunneling paradigms. A source node can choose a path without relying on hop-by-hop signaling protocols such as LDP or RSVP-TE. Each path is specified as a set of "segments" advertised by link-state routing protocols (IS-IS or OSPF). [I-D.ietf-spring-segment-routing] provides an introduction to SR technology. The corresponding IS-IS and OSPF extensions are specified in [I-D.ietf-isis-segment-routing-extensions] and [I-D.ietf-ospf-segment-routing-extensions], respectively.

A Segment Routed path (SR path) can be derived from an IGP Shortest Path Tree (SPT). Segment Routed Traffic Engineering paths (SR-TE paths) may not follow IGP SPT. Such paths may be chosen by a suitable network planning tool and provisioned on the source node of the SR-TE path.

It is possible to use a stateful PCE for computing one or more SR-TE paths taking into account various constraints and objective functions. Once a path is chosen, the stateful PCE can instantiate an SR-TE path on a PCC using PCEP extensions specified in [I-D.ietf-pce-pce-initiated-lsp] using the SR specific PCEP extensions described in [I-D.ietf-pce-segment-routing].

PCECC may further use PCEP protocol for SR label distribution instead of IGP extensions with some benefits.

Current MPLS label has local meaning. That is, MPLS label is always allocated by the downstream node to the upstream node. Then the MPLS label is only identified by the neighboring upstream node and downstream node. The label allocation is done locally and signaled through the LDP/RSVP-TE/BGP protocol. To ease the label allocation and signaling mechanism, PCE can be conveniently used as a central controller with Label download capability. Further PCE can also be used to manage the label range and SRGB etc.

The PCECC solution introduced in [I-D.zhao-pce-central-controller-user-cases] allow for a dynamic MPLS network that is eventually controlled and deployed without the deployment of RSVP-TE protocol or extended IGP protocol with node/ adjacency segment identifiers signaling capability while providing all the key MPLS functionalities needed by the service providers.

This draft specify the procedures and PCEP protocol extensions for using the PCE as the central controller and user cases where LSPs are calculated/setup/initiated/downloaded through extending the existing PCE architectures and PCEP.

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 following terminology is used in this document.

IGP: Interior Gateway Protocol. Either of the two routing protocols, Open Shortest Path First (OSPF) or Intermediate System to Intermediate System (IS-IS).

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.

TE: Traffic Engineering.

3. PCECC Modes

The following PCECC modes are supported -

- o Basic PCECC.
- o PCECC SR.
    - * PCECC SR-BE (Best Effort).

* PCECC SR-TE (Traffic Engineered).

In basic PCECC mode, the forwarding is similar to RSVP-TE signalled LSP without the RSVP-TE signaling. The PCECC allocates and download the label entries along the LSP. The rest of processing is similar to the existing stateful PCE mechanism.

In case of SR, there are two modes for SR-BE and SR-TE. For SR-BE, the forwarding is similar to LDP LSP without LDP signaling or IGP-SR extension. The SR Node label are allocated and distributed in the domain centrally by the PCE via PCEP. Each node (PCC) rely on local IGP for the nexthop calculation. For SR-TE, the forwarding uses label stack similar to IGP based SR-TE without IGP-SR extension. The SR node and adj labels are allocated and distributed in the domain centrally by the PCE via PCEP by PCECC. Rest of the processing is similar to existing stateful PCE with SR 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. Path Computation Client (PCC) supporting this draft MUST have a capability to communicate local label range or global label range or both to PCE.

3. Path Computation Element (PCE) supporting this draft SHOULD have the capability to negotiate a global label range for a group of clients and communicate the final global label range to PCC.

4. PCEP speaker not supporting this draft MUST be able to reject PCECC related message with a reason code that indicates no support for PCECC.

5. PCEP SHOULD provide a means to identify PCECC based LSP in the PCEP messages.

6. PCEP SHOULD provide a means to update (or cleanup) the label-download or label-map entry to the PCC.

5. Procedures for Using the PCE as the Central Controller (PCECC)

5.1. Stateful PCE Model

Active stateful PCE is described in [I-D.ietf-pce-stateful-pce]. 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:

(PCLRResv): a PCEP message sent by a PCC to a PCE to ask for the label range reservation or a PCE to a PCC to send the reserved label range. The PCLRResv message described in Section 6.1.

(PCLabelUpd): a PCEP message sent by a PCE to a PCC to download or cleanup the Label entry. The PCLabelUpd message described in Section 6.2.

(PCRpt): a PCEP message described in [I-D.ietf-pce-stateful-pce]. PCRpt message MAYBE used to send PCECC LSP Reports.

(PCInitiate): a PCEP message described in [I-D.ietf-pce-pce-initiated-lsp]. PCInitiate message is used to setup PCE-Initiated LSP based on PCECC mechanism.

(PCUpd): a PCEP message described in [I-D.ietf-pce-stateful-pce]. 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 Range Reservation	PCLRResv
Label entry Update	PCLabelUpd
Label entry Cleanup	PCLabelUpd
PCECC Initiated LSP	PCInitiate
PCECC LSP Update	PCUpd
PCECC LSP State Report	PCRpt
PCECC LSP Delegation	PCRpt

5.3. PCECC Capability Advertisement

During PCEP Initialization Phase, PCEP Speakers (PCE or PCC) advertise their support of PCECC extensions. A PCEP Speaker includes the "PCECC Capability" TLV, described in Section 7.1.1 of this document, in the OPEN Object to advertise its support for PCECC extensions.

The presence of the PCECC Capability TLV in PCC’s OPEN Object indicates that the PCC is willing to function as a PCECC client.

The presence of the PCECC Capability 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 PCECC Capability 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 LSP setup/download/label-range reservation if PCECC capability was not advertised) will be generated and the PCEP session will be terminated.

L flag and G flag defined in PCECC Capability TLV specifies the local and global label range reservation capability.

A PCC or a PCE MUST include both PCECC-CAPABILITY TLV and STATEFUL-PCE-CAPABILITY TLV in OPEN Object to support the extensions defined in this document. If PCECC-CAPABILITY 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. Label Range Reservation

After PCEP initial state synchronization, the label range is reserved.

If L flag is advertised in OPEN Object by PCEP speakers, a PCC reserves a local label range and is communicated using PCLRResv message to a PCE. The PCE maintains the local label range of each node and further during LSP setup, a label is assigned to each node from the corresponding local label range reserved.

If G flag is advertised in OPEN Object by PCEP speakers,a PCC reserves a global label range and is advertised in PCLRResv message to a PCE. The PCE MAY negotiate and reserves the global label range and also sends the negotiated global label range in PCLRResv message to the PCC. Please refer [I-D.li-mpls-global-label-framework] for MPLS global label allocation.

A PCC MUST send PCLRResv message immediately after the initial LSP synchronization completion. A PCE SHOULD not send PCLabelUpd message to a PCC before PCLRResv message received. If the PCC received PCLabelUpd message and not initiated label range reservation, it SHOULD send a PCErr message with Error-type=[TBD] (label range not reserved) and Error-value=[TBD].

The label range reservation sequence is shown below.

```
     +-+-+                           +-+-+
     |PCC|                           |PCE|
     +-+-+                           +-+-+
       |                               |
       |--- PCLRResv, label type=1 --->|local label range reserved
       |             (100-500)         |global label range negotiated
       |             label type=2      |
       |             (600-1000)        |
       |                               |
       |<--- PCLRResv, label type=2 ---|global label range reserved
       |             (700-1000)        |
       |                               |
```

[Editor’s Note: This section of the document would be updated with more details about Label Block Negotiation, Reservation, Adjustment etc in a future revision of the document.]

5.5. LSP Operations

The PCEP messages pertaining to PCECC MUST include PATH-SETUP-TYPE TLV [I-D.sivabalan-pce-lsp-setup-type] in the SRP object to clearly identify the PCECC LSP is intended.

5.5.1. Basic PCECC Mode

5.5.1.1. PCECC LSP Setup

Inorder to setup a LSP based on PCECC mechanism, a PCC MUST delegate the LSP by sending a PCRpt message with Path Setup Type set for basic PCECC (see Section 7.3) and D (Delegate) flag (see [I-D.ietf-pce-stateful-pce]) set in the LSP object.

The LSP-ID in LSP-IDENTIFIER TLV (which usually corresponds to the RSVP-TE LSP-ID) for PCECC LSP MUST always be generated by the PCE. In the first PCRpt message of PCECC LSP, LSP ID of LSP-IDENTIFIER TLV is set to zero.

When a PCE received PCRpt message with P and D flags set, it generates LSP ID; calculates the path and assign labels along the path; and setup the path by sending PCLabelUpd message to each node along the path of the LSP.

The PCE SHOULD send the PCUpd message with the same PLSP-ID to the Ingress PCC in response to the delegate PCRpt message.

The PCECC LSPs MUST be delegated to a PCE at all times.

LSP deletion operation for PCECC LSP is same as defined in [I-D.ietf-pce-stateful-pce]. If the PCE received PCRpt message for LSP deletion then it does Label cleanup operation as described in Section 5.5.1.3 for the corresponding LSP.

The Basic PCECC LSP setup sequence is as shown below.

```
                 +-------+                           +-------+
                 |PCC    |                           |  PCE  |
                 |Ingress|                           +-------+
          +------|       |                               |
          | PCC  +-------+                               |
          | Transit| |                                   |
   +------|        | |--- PCRpt,PLSP-ID=1, P=1, D=1  --->| PCECC LSP
   |PCC   +--------+ |       (LSP ID=0)                  |(LSPID=1)
   |Egress  |  |     |                                   |
   +--------+  |     |                                   |
       |       |     |                                   |
       |<------ PCLabelUpd,PLSP-ID=1  ------------------ | Label
       |       |     |       (LSP ID=1)                  | download
       |       |     |                                   |
       |       |<----- PCLabelUpd,PLSP-ID=1  ----------- | Label
       |       |     |      (LSP ID=1)                   | download
       |       |     |                                   |
       |       |     |<--- PCLabelUpd,PLSP-ID=1  ------- | Label
       |       |     |      (LSP ID=1)                   | download
       |       |     |                                   |
       |       |     |<-- PCUpd,PLSP-ID=1, P=1, D=1 ---- | PCECC LSP
       |       |     |      (LSP ID=1)                   | Update
       |       |     |                                   |
```

The 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.

5.5.1.2. Label Download

Inorder to setup an LSP based on PCECC, the PCE sends a PCLabelUpd message to each node of the LSP to download the Label entry as described in Section 5.5.1.1.

The LSP object in PCLabelUpd MUST include the LSP-IDENTIFIER TLV.

If a node (PCC) received a PCLabelUpd message but failed to download the Label entry, it MUST send a PCErr message with Error-type=[TBD] (label download failed) and Error-value=[TBD].

5.5.1.3. Label Cleanup

Inorder to delete an LSP based on PCECC, the PCE sends a PCLabelUpd message to each node along the path of the LSP to cleanup the Label entry.

If the PCC received a PCLabelUpd message but does not recognize the label, the PCC MUST generate a PCErr message with Error-Type 19(Invalid operation) and Error-Value=3, "Unknown Label".

The R flag in SRP object defined in [I-D.ietf-pce-pce-initiated-lsp] specifies the deletion of Label Entry in the PCLabelUpd message.

```
                 +-------+                           +-------+
                 |PCC    |                           |  PCE  |
                 |Ingress|                           +-------+
          +------|       |                               |
          | PCC  +-------+                               |
          | Transit| |                                   |
   +------|        | |--- PCRpt,PLSP-ID=1,P=1,D=1,R=1--->| PCECC LSP
   |PCC   +--------+ |       (LSP ID=1)                  | remove
   |Egress  |  |     |                                   |
   +--------+  |     |                                   |
       |       |     |                                   |
       |<------ PCLabelUpd, PLSP-ID=1 ------------------ | Label
       |       |     |       (LSP ID=1, R=1)             | cleanup
       |       |     |                                   |
       |       |<----- PCLabelUpd, PLSP-ID=1 ----------- | Label
       |       |     |       (LSP ID=1, R=1)             | cleanup
       |       |     |                                   |
       |       |     |<--- PCLabelUpd, PLSP-ID=1 ------- | Label
       |       |     |       (LSP ID=1, R=1)             | cleanup
       |       |     |                                   |
```

5.5.1.4. PCE Initiated PCECC LSP

The LSP Instantiation operation is same as defined in [I-D.ietf-pce-pce-initiated-lsp].

Inorder to setup a PCE Initiated LSP based on PCECC mechanism, a PCE sends PCInitiate message with Path Setup Type set for basic PCECC (see Section 7.3) to the Ingress PCC.

The Ingress PCC MUST also set D (Delegate) flag (see [I-D.ietf-pce-stateful-pce]) and C (Create) flag (see [I-D.ietf-pce-pce-initiated-lsp]) in LSP object of PCRpt message. The PCC responds with first PCRpt message with the status as "GOING-UP" and assigned PLSP-ID.

The rest of the PCECC LSP setup operations are same as those described in Section 5.5.1.1.

The LSP deletion operation for PCE Initiated PCECC LSP is same as defined in [I-D.ietf-pce-pce-initiated-lsp]. The PCE should further perform Label entry cleanup operation as described in Section 5.5.1.3 for the corresponding LSP.

The PCE Initiated PCECC LSP setup sequence is shown below.

```
                 +-------+                             +-------+
                 |PCC    |                             |  PCE  |
                 |Ingress|                             +-------+
          +------|       |                                 |
          | PCC  +-------+                                 |
          | Transit| |                                     |
   +------|        | |<---PCInitiate,PLSP-ID=0,P=1,D=1---| PCECC LSP
   |PCC   +--------+ |                                   | Initiate
   |Egress  |  |     |--- PCRpt,PLSP-ID=2,P=1,D=1,C=1--->| PCECC LSP
   +--------+  |     |       (LSP ID=0,GOING-UP)         |(LSPID=2
       |       |     |                                   | assigned)
       |<------ PCLabelUpd, PLSP-ID=2 ------------------ | Label
       |       |     |       (LSP ID=2)                  | download
       |       |     |                                   |
       |       |<----- PCLabelUpd, PLSP-ID=2 ----------- | Label
       |       |     |       (LSP ID=2)                  | download
       |       |     |                                   |
       |       |     |<--- PCLabelUpd, PLSP-ID=2 ------- | Label
       |       |     |       (LSP ID=2)                  | download
       |       |     |                                   |
       |       |     |<-- PCUpd, PLSP-ID=2, P=1, D=1---- | PCECC LSP
       |       |     |       (LSP ID=2)                  | Update
       |       |     |                                   |
```

5.5.1.5. PCECC LSP Update

Incase of a modification of PCECC LSP with a new path, a PCE sends a PCUpd message to the Ingress PCC.

When a PCC received a PCUpd message for an existing LSP, a PCC MUST follow the make-before-break procedure. On successful traffic switch over to the new LSP, PCC sends a PCRpt message to the PCE for the deletion of old LSP. Further the PCE does cleanup operation for the old LSP described in Section 5.5.1.3.

The PCECC LSP Update and make-before-break sequence is shown below.

```
                 +-------+                            +-------+
                 |PCC    |                            |  PCE  |
                 |Ingress|                            +-------+
          +------|       |                                |
          | PCC  +-------+                                |
          | Transit| |                                    |
   +------|        | |                                    |
   |PCC   +--------+ |                                    |
   |Egress  |  |     |                                    |
   +--------+  |     |                                    |
       |       |     |                                    | Modify LSP
       |<------ PCLabelUpd, PLSP-ID=1 ------------------  | (LSPID=3
       |       |     |       (LSP ID=3)                   | assigned)
       |       |     |                                    |
       |       |<----- PCLabelUpd, PLSP-ID=1 -----------  | Label
       |       |     |      (LSP ID=3)                    | download
       |       |     |                                    |
       |       |     |<--- PCLabelUpd, PLSP-ID=1 -------  | Label
       |       |     |      (LSP ID=3)                    | download
       |       |     |                                    |
       |       |     |<-- PCUpd, PLSP-ID=1, P=1, D=1----  | PCECC
       |       |     |       (LSP ID=3)                   | LSP Update
       |       |     |                                    |
       |       |     |--- PCRpt,PLSP-ID=1,P=1,D=1,R=1--->| Delete
       |       |     |       (LSP ID=1)                   | old LSP
       |       |     |                                    |
       |<------ PCLabelUpd, PLSP-ID=1 ------------------  | Label
       |       |     |       (LSP ID=1, R=1)              | cleanup
       |       |     |                                    |
       |       |<----- PCLabelUpd, PLSP-ID=1 -----------  | Label
       |       |     |      (LSP ID=1, R=1)               | cleanup
       |       |     |                                    |
       |       |     |<--- PCLabelUpd, PLSP-ID=1 -------  | Label
       |       |     |      (LSP ID=1, R=1)               | cleanup
```

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.

5.5.1.6. 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.

5.5.2. PCECC Segment Routing (SR)

Segment Routing (SR) as described in [I-D.ietf-spring-segment-routing] depends on "segments" that are advertised by Interior Gateway Protocols (IGPs). The SR-node allocate and advertise the SID (node, adj etc) and flood via the IGP. This document proposes a new mechanism where PCE allocate the SID (label) centrally and uses PCEP to advertise the SID. In some deployments PCE (and PCEP) are better suited than IGP because of centralized nature of PCE and direct TCP based PCEP session to the node.

5.5.2.1. PCECC SR-BE

Each node (PCC) is allocated a node-SID (label) by the PCECC. The PCECC sends PCLabelUpd to update the label map of each node to all the nodes in the domain. Each node (PCC) uses the local information to determines the next-hop and download the label forwarding instructions accordingly. The PCLabelUpd message in this case MUST not have LSP object but uses new FEC object.

```
                 +-------+                           +-------+
                 |PCC    |                           |  PCE  |
                 |3.3.3.3|                           +-------+
          +------|       |                               |
          | PCC  +-------+                               |
          | 2.2.2.2| |                                   |
   +------|        | |                                   |
   |PCC   +--------+ |                                   |
   |1.1.1.1 |  |     |                                   |
   +--------+  |     |                                   |
       |       |     |                                   |
       |<------ PCLabelUpd, FEC=1.1.1.1----------------- | Label Map
       |       |     |      Label=X                      | update
       |Find   |     |                                   |
       |Nexthop|<----- PCLabelUpd, FEC=1.1.1.1---------- | Label Map
       |locally|     |             Label=X               | update
       |       |     |                                   |
       |       |     |<--- PCLabelUpd, FEC=1.1.1.1------ | Label Map
       |       |     |                 Label=X           | update
       |       |     |                                   |
```

The forwarding behavior and the end result is similar to IGP based "Node-SID" in SR. Thus, from anywhere in the domain, it enforces the ECMP-aware shortest- path forwarding of the packet towards the related node.

PCE rely on the Node label cleanup using the same PCLabelUpd message.

5.5.2.2. PCECC SR-TE

A Segment Routed Best Effort path (SR-BE path) can be derived from an IGP Shortest Path Tree (SPT) as explained above. On the other hand, SR-TE paths may not follow IGP SPT. Such paths may be chosen by a PCE and provisioned on the source node of the SR-TE path.

[I-D.ietf-pce-segment-routing] extends PCEP to allow a stateful PCE to compute and initiate SR-TE paths, as well as a PCC to request a path subject to certain constraint(s) and optimization criteria in SR networks.

For SR-TE, apart from node-SID, Adj-SID is used where each adjacency is allocated an Adj-SID (label) by the PCECC. The PCECC sends PCLabelUpd to update the label map of each Adj to the corresponding nodes in the domain. Each node (PCC) download the label forwarding instructions accordingly. Similar to SR-BE, the PCLabelUpd message in this case MUST not have LSP object but uses new FEC object.

```
                 +-------+                           +-------+
                 |PCC    |                           |  PCE  |
                 |3.3.3.3|                           +-------+
          +------|       |                               |
          | PCC  +-------+                               |
          | 2.2.2.2| |                                   |
   +------|        | |                                   |
   |PCC   +--------+ |                                   |
   |1.1.1.1 |  |     |                                   |
   +--------+  |     |                                   |
       |       |     |                                   |
       |<------ PCLabelUpd, FEC=10.1.1.1 / ------------- | Label Map
       |       |     |          10.1.1.2                 | update
       |       |     |      Label=A                      |
       |       |     |                                   |
       |       |<----- PCLabelUpd, FEC=10.1.1.2--------- | Label Map
       |       |     |                 10.1.1.1          | update
       |       |     |             Label=B               |
       |       |     |                                   |
```

The forwarding behavior and the end result is similar to IGP based "Adj-SID" in SR.

The Path Setup Type MUST be set for PCECC SR-TE (see Section 7.3). The rest of the PCEP procedures and mechanism are similar to [I-D.ietf-pce-segment-routing].

PCE rely on the Adj label cleanup using the same PCLabelUpd message.

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.

6.1. The PCLRResv message

A Label Range Reservation message (also referred to as PCLRResv message) is a PCEP message sent by a PCC to a PCE for the reservation of label range or by PCE to PCC to send reserved label range for the network. The Message-Type field of the PCEP common header for the PCLRResv message is set to [TBD].

The format of a PCLRResv message is as follows:

```
PCLRResv Message>::= <Common Header>
                             <label-range>
   Where:

    <label-range> ::= <SRP>
                      <labelrange-list>

   Where
<labelrange-list>::=<LABEL-RANGE>[<labelrange-list>]
```

There are two mandatory objects that MUST be included within each <label-range> in the PCLRResv message: the SRP Object and LABEL-RANGE object.

SRP object is defined in [I-D.ietf-pce-stateful-pce] and this document extends the use of SRP object in PCLRResv message. If the SRP object is missing, the receiving PCE MUST send a PCErr message with Error-type=6 (Mandatory Object missing) and Error-value=10 (SRP object missing).

PCC generates the value of SRP-ID-number in SRP object of PCLRResv message send to a PCE. The PCE MUST include the same SRP-ID-number in SRP object of PCLRResv message sent to the PCC in response to PCLRResv message.

LABEL-RANGE object is defined in Section 7.2. If the LABEL-RANGE object is missing, the receiving PCE MUST send a PCErr message with Error-type=6 (Mandatory Object missing) and Error-value=[TBD] (Label object missing).

[Editor’s Note: This section of the document would be updated with more details about Label Block Negotiation, Reservation, Adjustment etc in a future revision of the document.]

6.2. The PCLabelUpd message

The Label Update Message (also referred to as PCLabelUpd) is a PCEP message sent by a PCE to a PCC to download label or update the label map. The same message is also used to cleanup the Label entry. The Message-Type field of the PCEP common header for the PCLabelUpd message is set to [TBD].

The format of the PCLabelUpd message is as follows:

```
<PCLabelUpd Message> ::= <Common Header>
                                 <pce-label-update-list>
Where:
<pce-label-update-list> ::= <pce-label-update>
                           [<pce-label-update-list>]

<pce-label-update> ::= (<pce-label-download>|<pce-label-map>)

Where:
<pce-label-download> ::= <SRP>
                                 <LSP>
                         <label-list>

<pce-label-map> ::= <SRP>
                        <LABEL>
                        <FEC>

<label-list > ::=  <LABEL>
                          [<label-list>]
```

The PCLabelUpd message is used to download label along the path of the LSP for the basic PCECC mode, as well as to update the label map for the Node and Adjacency Label in case of SR.

The SRP object is defined in [I-D.ietf-pce-stateful-pce] and this document extends the use of SRP object in PCLabelUpd message. The SRP object is mandatory and MUST be included in PCLabelUpd message. 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 [I-D.ietf-pce-stateful-pce] and this document extends the use of LSP object in PCLabelUpd message. The LSP is an optional object and used in the basic PCECC mode in PCLabelUpd message. LSP Identifiers TLV is defined in [I-D.ietf-pce-stateful-pce], it MUST be included in the LSP object in PCLabelUpd message. If the TLV is missing, the PCC will generate a PCErr message with Error-Type=6 (mandatory object missing) and Error-Value=11 (LSP-IDENTIFIERS TLV missing) and close the session.

The LABEL object is defined in Section 7.4. The LABEL is the mandatory object and MUST be included in PCLabelUpd message. If the LABEL object is missing, the receiving PCC MUST send a PCErr message with Error-type=6 (Mandatory Object missing) and Error-value=[TBD] (LABEL object missing). More than one LABEL object MAY be included in the PCLabelUpd message for the transit LSR.

The FEC object is defined in Section 7.5. The FEC is an optional object and used in PCECC SR mode in PCLabelUpd message. The FEC object encodes the Node and Adjacency information of the Label Map.

To cleanup the SRP object must set the R (remove) bit.

7. PCEP Objects

The PCEP objects defined in this document are compliant with the PCEP object format defined in [RFC5440]. The P flag and the I flag of the PCEP objects defined in this document MUST always be set to 0 on transmission and MUST be ignored on receipt since these flags are exclusively related to path computation requests.

7.1. OPEN Object

This document defines a new optional TLV for use in the OPEN Object.

7.1.1. PCECC Capability TLV

The PCECC-CAPABILITY TLV is an optional TLV for use in the OPEN Object for PCECC capability advertisement. 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:

```
 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]      |            Length=4           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                             Flags                         |G|L|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

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):

L (LOCAL-LABEL-RANGE-CAPABILITY - 1 bit): If set to 1 by a PCEP speaker, it indicates that the PCEP speaker is capable for local label range reservation.

G (GLOBAL-LABEL-RANGE-CAPABILITY - 1 bit): If set to 1 by a PCEP speaker, it indicates that the PCEP speaker capable for global label range reservation.

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

7.2. LABEL-RANGE Object

The LABEL-RANGE object MUST be carried within PCLRResv message. The LABEL-RANGE object is used to carry the label range information based on the label type.

LABEL-RANGE Object-Class is TBD.

LABEL-RANGE Object-Type is 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  label type  |                 range size                     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                            label  base                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
//                        Optional TLVs                        //
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

label type (8 bit): The values defined for label type are label type 1 specifies the local label. It means the label range is non negotiable. label type 2 specifies the global label. It means the label range is negotiable. Refer [I-D.li-mpls-global-label-framework] for global label.

Range size (24 bit): It specifies the size of label range.

Label base (32 bit): It specifies the minimum label of label range.

7.3. 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 = 2: Path is setup via Basic PCECC mode.

- o PST = 3: Path is setup via PCECC SR-TE mode.

On a PCRpt or PCInitiate message, the PST=2 in PATH-SETUP-TYPE TLV in SRP object indicates that this LSP was setup via a basic PCECC based mechanism; the PST=3 indicates that this LSP was setup via a PCECC SR-TE based mechanism.

7.4. Label Object

The LABEL Object is used to specify the Label information and MUST be carried within PCLabelUpd message.

LABEL Object-Class is TBD.

LABEL Object-Type is 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Reserved           |              Flags            |O|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                             Label                             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
//                         Optional TLV                        //
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The fields in the LABEL object are as follows:

Flags: is used to carry any additional information pertaining to the label. 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 nexthop information (via NEXTHOP-IPV4-ADDRESS TLV or NEXTHOP-IPV6-ADDRESS TLV or NEXTHOP-UNNUMBERED-IPV4-ID TLV in LABEL object).

Label (32-bit): The Label information encoded such that the 20 rightmost bits represent a label.

7.4.1. NextHop TLV

This document defines the following TLV for the LABEL object to associate the nexthop information incase of an outgoing label.

NEXTHOP-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          |  Length = 8                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        nexthop IPv4 address                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

NEXTHOP-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          |   Length = 20                 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
//                nexthop IPv6 address (16 bytes)              //
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

NEXTHOP-UNNUMBERED-IPV4-ID 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          |   Length = 12                 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                             Node-ID                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           Interface ID                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The NextHop TLVs are as follows:

NEXTHOP-IPV4-ADDRESS TLV: where Nexthop IPv4 address is specified as an IPv4 address of the Nexthop.

NEXTHOP-IPV6-ADDRESS TLV: where Nexthop IPv6 address is specified as an IPv6 address of the Nexthop.

NEXTHOP-UNNUMBERED-IPV4-ID TLV: where a pair of Node ID / Interface ID tuples is used for the Nexthop.

7.5. FEC Object

The FEC Object is used to specify the FEC information and MAY be carried within PCLabelUpd message.

FEC Object-Class is TBD.

FEC Object-Type is 1 'IPv4 Node 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        IPv4 Node ID                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

FEC Object-Type is 2 'IPv6 Node 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
//                      IPv6 Node ID (16 bytes)                //
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

FEC Object-Type is 3 'IPv4 Adjacency'.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Local IPv4 address                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     Remote IPv4 address                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

FEC Object-Type is 4 'IPv6 Adjacency'.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
//               Local IPv6 address (16 bytes)                 //
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
//               Remote IPv6 address (16 bytes)                //
```

```
   |                                                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

FEC Object-Type is 5 'Unnumbered Adjacency with IPv4 NodeIDs'.

```
    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
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                       Local Node-ID                           |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                     Local Interface ID                        |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                       Remote Node-ID                          |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                    Remote Interface ID                        |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The FEC objects are as follows:

IPv4 Node ID: where IPv4 Node ID is specified as an IPv4 address of the Node. FEC Object-type is 1, and the Object-Length is 4 in this case.

IPv6 Node ID: where IPv6 Node ID is specified as an IPv6 address of the Node. FEC Object-type is 2, and the Object-Length is 16 in this case.

IPv4 Adjacency: where Local and Remote IPv4 address is specified as pair of IPv4 address of the adjacency. FEC Object-type is 3, and the Object-Length is 8 in this case.

IPv6 Adjacency: where Local and Remote IPv6 address is specified as pair of IPv6 address of the adjacency. FEC Object-type is 4, and the Object-Length is 32 in this case.

Unnumbered Adjacency with IPv4 NodeID: where a pair of Node ID / Interface ID tuples is used. FEC Object-type is 5, and the Object-Length is 16 in this case.

8. Security Considerations

TBD

9. Manageability Considerations

9.1. Control of Function and Policy

TBD.

9.2. Information and Data Models

TBD.

9.3. Liveness Detection and Monitoring

TBD.

9.4. Verify Correct Operations

TBD.

9.5. Requirements On Other Protocols

TBD.

9.6. Impact On Network Operations

TBD.

10. IANA Considerations

TBD

11. Acknowledgments

We would like to thank Robert Tao, Changjing Yan, Tieying Huang 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, March 1997.

[RFC5440] Vasseur, JP. and JL. Le Roux, "Path Computation Element (PCE) Communication Protocol (PCEP)", RFC 5440, March 2009.

[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-10 (work in progress), October 2014.

[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-02 (work in progress), October 2014.

12.2. Informative References

[I-D.li-mpls-global-label-framework]
Li, Z., Zhao, Q., Chen, X., Yang, T., and R. Raszuk, "A Framework of MPLS Global Label", draft-li-mpls-global-label-framework-02 (work in progress), July 2014.

[I-D.zhao-pce-central-controller-user-cases]
Zhao, Q., Zhao, K., and Z. Ke, "The Use Cases for Using PCE as the Central Controller(PCECC) of LSPs", draft-zhao-pce-central-controller-user-cases-01 (work in progress), July 2014.

[I-D.ietf-pce-remote-initiated-gmpls-lsp]
Ali, Z., Sivabalan, S., Filsfils, C., Varga, R., Lopez, V., Dios, O., and X. Zhang, "Path Computation Element Communication Protocol (PCEP) Extensions for remote-initiated GMPLS LSP Setup", draft-ietf-pce-remote-initiated-gmpls-lsp-00 (work in progress), March 2014.

[I-D.ietf-spring-segment-routing]
Filsfils, C., Previdi, S., Bashandy, A., Decraene, B., Litkowski, S., Horneffer, M., Shakir, R., Tantsura, J., and E. Crabbe, "Segment Routing Architecture", draft-ietf-spring-segment-routing-01 (work in progress), February 2015.

[I-D.ietf-isis-segment-routing-extensions]
Previdi, S., Filsfils, C., Bashandy, A., Gredler, H., Litkowski, S., Decraene, B., and J. Tantsura, "IS-IS Extensions for Segment Routing", draft-ietf-isis-segment-routing-extensions-03 (work in progress), October 2014.

[I-D.ietf-ospf-segment-routing-extensions]
Psenak, P., Previdi, S., Filsfils, C., Gredler, H., Shakir, R., Henderickx, W., and J. Tantsura, "OSPF Extensions for Segment Routing", draft-ietf-ospf-segment-routing-extensions-04 (work in progress), February 2015.

[I-D.ietf-pce-segment-routing]
Sivabalan, S., Medved, J., Filsfils, C., Crabbe, E., Raszuk, R., Lopez, V., and J. Tantsura, "PCEP Extensions for Segment Routing", draft-ietf-pce-segment-routing-00 (work in progress), October 2014.

[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

Quintin Zhao
Huawei Technologies
125 Nagog Technology Park
Acton, MA 01719
USA

EMail: quintin.zhao@huawei.com

Katherine Zhao
Huawei Technologies
2330 Central Expressway
Santa Clara, CA 95050
USA

EMail: katherine.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 560037
India

EMail: dhruv.ietf@gmail.com

Udayasree Palle
Huawei Technologies
Divyashree Techno Park, Whitefield
Bangalore, Karnataka 560037
India

EMail: udayasree.palle@huawei.com

Boris Zhang
Telus Ltd.
Toronto
Canada

EMail: boris.zhang@telus.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

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 taker 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 specify 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.

```
                  +-------+                             +-------+
                  |PCC    |                             |  PCE  |
                  |Ingress|                             +-------+
           +------|       |                                 |
           | PCC  +-------+                                 |
           | Transit| |                                     |
    +------|        | |-- PCRpt,PLSP-ID=1, PST=TBD, D=1---->| PCECC LSP
    |PCC   +--------+ |                                     |
    |Egress  |  |     |                                     |
    +--------+  |     |                                     |
        |       |     |                                     |
        |<------ PCInitiate,CC-ID=X,PLSP-ID=1  ------------ | Label
        |       |     |                                     | download
        |------- PCRpt,CC-ID=X,PLSP-ID=1  ----------------->|
        |       |     |                                     |
        |       |<----- PCInitiate,CC-ID=Y,PLSP-ID=1  ----- | Label
        |       |     |                                     | download
        |       |-----  PCRpt,CC-ID=Y,PLSP-ID=1  ---------->|
        |       |     |                                     |
        |       |     |<--- PCInitiate,CC-ID=Z,PLSP-ID=1  - | Label
        |       |     |                                     | download
        |       |     |---- PCRpt,CC-ID=Z,PLSP-ID=1  ------>|
        |       |     |                                     |
        |       |     |<-- PCUpd,PLSP-ID=1,PST=TBD, D=1-----| PCECC LSP
        |       |     |                                     | Update
        |       |     |                                     |
```

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.

```
                  +-------+                              +-------+
                  |PCC    |                              |  PCE  |
                  |Ingress|                              +-------+
           +------|       |                                  |
           | PCC  +-------+                                  |
           | Transit| |                                      |
    +------|        | |-- PCRpt,PLSP-ID=1,PST=TBD,D=1,R=1--->| PCECC LSP
    |PCC   +--------+ |                                      | remove
    |Egress  |  |     |                                      |
    +--------+  |     |                                      |
        |       |     |                                      |
        |<------ PCInitiate,CC-ID=X,PLSP-ID=1  ------------  | Label
        |       |     |                    R=1               | cleanup
        |------- PCRpt,CC-ID=X,PLSP-ID=1  ------------------>|
        |       |     |                                      |
        |       |<----- PCInitiate,CC-ID=Y,PLSP-ID=1  ------ | Label
        |       |     |                           R=1        | cleanup
        |       |-----  PCRpt,CC-ID=Y,PLSP-ID=1  ----------->|
        |       |     |                                      |
        |       |     |<--- PCInitiate,CC-ID=Z,PLSP-ID=1  -- | Label
        |       |     |                               R=1    | cleanup
        |       |     |---- PCRpt,CC-ID=Z,PLSP-ID=1  ------->|
        |       |     |                                      |
```

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 -

```
              +-------+                              +-------+
              |PCC    |                              |  PCE  |
              |Ingress|                              +-------+
       +------|       |                                  |
       | PCC  +-------+                                  |
       | Transit| |                                      |
+------|        | |<--PCInitiate,PLSP-ID=0,PST=TBD,D=1---| PCECC LSP
|PCC   +--------+ |                                      | Initiate
|Egress  |  |     |--- PCRpt,PLSP-ID=2,P=1,D=1,C=1--->   | PCECC LSP
+--------+  |     |       (GOING-UP)                     |
    |       |     |                                      |
    |<------ PCInitiate,CC-ID=X,PLSP-ID=2 -------------- | Label
    |       |     |                                      | download
    |------- PCRpt,CC-ID=X,PLSP-ID=2  ------------------>|
    |       |     |                                      |
    |       |<----- PCInitiate,CC-ID=Y,PLSP-ID=2 ------- | Label
    |       |     |                                      | download
    |       |-----  PCRpt,CC-ID=Y,PLSP-ID=2  ----------->|
    |       |     |                                      |
    |       |     |<--- PCInitiate,CC-ID=Z,PLSP-ID=2 --- | Label
    |       |     |                                      | download
    |       |     |---- PCRpt,CC-ID=Z,PLSP-ID=2  ------->|
    |       |     |                                      |
    |       |     |<-- PCUpd, PLSP-ID=2, PST=TBD, D=1--- | PCECC LSP
    |       |     |      (UP)                            | Update
    |       |     |--- PCRpt,PLSP-ID=2,P=1,D=1,C=1--->   |
    |       |     |      (UP)                            |
```

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 -

```
                +-------+                             +-------+
                |PCC    |                             |  PCE  |
                |Ingress|                             +-------+
         +------|       |                                 |
         | PCC  +-------+                                 |
         | Transit| |                                     |
  +------|        | |                                     |
  |PCC   +--------+ |                                     |
  |Egress  |  |     |                                     |
  +--------+  |     |                                     |
      |       |     |                                     | New Path for
      |<------ PCInitiate,CC-ID=XX,PLSP-ID=1 -----------  | LSP trigger
      |       |     |                                     | new instruct
      |------- PCRpt,CC-ID=XX,PLSP-ID=1  ---------------->|
      |       |     |                                     |
      |       |<----- PCInitiate,CC-ID=YY,PLSP-ID=1------ | Label
      |       |     |                                     | download
      |       |-----  PCRpt,CC-ID=YY,PLSP-ID=1  --------->|
      |       |     |                                     |
      |       |     |<--- PCInitiate,CC-ID=ZZ,PLSP-ID=1 - | Label
      |       |     |                                     | download
      |       |     |---- PCRpt,CC-ID=ZZ,PLSP-ID=1  ----->|
      |       |     |                                     |
      |       |     |<-- PCUpd, PLSP-ID=1, PST=TBD, D=1-- | PCECC
      |       |     |    SRP=S                            | LSP Update
      |       |     |                                     |
      |       |     |-- PCRpt,PLSP-ID=1,PST=TBD,D=1    -->| Trigger
      |       |     |       (SRP=S)                       | Delete old
      |       |     |                                     | instruct
      |       |     |                                     |
      |<------ PCInitiate,CC-ID=X, PLSP-ID=1 -----------  | Label
      |       |     |                    R=1              | cleanup
      |------- PCRpt,CC-ID=X, PLSP-ID=1  ---------------->|
      |       |     |                                     |
      |       |<----- PCInitiate,CC-ID=Y, PLSP-ID=1 ----- | Label
      |       |     |                           R=1       | cleanup
      |       |-----  PCRpt,CC-ID=Y, PLSP-ID=1  --------->|
      |       |     |                                     |
      |       |     |<--- PCInitiate,CC-ID=Z, PLSP-ID=1 - | Label
      |       |     |                               R=1   | cleanup
      |       |     |---- PCRpt,CC-ID=Z, PLSP-ID=1  ----->|
      |       |     |                                     |
```

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:

```
 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        |           Length=4            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                             Flags                             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

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.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                            CC-ID                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Reserved             |               Flags         |O|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                 Label                 |     Reserved          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
//                        Optional TLV                         //
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

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          |  Length = 4                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        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          |   Length = 16                 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
//                 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          |   Length = 8                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                            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