

PCE Working Group  
Internet Draft  
Intended status: Standard Track  
Expires: July 31, 2013

Zafar Ali  
Siva Sivabalan  
Clarence Filsfils  
Cisco Systems

Robert Varga  
Pantheon Technologies

Victor Lopez  
Oscar Gonzalez de Dios  
Telefonica I+D

February 1, 2013

Path Computation Element Communication Protocol (PCEP)  
Extensions for remote-initiated GMPLS LSP Setup  
draft-ali-pce-remote-initiated-gmpls-lsp-00.txt

Status of this Memo

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

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

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

This Internet-Draft will expire on July 31, 2013.

Copyright Notice

Copyright (c) 2013 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.

This document may contain material from IETF Documents or IETF Contributions published or made publicly available before November 10, 2008. The person(s) controlling the copyright in some of this material may not have granted the IETF Trust the right to allow modifications of such material outside the IETF Standards Process. Without obtaining an adequate license from the person(s) controlling the copyright in such materials, this document may not be modified outside the IETF Standards Process, and derivative works of it may not be created outside the IETF Standards Process, except to format it for publication as an RFC or to translate it into languages other than English.

## Abstract

PCEP Extensions for PCE-initiated LSP Setup in a Stateful PCE Model draft [I-D. draft-crabbe-pce-pce-initiated-lsp] specifies procedures that can be used for creation and deletion of PCE-initiated LSPs under the active stateful PCE model. However, this specification is focused on MPLS networks, and does not cover remote instantiation of GMPLS paths. This document complements PCEP Extensions for PCE-initiated LSP Setup in a Stateful PCE Model draft by addressing the extensions required for GMPLS applications, for example for OTN and WSON networks.

When active stateful PCE is used for managing PCE-initiated LSP, PCC may not be aware of the intended usage of the LSP (e.g., in a multi-layer network). PCEP Extensions for PCE-initiated LSP Setup in a Stateful PCE Model draft does not address this requirement. This draft also addresses the requirement to specify on how PCC should use the PCEP initiated LSPs.

## Conventions used in this document

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

## Table of Contents

1. Introduction.....	3
2. Use Cases.....	4
2.1. Single-layer provisioning from Active stateful PCE....	4
2.2. Bandwidth-on-demand for multi-layer networks.....	5
2.3. Higher-layer signaling trigger.....	6
2.4. NMS-VNTM cooperation model (separated flavor).....	7

3. GMPLS Requirements for Remote-Initiated LSPs.....	9
4. Remote Initiated LSP Usage Requirement.....	9
5. PCEP Extensions for Remote-Initiated GMPLS LSPs.....	10
5.1. Generalized Endpoint in LSP Create Message.....	10
5.2. GENERALIZED-BANDWIDTH object in LSP Create Message...	11
5.3. Protection Attributes in LSP Create Message.....	11
5.4. ERO in LSP Create Object.....	12
5.4.1. ERO with explicit label control.....	12
5.4.2. ERO with Path Keys.....	12
5.4.3. Switch Layer Object .....	13
6. PCEP extension for PCEP Initiated LSP Usage Specification.	13
6.1. LSP_TUNNEL_INTERFACE_ID Object in LSP Create Message.	13
6.2. Communicating LSP usage to Egress node.....	15
6.3. LSP delegation and cleanup .....	15
7. Security Considerations.....	15
8. IANA Considerations.....	15
8.1. END-POINT Object.....	15
8.2. PCEP-Error Object.....	16
9. Acknowledgments.....	16
10. References.....	16
10.1. Normative References.....	16
10.2. Informative References.....	16

## 1. Introduction

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 draft [I-D. draft-ietf-pce-stateful-pce] describes a set of extensions to PCEP to enable active control of MPLS-TE and GMPLS tunnels.

[I-D. draft-crabbe-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 network that is centrally controlled and deployed. However, this specification is focused on MPLS networks, and does not cover the GMPLS networks (e.g., WSON, OTN, SONET/ SDH, etc. technologies). GMPLS requirements for PCEP initiated LSPs are outlined in Section 3. This document complements [I-D. draft-crabbe-pce-pce-initiated-lsp] by addressing the requirements for remote-initiated GMPLS LSPs. The PCEP extensions for PCEP initiated GMPLS LSPs are specified in Section 5. The mechanism described in this document is applicable not only to active PCEs initiating LSPs, but to any entity that initiates LSPs remotely.

When an active stateful PCE is used for managing remote-initiated LSP, the PCC may not be aware of the intended usage of the remote-initiated LSP. For example, the PCC may not know the target IGP instance in which the remote-initiated LSP is to be used. These requirements are outlined in Section 4. [RFC6107] defines LSP\_TUNNEL\_INTERFACE\_ID Object for communicating target IGP instance and usage of the forwarding and/ or routing adjacency from the ingress node to the egress node. However, current PCEP specifications do not include signaling of the LSP\_TUNNEL\_INTERFACE\_ID TLV in the PCEP message. Furthermore, [I-D. draft-crabbe-pce-pce-initiated-lsp] does not address this requirement. This draft also addresses the requirement to specify on how PCC should use the PCEP initiated LSPs. This is achieved by using LSP\_TUNNEL\_INTERFACE\_ID Object defined in [RFC6107] in PCEP, as detailed in Section 6.

## 2. Use Cases

### 2.1. Single-layer provisioning from active stateful PCE

Figure 1 shows a single-layer topology with optical nodes with a GMPLS control plane. In this scenario, the active PCE can dynamically create or delete L0 services between client interfaces. This process can be triggered by the deployment of a new network configuration or a re-optimization process. This operation can be human-driven (e.g. through an NMS) or an automatic process.

[Please refer to pdf version for the Figure]

Figure 1. Single-layer provisioning from active stateful PCE.

L0 PCE obtains resources information via control plane collecting LSAs messages. The request contains, at least, two optical transport interfaces (OT i/f), so PCE computes the path and sends a message to the optical equipment with ERO path information.

## 2.2. Bandwidth-on-demand for multi-layer networks

This use case assumes there is a multi-layer network composed by routers and optical equipment. In this scenario, there is an entity, which decides it needs extra bandwidth between two routers. This certain moment a GMPLS LSP connecting both routers via the optical network can be established on-the-fly. This entity can be a router, an active stateful PCE or even the NMS (with or without human intervention).

It is important to note that the bandwidth-on-demand interfaces and spare bandwidth in the optical network could be shared to cover many under capacity scenarios in the L3 network. For example, in this use-case, if we assume all interfaces are 10G and there is 10G of spare bandwidth available in the optical network, the spare bandwidth in the optical network can be used to connect any router, depending on bandwidth demand of the router network. For example, if there are three routers, it is not known a priori if the demand will make bandwidth-on-demand interface at R1 to be connected to bandwidth-on-demand interface at R2 or R3. For this reason, bandwidth-on-demand interfaces cannot be pre-provisioned with the IP services that are expected to carry.

According to [RFC5623], there are four options of Inter-Layer Path Computation and Inter-Layer Path Control Models: (1) PCE-VNTM cooperation, (2) Higher-layer signaling trigger, (3) NMS-VNTM cooperation model (integrated flavor) and (4) NMS-VNTM cooperation model (separated flavor). In all scenarios there is a certain moment when entities are using an interface to request for a path provisioning. In this document we have selected two use cases in a scenario with routers and optical equipment to obtain the requirements for this draft, but it is applicable to the four options.

[Please refer to pdf version for the Figure]

Figure 2. Use case higher-layer signaling trigger

### 2.3. Higher-layer signaling trigger

Figure 2 depicts a multi-layer network scenario similar to the presented in section 4.2.2. [RFC5623], with the difference that PCE is an active stateful PCE [I-D. draft-ietf-pce-stateful-pce].

In this example, O1, O2 and O3 are optical nodes that are connected with router nodes R1, R2 and R3, respectively. The network is designed such that the interface between R1-O1, R2-O2 and R3-O3 are setup to provide bandwidth-on-demand via the optical network.

The example assumes that an active stateful PCE is used for setting and tearing down bandwidth-on-demand connectivity. Although the simple use-case assumes a single PCE server (PCE1), the proposed technique is generalized to cover multiple co-operating PCE case. Similarly, although the use case assumes PCE1 only has knowledge of the L3 topology, the proposed technique is generalized to cover multi-layer PCE case.

The PCE server (PCE1) is assumed to be receiving L3 topology data. It is also assumed that PCE learns L0 (optical) addresses associated with bandwidth-on-demand interfaces R1-O1, R2-O2 and R3-O3. These addresses are referred by OTE-IP-R1 (optical TE

link R1-O1 address at R1), OTE-IP-R2 (optical TE link R2-O2 address at R2) and OTE-IP-R3 (optical TE link R3-O3 address at R3), respectively. How PCE learns the optical addresses associated with the bandwidth-on-demand interfaces is beyond the scope of this document.

How knowledge of the bandwidth-on-demand interfaces is utilized by the PCE is exemplified in the following. Suppose an application requests 8 Gbps from R1 to R2 (recall all interfaces in Figure 1 are assumed to be 10G). PCE1 satisfies this by establishing a tunnel using R1-R4-R2 path. PCEP initiated LSP using techniques specified in [I-D. draft-crabbe-pce-pce-initiated-lsp] can be used to establish a PSC tunnel using the R1-R4-R2 path. Now assume another application requests 7 Gbps service between R1 and R2. This request cannot be satisfied without establishing a GMPLS tunnel via optical network using bandwidth-on-demand interfaces. In this case, PCE1 initiates a GMPLS tunnel using R1-O1-O2-R2 path (this is referred as GMPLS tunnel1 in the following). The PCEP initiated LSP using techniques specified in document are used for this purpose.

As mentioned earlier, the GMPLS tunnel created on-the-fly to satisfy bandwidth demand of L3 applications cannot be pre-provisioned in IP network, as bandwidth-on-demand interfaces and spare bandwidth in the optical network are shared. Furthermore, in this example, as active stateful PCE is used for managing PCE-initiated LSP, PCC may not be aware of the intended usage of the PCE-initiated LSP. Specifically, when the PCE1 initiated GMPLS tunnel1, PCC does not know the IGP instance whose demand leads to establishment of the GMPLS tunnel1 and hence does not know the IGP instance in which the GMPLS tunnel1 needs to be advertised. Similarly, the PCC does not know IP address that should be assigned to the GMPLS tunnel1. In the above example, this IP address is labeled as TUN-IP-R1 (tunnel IP address at R1). The PCC also does not know if the tunnel needs to be advertised as forwarding and/ or routing adjacency and/or to be locally used by the target IGP instance. Similarly, egress node for GMPLS signaling (R2 node in this example) may not know the intended usage of the tunnel (tunnel1 in this example). For example, the R2 node does not know IP address that should be assigned to the GMPLS tunnel1. In the above example, this IP address is labeled as TUN-IP-R2 (tunnel IP address at R2). Section 6 of this draft addresses the requirement to specify on how PCC and egress node for signaling should use the PCEP initiated LSPs.

#### 2.4. NMS-VNTM cooperation model (separated flavor)

Figure 3 depicts NMS-VNTM cooperation model. This is the separated flavor, because NMS and VNTM are not in the same location.

A new L3 path is requested from NMS, because there is an automated process in the NMS or after human intervention. NMS does not have information about all network information, so it consults L3 PCE. For shake of simplicity L3-PCE is used, but any other multi-layer cooperating PCE model is applicable. In case that there are enough resources in the L3 layer, L3-PCE returns a L3 only path. On the other hand, if there is a lack of resources at the L3 layer, the response does not have any path or may contain a multilayer path with L3 and L0 (optical) information in case of a ML-PCE. In case of there is not a path in L3; NMS sends a message to the VNTM to create a GMPLS LSP in the lower layer. When the VNTM receives this message, based on the local policies, accepts the suggestion and sends a similar message to the router, which can create the lower layer LSP via UNI signaling in the routers, like in use case in section 2.3.1. Similarly, VNTM may talk with L0-PCE to set-up the path in the optical domain (section 2.2). This second option looks more complex, because it requires VNTM configuring inter-layer TE-links.

Requirements for the message from VNTM to the router are the same than in the previous use case (section 2.3.1). Regarding NMS to VNTM message, the requirements here depends on who has all the information. Three different addresses are required in this use case: (1) L3, (2) L0 and (3) inter-layer addressing. In case there is a non-cooperating L3-PCE, information about inter-layer connections have to be stored (or discovered) by VNTM. If there is a ML-PCE and this information is obtained from the network, the message would be the same than in section 2.3.1.



[Please refer to pdf version for the Figure]

Figure 3. Use case NMS-VNTM cooperation model

### 3. GMPLS Requirements for Remote-Initiated LSPs

[I-D. draft-crabbe-pce-pce-initiated-lsp] specifies procedures that can be used for creation and deletion of PCE-initiated LSPs under the active stateful PCE model. However, this specification does not address GMPLS requirements outlined in the following:

- GMPLS support multiple switching capabilities on per TE link basis. GMPLS LSP creation requires knowledge of LSP switching capability (e.g., TDM, L2SC, OTN-TDM, LSC, etc.) to be used [RFC3471], [RFC3473].
- GMPLS LSP creation requires knowledge of the encoding type (e.g., lambda photonic, Ethernet, SONET/ SDH, G709 OTN, etc.) to be used by the LSP [RFC3471], [RFC3473].
- GMPLS LSP creation requires information of the generalized payload (G-PID) to be carried by the LSP [RFC3471], [RFC3473].
- GMPLS LSP creation requires specification of data flow specific traffic parameters (also known as Tspec), which are technology specific.
- GMPLS also specifics support for asymmetric bandwidth requests [RFC6387].
- GMPLS extends the addressing to include unnumbered interface identifiers, as defined in [RFC3477].
- In some technologies path calculation is tightly coupled with label selection along the route. For example, path calculation in a WDM network may include lambda continuity and/ or lambda feasibility constraints and hence a path computed by the PCE is associated with a specific lambda (label). Hence, in such networks, the label information needs to be provided to a PCC in order for a PCE to initiate GMPLS LSPs under the active stateful PCE model. I.e., explicit label control may be required.
- GMPLS specifics protection context for the LSP, as defined in [RFC4872] and [RFC4873].

### 4. Remote Initiated LSP Usage Requirement

The requirement to specify usage of the LSP to the PCC includes but not limited to specification of the following information.

Internet-Draft      draft-ali-pce-remote-initiated-gmpls-lsp-00.txt

- The target IGP instance for the Remote-initiated LSP needs to be specified.
- In the target IGP instance, should the PCE-initiated LSP be advertised as a forwarding adjacency and/ or routing adjacency and/ or to be used locally by the PCC?
- Should the as Remote-initiated LSP be advertised an IPv4 FA/ RA, IPv6 FA/ RA or as unnumbered FA/ RA.
- If Remote-initiated LSP is to be advertised an IPv4 FA/ RA, IPv6 FA/ RA, what is the local and remote IP address is to be used for the advertisement.

## 5. PCEP Extensions for Remote-Initiated GMPLS LSPs

Section 3 outlines GMPLS and application requirements that need to be satisfied in order for a PCE to initiate GMPLS LSPs under the active stateful PCE model. The section provides PCEP protocol extensions required to meet these requirements.

LSP create message defined in [I-D. draft-crabbe-pce-pce-initiated-lsp] needs to be extended to include GMPLS specific PCEP objects as follows:

### 5.1. Generalized Endpoint in LSP Create Message

This document does not modify the usage of END-POINTS object for PCE initiated LSPs as specified in [I-D. draft-crabbe-pce-pce-initiated-lsp]. It augments the usage as specified below.

END-POINTS object has been extended by [I-D. draft-ietf-pcep-gmpls-ext] to include a new object type called "Generalized Endpoint". PCCreate message sent by a PCE to a PCC to trigger a GMPLS LSP instantiation SHOULD include the END-POINTS with Generalized Endpoint object type. Furthermore, the END-POINTS object MUST contain "label request" TLV. The label request TLV is used to specify the switching type, encoding type and GPID of the LSP being instantiated by the PCE.

As mentioned earlier, the PCE server is assumed to be receiving topology data. In the use case of higher-layer signaling trigger, the addresses associated with bandwidth-on-demand interfaces are included, e.g., OTE-IP-R1, OTE-IP-R2 and OTE-IP-R3, in the use case example. These addresses and R1, R2 and R3 router IDs are used to derive source and destination address of the END-POINT object. As previously mentioned, in the case of NMS-VNMT cooperation model with L3 PCE, VNTM must receive such inter-layer interface association to configure the whole path.

The unnumbered endpoint TLV can be used to specify unnumbered endpoint addresses for the LSP being instantiated by the PCE.

Internet-Draft      draft-ali-pce-remote-initiated-gmpls-lsp-00.txt

The END-POINTS MAY contain other TLVs defined in [I-D. draft-ietf-pcep-gmpls-ext].

If the END-POINTS Object of type Generalized Endpoint is missing the label request TLV, the PCC MUST send a PCErr message with Error-type=6 (Mandatory Object missing) and Error-value= TBA (LSP request TLV missing).

If the PCC does not support the END-POINTS Object of type Generalized Endpoint, the PCC MUST send a PCErr message with Error-type= TBA and Error-value= TBA. [may be already defined].

## 5.2. GENERALIZED-BANDWIDTH object in LSP Create Message

LSP create message defined in [I-D. draft-crabbe-pce-pce-initiated-lsp] can optionally include the BANDWIDTH object. However, the following possibilities cannot be represented in the BANDWIDTH object:

- Asymmetric bandwidth (different bandwidth in forward and reverse direction), as described in [RFC6387].
- Technology specific GMPLS parameters (e.g., Tspec for SDH/SONET, G.709, ATM, MEF, etc.) are not supported.

GENERALIZED-BANDWIDTH object has been defined in [I-D. draft-ietf-pcep-gmpls-ext] to address the above-mentioned limitation of the BANDWIDTH object.

This document specifies the use of GENERALIZED-BANDWIDTH object in PCCreate message. Specifically, GENERALIZED-BANDWIDTH object MAY be included in the PCCreate message. The GENERALIZED-BANDWIDTH object in PCCreate message is used to specify technology specific Tspec and asymmetrical bandwidth values for the LSP being instantiated by the PCE.

## 5.3. Protection Attributes in LSP Create Message

This document does not modify the usage of LSPA object for PCE initiated LSPs as specified in [I-D. draft-crabbe-pce-pce-initiated-lsp]. It augments the usage of LSPA object in LSP Create Message to carry the end-to-end protection context this also includes the protection state information.

The LSP Protection Information TLV of LSPA in the PCCreate message can be used to specify protection attributes of the LSP being instantiated by the PCE.

#### 5.4. ERO in LSP Create Object

This document does not modify the usage of ERO object for PCE initiated LSPs as specified in [I-D. draft-crabbe-pce-pce-initiated-lsp]. It augments the usage as specified in the following sections.

##### 5.4.1. ERO with explicit label control

As mentioned earlier, there are technologies and scenarios where active stateful PCE requires explicit label control in order to instantiate an LSP.

Explicit label control (ELC) is a procedure supported by RSVP-TE, where the outgoing label(s) is (are) encoded in the ERO. [I-D. draft-ietf-pcep-gmpls-ext] extends the <ERO> object of PCEP to include explicit label control. The ELC procedure enables the PCE to provide such label(s) directly in the path ERO.

The extended ERO object in PCCreate message can be used to specify label along with ERO to PCC for the LSP being instantiated by the active stateful PCE.

##### 5.4.2. ERO with Path Keys

There are many scenarios in packet and optical networks where the route information of an LSP may not be provided to the PCC for confidentiality reasons. A multi-domain or multi-layer network is an example of such networks. Similarly, a GMPLS User-Network Interface (UNI) [RFC4208] is also an example of such networks.

In such scenarios, ERO containing the entire route cannot be provided to PCC (by PCE). Instead, PCE provides an ERO with Path Keys to the PCC. For example, in the case UNI interface between the router and the optical nodes, the ERO in the LSP Create Message may be constructed as follows:

- The first hop is a strict hop that provides the egress interface information at PCC. This interface information is used to get to a network node that can extend the rest of the ERO. (Please note that in the cases where the network node is not directly connected with the PCC, this part of ERO may consist of multiple hops and may be loose).
- The following(s) hop in the ERO may provide the network node with the path key [RFC5520] that can be resolved to get the contents of the route towards the destination.

Internet-Draft      draft-ali-pce-remote-initiated-gmpls-lsp-00.txt

- There may be further hops but these hops may also be encoded with the path keys (if needed).

This document does not change encoding or processing roles for the path keys, which are defined in [RFC5520].

#### 5.4.3. Switch Layer Object

[draft-ietf-pce-inter-layer-ext-07] specifies the SWITCH-LAYER object which defines and specifies the switching layer (or layers) in which a path MUST or MUST NOT be established. A switching layer is expressed as a switching type and encoding type. [I-D. draft-ietf-pcep-gmpls-ext], which defines the GMPLS extensions for PCEP, suggests using the SWITCH-LAYER object. Thus, SWITCH-LAYER object can be used in the PCCreate message to specify the switching layer (or layers) of the LSP being remotely initiated.

### 6. PCEP extension for PCEP Initiated LSP Usage Specification

The requirement to specify on how PCC should use the PCEP initiated LSPs is outlined in Section 4. This subsection specifies PCEP extension used to satisfy this requirement.

PCEP extensions specified in this section are equally applicable to PCEP initiated MPLS as well as GMPLS LSPs.

#### 6.1. LSP\_TUNNEL\_INTERFACE\_ID Object in LSP Create Message

[RFC6107] defines LSP\_TUNNEL\_INTERFACE\_ID Object for communicating usage of the forwarding or routing adjacency from the ingress node to the egress node. This document extends the LSP Create Message to include LSP\_TUNNEL\_INTERFACE\_ID object defined in [RFC6107]. Object class and type for the LSP\_TUNNEL\_INTERFACE\_ID object are as follows:

Object Name: LSP\_TUNNEL\_INTERFACE\_ID

Object-Class Value: TBA by Iana (suggested value: 40)

Object-type: 1

The contents of this object are identical in encoding to the contents of the RSVP-TE LSP\_TUNNEL\_INTERFACE\_ID object defined in [RFC6107] and [RFC3477]. The following TLVs of RSVP-TE LSP\_TUNNEL\_INTERFACE\_ID object are acceptable in this object.

The PCEP LSP\_TUNNEL\_INTERFACE\_ID object's TLV types correspond to RSVP-TE LSP\_TUNNEL\_INTERFACE\_ID object's TLV types. Please note that use of TLV type 1 defined in [RFC3477] is not specified by this document.

TLV Type	TLV Description	Reference
2	IPv4 interface identifier with target IGP instance	[RFC6107]
3	IPv6 interface identifier with target IGP instance	[RFC6107]
4	Unnumbered interface with target IGP instance	[RFC6107]

The meanings of the fields of PCEP LSP\_TUNNEL\_INTERFACE\_ID object are identical to those defined for the RSVP-TE LSP\_TUNNEL\_INTERFACE\_ID object. Similarly, meanings of the fields of PCEP LSP\_TUNNEL\_INTERFACE\_ID object's supported TLV are identical to those defined for the corresponding RSVP-TE LSP\_TUNNEL\_INTERFACE\_ID object's TLVs. The following fields have slightly different usage.

- IPv4 Interface Address field in IPv4 interface identifier with target IGP instance TLV: This field indicates the local IPv4 address to be assigned to the tunnel at the PCC (ingress node for RSVP-TE signaling). In the example use case of Section 2, IP address TUN-IP-R1 (tunnel IP address at R1) is carried in this field (if TUN-IP-R1 is a v4 address).
- IPv6 Interface Address field in IPv4 interface identifier with target IGP instance TLV: This field indicates the local IPv6 address to be assigned to the tunnel at the PCC (ingress node for RSVP-TE signaling). In the example use case of Section 2, IP address TUN-IP-R1 (tunnel IP address at R1) is carried in this field (if TUN-IP-R1 is a v6 address).
- LSR's Router ID field in Unnumbered interface with target IGP instance: The PCC SHOULD use the LSR's Router ID in Unnumbered interface with target IGP instance in advertising the LSP being initiated by the PCE. In the example use case of Section 2, this field carries router-id of R1 in the target IGP instance.
- Interface ID (32 bits) field in unnumbered interface with target IGP instance: All bits of this field MUST be set to 0 by the PCE server and MUST be ignored by PCC. PCC SHOULD allocate an Interface ID that fulfills Interface ID requirements specified in [RFC3477].

When the Ingress PCC receives an LPS Request Message with LSP\_TUNNEL\_INTERFACE\_ID TLV, it uses the information contained in the TLV to drive the IGP instance, treatment of the LSP being

initiated in the target IGP instance (e.g., FA, RA or local usage), the local IPv4 or IPv6 address or router-id for unnumbered case to be used for advertisement of the LSP being instantiated.

## 6.2. Communicating LSP usage to Egress node

PCE does not need to send LSP Create message to egress node (node R2 in the example of section 2) to communicate LSP usage information. Instead PCC (Ingress signaling node) uses RSVP-TE signaling mechanism specified in [RFC6107] to send the LSP usage to Egress node. Specifically, when the Ingress PCC receives an LPS Request Message with LSP\_TUNNEL\_INTERFACE\_ID TLV, it SHOULD add LSP\_TUNNEL\_INTERFACE\_ID object in RSVP TE Path message. For this purpose, it is RECOMMENDED that the ingress PCC uses content of the LSP\_TUNNEL\_INTERFACE\_ID TLV in LSP Create Message in PCEP to drive LSP\_TUNNEL\_INTERFACE\_ID object in RSVP-TE. This document does not modify usage of LSP\_TUNNEL\_INTERFACE\_ID Object in RSVP-TE signaling as specified in [RFC6107].

The egress node uses information contained in the LSP\_TUNNEL\_INTERFACE\_ID object in RSVP-TE Path message to drive the IGP instance, treatment of the LSP being initiated in the target IGP instance (e.g., FA, RA or local usage), the local IPv4 or IPv6 address or router-id for unnumbered case to be used for advertisement of the LSP being instantiated.

## 6.3. LSP delegation and cleanup

LSP delegation and cleanup procedure specified in [I-D. draft-ietf-pcep-gmpls-ext] are equally applicable to GMPLS LSPs and this document does not modify the associated usage.

## 7. Security Considerations

To be added in future revision of this document.

## 8. IANA Considerations

### 8.1. END-POINT Object

This document extends the LSP Create Message to include LSP\_TUNNEL\_INTERFACE\_ID object defined in [RFC6107]. Object class and type for the LSP\_TUNNEL\_INTERFACE\_ID object are as follows:

Name	Class value	Type
----	-----	----
LSP_TUNNEL_INTERFACE_ID	TBA by Iana (Suggested:40)	1

Internet-Draft     draft-ali-pce-remote-initiated-gmpls-lsp-00.txt

## 8.2. PCEP-Error Object

This document defines the following new Error-Value:

Error-Type	Error Value
------------	-------------

6	Error-value=TBA: LSP Request TLV missing
---	--

## 9. Acknowledgments

The authors would like to thank George Swallow and Jan Medved for their comments.

## 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.
- [I-D. draft-crabbe-pce-pce-initiated-lsp] Crabbe, E., Minei, I., Sivabalan, S., Varga, R., "PCEP Extensions for PCE-initiated LSP Setup in a Stateful PCE Model", draft-crabbe-pce-pce-initiated-lsp, work in progress.
- [RFC5440] Vasseur, JP., Ed., and JL. Le Roux, Ed., "Path Computation Element (PCE) Communication Protocol (PCEP)", RFC 5440, March 2009.
- [RFC5623] Oki, E., Takeda, T., Le Roux, JL., and A. Farrel, "Framework for PCE-Based Inter-Layer MPLS and GMPLS Traffic Engineering", RFC 5623, September 2009.
- [RFC 6107] Shiomoto, K., Ed., and A. Farrel, Ed., "Procedures for Dynamically Signaled Hierarchical Label Switched Paths", RFC 6107, February 2011.

### 10.2. Informative References

- [RFC3471] Berger, L., Ed., "Generalized Multi-Protocol Label Switching (GMPLS) Signaling Functional Description", RFC 3471, January 2003.
- [RFC3473] Berger, L., Ed., "Generalized Multi-Protocol Label Switching (GMPLS) Signaling Resource ReserVation Protocol-Traffic Engineering (RSVP-TE) Extensions", RFC 3473, January 2003.



Internet-Draft     draft-ali-pce-remote-initiated-gmpls-lsp-00.txt

[RFC 5467] Berger, L., Takacs, A., Caviglia, D., Fedyk, D., and J. Meuric, "GMPLS Asymmetric Bandwidth Bidirectional Label Switched Paths (LSPs)", RFC 5467, March 2009.

[RFC3477] Kompella, K. and Y. Rekhter, "Signalling Unnumbered Links in Resource ReSerVation Protocol - Traffic Engineering (RSVP-TE)", RFC 3477, January 2003.

[RFC4872] Lang, J., Ed., Rekhter, Y., Ed., and D. Papadimitriou, Ed., "RSVP-TE Extensions in Support of End-to-End Generalized Multi-Protocol Label Switching (GMPLS) Recovery", RFC 4872, May 2007.

[RFC4873] Berger, L., Bryskin, I., Papadimitriou, D., and A. Farrel, "GMPLS Segment Recovery", RFC 4873, May 2007.

[RFC4208] Swallow, G., Drake, J., Ishimatsu, H., and Y. Rekhter, "Generalized Multiprotocol Label Switching (GMPLS) User-Network Interface (UNI): Resource ReserVation Protocol-Traffic Engineering (RSVP-TE) Support for the Overlay Model", RFC 4208, October 2005.

[RFC5520] Bradford, R., Ed., Vasseur, JP., and A. Farrel, "Preserving Topology Confidentiality in Inter-Domain Path Computation Using a Path-Key-Based Mechanism", RFC 5520, July 2009.

#### Authors' Addresses

Zafar Ali  
Cisco Systems  
Email: zali@cisco.com

Siva Sivabalan  
Cisco Systems  
Email: msiva@cisco.com

Clarence Filsfils  
Cisco Systems  
Email: cfilsfil@cisco.com

Internet-Draft     draft-ali-pce-remote-initiated-gmpls-lsp-00.txt

Robert Varga  
Pantheon Technologies

Victor Lopez  
Telefonica I+D  
Email: vlopez@tid.es

Oscar Gonzalez de Dios  
Telefonica I+D  
Email: ogondio@tid.es

PCE Working Group  
Internet Draft  
Intended status: Standard Track  
Expires: August 13, 2014

Zafar Ali  
Siva Sivabalan  
Clarence Filsfils  
Cisco Systems  
Robert Varga  
Pantheon Technologies  
Victor Lopez  
Oscar Gonzalez de Dios  
Telefonica I+D  
Xian Zhang  
Huawei  
February 14, 2014

Path Computation Element Communication Protocol (PCEP)  
Extensions for remote-initiated GMPLS LSP Setup  
draft-ali-pce-remote-initiated-gmpls-lsp-03.txt

#### 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 13, 2014.

#### 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.

This document may contain material from IETF Documents or IETF Contributions published or made publicly available before November 10, 2008. The person(s) controlling the copyright in some of this material may not have granted the IETF Trust the right to allow modifications of such material outside the IETF Standards Process. Without obtaining an adequate license from the person(s) controlling the copyright in such materials, this document may not be modified outside the IETF Standards Process, and derivative works of it may not be created outside the IETF Standards Process, except to format it for publication as an RFC or to translate it into languages other than English.

#### Abstract

Draft [I-D. draft-crabbe-pce-pce-initiated-lsp] specifies procedures that can be used for creation and deletion of PCE-initiated LSPs in the active stateful PCE model. However, this specification focuses on MPLS networks, and does not cover remote instantiation of paths in GMPLS-controlled networks. This document complements [I-D. draft-crabbe-pce-pce-initiated-lsp] by addressing the requirements for remote-initiated GMPLS LSPs.

#### Conventions used in this document

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

#### Table of Contents

1. Introduction .....	3
2. Requirements for Remote-Initiated GMPLS LSPs .....	3
3. PCEP Extensions for Remote-Initiated GMPLS LSPs .....	4
3.1. Generalized Endpoint in LSP Initiate Message .....	4
3.2. GENERALIZED-BANDWIDTH object in LSP Initiate Message ..	5
3.3. Protection Attributes in LSP Initiate Message .....	5
3.4. ERO in LSP Initiate Object .....	5
3.4.1. ERO with explicit label control .....	5
3.4.2. ERO with Path Keys .....	6
3.4.3. Switch Layer Object .....	6
3.5. LSP delegation and cleanup .....	7
4. Security Considerations .....	7
5. IANA Considerations .....	7
5.1. PCEP-Error Object .....	7
6. Acknowledgments .....	7
7. References .....	7
7.1. Normative References .....	7

7.2. Informative References .....8

## 1. Introduction

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 draft [I-D. draft-ietf-pce-stateful-pce] describes a set of extensions to PCEP to enable active control of MPLS-TE and GMPLS network.

[I-D. draft-crabbe-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. This enables realization of a dynamic network that is centrally controlled and deployed. However, this specification is focused on MPLS networks, and does not cover the GMPLS networks (e.g., WSON, OTN, SONET/ SDH, etc. technologies). This document complements [I-D. draft-crabbe-pce-pce-initiated-lsp] by addressing the requirements for remote-initiated GMPLS LSPs. These requirements are covered in Section 2 of this draft. The PCEP extensions for remote initiated GMPLS LSPs are specified in Section 3.

## 2. Requirements for Remote-Initiated GMPLS LSPs

[I-D. draft-crabbe-pce-pce-initiated-lsp] specifies procedures that can be used for creation and deletion of PCE-initiated LSPs under the active stateful PCE model. However, this specification does not address GMPLS requirements outlined in the following:

- GMPLS support multiple switching capabilities on per TE link basis. GMPLS LSP creation requires knowledge of LSP switching capability (e.g., TDM, L2SC, OTN-TDM, LSC, etc.) to be used [RFC3471], [RFC3473].
- GMPLS LSP creation requires knowledge of the encoding type (e.g., lambda photonic, Ethernet, SONET/ SDH, G709 OTN, etc.) to be used by the LSP [RFC3471], [RFC3473].
- GMPLS LSP creation requires information of the generalized payload (G-PID) to be carried by the LSP [RFC3471], [RFC3473].
- GMPLS LSP creation requires specification of data flow specific traffic parameters (also known as Tspec), which are technology specific.

- GMPLS also specifics support for asymmetric bandwidth requests [RFC6387].
- GMPLS extends the addressing to include unnumbered interface identifiers, as defined in [RFC3477].
- In some technologies path calculation is tightly coupled with label selection along the route. For example, path calculation in a WDM network may include lambda continuity and/ or lambda feasibility constraints and hence a path computed by the PCE is associated with a specific lambda (label). Hence, in such networks, the label information needs to be provided to a PCC in order for a PCE to initiate GMPLS LSPs under the active stateful PCE model. I.e., explicit label control may be required.
- GMPLS specifics protection context for the LSP, as defined in [RFC4872] and [RFC4873].

### 3. PCEP Extensions for Remote-Initiated GMPLS LSPs

LSP initiate (PCInitiate) message defined in [I-D. draft-crabbe-pce-pce-initiated-lsp] needs to be extended to include GMPLS specific PCEP objects as follows:

#### 3.1. Generalized Endpoint in LSP Initiate Message

This document does not modify the usage of END-POINTS object for PCE initiated LSPs as specified in [I-D. draft-crabbe-pce-pce-initiated-lsp]. It augments the usage as specified below.

END-POINTS object has been extended by [I-D. draft-ietf-pcep-gmpls-ext] to include a new object type called "Generalized Endpoint". PCInitiate message sent by a PCE to a PCC to trigger a GMPLS LSP instantiation SHOULD include the END-POINTS with Generalized Endpoint object type. Furthermore, the END-POINTS object MUST contain "label request" TLV. The label request TLV is used to specify the switching type, encoding type and GPID of the LSP being instantiated by the PCE.

The unnumbered endpoint TLV can be used to specify unnumbered endpoint addresses for the LSP being instantiated by the PCE. The END-POINTS MAY contain other TLVs defined in [I-D. draft-ietf-pcep-gmpls-ext].

If the END-POINTS Object of type Generalized Endpoint is missing the label request TLV, the PCC MUST send a PCErr message with Error-type=6 (Mandatory Object missing) and Error-value= TBA (LSP request TLV missing).

If the PCC does not support the END-POINTS Object of type Generalized Endpoint, the PCC MUST send a PCErr message with

Internet-Draft      draft-ali-pce-remote-initiated-gmpls-lsp-02.txt

Error-type = 3 (Unknown Object), Error-value = 2(unknown object type).

### 3.2. GENERALIZED-BANDWIDTH object in LSP Initiate Message

LSP initiate message defined in [I-D. draft-crabbe-pce-pce-initiated-lsp] can optionally include the BANDWIDTH object. However, the following possibilities cannot be represented in the BANDWIDTH object:

- Asymmetric bandwidth (different bandwidth in forward and reverse direction), as described in [RFC6387].

- Technology specific GMPLS parameters (e.g., Tspec for SDH/SONET, G.709, ATM, MEF, etc.) are not supported.

GENERALIZED-BANDWIDTH object has been defined in [I-D. draft-ietf-pcep-gmpls-ext] to address the above-mentioned limitation of the BANDWIDTH object.

This document specifies the use of GENERALIZED-BANDWIDTH object in PCInitiate message. Specifically, GENERALIZED-BANDWIDTH object MAY be included in the PCInitiate message. The GENERALIZED-BANDWIDTH object in PCInitiate message is used to specify technology specific Tspec and asymmetrical bandwidth values for the LSP being instantiated by the PCE.

### 3.3. Protection Attributes in LSP Initiate Message

This document does not modify the usage of LSPA object for PCE initiated LSPs as specified in [I-D. draft-crabbe-pce-pce-initiated-lsp]. It augments the usage of LSPA object in LSP Initiate Message to carry the end-to-end protection context this also includes the protection state information.

The LSP Protection Information TLV of LSPA in the PCInitiate message can be used to specify protection attributes of the LSP being instantiated by the PCE.

### 3.4. ERO in LSP Initiate Object

This document does not modify the usage of ERO object for PCE initiated LSPs as specified in [I-D. draft-crabbe-pce-pce-initiated-lsp]. It augments the usage as specified in the following sections.

#### 3.4.1. ERO with explicit label control

As mentioned earlier, there are technologies and scenarios where active stateful PCE requires explicit label control in order to instantiate an LSP.

Explicit label control (ELC) is a procedure supported by RSVP-TE, where the outgoing label(s) is (are) encoded in the ERO. [I-D. draft-ietf-pcep-gmpls-ext] extends the <ERO> object of PCEP to include explicit label control. The ELC procedure enables the PCE to provide such label(s) directly in the path ERO.

The extended ERO object in PCInitiate message can be used to specify label along with ERO to PCC for the LSP being instantiated by the active stateful PCE.

### 3.4.2. ERO with Path Keys

There are many scenarios in packet and optical networks where the route information of an LSP may not be provided to the PCC for confidentiality reasons. A multi-domain or multi-layer network is an example of such networks. Similarly, a GMPLS User-Network Interface (UNI) [RFC4208] is also an example of such networks.

In such scenarios, ERO containing the entire route cannot be provided to PCC (by PCE). Instead, PCE provides an ERO with Path Keys to the PCC. For example, in the case UNI interface between the router and the optical nodes, the ERO in the LSP Initiate Message may be constructed as follows:

- The first hop is a strict hop that provides the egress interface information at PCC. This interface information is used to get to a network node that can extend the rest of the ERO. (Please note that in the cases where the network node is not directly connected with the PCC, this part of ERO may consist of multiple hops and may be loose).
- The following(s) hop in the ERO may provide the network node with the path key [RFC5520] that can be resolved to get the contents of the route towards the destination.
- There may be further hops but these hops may also be encoded with the path keys (if needed).

This document does not change encoding or processing roles for the path keys, which are defined in [RFC5520].

### 3.4.3. Switch Layer Object

[draft-ietf-pce-inter-layer-ext-07] specifies the SWITCH-LAYER object which defines and specifies the switching layer (or layers) in which a path MUST or MUST NOT be established. A switching layer is expressed as a switching type and encoding type. [I-D. draft-ietf-pcep-gmpls-ext], which defines the GMPLS



Internet-Draft      draft-ali-pce-remote-initiated-gmpls-lsp-02.txt

extensions for PCEP, suggests using the SWITCH-LAYER object. Thus, SWITCH-LAYER object can be used in the PCInitiate message to specify the switching layer (or layers) of the LSP being remotely initiated.

### 3.5. LSP delegation and cleanup

LSP delegation and cleanup procedure specified in [I-D. draft-ietf-pcep-gmpls-ext] are equally applicable to GMPLS LSPs and this document does not modify the associated usage.

## 4. Security Considerations

To be added in future revision of this document.

## 5. IANA Considerations

### 5.1. PCEP-Error Object

This document defines the following new Error-Value:

Error-Type	Error Value
------------	-------------

6	Error-value=TBA: LSP Request TLV missing
---	--

## 6. Acknowledgments

The authors would like to thank George Swallow and Jan Medved for their comments.

## 7. References

### 7.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [I-D. draft-crabbe-pce-pce-initiated-lsp] Crabbe, E., Minei, I., Sivabalan, S., Varga, R., "PCEP Extensions for PCE-initiated LSP Setup in a Stateful PCE Model", draft-crabbe-pce-pce-initiated-lsp, work in progress.
- [RFC5440] Vasseur, JP., Ed., and JL. Le Roux, Ed., "Path Computation Element (PCE) Communication Protocol (PCEP)", RFC 5440, March 2009.
- [RFC5623] Oki, E., Takeda, T., Le Roux, JL., and A. Farrel, "Framework for PCE-Based Inter-Layer MPLS and GMPLS Traffic Engineering", RFC 5623, September 2009.

Internet-Draft      draft-ali-pce-remote-initiated-gmpls-lsp-02.txt

- [RFC 6107] Shiomoto, K., Ed., and A. Farrel, Ed., "Procedures for Dynamically Signaled Hierarchical Label Switched Paths", RFC 6107, February 2011.

## 7.2. Informative References

- [RFC3471] Berger, L., Ed., "Generalized Multi-Protocol Label Switching (GMPLS) Signaling Functional Description", RFC 3471, January 2003.
- [RFC3473] Berger, L., Ed., "Generalized Multi-Protocol Label Switching (GMPLS) Signaling Resource ReserVation Protocol-Traffic Engineering (RSVP-TE) Extensions", RFC 3473, January 2003.
- [RFC3477] Kompella, K. and Y. Rekhter, "Signalling Unnumbered Links in Resource ReSerVation Protocol - Traffic Engineering (RSVP-TE)", RFC 3477, January 2003.
- [RFC4872] Lang, J., Ed., Rekhter, Y., Ed., and D. Papadimitriou, Ed., "RSVP-TE Extensions in Support of End-to-End Generalized Multi-Protocol Label Switching (GMPLS) Recovery", RFC 4872, May 2007.
- [RFC4873] Berger, L., Bryskin, I., Papadimitriou, D., and A. Farrel, "GMPLS Segment Recovery", RFC 4873, May 2007.
- [RFC4208] Swallow, G., Drake, J., Ishimatsu, H., and Y. Rekhter, "Generalized Multiprotocol Label Switching (GMPLS) User-Network Interface (UNI): Resource ReserVation Protocol-Traffic Engineering (RSVP-TE) Support for the Overlay Model", RFC 4208, October 2005.
- [RFC5520] Bradford, R., Ed., Vasseur, JP., and A. Farrel, "Preserving Topology Confidentiality in Inter-Domain Path Computation Using a Path-Key-Based Mechanism", RFC 5520, April 2009.

## Author's Addresses

Zafar Ali  
Cisco Systems  
Email: zali@cisco.com

Siva Sivabalan  
Cisco Systems  
Email: msiva@cisco.com

Clarence Filsfils

Internet-Draft      draft-ali-pce-remote-initiated-gmpls-lsp-02.txt

Cisco Systems  
Email: cfilsfil@cisco.com

Robert Varga  
Pantheon Technologies

Victor Lopez  
Telefonica I+D  
Email: vlopez@tid.es

Oscar Gonzalez de Dios  
Telefonica I+D  
Email: ogondio@tid.es

Xian Zhang  
Huawei Technologies  
Email: zhang.xian@huawei.com

Network Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: April 12, 2013

E. Crabbe  
Google, Inc.  
I. Minei  
Juniper Networks, Inc.  
S. Sivabalan  
Cisco Systems, Inc.  
R. Varga  
Pantheon Technologies SRO  
October 9, 2012

PCEP Extensions for PCE-initiated LSP Setup in a Stateful PCE Model  
draft-crabbe-pce-pce-initiated-lsp-00

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 extensions described in [I-D.ietf-pce-stateful-pce] 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 creation and deletion of PCE-initiated LSPs under 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 12, 2013.

#### Copyright Notice

Copyright (c) 2012 IETF Trust and the persons identified as the document authors. All rights reserved.

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

## Table of Contents

1. Introduction . . . . .	4
2. Terminology . . . . .	4
3. Architectural Overview . . . . .	4
3.1. Motivation . . . . .	5
3.2. Operation overview . . . . .	5
4. Support of PCE-initiated LSPs . . . . .	6
4.1. Stateful PCE Capability TLV . . . . .	6
5. PCE-initiated LSP creation . . . . .	7
5.1. The LSP Create Message . . . . .	7
6. LSP delegation and cleanup . . . . .	9
6.1. LSP delegation procedures . . . . .	9
6.2. LSP cleanup procedures . . . . .	9
6.2.1. LSP-CLEANUP TLV . . . . .	10
7. IANA considerations . . . . .	10
7.1. PCEP-Error Object . . . . .	11
7.2. PCEP TLV Type Indicators . . . . .	11
8. Security Considerations . . . . .	11
8.1. Malicious PCE . . . . .	11
9. Acknowledgements . . . . .	12
10. References . . . . .	12
10.1. Normative References . . . . .	12
10.2. Informative References . . . . .	13
Authors' Addresses . . . . .	13

## 1. Introduction

[RFC5440] describes the Path Computation Element Protocol PCEP. PCEP defines the communication between a Path Computation Client (PCC) and a Path Control Element (PCE), or between PCE and PCE, enabling computation of Multiprotocol Label Switching (MPLS) for Traffic Engineering Label Switched Path (TE LSP) characteristics.

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

This document describes the setup and teardown of PCE-initiated LSPs under the stateful PCE model, without the need for local configuration on the PCC, thus allowing for a dynamic network that is centrally controlled and deployed.

## 2. Terminology

This document uses the following terms defined in [RFC5440]: PCC, PCE, PCEP Peer.

This document uses the following terms defined in [I-D.ietf-pce-stateful-pce]: Stateful PCE, Delegation, Delegation Timeout Interval, LSP State Report, LSP Update Request.

The following terms are defined in this document:

PCE-initiated LSP: LSP that is instantiated as a result of a request from the PCE.

LSP cleanup timer: PCE-defined timer for cleanup of PCE-initiated LSPs that are no longer delegated to a PCE.

The message formats in this document are specified using Routing Backus-Naur Format (RBNF) encoding as specified in [RFC5511].

## 3. Architectural Overview

### 3.1. Motivation

[I-D.ietf-pce-stateful-pce] provides stateful control over LSPs that are locally configured on the PCC. This model relies on the LER taking an active role in delegating locally configured LSPs to the PCE, and is well suited in environments where the LSP placement is fairly static. However, in environments where the LSP placement needs to change in response to application demands, it is useful to support dynamic creation and tear down of LSPs. The ability for a PCE to trigger the creation of LSPs on demand can make possible agile software-driven network operation, and can be seamlessly integrated into a controller-based network architecture, where intelligence in the controller can determine when and where to set up paths.

A possible use case is one of a software-driven network, where applications request network resources and paths from the network infrastructure. For example, an application can request a path with certain constraints between two LSRs by contacting the PCE. The PCE can compute a path satisfying the constraints, and instruct the head end LSR to create and signal it. When the path is no longer required by the application, the PCE can request its teardown.

Another use case is that of demand engineering, where a PCE with visibility into both the network state and the demand matrix can anticipate and optimize how traffic is distributed across the infrastructure. Such optimizations may require creating new paths across the infrastructure.

### 3.2. Operation overview

A PCC indicates its ability to support PCE provisioned dynamic LSPs during the PCEP Initialization Phase via a new flag in the STATEFUL-PCE-CAPABILITY TLV (see details in Section 4.1).

The decision when to create a PCE-initiated LSP is out of the scope of this document. To instantiate an LSP, the PCE sends a new message, the LSP Create Request (PCCreate) message to the PCC. The LSP Create Request MUST include the END-POINTS and LSPA objects, and the LSPA object MUST include the SYMBOLIC-PATH-NAME TLV. The PCC creates the LSP using the attributes communicated by the PCE, and local values for the unspecified parameters. It assigns a unique LSP-ID for the LSP and automatically delegates the LSP to the PCE. It then generates an LSP State Report (PCRpt) for the LSP, carrying the LSP-ID and the delegation bit. The PCE may update the attributes of the LSP via subsequent PCUpd messages.

Subsequent LSP State Report and LSP Update Request for the LSP will carry the PCC-assigned LSP-ID, which uniquely identifies the LSP.



The LSPA Object included in these messages MUST carry the SYMBOLIC-PATH-NAME TLV which will be used to correlate between the PCC-assigned LSP-ID and the LSP. See details in Section 5.

Removal of PCE-initiated LSPs is done by the PCE by setting the R flag in the LSP Object in the PCUpd message. Upon receiving the PCUpd message with the R Flag set, the PCC deletes the LSP. See details in Section 5.

Once instantiated, a PCRpt is generated for the LSP, with the delegation bit set. After this, the delegation procedures for PCE-initiated LSPs are the same as for PCC initiated LSPs. Upon session failure, PCE-initiated LSPs are not immediately removed, in order to avoid LSP flap and service interruption. However, to allow for network cleanup without manual intervention, such "orphan" PCE-initiated LSPs must be either adopted by a different PCE or cleaned up within a time interval. This time is negotiated between PCE and PCC at session initialization time. See details in Section 6.

#### 4. Support of PCE-initiated LSPs

A PCC indicates its ability to support PCE provisioned dynamic LSPs during the PCEP Initialization Phase. The Open Object in the Open message contains the "Stateful PCE Capability" TLV, defined in [I-D.ietf-pce-stateful-pce].

A new flag, the I (LSP-INSTANTIATION-CAPABILITY) flag is introduced to indicate support for instantiation of PCE-initiated LSPs. A PCE wishing to initiate LSPs, can do so only for PCCs that advertised this capability and a PCC will follow the procedures described in this document only on sessions where the PCE advertised the I flag. A PCE or PCC that advertise support of LSP initiation MUST also advertise a cleanup time for the removal of such LSPs. The cleanup time is advertised via a new TLV in the Open Object, the LSP-CLEANUP TLV, discussed in Section 6, and the value is negotiated to the lower one advertised on a session.

##### 4.1. Stateful PCE Capability TLV

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

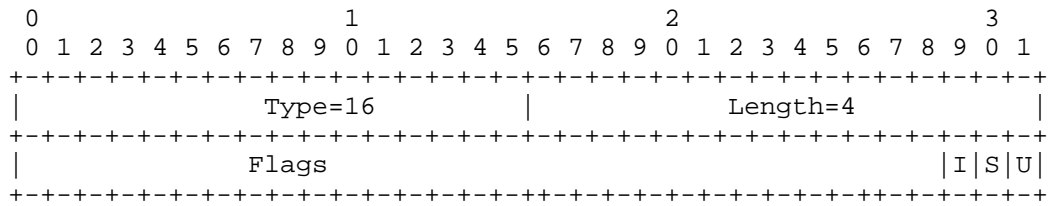


Figure 1: STATEFUL-PCE-CAPABILITY TLV format

The type of the TLV is defined in [I-D.ietf-pce-stateful-pce] and it has a fixed length of 4 octets.

The value comprises a single field - Flags (32 bits). The U and S bits are defined in [I-D.ietf-pce-stateful-pce].

If set to 1 by a PCC, the I Flag indicates that the PCC allows instantiation of an LSP by a PCE. If set to 1 by a PCE, the I flag indicates that the PCE will attempt to instantiate LSPs. The LSP-INstantiation-CAPABILITY flag must be set by both PCC and PCE in order to support PCE-initiated LSP instantiation.

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

## 5. PCE-initiated LSP creation

To create a PCE-initiated LSP, a PCE sends a PCCreate message to a PCC, which include a set of objects and TLVs describing the LSP to be instantiated. The message format, the objects and TLVs are discussed separately below.

### 5.1. The LSP Create Message

A Path Computation LSP Create message (also referred to as PCCreate message) is a PCEP message sent by a PCE to a PCC to trigger an LSP instantiation. The Message-Type field of the PCEP common header for the PCCreate message is set to [TBD].

The PCCreate message MUST include the END-POINTS and the LSPA objects. In the LSPA object, it MUST include the SYMBOLIC-PATH-NAME TLV for the LSP. The PCCreate message MAY include other attributes for the LSP. If specified, the PCC MUST use them for the LSP instantiation, otherwise it MUST use its locally configured values. The error messages will be specified in a future version of this document.

The format of a PCCreate message is as follows:

```
<PCCreate Message> ::= <Common Header>
                        <lsp-instantiation-list>
```

Where:

```
<lsp-instantiation-list> ::= <lsp-instantiation-request>[<lsp-instantiation-1
ist>]
```

```
<lsp-instantiation-request> ::= <END-POINTS>
                                <LSPA>
                                [<ERO>]
                                [<BANDWIDTH>]
                                [<metric-list>]
```

Where:

```
<metric-list> ::= <METRIC>[<metric-list>]
```

The END-POINTS Object contains the source and destination addresses for provisioning the PCE-initiated LSP. If the END-POINTS Object is missing, the PCC MUST send a PCErr message with Error-type=6 (Mandatory Object missing) and Error-value=3 (END-POINTS Object missing).

The LSPA Object MUST include the SYMBOLIC-PATH-NAME TLV, which will be used to correlate between the PCC-assigned LSP-ID and the LSP. The symbolic name used for provisioning PCE-initiated LSPs must not have conflict with the LSP name of any existing LSP in the PCC. (Existing LSPs may be either statically configured, or initiated by another PCE). If there is conflict with the LSP name, the PCC MUST send a PCErr message with Error-type=TBD (Invalid Parameter) and Error-value=TBD (Bad Symbolic Path Name). The only exception to this rule is for LSPs for which the LSP-cleanup timer is running (see Section 6).

PCE-initiated removal of a PCE-initiated LSP is done by setting the R (remove) flag in the LSP Object in the PCUpd request from the PCE. The definition of the R bit is updated as follows:

R (Remove - 1 bit): On PCRpt messages the R Flag indicates that the LSP has been removed from the PCC. Upon receiving a PCRpt message with the R Flag set to 1, the PCE SHOULD remove all state related to the LSP from its database. In PCUpd messages the R flag indicates that the PCE wishes to disable the LSP. Upon receiving the PCUpd message with the R Flag set for a PCE-initiated LSP, the PCC tears

down the LSP and removes its state.

A PCC SHOULD be able to place a limit on either the number of LSPs or the percentage of resources that are allocated to honor PCE-initiated LSP requests. As soon as that limit is reached, the PCC MUST send a PCErr message of type 19 (Invalid Operation) and value TBD "PCE-initiated limit reached" and is free to drop any incoming PCUpd messages without additional processing.

A PCC SHOULD relay to the PCE errors it encounters in the setup of PCE-initiated LSP. The error codes and error processing will be detailed in a future version of this document.

## 6. LSP delegation and cleanup

### 6.1. LSP delegation procedures

PCE-initiated LSPs are automatically delegated by the PCC to the PCE upon instantiation. The PCC MUST delegate the LSP to the PCE by setting the delegation bit to 1 in the PCRpt that includes the assigned LSP-Id. All subsequent messages from the PCC must have the delegation bit set to 1. The PCC cannot revoke the delegation for PCE-initiated LSPs for an active PCEP session. Sending a PCRpt message with the delegation bit set to 0 results in a PCErr message of type 19 (Invalid Operation) and value TBD "Delegation for PCE-initiated LSP cannot be revoked".

A PCE MAY return a delegation to the PCC, to allow for LSP transfer between PCEs. Doing so MUST trigger the LSP cleanup timer described in Section 6.2.

Control over PCE-initiated LSPs reverts to the PCC at the expiration of the delegation timeout. To obtain control of a PCE-initiated LSP, a PCE (either the original or one of its backups) sends a PCCreate message specifying the endpoints and symbolic name (the same process used when initiating an LSP from the PCE). See more in the next section.

### 6.2. LSP cleanup procedures

The LSP cleanup timer ensures that a PCE crash does not result in automatic and immediate disruption for the services using PCE-initiated LSPs. PCE-initiated LSPs are not be 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 LSP cleanup timer is advertised in the session open message via a mandatory TLV for sessions where PCE-initiated

LSPs are supported. The timer is started upon PCEP session failure and is stopped when the LSP is delegated to a PCE. Both PCE and PCC advertise a value for this timer, and the timer value is negotiated to the lower value of the two.

#### 6.2.1. LSP-CLEANUP TLV

The LSP-CLEANUP TLV is advertised in the Open Object and is mandatory when the I flag is set in the STATEFUL-PCE-CAPABILITY TLV. The LSP-CLEANUP TLV contains the time in seconds that the PCC has to wait before cleaning up any PCE-initiated LSPs belonging to a particular PCEP session when a PCEP session terminates. Both PCE and PCC advertise a value for the cleanup time, and the cleanup timer is set to the lower of the two. The timer is triggered on PCEP session failure and reset when the LSP is delegated to a PCE.

Failure to include the mandatory LSP-CLEANUP TLV in the Open Object when the I flag is set MUST trigger PCerr of type 6 (Mandatory Object missing) and value TBD (LSP-CLEANUP TLV missing).

The format of the LSP-CLEANUP TLV is shown in the following figure:

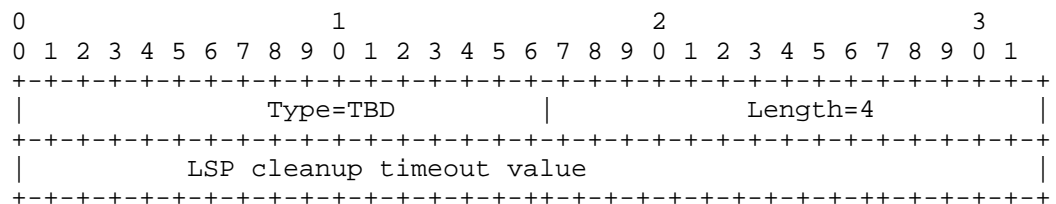


Figure 2: LSP-CLEANUP TLV format

The type of the TLV is TBD and it has a fixed length of 4 octets.

The value comprises a single field, the LSP cleanup timeout value.

The time in seconds to wait before cleaning up PCE-initiated LSPs. Zero means immediate removal. The value 0xFFFFFFFF is reserved.

A PCE may take control of the dynamic LSPs for which the LSP cleanup timer is running by sending an PCCreate request for the LSP. In this case, the "Bad Symbolic Path Name" error MUST NOT be generated, the LSP MUST be delegated and the cleanup timer MUST be stopped.

## 7. IANA considerations

### 7.1. PCEP-Error Object

This document defines new Error-Type and Error-Value for the following new error conditions:

Error-Type	Meaning
6	Mandatory Object missing Error-value=8: LSP cleanup TLV missing
19	Invalid operation Error-value=TBD: PCE-initiated LSP limit reached Error-value=TBD: Delegation for PCE-initiated LSP cannot be revoked

### 7.2. PCEP TLV Type Indicators

This document defines the following new PCEP TLVs:

Value	Meaning	Reference
???	LSP cleanup	This document

## 8. Security Considerations

The security considerations described in [I-D.ietf-pce-stateful-pce] apply to the extensions described in this document. Additional considerations related to a malicious PCE are introduced.

### 8.1. Malicious PCE

The LSP instantiation mechanism described in this document allows a PCE to generate state on the PCC and throughout the network. As a result, it introduces a new attack vector: an attacker may flood the PCC with LSP instantiation requests and consume network and LSR resources, either by spoofing messages or by compromising the PCE itself.

A PCC can protect itself from such an attack by imposing a limit on either the number of LSPs or the percentage of resources that are allocated to honor PCE-initiated LSP requests. As soon as that limit is reached, the PCC MUST send a PCErr message of type 19 (Invalid Operation) and value TBD "PCE-initiated LSP limit reached" (XXX TBD add to the IANA section) and is free to drop any incoming PCUpd messages without additional processing.

Rapid flaps triggered by the PCE can also be an attack vector. This will be discussed in a future version of this document.

## 9. Acknowledgements

We would like to thank Jan Medved, Ambrose Kwong and Raveendra Trovi for their contributions to this document.

## 10. References

### 10.1. Normative References

- [I-D.ietf-pce-stateful-pce]  
Crabbe, E., Medved, J., Varga, R., and I. Minei, "PCEP Extensions for Stateful PCE",  
draft-ietf-pce-stateful-pce-01 (work in progress),  
July 2012.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC2205] Braden, B., Zhang, L., Berson, S., Herzog, S., and S. Jamin, "Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification", RFC 2205, September 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.
- [RFC4090] Pan, P., Swallow, G., and A. Atlas, "Fast Reroute Extensions to RSVP-TE for LSP Tunnels", RFC 4090, May 2005.
- [RFC5088] Le Roux, JL., Vasseur, JP., Ikejiri, Y., and R. Zhang, "OSPF Protocol Extensions for Path Computation Element (PCE) Discovery", RFC 5088, January 2008.
- [RFC5089] Le Roux, JL., Vasseur, JP., Ikejiri, Y., and R. Zhang, "IS-IS Protocol Extensions for Path Computation Element (PCE) Discovery", RFC 5089, January 2008.
- [RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, May 2008.
- [RFC5440] Vasseur, JP. and JL. Le Roux, "Path Computation Element (PCE) Communication Protocol (PCEP)", RFC 5440, March 2009.
- [RFC5511] Farrel, A., "Routing Backus-Naur Form (RBNF): A Syntax

Used to Form Encoding Rules in Various Routing Protocol Specifications", RFC 5511, April 2009.

## 10.2. Informative References

- [RFC2702] Awduche, D., Malcolm, J., Agogbua, J., O'Dell, M., and J. McManus, "Requirements for Traffic Engineering Over MPLS", RFC 2702, September 1999.
- [RFC3031] Rosen, E., Viswanathan, A., and R. Callon, "Multiprotocol Label Switching Architecture", RFC 3031, January 2001.
- [RFC3346] Boyle, J., Gill, V., Hannan, A., Cooper, D., Awduche, D., Christian, B., and W. Lai, "Applicability Statement for Traffic Engineering with MPLS", RFC 3346, August 2002.
- [RFC3630] Katz, D., Kompella, K., and D. Yeung, "Traffic Engineering (TE) Extensions to OSPF Version 2", RFC 3630, September 2003.
- [RFC4655] Farrel, A., Vasseur, J., and J. Ash, "A Path Computation Element (PCE)-Based Architecture", RFC 4655, August 2006.
- [RFC4657] Ash, J. and J. Le Roux, "Path Computation Element (PCE) Communication Protocol Generic Requirements", RFC 4657, September 2006.
- [RFC5305] Li, T. and H. Smit, "IS-IS Extensions for Traffic Engineering", RFC 5305, October 2008.
- [RFC5394] Bryskin, I., Papadimitriou, D., Berger, L., and J. Ash, "Policy-Enabled Path Computation Framework", RFC 5394, December 2008.
- [RFC5557] Lee, Y., Le Roux, J.L., King, D., and E. Oki, "Path Computation Element Communication Protocol (PCEP) Requirements and Protocol Extensions in Support of Global Concurrent Optimization", RFC 5557, July 2009.



Authors' Addresses

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

Email: [edc@google.com](mailto:edc@google.com)

Ina Minei  
Juniper Networks, Inc.  
1194 N. Mathilda Ave.  
Sunnyvale, CA 94089  
US

Email: [ina@juniper.net](mailto:ina@juniper.net)

Siva Sivabalan  
Cisco Systems, Inc.  
170 West Tasman Dr.  
San Jose, CA 95134  
US

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

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

Email: [robert.varga@pantheon.sk](mailto:robert.varga@pantheon.sk)



Network Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: April 13, 2014

E. Crabbe  
Google, Inc.  
I. Minei  
Juniper Networks, Inc.  
S. Sivabalan  
Cisco Systems, Inc.  
R. Varga  
Pantheon Technologies SRO  
October 10, 2013

PCEP Extensions for PCE-initiated LSP Setup in a Stateful PCE Model  
draft-crabbe-pce-pce-initiated-lsp-03

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 extensions described in [I-D.ietf-pce-stateful-pce] 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 creation and deletion of PCE-initiated LSPs under 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 13, 2014.

#### Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

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

## Table of Contents

1. Introduction . . . . .	4
2. Terminology . . . . .	4
3. Architectural Overview . . . . .	4
3.1. Motivation . . . . .	4
3.2. Operation overview . . . . .	5
4. Support of PCE-initiated LSPs . . . . .	6
4.1. Stateful PCE Capability TLV . . . . .	7
5. PCE-initiated LSP instantiation and deletion . . . . .	7
5.1. The LSP Initiate Message . . . . .	7
5.2. The R flag in the SRP Object . . . . .	8
5.3. LSP instantiation . . . . .	9
5.3.1. The Create flag . . . . .	11
5.4. LSP deletion . . . . .	11
6. LSP delegation and cleanup . . . . .	12
7. Implementation status . . . . .	12
8. IANA considerations . . . . .	13
8.1. PCEP Messages . . . . .	13
8.2. LSP Object . . . . .	13
8.3. PCEP-Error Object . . . . .	14
9. Security Considerations . . . . .	14
9.1. Malicious PCE . . . . .	14
9.2. Malicious PCC . . . . .	15
10. Acknowledgements . . . . .	15
11. References . . . . .	15
11.1. Normative References . . . . .	15
11.2. Informative References . . . . .	16
Authors' Addresses . . . . .	17

## 1. Introduction

[RFC5440] describes the Path Computation Element Protocol PCEP. PCEP defines the communication between a Path Computation Client (PCC) and a Path Control Element (PCE), or between PCE and PCE, enabling computation of Multiprotocol Label Switching (MPLS) for Traffic Engineering Label Switched Path (TE LSP) characteristics.

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

This document describes the setup, maintenance and teardown of PCE-initiated LSPs under the stateful PCE model, without the need for local configuration on the PCC, thus allowing for a dynamic network that is centrally controlled and deployed.

## 2. Terminology

This document uses the following terms defined in [RFC5440]: PCC, PCE, PCEP Peer.

This document uses the following terms defined in [I-D.ietf-pce-stateful-pce]: Stateful PCE, Delegation, Redelegation Timeout, State Timeout Interval LSP State Report, LSP Update Request.

The following terms are defined in this document:

PCE-initiated LSP: LSP that is instantiated as a result of a request from the PCE.

The message formats in this document are specified using Routing Backus-Naur Format (RBNF) encoding as specified in [RFC5511].

## 3. Architectural Overview

### 3.1. Motivation

[I-D.ietf-pce-stateful-pce] provides stateful control over LSPs that are locally configured on the PCC. This model relies on the LER taking an active role in delegating locally configured LSPs to the

PCE, and is well suited in environments where the LSP placement is fairly static. However, in environments where the LSP placement needs to change in response to application demands, it is useful to support dynamic creation and tear down of LSPs. The ability for a PCE to trigger the creation of LSPs on demand can make possible agile software-driven network operation, and can be seamlessly integrated into a controller-based network architecture, where intelligence in the controller can determine when and where to set up paths.

A possible use case is one of a software-driven network, where applications request network resources and paths from the network infrastructure. For example, an application can request a path with certain constraints between two LSRs by contacting the PCE. The PCE can compute a path satisfying the constraints, and instruct the head end LSR to instantiate and signal it. When the path is no longer required by the application, the PCE can request its teardown.

Another use case is one of dynamically adjusting aggregate bandwidth between two points in the network using multiple LSPs. This functionality is very similar to auto-bandwidth, but allows for providing the desired capacity through multiple LSPs. This approach overcomes two of the limitations auto-bandwidth can experience: 1) growing the capacity between the endpoints beyond the capacity of individual links in the path and 2) achieving good bin-packing through use of several small LSPs instead of a single large one. The number of LSPs varies based on the demand, and LSPs are created and deleted dynamically to satisfy the bandwidth requirements.

Another use case is that of demand engineering, where a PCE with visibility into both the network state and the demand matrix can anticipate and optimize how traffic is distributed across the infrastructure. Such optimizations may require creating new paths across the infrastructure.

### 3.2. Operation overview

A PCC or PCE indicates its ability to support PCE provisioned dynamic LSPs during the PCEP Initialization Phase via a new flag in the STATEFUL-PCE-CAPABILITY TLV (see details in Section 4.1).

The decision when to instantiate or delete a PCE-initiated LSP is out of the scope of this document. To instantiate or delete an LSP, the PCE sends a new message, the Path Computation LSP Initiate Request (PCInitiate) message to the PCC. The LSP Initiate Request MUST include the SRP and LSP objects, and the LSP object MUST include the Symbolic Path Name TLV and MUST have a PLSP-ID of 0.

For an instantiation operation, the PCE MUST include the ERO and END-

POINTS object and may include various attributes as per [RFC5440]. The PCC creates the LSP using the attributes communicated by the PCE, and local values for the unspecified parameters. It assigns a unique PLSP-ID for the LSP and automatically delegates the LSP to the PCE. It also generates an LSP State Report (PCRpt) for the LSP, carrying the newly assigned PLSP-ID and indicating the delegation via the Delegate flag in the LSP object. In addition to the Delegate flag, the PCC also sets the Create flag in the LSP object (see Section 5.3.1), to indicate that the LSP was created as a result of a PCInitiate message. This PCRpt message MUST include the SRP object, with the SRP-id-number used in the SRP object of the PCInitiate message. The PCE may update the attributes of the LSP via subsequent PCUpd messages. Subsequent LSP State Report and LSP Update Request for the LSP will carry the PCC-assigned PLSP-ID, which uniquely identifies the LSP. See details in Section 5.3.

Once instantiated, the delegation procedures for PCE-initiated LSPs are the same as for PCC initiated LSPs as described in [I-D.ietf-pce-stateful-pce]. This applies to the case of a PCE failure as well. In order to allow for network cleanup without manual intervention, the PCC SHOULD support removal of PCE-initiated LSPs as one of the behaviors applied on expiration of the State Timeout Interval [I-D.ietf-pce-stateful-pce]. The behavior SHOULD be picked based on local policy, and can result either in LSP removal, or into reverting to operator-defined default parameters. See details in Section 6. A PCE may return a delegation to the PCC in order to facilitate re-delegation of its LSPs to an alternate PCE.

To indicate a delete operation, the PCE MUST use the R flag in the SRP object in a PCUpd message. As a result of the deletion request, the PCC MUST remove all state related to the LSP, and send a PCRpt with the R flag set in the LSP object for the removed state. See details in Section 5.4.

#### 4. Support of PCE-initiated LSPs

A PCC indicates its ability to support PCE provisioned dynamic LSPs during the PCEP Initialization phase. The Open Object in the Open message contains the "Stateful PCE Capability" TLV, defined in [I-D.ietf-pce-stateful-pce]. A new flag, the I (LSP-INSTANTIATION-CAPABILITY) flag is introduced to indicate support for instantiation of PCE-initiated LSPs. A PCE can initiate LSPs only for PCCs that advertised this capability and a PCC will follow the procedures described in this document only on sessions where the PCE advertised the I flag.



#### 4.1. Stateful PCE Capability TLV

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

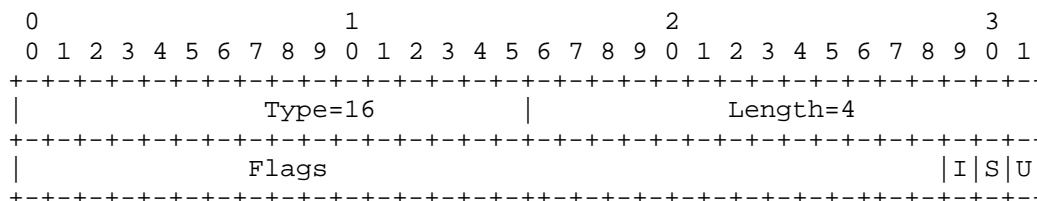


Figure 1: STATEFUL-PCE-CAPABILITY TLV format

The type of the TLV is defined in [I-D.ietf-pce-stateful-pce] and it has a fixed length of 4 octets.

The value comprises a single field - Flags (32 bits). The U and S bits are defined in [I-D.ietf-pce-stateful-pce].

I (LSP-INSTANTIATION-CAPABILITY - 1 bit): If set to 1 by a PCC, the I Flag indicates that the PCC allows instantiation of an LSP by a PCE. If set to 1 by a PCE, the I flag indicates that the PCE will attempt to instantiate LSPs. The LSP-INSTANTIATION-CAPABILITY flag must be set by both PCC and PCE in order to support PCE-initiated LSP instantiation.

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

#### 5. PCE-initiated LSP instantiation and deletion

To initiate an LSP, a PCE sends a PCInitiate message to a PCC. The message format, objects and TLVs are discussed separately below for the creation and the deletion cases.

##### 5.1. The LSP Initiate Message

A Path Computation LSP Initiate Message (also referred to as PCInitiate message) is a PCEP message sent by a PCE to a PCC to trigger LSP instantiation or deletion. The Message-Type field of the PCEP common header for the PCInitiate message is set to [TBD]. The PCInitiate message MUST include the SRP and the LSP objects, and may contain other objects, as discussed later in this section. If either the SRP or the LSP object is missing, the PCC MUST send a PCErr as described in [I-D.ietf-pce-stateful-pce]. LSP instantiation is done

by sending an LSP Initiate Message with an LSP object with the reserved PLSP-ID 0. LSP deletion is done by sending an LSP Initiate Message with an LSP object carrying the PLSP-ID of the LSP to be removed and an SRP object with the R flag set (see Section 5.2).

The format of a PCInitiate message for LSP instantiation is as follows:

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

Where:

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

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

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

```
<PCE-initiated-lsp-deletion> ::= <SRP>
                                   <LSP>
```

Where:

<attribute-list> is defined in [RFC5440] and extended by PCEP extensions.

The SRP object is used to correlate between initiation requests sent by the PCE and the error reports and state reports sent by the PCC. Every request from the PCE receives a new SRP-ID-number. This number is unique per PCEP session and is incremented each time an operation (initiation, update, etc) is requested from the PCE. The value of the SRP-ID-number MUST be echoed back by the PCC in PCErr and PCrpt messages to allow for correlation between requests made by the PCE and errors or state reports generated by the PCC. Details of the SRP object and its use can be found in [I-D.ietf-pce-stateful-pce].

## 5.2. The R flag in the SRP Object

The format of the SRP object is shown Figure 2:

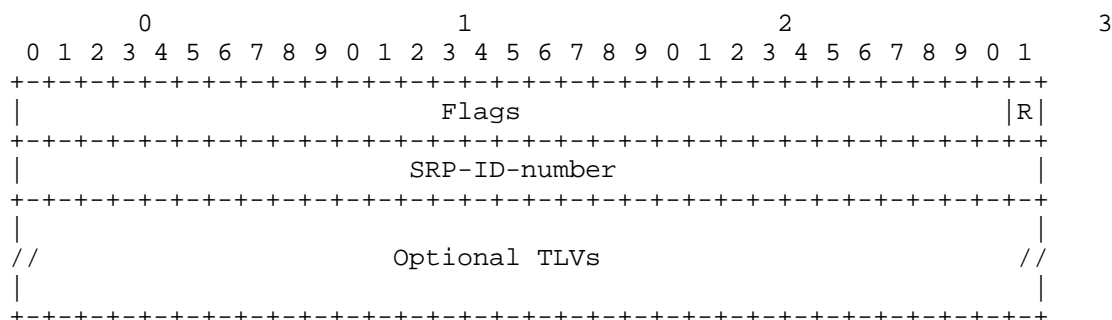


Figure 2: The SRP Object format

The type object is defined in [I-D.ietf-pce-stateful-pce].

A new flag is defined to indicate a delete operation initiated by the PCE:

R (LSP-REMOVE - 1 bit): If set to 1, it indicates a removal request initiated by the PCE.

### 5.3. LSP instantiation

LSP instantiation is done by sending an LSP Initiate Message with an LSP object with the reserved PLSP-ID 0. The LSP is set up using RSVP-TE, extensions for other setup methods are outside the scope of this draft.

Receipt of a PCInitiate Message with a non-zero PLSP-ID and the R flag in the SRP object set to zero results in a PCErr message of type 19 (Invalid Operation) and value 8 (non-zero PLSP-ID in LSP initiation request).

The END-POINTS Object is mandatory for an instantiation request of an RSVP-signaled LSP. It contains the source and destination addresses for provisioning the LSP. If the END-POINTS Object is missing, the PCC MUST send a PCErr message with Error-type=6 (Mandatory Object missing) and Error-value=3 (END-POINTS Object missing).

The ERO Object is mandatory for an instantiation request. It contains the ERO for the LSP. If the ERO Object is missing, the PCC MUST send a PCErr message with Error-type=6 (Mandatory Object missing) and Error-value=9 (ERO Object missing).

The LSP Object MUST include the SYMBOLIC-PATH-NAME TLV, which will be used to correlate between the PCC-assigned PLSP-ID and the LSP. If

the TLV is missing, the PCC MUST send a PCErr message with Error-type=6(Mandatory object missing) and Error-value=14 (SYMBOLIC-PATH-NAME TLV missing). The symbolic name used for provisioning PCE-initiated LSPs must not have conflict with the LSP name of any existing LSP in the PCC. (Existing LSPs may be either statically configured, or initiated by another PCE). If there is conflict with the LSP name, the PCC MUST send a PCErr message with Error-type=23 (Bad Parameter value) and Error-value=1 (SYMBOLIC-PATH-NAME in use). The only exception to this rule is for LSPs for which the State timeout timer is running (see Section 6).

The PCE MAY include various attributes as per [RFC5440]. The PCC MUST use these values in the LSP instantiation, and local values for unspecified parameters. After the LSP setup, the PCC MUST send a PCRpt to the PCE, reflecting these values. The SRP object in the PCRpt message MUST echo the value of the PCInitiate message that triggered the setup. LSPs that were instantiated as a result of a PCInitiate message MUST have the C flag set in the LSP object.

If the PCC determines that the LSP parameters proposed in the PCInitiate message are unacceptable, it MUST trigger a PCErr with error-type=TBD (PCE instantiation error) and error-value=1 (Unacceptable instantiation parameters). If the PCC encounters an internal error during the processing of the PCInitiate message, it MUST trigger a PCErr with error-type=TBD (PCE instantiation error) and error-value=2 (Internal error).

A PCC MUST relay to the PCE errors it encounters in the setup of PCE-initiated LSP by sending a PCErr with error-type=TBD (PCE instantiation error) and error-value=3 (RSVP signaling error). The PCErr MUST echo the SRP-id-number of the PCInitiate message. The PCEP-ERROR object SHOULD include the RSVP Error Spec TLV (if an ERROR SPEC was returned to the PCC by a downstream node). After the LSP is set up, errors in RSVP signaling are reported in PCRpt messages, as described in [I-D.ietf-pce-stateful-pce].

A PCC SHOULD be able to place a limit on either the number of LSPs or the percentage of resources that are allocated to honor PCE-initiated LSP requests. As soon as that limit is reached, the PCC MUST send a PCErr message of type 19 (Invalid Operation) and value TBD "PCE-initiated limit reached" and is free to drop any incoming PCInitiate messages without additional processing.

Similarly, the PCE SHOULD be able to place a limit on either the number of LSP initiation requests pending for a particular PCC, or on the time it waits for a response (positive or negative) to a PCInitiate request from a PCC and MAY take further action (such as closing the session or removing all its LSPs) if this limit is

reached.

On successful completion of the LSP instantiation, the PCC assigns a PLSP-ID, and immediately delegates the LSP to the PCE by sending a PCRpt with the Delegate flag set. The PCRpt MUST include the SRP-ID-number of the PCInitiate request that triggered its creation. PCE-initiated LSPs are identified with the Create flag in the LSP Object.

#### 5.3.1. The Create flag

The LSP object is defined in [I-D.ietf-pce-stateful-pce] and included here for easy reference.

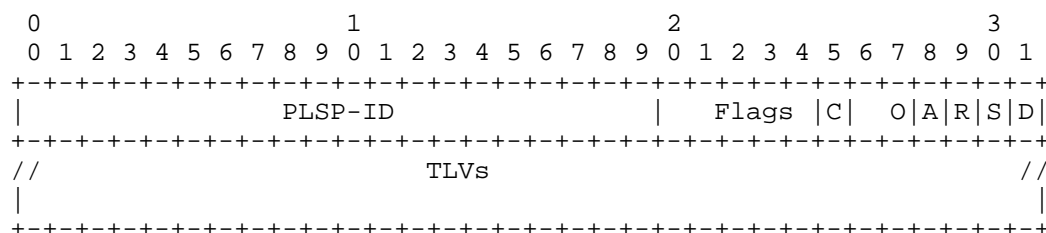


Figure 3: The LSP Object format

A new flag, the Create (C) flag is introduced. On a PCRpt message, the C Flag set to 1 indicates that this LSP was created via a PCInitiate message. The C Flag MUST be set to 1 on each PCRpt message for the duration of existence of the LSP. The Create flag allows PCEs to be aware of which LSPs were PCE-initiated (a state that would otherwise only be known by the PCC and the PCE that initiated them).

#### 5.4. LSP deletion

PCE-initiated removal of a PCE-initiated LSP is done by setting the R (remove) flag in the SRP Object in the PCInitiate message from the PCE. The LSP is identified by the PLSP-ID in the LSP object. If the PLSP-ID is unknown, the PCC MUST generate a PCErr with error type 19, error value 3, "Unknown PLSP-ID". A PLSP-ID of zero removes all LSPs that were initiated by the PCE. If the PLSP-ID specified in the PCInitiate message is not delegated to the PCE, the PCC MUST send a PCErr message indicating "LSP is not delegated" (Error code 19, error value 1 ([I-D.ietf-pce-stateful-pce])). If the PLSP-ID specified in the PCInitiate message was not created by the PCE, the PCC MUST send a PCErr message indicating "LSP is not PCE initiated" (Error code 19, error value TBD). Following the removal of the LSP, the PCC MUST send a PCRpt as described in [I-D.ietf-pce-stateful-pce]. 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 SHOULD be set.

## 6. LSP delegation and cleanup

PCE-initiated LSPs are automatically delegated by the PCC to the PCE upon instantiation. The PCC MUST delegate the LSP to the PCE by setting the delegation bit to 1 in the PCRpt that includes the assigned PLSP-ID. All subsequent messages from the PCC must have the delegation bit set to 1. The PCC cannot revoke the delegation for PCE-initiated LSPs for an active PCEP session. Sending a PCRpt message with the delegation bit set to 0 results in a PCErr message of type 19 (Invalid Operation) and value TBD "Delegation for PCE-initiated LSP cannot be revoked". The PCE MAY further react by closing the session.

A PCE MAY return a delegation to the PCC, to allow for LSP transfer between PCEs. Doing so MUST trigger the State Timeout Interval timer ([I-D.ietf-pce-stateful-pce]).

In case of PCEP session failure, control over PCE-initiated LSPs reverts to the PCC at the expiration of the redelegation timeout. To obtain control of a PCE-initiated LSP, a PCE (either the original or one of its backups) sends a PCInitiate message, including just the SRP and LSP objects, and carrying the PLSP-ID of the LSP it wants to take control of. Receipt of a PCInitiate message with a non-zero PLSP-ID normally results in the generation of a PCErr. If the State Timeout timer is running, the PCC MUST NOT generate an error and redelegate the LSP to the PCE. The State Timeout timer is stopped upon the redelegation. After obtaining control of the LSP, the PCE may remove it using the procedures described in this document.

The State Timeout timer ensures that a PCE crash does not result in automatic and immediate disruption for the services using PCE-initiated LSPs. PCE-initiated LSPs are not be 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 SHOULD support removal of PCE-initiated LSPs as one of the behaviors applied on expiration of the State Timeout Interval [I-D.ietf-pce-stateful-pce]. The behavior SHOULD be picked based on local policy, and can result either in LSP removal, or into reverting to operator-defined default parameters.

## 7. Implementation status

This section to be removed by the RFC editor.

This section records the status of known implementations of the protocol defined by this specification at the time of posting of this Internet-Draft, and is based on a proposal described in RFC 6982. The description of implementations in this section is intended to assist the IETF in its decision processes in progressing drafts to RFCs. Please note that the listing of any individual implementation here does not imply endorsement by the IETF. Furthermore, no effort has been spent to verify the information presented here that was supplied by IETF contributors. This is not intended as, and must not be construed to be, a catalog of available implementations or their features. Readers are advised to note that other implementations may exist.

According to RFC 6982, "this will allow reviewers and working groups to assign due consideration to documents that have the benefit of running code, which may serve as evidence of valuable experimentation and feedback that have made the implemented protocols more mature. It is up to the individual working groups to use this information as they see fit".

Two vendors are implementing the extensions described in this draft and have included the functionality in releases that will be shipping in the near future. An additional entity is working on implementing these extensions in the scope of research projects.

## 8. IANA considerations

### 8.1. PCEP Messages

This document defines the following new PCEP messages:

Value	Meaning	Reference
12	Initiate	This document

### 8.2. LSP Object

The following values are defined in this document for the Flags field in the LSP Object.

Bit	Description	Reference
24	Create	This document

### 8.3. PCEP-Error Object

This document defines new Error-Type and Error-Value for the following new error conditions:

Error-Type	Meaning
6	Mandatory Object missing Error-value=13: LSP cleanup TLV missing Error-value=14: SYMBOLIC-PATH-NAME TLV missing
19	Invalid operation Error-value=6: PCE-initiated LSP limit reached Error-value=7: Delegation for PCE-initiated LSP cannot be revoked Error-value=8: Non-zero PLSP-ID in LSP initiation request
23	Bad parameter value Error-value=1: SYMBOLIC-PATH-NAME in use
24	LSP instantiation error Error-value=1: Unacceptable instantiation parameters Error-value=2: Internal error Error-value=3: RSVP signaling error

## 9. Security Considerations

The security considerations described in [I-D.ietf-pce-stateful-pce] apply to the extensions described in this document. Additional considerations related to a malicious PCE are introduced.

### 9.1. Malicious PCE

The LSP instantiation mechanism described in this document allows a PCE to generate state on the PCC and throughout the network. As a result, it introduces a new attack vector: an attacker may flood the PCC with LSP instantiation requests and consume network and LSR resources, either by spoofing messages or by compromising the PCE itself.

A PCC can protect itself from such an attack by imposing a limit on either the number of LSPs or the percentage of resources that are allocated to honor PCE-initiated LSP requests. As soon as that limit is reached, the PCC MUST send a PCErr message of type 19 (Invalid Operation) and value 3 "PCE-initiated LSP limit reached" and is free to drop any incoming PCInitiate messages for LSP instantiation without additional processing.

Rapid flaps triggered by the PCE can also be an attack vector. This will be discussed in a future version of this document.



## 9.2. Malicious PCC

The LSP instantiation mechanism described in this document requires the PCE to keep state for LSPs that it instantiates and relies on the PCC responding (with either a state report or an error message) to requests for LSP instantiation. A malicious PCC or one that reached the limit of the number of PCE-initiated LSPs, can ignore PCE requests and consume PCE resources. A PCE can protect itself by imposing a limit on the number of requests pending, or by setting a timeout and it MAY take further action such as closing the session or removing all the LSPs it initiated.

## 10. Acknowledgements

We would like to thank Jan Medved, Ambrose Kwong, Ramon Casellas, Dhruv Dhody, and Raveendra Trovi for their contributions to this document.

## 11. References

### 11.1. Normative References

- [I-D.ietf-pce-stateful-pce]  
Crabbe, E., Medved, J., Minei, I., and R. Varga, "PCEP Extensions for Stateful PCE",  
draft-ietf-pce-stateful-pce-07 (work in progress),  
October 2013.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC2205] Braden, B., Zhang, L., Berson, S., Herzog, S., and S. Jamin, "Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification", RFC 2205, September 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.
- [RFC4090] Pan, P., Swallow, G., and A. Atlas, "Fast Reroute Extensions to RSVP-TE for LSP Tunnels", RFC 4090, May 2005.
- [RFC5088] Le Roux, J.L., Vasseur, J.P., Ikejiri, Y., and R. Zhang, "OSPF Protocol Extensions for Path Computation Element (PCE) Discovery", RFC 5088, January 2008.

- [RFC5089] Le Roux, JL., Vasseur, JP., Ikejiri, Y., and R. Zhang, "IS-IS Protocol Extensions for Path Computation Element (PCE) Discovery", RFC 5089, January 2008.
- [RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, May 2008.
- [RFC5440] Vasseur, JP. and JL. Le Roux, "Path Computation Element (PCE) Communication Protocol (PCEP)", RFC 5440, March 2009.
- [RFC5511] Farrel, A., "Routing Backus-Naur Form (RBNF): A Syntax Used to Form Encoding Rules in Various Routing Protocol Specifications", RFC 5511, April 2009.

#### 11.2. Informative References

- [RFC2702] Awduche, D., Malcolm, J., Agogbua, J., O'Dell, M., and J. McManus, "Requirements for Traffic Engineering Over MPLS", RFC 2702, September 1999.
- [RFC3031] Rosen, E., Viswanathan, A., and R. Callon, "Multiprotocol Label Switching Architecture", RFC 3031, January 2001.
- [RFC3346] Boyle, J., Gill, V., Hannan, A., Cooper, D., Awduche, D., Christian, B., and W. Lai, "Applicability Statement for Traffic Engineering with MPLS", RFC 3346, August 2002.
- [RFC3630] Katz, D., Kompella, K., and D. Yeung, "Traffic Engineering (TE) Extensions to OSPF Version 2", RFC 3630, September 2003.
- [RFC4655] Farrel, A., Vasseur, J., and J. Ash, "A Path Computation Element (PCE)-Based Architecture", RFC 4655, August 2006.
- [RFC4657] Ash, J. and J. Le Roux, "Path Computation Element (PCE) Communication Protocol Generic Requirements", RFC 4657, September 2006.
- [RFC5305] Li, T. and H. Smit, "IS-IS Extensions for Traffic Engineering", RFC 5305, October 2008.
- [RFC5394] Bryskin, I., Papadimitriou, D., Berger, L., and J. Ash, "Policy-Enabled Path Computation Framework", RFC 5394, December 2008.
- [RFC5557] Lee, Y., Le Roux, JL., King, D., and E. Oki, "Path

Computation Element Communication Protocol (PCEP)  
Requirements and Protocol Extensions in Support of Global  
Concurrent Optimization", RFC 5557, July 2009.

[RFC6982] Sheffer, Y. and A. Farrel, "Improving Awareness of Running  
Code: The Implementation Status Section", RFC 6982,  
July 2013.

#### Authors' Addresses

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

Email: edc@google.com

Ina Minei  
Juniper Networks, Inc.  
1194 N. Mathilda Ave.  
Sunnyvale, CA 94089  
US

Email: ina@juniper.net

Siva Sivabalan  
Cisco Systems, Inc.  
170 West Tasman Dr.  
San Jose, CA 95134  
US

Email: msiva@cisco.com

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

Email: robert.varga@pantheon.sk



Network Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: April 18, 2013

E. Crabbe  
Google, Inc.  
J. Medved  
Cisco Systems, Inc.  
I. Minei  
Juniper Networks, Inc.  
R. Varga  
Pantheon Technologies SRO  
October 15, 2012

Stateful PCE extensions for MPLS-TE LSPs  
draft-crabbe-pce-stateful-pce-mpls-te-00

Abstract

The Path Computation Element Communication Protocol (PCEP) provides mechanisms for Path Computation Elements (PCEs) to perform path computations in response to Path Computation Clients (PCCs) requests.

[I-D.ietf-pce-stateful-pce] describes a set of extensions to PCEP to provide stateful control. This document describes the objects and TLVs to be used with these PCEP extensions to control Multiprotocol Label Switching (MPLS) Traffic Engineering Label Switched Paths (TE LSP) via a stateful PCE.

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 18, 2013.

## Copyright Notice

Copyright (c) 2012 IETF Trust and the persons identified as the document authors. All rights reserved.

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

## Table of Contents

1. Introduction . . . . .	3
2. Terminology . . . . .	3
3. Motivation . . . . .	3
4. MPLS-TE specific descriptors used in PCEP Messages . . . . .	3
4.1. MPLS-TE specific descriptors for the PCRpt Message . . . . .	4
4.2. MPLS-TE specific descriptors for the PCUpd Message . . . . .	4
4.3. MPLS-TE specific encoding for the PCReq Message for stateful PCE . . . . .	6
4.4. MPLS-TE specific encoding for the PCRep Message for stateful PCE . . . . .	7
5. Object and TLV Formats . . . . .	8
5.1. LSP Identifiers TLVs . . . . .	8
5.2. Tunnel ID TLV . . . . .	11
5.3. LSP Update Error Code TLV . . . . .	11
6. IANA Considerations . . . . .	12
6.1. PCEP Objects . . . . .	12
6.2. PCEP-Error Object . . . . .	12
6.3. PCEP TLV Type Indicators . . . . .	12
7. Security Considerations . . . . .	13
8. Acknowledgements . . . . .	13
9. References . . . . .	13
9.1. Normative References . . . . .	13
9.2. Informative References . . . . .	14
Authors' Addresses . . . . .	15

## 1. Introduction

The Path Computation Element Communication Protocol (PCEP) provides mechanisms for Path Computation Elements (PCEs) to perform path computations in response to Path Computation Clients (PCCs) requests.

[I-D.ietf-pce-stateful-pce] describes a set of extensions to PCEP to provide stateful control. This document describes the objects and TLVs to be used with these PCEP extensions to control Multiprotocol Label Switching (MPLS) Traffic Engineering Label Switched Paths (TE LSP) via a stateful PCE.

## 2. Terminology

This document uses the following terms defined in [RFC5440]: PCC, PCE, PCEP Peer.

This document uses the following terms defined in [RFC4090]: MPLS TE Fast Reroute (FRR), FRR One-to-One Backup, FRR Facility Backup.

This document uses the following terms defined in [I-D.ietf-pce-stateful-pce] : Passive Stateful PCE, Active Stateful PCE, Delegation, Delegation Timeout Interval, LSP State Report, LSP Update Request, LSP Priority, LSP State Database, Revocation.

Within this document, when describing PCE-PCE communications, the requesting PCE fills the role of a PCC. This provides a saving in documentation without loss of function.

The message formats in this document are specified using Routing Backus-Naur Format (RBNF) encoding as specified in [RFC5511].

## 3. Motivation

Several use cases for stateful PCE in an MPLS-TE network are included in [I-D.ietf-pce-stateful-pce].

## 4. MPLS-TE specific descriptors used in 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. [I-D.ietf-pce-stateful-pce] describes the messages and objects needed in support of stateful PCE. The following sections contain MPLS-TE specific descriptors used in some of these messages.

## 4.1.    MPLS-TE specific descriptors for the PCRpt Message

The format of the PCRpt message is defined in [I-D.ietf-pce-stateful-pce] as follows, and included here for easy reference:

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

Where:

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

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

Where:

```
<path-list> ::= <path>[<path-list>]
```

For MPLS-TE LSPs, the path descriptor is defined as follows:

```
<path> ::= <ERO><attribute-list>
```

Where:

```
<attribute-list> ::= [<LSPA>]
                      [<BANDWIDTH>]
                      [<RRO>]
                      [<metric-list>]
```

```
<metric-list> ::= <METRIC>[<metric-list>]
```

The LSP State Report MAY contain a path descriptor for the primary path and one or more path descriptors for backup paths. A path descriptor MUST contain an ERO object as it was specified by a PCE or an operator. A path descriptor MUST contain the RRO object if a primary or secondary LSP is set up along the path in the network. A path descriptor MAY contain the LSPA, BANDWIDTH, and METRIC objects. The ERO, LSPA, BANDWIDTH, METRIC, and RRO objects are defined in [RFC5440].

## 4.2.    MPLS-TE specific descriptors for the PCUpd Message

A Path Computation LSP Update Request message (also referred to as PCUpd message) is a PCEP message sent by a PCE to a PCC to update attributes of an LSP. A PCUpd message can carry more than one LSP Update Request. The Message-Type field of the PCEP common header for the PCUpd message is set to [TBD].



The format of the PCUpd message is defined in [I-D.ietf-pce-stateful-pce] and included here for easy reference:

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

Where:

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

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

Where:

```
<path-list> ::= <path>[<path-list>]
```

For MPLS-TE LSPs, the encoding of path descriptor is defined as follows:

```
<path> ::= <ERO><attribute-list>
```

Where:

```
<path> ::= <ERO><attribute-list>
```

Where:

```
<attribute-list> ::= [<LSPA>]
                        [<BANDWIDTH>]
                        [<metric-list>]
```

```
<metric-list> ::= <METRIC>[<metric-list>]
```

There is one mandatory object that MUST be included within each LSP Update Request in the PCUpd message: the LSP object (see [I-D.ietf-pce-stateful-pce]). If the LSP object is missing, the receiving PCE MUST send a PCErr message with Error-type=6 (Mandatory Object missing) and Error-value=[TBD] (LSP object missing).

The LSP Update Request MUST contain a path descriptor for the primary path, and MAY contain one or more path descriptors for backup paths. A path descriptor MUST contain an ERO object. A path descriptor MAY further contain the BANDWIDTH, IRO, and METRIC objects. The ERO, LSPA, BANDWIDTH, METRIC, and IRO objects are defined in [RFC5440].

Each LSP Update Request results in a separate LSP setup operation at a PCC. An LSP Update Request MUST contain all LSP parameters that a PCC wishes to set for the LSP. A PCC MAY set missing parameters from

locally configured defaults. If the LSP specified the Update Request is already up, it will be re-signaled. The PCC will use make-before-break whenever possible in the re-signaling operation.

A PCC MUST respond with an LSP State Report to each LSP Update Request to indicate the resulting state of the LSP in the network. A PCC MAY respond with multiple LSP State Reports to report LSP setup progress of a single LSP.

If the rate of PCUpd messages sent to a PCC for the same target LSP exceeds the rate at which the PCC can signal LSPs into the network, the PCC MAY perform state compression and only re-signal the last modification in its queue.

Note that a PCC MUST process all LSP Update Requests - for example, an LSP Update Request is sent when a PCE returns delegation or puts an LSP into non-operational state. The protocol relies on TCP for message-level flow control.

Note also that it's up to the PCE to handle inter-LSP dependencies; for example, if ordering of LSP set-ups is required, the PCE has to wait for an LSP State Report for a previous LSP before triggering the LSP setup of a next LSP.

#### 4.3. MPLS-TE specific encoding for the PCReq Message for stateful PCE

A PCC MAY include the LSP object defined in [I-D.ietf-pce-stateful-pce] in the PCReq message if the stateful PCE capability has been negotiated on a PCEP session between the PCC and a PCE. The definition of the PCReq message (see [RFC5440], Section 6.4) is then extended as follows:

```
<PCReq Message> ::= <Common Header>
                        [<svec-list>]
                        <request-list>
```

Where:

```
<svec-list> ::= <SVEC> [<svec-list>]
<request-list> ::= <request> [<request-list>]

<request> ::= <RP>
                <END-POINTS>
                [<LSP>]                <--- New Object
                [<LSPA>]
                [<BANDWIDTH>]
                [<metric-list>]
                [<RRO> [<BANDWIDTH>]]
                [<IRO>]
                [<LOAD-BALANCING>]
```

Where:

```
<metric-list> ::= <METRIC> [<metric-list>]
```

#### 4.4. MPLS-TE specific encoding for the PCRep Message for stateful PCE

A PCE MAY include the LSP object defined in [I-D.ietf-pce-stateful-pce] in the PCRep message if the stateful PCE capability has been negotiated on a PCEP session between the PCC and the PCE and the LSP object was included in the corresponding PCReq message from the PCC. The definition of the PCRep message (see [RFC5440], Section 6.5) is then extended as follows

```

<PCRep Message> ::= <Common Header>
                    <response-list>

```

Where:

```

<response-list> ::= <response> [<response-list>]

<response> ::= <RP>
               [<LSP>]                <--- New Object
               [<NO-PATH>]
               [<attribute-list>]
               [<path-list>]

<path-list> ::= <path> [<path-list>]

<path> ::= <ERO> <attribute-list>

```

Where:

```

<attribute-list> ::= [<LSPA>]
                    [<BANDWIDTH>]
                    [<metric-list>]
                    [<IRO>]

<metric-list> ::= <METRIC> [<metric-list>]

```

## 5. Object and TLV Formats

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.

### 5.1. LSP Identifiers TLVs

Whenever the value of an LSP identifier changes, a PCC MUST send out an LSP State Report, where the LSP Object carries the LSP Identifiers TLV that contains the new value. The LSP Identifiers TLV MUST also be included in the LSP object during state synchronization. There are two LSP Identifiers TLVs, one for IPv4 and one for IPv6.

The format of the IPV4-LSP-IDENTIFIERS TLV is shown in the following figure:

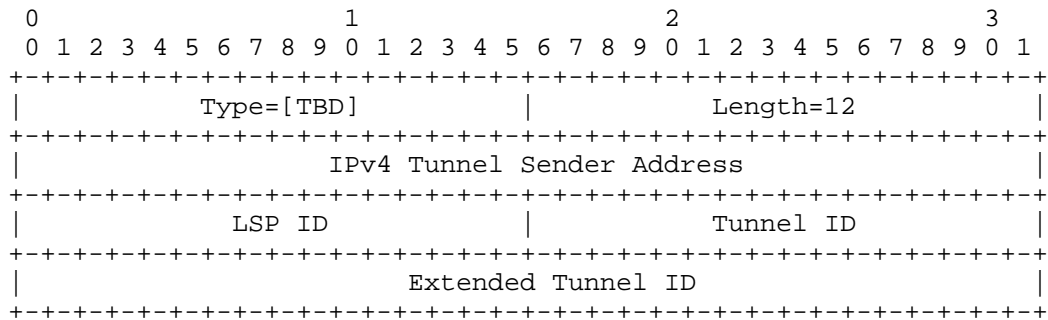


Figure 1: IPV4-LSP-IDENTIFIERS TLV format

The type of the TLV is [TBD] and it has a fixed length of 12 octets. The value contains the following fields:

IPv4 Tunnel Sender Address: contains the sender node's IPv4 address, as defined in [RFC3209], Section 4.6.2.1 for the LSP\_TUNNEL\_IPv4 Sender Template Object.

LSP ID: contains the 16-bit 'LSP ID' identifier defined in [RFC3209], Section 4.6.2.1 for the LSP\_TUNNEL\_IPv4 Sender Template Object.

Tunnel ID: contains the 16-bit 'Tunnel ID' identifier defined in [RFC3209], Section 4.6.1.1 for the LSP\_TUNNEL\_IPv4 Session Object. Tunnel ID remains constant over the life time of a tunnel. However, when Global Path Protection or Global Default Restoration is used, both the primary and secondary LSPs have their own Tunnel IDs. A PCC will report a change in Tunnel ID when traffic switches over from primary LSP to secondary LSP (or vice versa).

Extended Tunnel ID: contains the 32-bit 'Extended Tunnel ID' identifier defined in [RFC3209], Section 4.6.1.1 for the LSP\_TUNNEL\_IPv4 Session Object.

The format of the IPV6-LSP-IDENTIFIERS TLV is shown in 1 following figure:

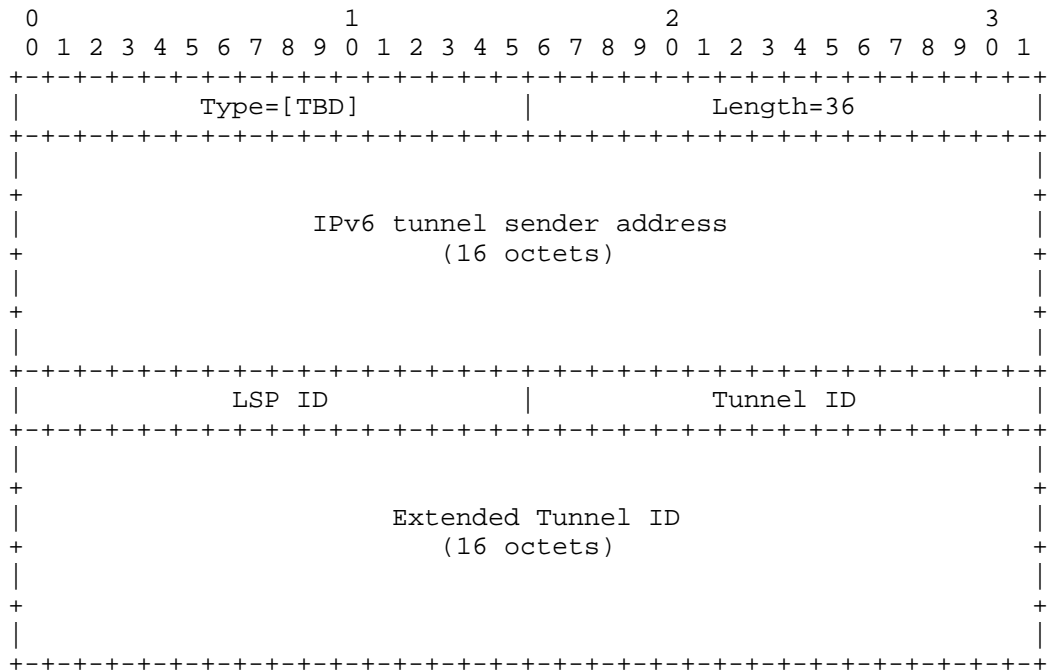


Figure 2: IPV6-LSP-IDENTIFIERS TLV format

The type of the TLV is [TBD] and it has a fixed length of 36 octets. The value contains the following fields:

IPv6 Tunnel Sender Address: contains the sender node's IPv6 address, as defined in [RFC3209], Section 4.6.2.2 for the LSP\_TUNNEL\_IPv6 Sender Template Object.

LSP ID: contains the 16-bit 'LSP ID' identifier defined in [RFC3209], Section 4.6.2.2 for the LSP\_TUNNEL\_IPv6 Sender Template Object.

Tunnel ID: contains the 16-bit 'Tunnel ID' identifier defined in [RFC3209], Section 4.6.1.2 for the LSP\_TUNNEL\_IPv6 Session Object. Tunnel ID remains constant over the life time of a tunnel. However, when Global Path Protection or Global Default Restoration is used, both the primary and secondary LSPs have their own Tunnel IDs. A PCC will report a change in Tunnel ID when traffic switches over from primary LSP to secondary LSP (or vice versa).

Extended Tunnel ID: contains the 128-bit 'Extended Tunnel ID' identifier defined in [RFC3209], Section 4.6.1.2 for the LSP\_TUNNEL\_IPv6 Session Object.

## 5.2. Tunnel ID TLV

The Tunnel ID TLV MAY be included in the LSPA object.

The format of the TUNNEL TLV is shown in the following figure:

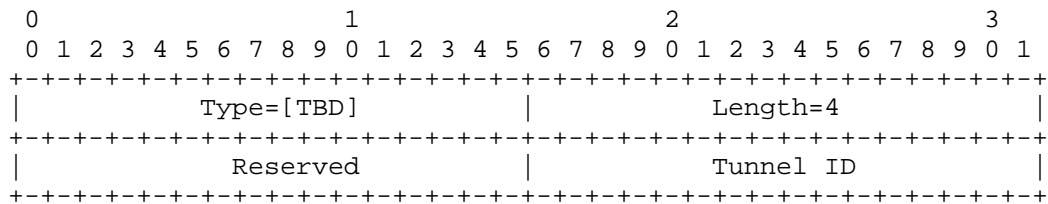


Figure 3: Tunnel-ID TLV format

The type of the TLV is [TBD] and it has a fixed length of 4 octets. The value contains a single field:

Tunnel ID: contains the 16-bit 'Tunnel ID' identifier defined in [RFC3209], Section 4.6.1.1 for the LSP\_TUNNEL\_IPv4 Session Object. Tunnel ID remains constant over the life time of a tunnel. However, when Global Path Protection or Global Default Restoration is used, both the primary and secondary LSPs have their own Tunnel IDs.

## 5.3. LSP Update Error Code TLV

If an LSP Update Request failed, an LSP State Report MUST be sent to all connected stateful PCEs. LSP State Report MUST contain the LSP Update Error Code TLV, indicating the cause of the failure.

The format of the LSP-UPDATE-ERROR-CODE TLV is shown in the following figure:

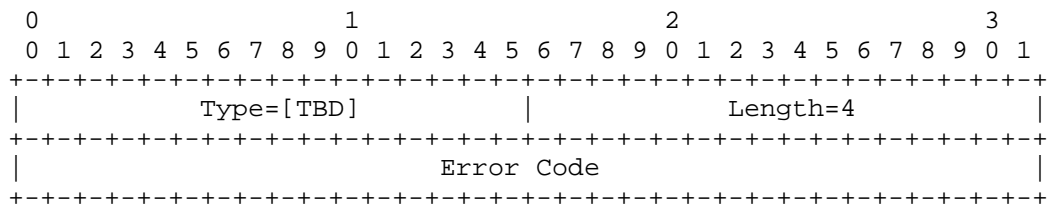


Figure 4: LSP-UPDATE-ERROR-CODE TLV format

The type of the TLV is [TBD] and it has a fixed length of 4 octets. The value contains the error code that indicates the cause of the LSP setup failure. Error codes will be defined in a later revision of this document.

## 6. IANA Considerations

This document requests IANA actions to allocate code points for the protocol elements defined in this document. Values shown here are suggested for use by IANA.

### 6.1. PCEP Objects

This document defines the following new PCEP Object-classes and Object-values:

Object-Class Value	Name	Reference
32	LSP Object-Type 1	This document

### 6.2. PCEP-Error Object

This document defines new Error-Type and Error-Value for the following new error conditions:

Error-Type	Meaning
6	Mandatory Object missing
Error-value=9:	ERO Object missing for a path in an LSP Update Request where TE-LSP setup is requested
Error-value=10:	BANDWIDTH Object missing for a path in an LSP Update Request where TE-LSP setup is requested
Error-value=11:	LSPA Object missing for a path in an LSP Update Request where TE-LSP setup is requested

### 6.3. PCEP TLV Type Indicators

This document defines the following new PCEP TLVs:



Value	Meaning	Reference
18	IPV4-LSP-IDENTIFIERS	This document
19	IPV6-LSP-IDENTIFIERS	This document
20	LSP-UPDATE-ERROR-CODE	This document
24	TUNNEL-ID	This document

## 7. Security Considerations

The security considerations listed in [I-D.ietf-pce-stateful-pce] apply to this document as well.

## 8. Acknowledgements

We would like to thank Adrian Farrel, Cyril Margaria and Ramon Casellas for their contributions to this document.

We would like to thank Shane Amante, Julien Meuric, Kohei Shiimoto, Paul Schultz and Raveendra Torvi for their comments and suggestions. Thanks also to Dhruv Dhoddy, Oscar Gonzales de Dios, Tomas Janciga, Stefan Kobza and Kexin Tang for helpful discussions.

## 9. References

### 9.1. Normative References

- [I-D.ietf-pce-stateful-pce]  
Crabbe, E., Medved, J., Minei, I., and R. Varga, "PCEP Extensions for Stateful PCE",  
draft-ietf-pce-stateful-pce-02 (work in progress),  
October 2012.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC2205] Braden, B., Zhang, L., Berson, S., Herzog, S., and S. Jamin, "Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification", RFC 2205, September 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.
- [RFC4090] Pan, P., Swallow, G., and A. Atlas, "Fast Reroute Extensions to RSVP-TE for LSP Tunnels", RFC 4090, May 2005.

- [RFC5088] Le Roux, JL., Vasseur, JP., Ikejiri, Y., and R. Zhang, "OSPF Protocol Extensions for Path Computation Element (PCE) Discovery", RFC 5088, January 2008.
- [RFC5089] Le Roux, JL., Vasseur, JP., Ikejiri, Y., and R. Zhang, "IS-IS Protocol Extensions for Path Computation Element (PCE) Discovery", RFC 5089, January 2008.
- [RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, May 2008.
- [RFC5440] Vasseur, JP. and JL. Le Roux, "Path Computation Element (PCE) Communication Protocol (PCEP)", RFC 5440, March 2009.
- [RFC5511] Farrel, A., "Routing Backus-Naur Form (RBNF): A Syntax Used to Form Encoding Rules in Various Routing Protocol Specifications", RFC 5511, April 2009.

## 9.2. Informative References

- [RFC2702] Awduche, D., Malcolm, J., Agogbua, J., O'Dell, M., and J. McManus, "Requirements for Traffic Engineering Over MPLS", RFC 2702, September 1999.
- [RFC3031] Rosen, E., Viswanathan, A., and R. Callon, "Multiprotocol Label Switching Architecture", RFC 3031, January 2001.
- [RFC3346] Boyle, J., Gill, V., Hannan, A., Cooper, D., Awduche, D., Christian, B., and W. Lai, "Applicability Statement for Traffic Engineering with MPLS", RFC 3346, August 2002.
- [RFC3630] Katz, D., Kompella, K., and D. Yeung, "Traffic Engineering (TE) Extensions to OSPF Version 2", RFC 3630, September 2003.
- [RFC4655] Farrel, A., Vasseur, J., and J. Ash, "A Path Computation Element (PCE)-Based Architecture", RFC 4655, August 2006.
- [RFC4657] Ash, J. and J. Le Roux, "Path Computation Element (PCE) Communication Protocol Generic Requirements", RFC 4657, September 2006.
- [RFC5305] Li, T. and H. Smit, "IS-IS Extensions for Traffic Engineering", RFC 5305, October 2008.
- [RFC5394] Bryskin, I., Papadimitriou, D., Berger, L., and J. Ash,

"Policy-Enabled Path Computation Framework", RFC 5394,  
December 2008.

[RFC5557] Lee, Y., Le Roux, J.L., King, D., and E. Oki, "Path  
Computation Element Communication Protocol (PCEP)  
Requirements and Protocol Extensions in Support of Global  
Concurrent Optimization", RFC 5557, July 2009.

#### Authors' Addresses

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

Email: [edc@google.com](mailto:edc@google.com)

Jan Medved  
Cisco Systems, Inc.  
170 West Tasman Dr.  
San Jose, CA 95134  
US

Email: [jmedved@cisco.com](mailto:jmedved@cisco.com)

Ina Minei  
Juniper Networks, Inc.  
1194 N. Mathilda Ave.  
Sunnyvale, CA 94089  
US

Email: [ina@juniper.net](mailto:ina@juniper.net)

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

Email: [robert.varga@pantheon.sk](mailto:robert.varga@pantheon.sk)



Network Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: November 9, 2013

E. Crabbe  
Google, Inc.  
J. Medved  
Cisco Systems, Inc.  
I. Minei  
Juniper Networks, Inc.  
R. Varga  
Pantheon Technologies SRO  
May 8, 2013

Stateful PCE extensions for MPLS-TE LSPs  
draft-crabbe-pce-stateful-pce-mpls-te-01

## Abstract

The Path Computation Element Communication Protocol (PCEP) provides mechanisms for Path Computation Elements (PCEs) to perform path computations in response to Path Computation Clients (PCCs) requests.

[I-D.ietf-pce-stateful-pce] describes a set of extensions to PCEP to provide stateful control. This document describes the objects and TLVs to be used with these PCEP extensions to control Multiprotocol Label Switching (MPLS) Traffic Engineering Label Switched Paths (TE LSP) via a stateful PCE.

## 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 November 9, 2013.

## Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

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

## Table of Contents

1. Introduction . . . . .	3
2. Terminology . . . . .	3
3. MPLS-TE specific descriptors used in PCEP Messages . . . . .	3
3.1. MPLS-TE specific descriptors for the PCRpt Message . . . . .	3
3.2. MPLS-TE specific descriptors for the PCUpd Message . . . . .	4
3.3. MPLS-TE specific encoding for the PCReq Message for stateful PCE . . . . .	6
3.4. MPLS-TE specific encoding for the PCRep Message for stateful PCE . . . . .	7
4. IANA Considerations . . . . .	8
4.1. PCEP-Error Object . . . . .	8
5. Security Considerations . . . . .	9
6. Acknowledgements . . . . .	9
7. References . . . . .	9
7.1. Normative References . . . . .	9
7.2. Informative References . . . . .	10
Authors' Addresses . . . . .	11

## 1. Introduction

The Path Computation Element Communication Protocol (PCEP) provides mechanisms for Path Computation Elements (PCEs) to perform path computations in response to Path Computation Clients (PCCs) requests.

[I-D.ietf-pce-stateful-pce] describes a set of extensions to PCEP to provide stateful control. This document describes the objects and TLVs to be used with these PCEP extensions to control Multiprotocol Label Switching (MPLS) Traffic Engineering Label Switched Paths (TE LSP) via a stateful PCE.

## 2. Terminology

This document uses the following terms defined in [RFC5440]: PCC, PCE, PCEP Peer.

This document uses the following terms defined in [I-D.ietf-pce-stateful-pce] : Passive Stateful PCE, Active Stateful PCE, Delegation, Delegation Timeout Interval, LSP State Report, LSP Update Request, LSP Priority, LSP State Database, Revocation.

Within this document, when describing PCE-PCE communications, the requesting PCE fills the role of a PCC. This provides a saving in documentation without loss of function.

The message formats in this document are specified using Routing Backus-Naur Format (RBNF) encoding as specified in [RFC5511].

## 3. MPLS-TE specific descriptors used in 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. [I-D.ietf-pce-stateful-pce] describes the messages and objects needed in support of stateful PCE. The following sections contain MPLS-TE specific descriptors used in some of these messages.

### 3.1. MPLS-TE specific descriptors for the PCRpt Message

The format of the PCRpt message is defined in [I-D.ietf-pce-stateful-pce] as follows, and included here for easy reference:

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

Where:

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

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

Where:

```
<path-list> ::= <path>[<path-list>]
```

For MPLS-TE LSPs, the path descriptor is defined as follows:

```
<path> ::= <ERO><attribute-list>
```

Where:

```
<attribute-list> ::= [<LSPA>]
                    [<BANDWIDTH>]
                    [<RRO>]
                    [<metric-list>]
```

```
<metric-list> ::= <METRIC>[<metric-list>]
```

The LSP State Report MAY contain a path descriptor for the primary path and one or more path descriptors for backup paths. A path descriptor MUST contain an ERO object as it was specified by a PCE or an operator. A path descriptor MUST contain the RRO object if a primary or secondary LSP is set up along the path in the network. A path descriptor MAY contain the LSPA, BANDWIDTH, and METRIC objects. The ERO, LSPA, BANDWIDTH, METRIC, and RRO objects are defined in [RFC5440].

### 3.2. MPLS-TE specific descriptors for the PCUpd Message

A Path Computation LSP Update Request message (also referred to as PCUpd message) is a PCEP message sent by a PCE to a PCC to update attributes of an LSP. A PCUpd message can carry more than one LSP Update Request. The Message-Type field of the PCEP common header for the PCUpd message is set to [TBD].

The format of the PCUpd message is defined in [I-D.ietf-pce-stateful-pce] and included here for easy reference:



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

Where:

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

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

Where:

```
<path-list> ::= <path>[<path-list>]
```

For MPLS-TE LSPs, the encoding of path descriptor is defined as follows:

```
<path> ::= <ERO><attribute-list>
```

Where:

```
<path> ::= <ERO><attribute-list>
```

Where:

```
<attribute-list> ::= [<LSPA>]
                     [<BANDWIDTH>]
                     [<metric-list>]
```

```
<metric-list> ::= <METRIC>[<metric-list>]
```

There is one mandatory object that MUST be included within each LSP Update Request in the PCUpd message: the LSP object (see [I-D.ietf-pce-stateful-pce]). If the LSP object is missing, the receiving PCE MUST send a PCErr message with Error-type=6 (Mandatory Object missing) and Error-value=[TBD] (LSP object missing).

The LSP Update Request MUST contain a path descriptor for the primary path, and MAY contain one or more path descriptors for backup paths. A path descriptor MUST contain an ERO object. A path descriptor MAY further contain the BANDWIDTH, IRO, and METRIC objects. The ERO, LSPA, BANDWIDTH, METRIC, and IRO objects are defined in [RFC5440].

Each LSP Update Request results in a separate LSP setup operation at a PCC. An LSP Update Request MUST contain all LSP parameters that a PCC wishes to set for the LSP. A PCC MAY set missing parameters from locally configured defaults. If the LSP specified the Update Request is already up, it will be re-signaled. The PCC will use make-before-break whenever possible in the re-signaling operation.

A PCC MUST respond with an LSP State Report to each LSP Update Request to indicate the resulting state of the LSP in the network. A PCC MAY respond with multiple LSP State Reports to report LSP setup progress of a single LSP.

If the rate of PCUpd messages sent to a PCC for the same target LSP exceeds the rate at which the PCC can signal LSPs into the network, the PCC MAY perform state compression and only re-signal the last modification in its queue.

Note that a PCC MUST process all LSP Update Requests - for example, an LSP Update Request is sent when a PCE returns delegation or puts an LSP into non-operational state. The protocol relies on TCP for message-level flow control.

Note also that it's up to the PCE to handle inter-LSP dependencies; for example, if ordering of LSP set-ups is required, the PCE has to wait for an LSP State Report for a previous LSP before triggering the LSP setup of a next LSP.

### 3.3. MPLS-TE specific encoding for the PCReq Message for stateful PCE

A PCC MAY include the LSP object defined in [I-D.ietf-pce-stateful-pce] in the PCReq message if the stateful PCE capability has been negotiated on a PCEP session between the PCC and a PCE. The definition of the PCReq message (see [RFC5440], Section 6.4) is then extended as follows:

```
<PCReq Message>::= <Common Header>
                    [<svec-list>]
                    <request-list>
```

Where:

```
<svec-list>::=<SVEC>[<svec-list>]
<request-list>::=<request>[<request-list>]

<request>::= <RP>
              <END-POINTS>
              [<LSP>]                <--- New Object
              [<LSPA>]
              [<BANDWIDTH>]
              [<metric-list>]
              [<RRO>[<BANDWIDTH>]]
              [<IRO>]
              [<LOAD-BALANCING>]
```

Where:

```
<metric-list>::=<METRIC>[<metric-list>]
```

#### 3.4. MPLS-TE specific encoding for the PCRep Message for stateful PCE

A PCE MAY include the LSP object defined in [I-D.ietf-pce-stateful-pce] in the PCRep message if the stateful PCE capability has been negotiated on a PCEP session between the PCC and the PCE and the LSP object was included in the corresponding PCReq message from the PCC. The definition of the PCRep message (see [RFC5440], Section 6.5) is then extended as follows

```

<PCRep Message> ::= <Common Header>
                    <response-list>

```

Where:

```

<response-list> ::= <response> [<response-list>]

<response> ::= <RP>
               [<LSP>]                <--- New Object
               [<NO-PATH>]
               [<attribute-list>]
               [<path-list>]

<path-list> ::= <path> [<path-list>]

<path> ::= <ERO> <attribute-list>

```

Where:

```

<attribute-list> ::= [<LSPA>]
                    [<BANDWIDTH>]
                    [<metric-list>]
                    [<IRO>]

<metric-list> ::= <METRIC> [<metric-list>]

```

#### 4. IANA Considerations

This document requests IANA actions to allocate code points for the protocol elements defined in this document. Values shown here are suggested for use by IANA.

##### 4.1. PCEP-Error Object

This document defines new Error-Type and Error-Value for the following new error conditions:

Error-Type	Meaning
6	Mandatory Object missing
Error-value=9:	ERO Object missing for a path in an LSP Update Request where TE-LSP setup is requested
Error-value=10:	BANDWIDTH Object missing for a path in an LSP Update Request where TE-LSP setup is requested

Error-value=11: LSPA Object missing for a path in an LSP Update Request where TE-LSP setup is requested

## 5. Security Considerations

The security considerations listed in [I-D.ietf-pce-stateful-pce] apply to this document as well.

## 6. Acknowledgements

We would like to thank Adrian Farrel, Cyril Margaria and Ramon Casellas for their contributions to this document.

We would like to thank Shane Amante, Julien Meuric, Kohei Shiimoto, Paul Schultz and Raveendra Torvi for their comments and suggestions. Thanks also to Dhruv Dhoddy, Oscar Gonzales de Dios, Tomas Janciga, Stefan Kobza and Kexin Tang for helpful discussions.

## 7. References

### 7.1. Normative References

- [I-D.ietf-pce-stateful-pce]  
Crabbe, E., Medved, J., Minei, I., and R. Varga, "PCEP Extensions for Stateful PCE", draft-ietf-pce-stateful-pce-03 (work in progress), March 2013.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC2205] Braden, B., Zhang, L., Berson, S., Herzog, S., and S. Jamin, "Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification", RFC 2205, September 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.
- [RFC4090] Pan, P., Swallow, G., and A. Atlas, "Fast Reroute Extensions to RSVP-TE for LSP Tunnels", RFC 4090, May 2005.
- [RFC5088] Le Roux, JL., Vasseur, JP., Ikejiri, Y., and R. Zhang,

"OSPF Protocol Extensions for Path Computation Element (PCE) Discovery", RFC 5088, January 2008.

- [RFC5089] Le Roux, J.L., Vasseur, J.P., Ikejiri, Y., and R. Zhang, "IS-IS Protocol Extensions for Path Computation Element (PCE) Discovery", RFC 5089, January 2008.
- [RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, May 2008.
- [RFC5440] Vasseur, J.P. and J.L. Le Roux, "Path Computation Element (PCE) Communication Protocol (PCEP)", RFC 5440, March 2009.
- [RFC5511] Farrel, A., "Routing Backus-Naur Form (RBNF): A Syntax Used to Form Encoding Rules in Various Routing Protocol Specifications", RFC 5511, April 2009.

## 7.2. Informative References

- [RFC2702] Awduche, D., Malcolm, J., Agogbua, J., O'Dell, M., and J. McManus, "Requirements for Traffic Engineering Over MPLS", RFC 2702, September 1999.
- [RFC3031] Rosen, E., Viswanathan, A., and R. Callon, "Multiprotocol Label Switching Architecture", RFC 3031, January 2001.
- [RFC3346] Boyle, J., Gill, V., Hannan, A., Cooper, D., Awduche, D., Christian, B., and W. Lai, "Applicability Statement for Traffic Engineering with MPLS", RFC 3346, August 2002.
- [RFC3630] Katz, D., Kompella, K., and D. Yeung, "Traffic Engineering (TE) Extensions to OSPF Version 2", RFC 3630, September 2003.
- [RFC4655] Farrel, A., Vasseur, J., and J. Ash, "A Path Computation Element (PCE)-Based Architecture", RFC 4655, August 2006.
- [RFC4657] Ash, J. and J. Le Roux, "Path Computation Element (PCE) Communication Protocol Generic Requirements", RFC 4657, September 2006.
- [RFC5305] Li, T. and H. Smit, "IS-IS Extensions for Traffic Engineering", RFC 5305, October 2008.
- [RFC5394] Bryskin, I., Papadimitriou, D., Berger, L., and J. Ash, "Policy-Enabled Path Computation Framework", RFC 5394,

December 2008.

[RFC5557] Lee, Y., Le Roux, JL., King, D., and E. Oki, "Path Computation Element Communication Protocol (PCEP) Requirements and Protocol Extensions in Support of Global Concurrent Optimization", RFC 5557, July 2009.

#### Authors' Addresses

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

Email: [edc@google.com](mailto:edc@google.com)

Jan Medved  
Cisco Systems, Inc.  
170 West Tasman Dr.  
San Jose, CA 95134  
US

Email: [jmedved@cisco.com](mailto:jmedved@cisco.com)

Ina Minei  
Juniper Networks, Inc.  
1194 N. Mathilda Ave.  
Sunnyvale, CA 94089  
US

Email: [ina@juniper.net](mailto:ina@juniper.net)

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

Email: [robert.varga@pantheon.sk](mailto:robert.varga@pantheon.sk)





Network Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: April 15, 2013

E. Crabbe  
Google, Inc.  
J. Medved  
Cisco Systems, Inc.  
I. Minei  
R. Torvi  
Juniper Networks, Inc.  
October 12, 2012

PCEP Extensions for MPLS-TE LSP protection with stateful PCE  
draft-crabbe-pce-stateful-pce-protection-00

## Abstract

Stateful PCE [I-D.ietf-pce-stateful-pce] can apply global concurrent optimizations to optimize LSP placement. In a deployment where a PCE is used to compute all the paths, it may be beneficial for the protection paths to also be computed by the PCE. This document defines extensions needed for the setup and management of MPLS-TE protection paths by the PCE.

## 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 15, 2013.

## Copyright Notice

Copyright (c) 2012 IETF Trust and the persons identified as the

document authors. All rights reserved.

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

## Table of Contents

1. Introduction . . . . .	3
2. Terminology . . . . .	3
3. Architectural Overview . . . . .	3
3.1. Path Protection Overview . . . . .	3
3.2. Local Protection Overview . . . . .	4
4. Extensions for the LSPA object . . . . .	5
4.1. The Standby flag in the LSPA object . . . . .	5
4.2. The Weight TLV . . . . .	6
4.3. The Bypass TLV . . . . .	6
4.4. The LOCALLY-PROTECTED-LSPS TLV . . . . .	7
5. IANA considerations . . . . .	9
5.1. PCEP-Error Object . . . . .	9
5.2. PCEP TLV Type Indicators . . . . .	9
6. Security Considerations . . . . .	9
7. Acknowledgements . . . . .	9
8. References . . . . .	10
8.1. Normative References . . . . .	10
8.2. Informative References . . . . .	11
Authors' Addresses . . . . .	11

## 1. Introduction

[RFC5440] describes the Path Computation Element Protocol PCEP. PCEP defines the communication between a Path Computation Client (PCC) and a Path Control Element (PCE), or between PCE and PCE, enabling computation of Multiprotocol Label Switching (MPLS) for Traffic Engineering Label Switched Path (TE LSP) characteristics.

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

Stateful PCE can apply global concurrent optimizations to optimize LSP placement. In a deployment where a PCE is used to compute all the paths, it may be beneficial for the protection paths to also be controlled through the PCE. This document defines extensions needed for the setup and management of protection paths by the PCE.

Benefits of controlling the protection paths include: better control over traffic after a failure and more deterministic path computation (paths not affected by overload after a failure).

## 2. Terminology

This document uses the following terms defined in [RFC5440]: PCC, PCE, PCEP Peer.

This document uses the following terms defined in [I-D.ietf-pce-stateful-pce]: Stateful PCE, Delegation, Delegation Timeout Interval, LSP State Report, LSP Update Request.

The message formats in this document are specified using Routing Backus-Naur Format (RBNF) encoding as specified in [RFC5511].

## 3. Architectural Overview

### 3.1. Path Protection Overview

Path protection refers to switching to a new path on failure. Several cases exist:

- (1) MPLS-TE Global Default Restoration - protection paths are computed dynamically by the LSR after the failure. This can be supported without any PCEP protocol changes by specifying a secondary path with an ERO of just the end points of the LSP. Once reestablished, the path is communicated to the PCE via the LSP State Report message.
- (2) MPLS-TE Global Path Protection - protection paths are fully specified ahead of the failure. The base Stateful PCE specification [I-D.ietf-pce-stateful-pce] supports sending multiple fully-specified paths in the PCUpd requests. There are 2 further sub-cases:
  - (a) Protection paths are pre-signaled ahead of the failure (standby paths).
  - (b) Protection paths are set up after the failure.

The protection path setup regimen (standby or not) is specified in the path using a new per-path flag in the LSPA object, the S (standby) flag (see section Section 4.1). Paths for which the S flag is set MUST have a name associated with them, specified using the SYMBOLIC-PATH-NAME TLV in the LSPA object.

Because multiple secondary standby paths are possible, there is also a need for the PCE to be able to specify the relative priorities between the paths (which one to take if there are 3 available). This is done through a weight assigned to each path. See details in Section 4.2.

Reversion from protection paths to the primary path when possible will be controlled by the PCE, by sending a new LSP Update Request. If the primary can be successfully signaled and the secondary does not have the S flag set, then the secondary MUST be torn down. Thus, there is no need to signal the desire for revertive behavior.

### 3.2. Local Protection Overview

Local protection refers to the ability to locally route around failure of an LSP. Two types of local protection are possible:

- (1) 1:1 protection - the protection path protects a single LSP.
- (2) 1:N protection - the protection path protects multiple LSPs traversing the protected resource.

It is assumed that the PCE knows what resources require protection through mechanisms outside the scope of this document. In a PCE-

controlled deployment, support of 1:1 protection has limited applicability, and can be achieved as a degenerate case of 1:N protection. For this reason, local protection will be discussed only for the 1:N case.

Local protection requires the setup of a bypass at the PLR. This bypass can be locally initiated and delegated, or PCE-initiated. In either case, the PLR must maintain a PCEP session to the PCE. A bypass identifier (the name of the bypass) is required for disambiguation as multiple bypasses are possible at the PLR. Mapping of LSPs to bypass is done through a new TLV, the LOCALLY-PROTECTED-LSPS TLV in the LSP Update message from PCE to PLR. See section Section 4.4. When an LSP requiring protection is set up through the PLR, the PLR checks if it has a mapping to a bypass and only provides protection if such a mapping exists. The status of bypasses and what LSPs are protected by them is communicated to the PCE via LSP Status Report messages.

#### 4. Extensions for the LSPA object

##### 4.1. The Standby flag in the LSPA object

The LSPA object is defined in [RFC5440] and replicated below for easy reference. This document defines a new flag, the S flag in the flags field of the LSPA object.

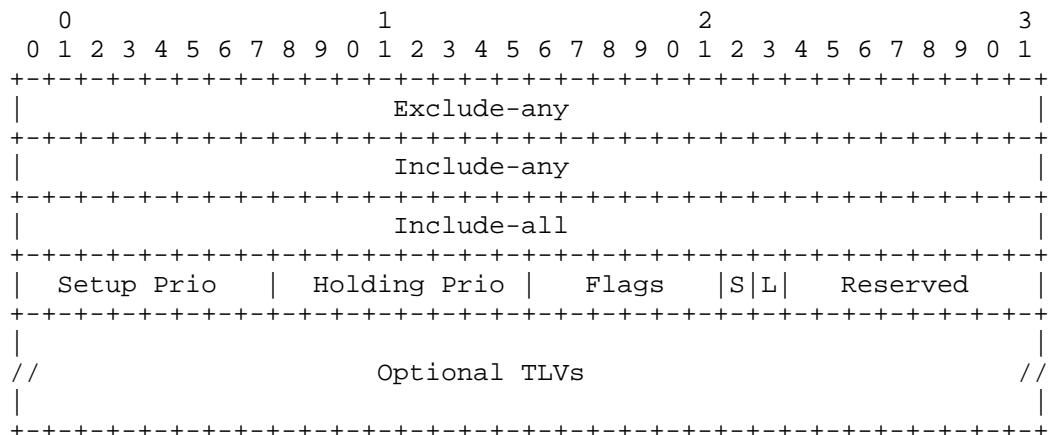


Figure 1: STATEFUL-PCE-CAPABILITY TLV format

The L flag is defined in [RFC5440].

If set to 1, the S Flag indicates this is a standby path.

If the S flag is set, the LSPA object MUST also carry the SYMBOLIC-PATH-NAME TLV as one of the optional TLVs. Failure to include the mandatory SYMBOLIC-PATH-NAME TLV when the S flag is set MUST trigger PCErr of type 6 (Mandatory Object missing) and value TBD (SYMBOLIC-PATH-NAME TLV missing for standby LSP).

#### 4.2. The Weight TLV

This TLV will be discussed in a future version of this document.

#### 4.3. The Bypass TLV

The facility backup method creates a bypass tunnel to protect a potential failure point. The bypass tunnel protects a set of LSPs with similar backup constraints [RFC4090].

A PCC can delegate a bypass tunnel to PCE control or a PCE can provision the bypass tunnel via a PCC. The procedures for bypass instantiation rely on the extensions defined in [I-D.crabbe-pce-pce-initiated-lsp] and will be detailed in a future version of this document.

The Bypass TLV carries information about the bypass tunnel. It is included in the LSPA Object in LSP State Report and LSP Update Request messages.

The format of the Bypass TLV is shown in the following figure:

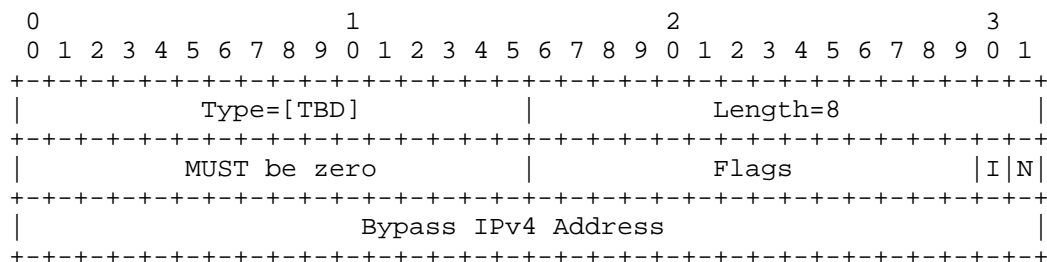


Figure 2: Bypass TLV format

The type of the TLV is [TBD] and it has a fixed length of 8 octets. The value contains the following fields:

## Flags

N (Node Protection - 1 bit): The N Flag indicates whether the Bypass is used for node-protection. If the N flag is set to 1, the Bypass is used for node-protection. If the N flag is 0, the Bypass is used for link-protection.

I (Local Protection In Use - 1 bit): The I Flag indicates that local repair mechanism is in use.

Bypass IPv4 address: For link protection, the Bypass IPv4 Address is the nexthop address of the protected link in the paths of the protected LSPs. For node protection, the Bypass IPv4 Address is the node addresses of the protected node.

If the Bypass TLV is included, then the LSPA object MUST also carry the SYMBOLIC-PATH-NAME TLV as one of the optional TLVs. Failure to include the mandatory SYMBOLIC-PATH-NAME TLV MUST trigger PCErr of type 6 (Mandatory Object missing) and value TBD (SYMBOLIC-PATH-NAME TLV missing for bypass LSP)

### 4.4. The LOCALLY-PROTECTED-LSPS TLV

The LOCALLY-PROTECTED-LSPS TLV in the LSPA Object contains a list of LSPs protected by the bypass tunnel.

The format of the Bypass TLV is shown in the following figure:

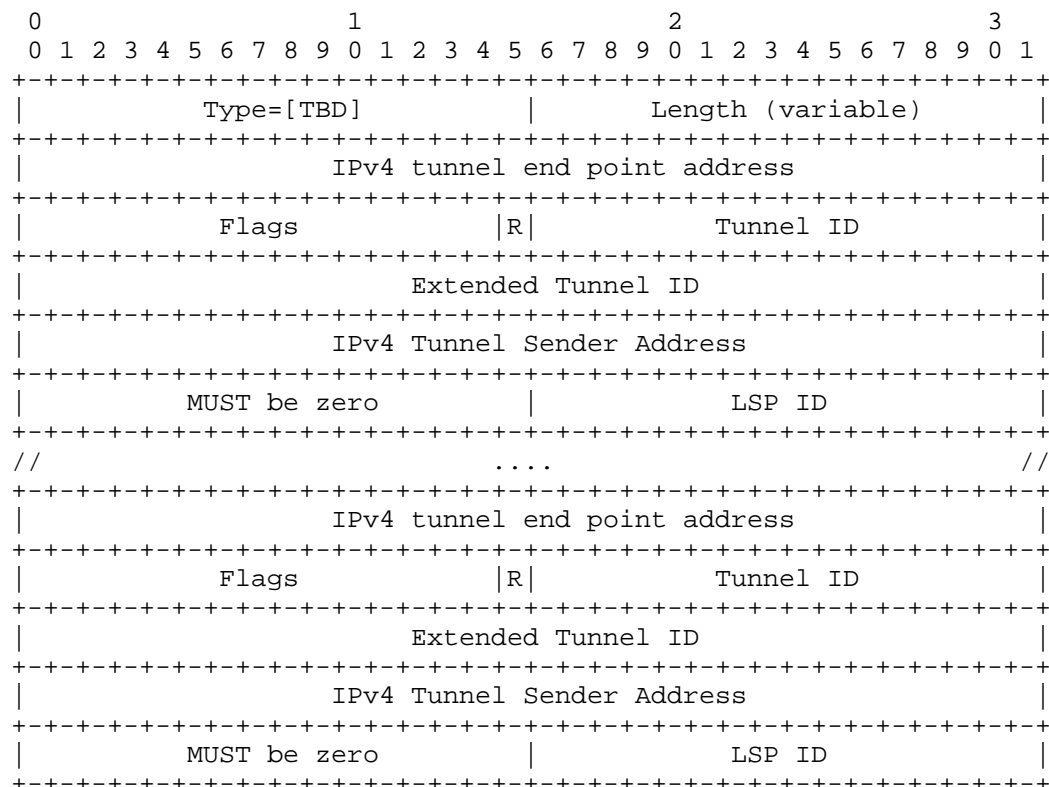


Figure 3: Locally protected LSPs TLV format

The type of the TLV is [TBD] and it is of variable length. The value contains one or more LSP descriptors including the following fields filled per [RFC3209].

IPv4 Tunnel end point address: [RFC3209]

Flags

R(Remove - 1 bit): The R Flag indicates that the LSP has been removed from the list of LSPs protected by the bypass tunnel.

Tunnel ID: [RFC3209]



Extended Tunnel ID: [RFC3209]

IPv4 Tunnel Sender address: [RFC3209]

LSP ID: [RFC3209]

## 5. IANA considerations

### 5.1. PCEP-Error Object

This document defines new Error-Type and Error-Value for the following new error conditions:

Error-Type	Meaning
6	Mandatory Object missing
Error-value=TBD:	SYMBOLIC-PATH-NAME TLV missing for a path where the S-bit is set in the LSPA object.
Error-value=TBD:	SYMBOLIC-PATH-NAME TLV missing for a bypass path.

### 5.2. PCEP TLV Type Indicators

This document defines the following new PCEP TLVs:

Value	Meaning	Reference
???	Bypass	This document
???	weight	This document
???	LOCALLY-PROTECTED-LSPS	This document

## 6. Security Considerations

The same security considerations apply at the PLR as those describe for the head end in [I-D.crabbe-pce-pce-initiated-lsp].

## 7. Acknowledgements

We would like to thank Ambrose Kwong for his contributions to this document.

## 8. References

## 8.1. Normative References

- [I-D.crabbe-pce-pce-initiated-lsp]  
Crabbe, E., Minei, I., Sivabalan, S., and R. Varga, "PCEP Extensions for PCE-initiated LSP Setup in a Stateful PCE Model", draft-crabbe-pce-pce-initiated-lsp-00 (work in progress), October 2012.
- [I-D.ietf-pce-stateful-pce]  
Crabbe, E., Medved, J., Varga, R., and I. Minei, "PCEP Extensions for Stateful PCE", draft-ietf-pce-stateful-pce-01 (work in progress), July 2012.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC2205] Braden, B., Zhang, L., Berson, S., Herzog, S., and S. Jamin, "Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification", RFC 2205, September 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.
- [RFC4090] Pan, P., Swallow, G., and A. Atlas, "Fast Reroute Extensions to RSVP-TE for LSP Tunnels", RFC 4090, May 2005.
- [RFC5088] Le Roux, JL., Vasseur, JP., Ikejiri, Y., and R. Zhang, "OSPF Protocol Extensions for Path Computation Element (PCE) Discovery", RFC 5088, January 2008.
- [RFC5089] Le Roux, JL., Vasseur, JP., Ikejiri, Y., and R. Zhang, "IS-IS Protocol Extensions for Path Computation Element (PCE) Discovery", RFC 5089, January 2008.
- [RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, May 2008.
- [RFC5440] Vasseur, JP. and JL. Le Roux, "Path Computation Element (PCE) Communication Protocol (PCEP)", RFC 5440, March 2009.
- [RFC5511] Farrel, A., "Routing Backus-Naur Form (RBNF): A Syntax Used to Form Encoding Rules in Various Routing Protocol Specifications", RFC 5511, April 2009.

## 8.2. Informative References

- [RFC2702] Awduche, D., Malcolm, J., Agogbua, J., O'Dell, M., and J. McManus, "Requirements for Traffic Engineering Over MPLS", RFC 2702, September 1999.
- [RFC3031] Rosen, E., Viswanathan, A., and R. Callon, "Multiprotocol Label Switching Architecture", RFC 3031, January 2001.
- [RFC3346] Boyle, J., Gill, V., Hannan, A., Cooper, D., Awduche, D., Christian, B., and W. Lai, "Applicability Statement for Traffic Engineering with MPLS", RFC 3346, August 2002.
- [RFC3630] Katz, D., Kompella, K., and D. Yeung, "Traffic Engineering (TE) Extensions to OSPF Version 2", RFC 3630, September 2003.
- [RFC4655] Farrel, A., Vasseur, J., and J. Ash, "A Path Computation Element (PCE)-Based Architecture", RFC 4655, August 2006.
- [RFC4657] Ash, J. and J. Le Roux, "Path Computation Element (PCE) Communication Protocol Generic Requirements", RFC 4657, September 2006.
- [RFC5305] Li, T. and H. Smit, "IS-IS Extensions for Traffic Engineering", RFC 5305, October 2008.
- [RFC5394] Bryskin, I., Papadimitriou, D., Berger, L., and J. Ash, "Policy-Enabled Path Computation Framework", RFC 5394, December 2008.
- [RFC5557] Lee, Y., Le Roux, J.L., King, D., and E. Oki, "Path Computation Element Communication Protocol (PCEP) Requirements and Protocol Extensions in Support of Global Concurrent Optimization", RFC 5557, July 2009.

## Authors' Addresses

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

Email: edc@google.com

Jan Medved  
Cisco Systems, Inc.  
170 West Tasman Dr.  
San Jose, CA 95134  
US

Email: [jmedved@cisco.com](mailto:jmedved@cisco.com)

Ina Minei  
Juniper Networks, Inc.  
1194 N. Mathilda Ave.  
Sunnyvale, CA 94089  
US

Email: [ina@juniper.net](mailto:ina@juniper.net)

Raveendra Torvi  
Juniper Networks, Inc.  
1194 N. Mathilda Ave.  
Sunnyvale, CA 94089  
US

Email: [rtorvi@juniper.net](mailto:rtorvi@juniper.net)



PCE Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: August 24, 2013

D. Dhody  
Huawei Technologies India Pvt  
Ltd  
V. Manral  
Hewlett-Packard Corp.  
Z. Ali  
G. Swallow  
Cisco Systems  
K. Kumaki  
KDDI Corporation  
February 25, 2013

Extensions to the Path Computation Element Communication Protocol (PCEP)  
to compute service aware Label Switched Path (LSP).  
draft-dhody-pce-pcep-service-aware-05

## Abstract

In certain networks like financial information network (stock/commodity trading) and enterprises using cloud based applications, Latency (delay), Latency-Variation (jitter) and Packet loss is becoming a key requirement for path computation along with other constraints and metrics. Latency, Latency-Variation and Packet Loss is associated with the Service Level Agreement (SLA) between customers and service providers.

[MPLS-DELAY-FWK] describes MPLS architecture to allow Latency (delay), Latency-Variation (jitter) and Packet loss as properties. [OSPF-TE-EXPRESS] and [ISIS-TE-EXPRESS] describes mechanisms with which network performance information is distributed via OSPF and ISIS respectively. This document describes the extension to PCEP to carry Latency, Latency-Variation and Loss as constraints for end to end path computation.

## 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 June 4, 2013.

#### Copyright Notice

Copyright (c) 2012 IETF Trust and the persons identified as the document authors. All rights reserved.

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

## Table of Contents

1. Introduction . . . . .	4
1.1. Requirements Language . . . . .	4
2. Terminology . . . . .	4
3. PCEP Requirements . . . . .	5
4. PCEP extensions . . . . .	5
4.1. Latency (Delay) Metric . . . . .	6
4.1.1. Latency (Delay) Metric Value . . . . .	6
4.2. Latency Variation (Jitter) Metric . . . . .	7
4.2.1. Latency Variation (Jitter) Metric Value . . . . .	7
4.3. Packet Loss Metric . . . . .	8
4.3.1. Packet Loss Metric Value . . . . .	9
4.4. Non-Understanding / Non-Support of Service Aware Path Computation . . . . .	9
4.5. Mode of Operation . . . . .	9
4.5.1. Examples . . . . .	10
5. Relationship with Objective function . . . . .	11
6. Protocol Consideration . . . . .	11
6.1. Inter domain Consideration . . . . .	11
6.1.1. Inter-AS Link . . . . .	12
6.1.2. Inter-Layer Consideration . . . . .	12
6.2. Reoptimization Consideration . . . . .	12
6.3. Point-to-Multipoint (P2MP) . . . . .	12
6.3.1. P2MP Latency Metric . . . . .	12
6.3.2. P2MP Latency Variation Metric . . . . .	13
7. IANA Considerations . . . . .	13
8. Security Considerations . . . . .	13
9. Manageability Considerations . . . . .	14
9.1. Control of Function and Policy . . . . .	14
9.2. Information and Data Models . . . . .	14
9.3. Liveness Detection and Monitoring . . . . .	14
9.4. Verify Correct Operations . . . . .	14
9.5. Requirements On Other Protocols . . . . .	14
9.6. Impact On Network Operations . . . . .	14
10. Acknowledgments . . . . .	14
11. References . . . . .	15
11.1. Normative References . . . . .	15
11.2. Informative References . . . . .	15
Appendix A. Contributor Addresses . . . . .	16



## 1. Introduction

Real time Network Performance is becoming a critical in the path computation in some networks. There exist mechanism described in [RFC6374] to measure latency, latency-Variation and packet loss after the LSP has been established, which is inefficient. It is important that latency, latency-variation and packet loss are considered during path selection process, even before the LSP is setup.

TED is populated with network performance information like link latency, latency variation and packet loss through [OSPF-TE-EXPRESS] or [ISIS-TE-EXPRESS]. Path Computation Client (PCC) can request Path Computation Element (PCE) to provide a path meeting end to end network performance criteria. This document extends Path Computation Element Communication Protocol (PCEP) [RFC5440] to handle network performance constraint.

PCE MAY use mechanism described in [MPLS-TE-EXPRESS-PATH] on how to use the link latency, latency variation and packet loss information for end to end path selection.

### 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).

IS-IS: Intermediate System to Intermediate System.

OSPF: Open Shortest Path First.

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. PCEP Requirements

End-to-end service optimization based on latency, latency-variation and packet loss is a key requirement for service provider. Following key requirements associated with latency, latency-variation and loss are identified for PCEP:

1. Path Computation Element (PCE) supporting this draft MUST have the capability to compute end-to-end path with latency, latency-variation and packet loss constraints. It MUST also support the combination of network performance constraint (latency, latency-variation, loss...) with existing constraints (cost, hop-limit...)
2. Path Computation Client (PCC) MUST be able to request for network performance constraint in path request message as the key constraint to be optimized or to suggest boundary condition that should not be crossed.
3. PCEs are not required to support service aware path computation. Therefore, it MUST be possible for a PCE to reject a Path Computation Request message with a reason code that indicates no support for service-aware path computation.
4. PCEP SHOULD provide a means to return end to end network performance information of the computed path in the reply message.
5. PCEP SHOULD provide mechanism to compute multi-domain (e.g., Inter-AS, Inter-Area or Multi-Layer) service aware paths.

It is assumed that such constraints are only meaningful if used consistently: for instance, if the delay of a computed path segment is exchanged between two PCEs residing in different domains, consistent ways of defining the delay must be used.

### 4. PCEP extensions

This section defines PCEP extensions (see [RFC5440]) for requirements outlined in Section 3. The proposed solution is used to support network performance and service aware path computation.

This document defines the following optional types for the METRIC object defined in [RFC5440].

For explanation of these metrics, the following terminology is used

and expanded along the way.

- A network comprises of a set of N links  $\{L_i, (i=1\dots N)\}$ .
- A path P of a P2P LSP is a list of K links  $\{L_{pi}, (i=1\dots K)\}$ .

#### 4.1. Latency (Delay) Metric

Link delay metric is defined in [OSPF-TE-EXPRESS] and [ISIS-TE-EXPRESS]. P2P latency metric type of METRIC object in PCEP encodes the sum of the link delay metric of all links along a P2P Path. Specifically, extending on the above mentioned terminology:

- A Link delay metric of link L is denoted  $D(L)$ .
- A P2P latency metric for the Path  $P = \text{Sum } \{D(L_{pi}), (i=1\dots K)\}$ .

\* T=13(IANA): Latency metric

PCC MAY use this latency metric In PCReq to request a path meeting the end to end latency requirement. In this case B bit MUST be set to suggest a bound (a maximum) for the path latency metric that must not be exceeded for the PCC to consider the computed path as acceptable. The path metric must be less than or equal to the value specified in the metric-value field.

PCC MAY also use this metric to ask PCE to optimize delay during path computation, in this case B flag will be cleared.

PCE MAY use this latency metric In PCRep along with NO-PATH object incase PCE cannot compute a path meeting this constraint. PCE MAY also use this metric to reply the computed end to end latency metric to PCC.

##### 4.1.1. Latency (Delay) Metric Value

[OSPF-TE-EXPRESS] and [ISIS-TE-EXPRESS] defines "Unidirectional Link Delay Sub-TLV" in a 24-bit field. [RFC5440] defines the METRIC object with 32-bit metric value. Consequently, encoding for Latency (Delay) Metric Value is defined as follows:

```

  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                | Latency (Delay) Metric                |
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

```

Reserved (8 bits): Reserved field. This field MUST be set to zero on

transmission and MUST be ignored on receipt.

Latency (Delay) Metric (24 bits): Represents the end to end Latency (delay) quantified in units of microseconds and MUST be encoded as integer value. With the maximum value 16,777,215 representing 16.777215 sec.

#### 4.2. Latency Variation (Jitter) Metric

Link delay variation metric is defined in [OSPF-TE-EXPRESS] and [ISIS-TE-EXPRESS]. P2P latency variation metric type of METRIC object in PCEP encodes a function of the link delay variation metric of all links along a P2P Path. Specifically, extending on the above mentioned terminology:

- A Latency variation of link L is denoted DV(L).
- A P2P latency variation metric for the Path P = function {DV(L<sub>p</sub>i), (i=1...K)}.

Specification of the "Function" used to drive latency variation metric of a path from latency variation metrics of individual links along the path is beyond the scope of this document.

\* T=14(IANA): Latency Variation metric

PCC MAY use this latency variation metric In PCReq to request a path meeting the end to end latency variation requirement. In this case B bit MUST be set to suggest a bound (a maximum) for the path latency variation metric that must not be exceeded for the PCC to consider the computed path as acceptable. The path metric must be less than or equal to the value specified in the metric-value field.

PCC MAY also use this metric to ask PCE to optimize jitter during path computation, in this case B flag will be cleared.

PCE MAY use this latency variation metric In PCRep along with NO-PATH object incase PCE cannot compute a path meeting this constraint. PCE MAY also use this metric to reply the computed end to end latency variation metric to PCC.

##### 4.2.1. Latency Variation (Jitter) Metric Value

[OSPF-TE-EXPRESS] and [ISIS-TE-EXPRESS] defines "Unidirectional Delay Variation Sub-TLV" in a 24-bit field. [RFC5440] defines the METRIC object with 32-bit metric value. Consequently, encoding for Latency Variation (Jitter) Metric Value is defined as follows:

```

  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   |   Latency variation (jitter) Metric   |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

Reserved (8 bits): Reserved field. This field MUST be set to zero on transmission and MUST be ignored on receipt.

Latency variation (jitter) Metric (24 bits): Represents the end to end Latency variation (jitter) quantified in units of microseconds and MUST be encoded as integer value. With the maximum value 16,777,215 representing 16.777215 sec.

#### 4.3. Packet Loss Metric

[OSPF-TE-EXPRESS] and [ISIS-TE-EXPRESS] defines "Unidirectional Link Loss". Packet Loss Metric metric type of METRIC object in PCEP encodes a function of the link's unidirectional loss metric of all links along a P2P Path. Specifically, extending on the above mentioned terminology:

The end to end Packet Loss for the path is represented by this metric.

- A Packet loss of link L is denoted PL(L).

- A P2P packet loss metric for the Path P = function {PL(L<sub>pi</sub>), (i=1...K)}.

Specification of the "Function" used to drive end to end packet loss metric of a path from packet loss metrics of individual links along the path is beyond the scope of this document.

\* T=15(IANA): Packet Loss metric

PCC MAY use this packet loss metric In PCReq to request a path meeting the end to end packet loss requirement. In this case B bit MUST be set to suggest a bound (a maximum) for the path packet loss metric that must not be exceeded for the PCC to consider the computed path as acceptable. The path metric must be less than or equal to the value specified in the metric-value field.

PCC MAY also use this metric to ask PCE to optimize packet loss during path computation, in this case B flag will be cleared.

PCE MAY use this packet loss metric In PCRep along with NO-PATH object incase PCE cannot compute a path meeting this constraint. PCE

MAY also use this metric to reply the computed end to end packet loss metric to PCC.

#### 4.3.1. Packet Loss Metric Value

[OSPF-TE-EXPRESS] and [ISIS-TE-EXPRESS] defines "Unidirectional Link Loss Sub-TLV" in a 24-bit field. [RFC5440] defines the METRIC object with 32-bit metric value. Consequently, encoding for Packet Loss Metric Value is defined as follows:

```

  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      |                               Packet loss Metric |
  +-----+-----+-----+-----+-----+-----+-----+-----+

```

Reserved (8 bits): Reserved field. This field MUST be set to zero on transmission and MUST be ignored on receipt.

Packet loss Metric (24 bits): Represents the end to end packet loss quantified as a percentage of packets lost and MUST be encoded as integer. The basic unit is 0.000003%, with the maximum value 16,777,215 representing 50.331645% ( $16,777,215 * 0.000003\%$ ). This value is the highest packet loss percentage that can be expressed.

#### 4.4. Non-Understanding / Non-Support of Service Aware Path Computation

If the P bit is clear in the object header and PCE does not understand or does not support service aware path computation it SHOULD simply ignore this METRIC.

If the P Bit is set in the object header and PCE receives new METRIC type in path request and it understands the METRIC type, but the PCE is not capable of service aware path computation, the PCE MUST send a PCErr message with a PCEP-ERROR Object Error-Type = 4 (Not supported object) [RFC5440]. The path computation request MUST then be cancelled.

If the PCE does not understand the new METRIC type, then the PCE MUST send a PCErr message with a PCEP-ERROR Object Error-Type = 3 (Unknown object) [RFC5440].

#### 4.5. Mode of Operation

As explained in [RFC5440], The METRIC object is optional and can be used for several purposes. In a PCReq message, a PCC MAY insert one or more METRIC objects:

- o To indicate the metric that MUST be optimized by the path computation algorithm (Latency, Latency-Variation or Loss)
- o To indicate a bound on the path METRIC (Latency, Latency-Variation or Loss) that MUST NOT be exceeded for the path to be considered as acceptable by the PCC.

In a PCRep message, the METRIC object MAY be inserted so as to provide the METRIC (Latency, Latency-Variation or Loss) for the computed path. It MAY also be inserted within a PCRep with the NO-PATH object to indicate that the metric constraint could not be satisfied.

The path computation algorithmic aspects used by the PCE to optimize a path with respect to a specific metric are outside the scope of this document.

All the rules of processing METRIC object as explained in [RFC5440] are applicable to the new metric types as well.

In a PCReq message, a PCC MAY insert more than one METRIC object to be optimized, in such a case PCE should find the path that is optimal when both the metrics are considered together.

#### 4.5.1. Examples

Example 1: If a PCC sends a path computation request to a PCE where two metric to optimize are the latency and the packet loss, two METRIC objects are inserted in the PCReq message:

- o First METRIC object with B=0, T=13 (TBA - IANA), C=1, metric-value=0x0000
- o Second METRIC object with B=0, T=15 (TBA - IANA), C=1, metric-value=0x0000

PCE in such a case should try to optimize both the metrics and find a path with the minimum latency and packet loss, if a path can be found by the PCE and there is no policy that prevents the return of the computed metric, the PCE inserts two METRIC object with B=0, T=13 (TBA - IANA), metric-value= computed end to end latency and second METRIC object with B=1, T=15 (TBA - IANA), metric-value= computed end to end packet loss.

Example 2: If a PCC sends a path computation request to a PCE where the metric to optimize is the latency and the packet loss must not exceed the value of M, two METRIC objects are inserted in the PCReq message:

- o First METRIC object with B=0, T=13 (TBA - IANA), C=1, metric-value=0x0000
- o Second METRIC object with B=1, T=15 (TBA - IANA), metric-value=M

If a path satisfying the set of constraints can be found by the PCE and there is no policy that prevents the return of the computed metric, the PCE inserts one METRIC object with B=0, T=13 (TBA - IANA), metric-value= computed end to end latency. Additionally, the PCE may insert a second METRIC object with B=1, T=15 (TBA - IANA), metric-value= computed end to end packet loss.

## 5. Relationship with Objective function

[RFC5541] defines mechanism to specify an optimization criteria, referred to as objective functions. The new metric types specified in this document can continue to use the existing Objective function.

Minimum Cost Path (MCP) is one such objective function.

- o A network comprises a set of N links  $\{L_i, (i=1\dots N)\}$ .
- o A path P is a list of K links  $\{L_{pi}, (i=1\dots K)\}$ .
- o Metric of link L is denoted M(L). This can be any metric, including the ones defined in this document.
- o The cost of a path P is denoted C(P), where  $C(P) = \sum \{M(L_{pi}), (i=1\dots K)\}$ .

Name: Minimum Cost Path (MCP)

Description: Find a path P such that C(P) is minimized.

The new metric types for example latency (delay) can continue to use the above objective function to find the minimum cost path where cost is latency (delay). At the same time new objective functions can be defined in future to optimize these new metric types.

## 6. Protocol Consideration

There is no change in the message format of Path Request and Reply Message.

### 6.1. Inter domain Consideration

[RFC5441] describes the BRPC procedure to compute end to end optimized inter domain path by cooperating PCEs. The network



performance constraints can be applied end to end in similar manner as IGP or TE cost.

All domains should have the same understanding of the METRIC (Latency-Variation etc) for end-to-end inter-domain path computation to make sense. Otherwise some form of Metric Normalization as described in [RFC5441] MAY need to be applied.

#### 6.1.1. Inter-AS Link

The IGP in each neighbor domain can advertise its inter-domain TE link capabilities, this has been described in [RFC5316] (ISIS) and [RFC5392] (OSPF). The network performance link properties are described in [OSPF-TE-EXPRESS] and [ISIS-TE-EXPRESS], the same properties must be advertised using the mechanism described in [RFC5392] (OSPF) and [RFC5316] (ISIS).

#### 6.1.2. Inter-Layer Consideration

PCEP supporting this draft SHOULD provide mechanism to support different Metric requirements for different Layers. This is important as the network performance metric would be different for Packet and Optical (TDM, LSC etc) Layers. In order to allow different Metric-Value to be applied within different network layers, multiple METRIC objects of the same type MAY be present. In such a case, the first METRIC object specifies a metric for the higher-layer network, and subsequent METRIC objects specify objection functions of the subsequent lower-layer networks.

#### 6.2. Reoptimization Consideration

PCC can monitor the setup LSPs and in case of degradation of network performance constraints, it MAY ask PCE for reoptimization as per [RFC5440].

#### 6.3. Point-to-Multipoint (P2MP)

This document defines the following optional types for the METRIC object defined in [RFC5440] for P2MP TE LSPs. Additional metric types for P2MP TE LSPs are to be added in a future revision

##### 6.3.1. P2MP Latency Metric

P2MP latency metric type of METRIC object in PCEP encodes the path latency metric for destination that observes the worst latency metric among all destination of the P2MP tree. Specifically, extending on the above mentioned terminology:

- A P2MP Tree T comprises of a set of M destinations {Dest\_j, (j=1...M)}
- P2P latency metric of the Path to destination Dest\_j is denoted by LM(Dest\_j).
- P2MP latency metric for the P2MP tree T = Maximum {LM(Dest\_j), (j=1...M)}.

Value for P2MP latency metric is to be assigned by IANA

#### 6.3.2. P2MP Latency Variation Metric

P2MP latency variation metric type of METRIC object in PCEP encodes the path latency variation metric for destination that observes the worst latency variation metric among all destination of the P2MP tree. Specifically, extending on the above mentioned terminology:

- A P2MP Tree T comprises of a set of M destinations {Dest\_j, (j=1...M)}
- P2P latency variation metric of the Path to destination Dest\_j is denoted by LVM(Dest\_j).
- P2MP latency variation metric for the P2MP tree T = Maximum {LVM(Dest\_j), (j=1...M)}.

Value for P2MP latency variation metric is to be assigned by IANA

### 7. IANA Considerations

IANA has defined a registry for new METRIC type.

Type	Meaning
13(TBD)	Latency (delay) metric
14(TBD)	Latency Variation (jitter) metric
15(TBD)	Packet Loss metric
16(TBD)	P2MP latency metric
17(TBD)	P2MP latency variation metric

### 8. Security Considerations

This document defines three new METRIC Types which does not add any new security concerns to PCEP protocol.

## 9. Manageability Considerations

### 9.1. Control of Function and Policy

The only configurable item is the support of the new service-aware METRICS on a PCE which MAY be controlled by a policy module. If the new METRIC is not supported/allowed on a PCE, it MUST send a PCErr message as specified in Section 4.4.

### 9.2. Information and Data Models

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

### 9.3. Liveness Detection and Monitoring

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

### 9.4. Verify Correct Operations

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

### 9.5. Requirements On Other Protocols

PCE requires the TED to be populated with network performance information like link latency, latency variation and packet loss. This mechanism is described in [OSPF-TE-EXPRESS] or [ISIS-TE-EXPRESS].

### 9.6. Impact On Network Operations

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

## 10. Acknowledgments

We would like to thank Young Lee, Venugopal Reddy, Reeja Paul, Sandeep Kumar Boina, Suresh babu, Quintin Zhao and Chen Huaimo for their useful comments and suggestions.

## 11. References

## 11.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC5440] Vasseur, JP. and JL. Le Roux, "Path Computation Element (PCE) Communication Protocol (PCEP)", RFC 5440, March 2009.

## 11.2. Informative References

- [RFC5441] Vasseur, JP., Zhang, R., Bitar, N., and JL. Le Roux, "A Backward-Recursive PCE-Based Computation (BRPC) Procedure to Compute Shortest Constrained Inter-Domain Traffic Engineering Label Switched Paths", RFC 5441, April 2009.
- [RFC5316] Chen, M., Zhang, R., and X. Duan, "ISIS Extensions in Support of Inter-Autonomous System (AS) MPLS and GMPLS Traffic Engineering", RFC 5316, December 2008.
- [RFC5392] Chen, M., Zhang, R., and X. Duan, "OSPF Extensions in Support of Inter-Autonomous System (AS) MPLS and GMPLS Traffic Engineering", RFC 5392, January 2009.
- [RFC5541] Le Roux, JL., Vasseur, JP., and Y. Lee, "Encoding of Objective Functions in the Path Computation Element Communication Protocol (PCEP)", RFC 5541, June 2009.
- [RFC6374] Frost, D. and S. Bryant, "Packet Loss and Delay Measurement for MPLS Networks", RFC 6374, September 2011.
- [MPLS-DELAY-FWK] Fu, X., Manral, V., McDysan, D., Malis, A., Giacalone, S., Betts, M., Wang, Q., and J. Drake, "Traffic Engineering architecture for services aware MPLS [draft-fuxh-mpls-delay-loss-te-framework]", Oct 2012.
- [OSPF-TE-EXPRESS] Giacalone, S., Ward, D., Drake, J., Atlas, A., and S. Previdi, "OSPF Traffic Engineering (TE) Metric Extensions

[draft-ietf-ospf-te-metric-extensions]",  
May 2012.

- [ISIS-TE-EXPRESS] Previdi, S., Giacalone, S., Ward, D., Drake, J., Atlas, A., and C. Filsfils, "IS-IS Traffic Engineering (TE) Metric Extensions [draft-previdi-isis-te-metric-extensions]", Oct 2012.
- [MPLS-TE-EXPRESS-PATH] Atlas, A., Drake, J., Ward, D., Giacalone, S., Previdi, S., and C. Filsfils, "Performance-based Path Selection for Explicitly Routed LSPs [draft-atlas-mpls-te-express-path]", June 2012.
- [PCEP-MIB] Kiran Koushik, A S., Stephan, E., Zhao, Q., King, D., and J. Hardwick, "PCE communication protocol(PCEP) Management Information Base [draft-ietf-pce-pcep-mib]", July 2012.

#### Appendix A. Contributor Addresses

Clarence Filsfils  
Cisco Systems  
EMail: cfilsfil@cisco.com

Siva Sivabalan  
Cisco Systems  
EMail: msiva@cisco.com

Stefano Previdi  
Cisco Systems  
EMail: sprevidi@cisco.com

Udayasree Palle  
Huawei Technologies India Pvt Ltd  
Leela Palace  
Bangalore, Karnataka 560008  
INDIA  
EMail: udayasree.palle@huawei.com

Authors' Addresses

Dhruv Dhody  
Huawei Technologies India Pvt Ltd  
Leela Palace  
Bangalore, Karnataka 560008  
INDIA

EMail: dhruv.dhody@huawei.com

Vishwas Manral  
Hewlett-Packard Corp.  
191111 Pruneridge Ave.  
Cupertino, CA 95014  
USA

EMail: vishwas.manral@hp.com

Zafar Ali  
Cisco Systems

EMail: zali@cisco.com

George Swallow  
Cisco Systems

EMail: swallow@cisco.com

Kenji Kumaki  
KDDI Corporation

EMail: ke-kumaki@kddi.com



PCE Working Group  
Internet-Draft  
Intended status: Experimental  
Expires: December 17, 2014

Q. Zhao  
D. Dhody  
Huawei Technology  
D. King  
Old Dog Consulting  
Z. Ali  
Cisco Systems  
R. Casellas  
CTTC  
June 17, 2014

PCE-based Computation Procedure To Compute Shortest Constrained P2MP  
Inter-domain Traffic Engineering Label Switched Paths  
draft-ietf-pce-pcep-inter-domain-p2mp-procedures-08

## Abstract

The ability to compute paths for constrained point-to-multipoint (P2MP) Traffic Engineering Label Switched Paths (TE LSPs) across multiple domains has been identified as a key requirement for the deployment of P2MP services in MPLS and GMPLS-controlled networks. The Path Computation Element (PCE) has been recognized as an appropriate technology for the determination of inter-domain paths of P2MP TE LSPs.

This document describes an experiment to provide procedures and extensions to the PCE communication Protocol (PCEP) for the computation of inter-domain paths for P2MP TE LSPs.

## Status of This Memo

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

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

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

This Internet-Draft will expire on December 17, 2014.

## Copyright Notice



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

## Table of Contents

1. Introduction . . . . .	.2
1.1. Scope . . . . .	.2
1.2. Requirements Language . . . . .	.2
2. Terminology . . . . .	.2
3. Examination of Existing Mechanisms . . . . .	.3
4. Assumptions . . . . .	.5
5. Requirements . . . . .	.5
6. Objective Functions and Constraints. . . . .	.7
7. P2MP Path Computation Procedures . . . . .	.8
7.1. General . . . . .	.8
7.2. Core-Trees . . . . .	.9
7.3. Optimal Core-Tree Computation Procedure. . . . .	.12
7.4. Sub-tree Computation Procedures . . . . .	.13
7.5. PCEP Protocol Extensions . . . . .	.13
7.5.1. The Extension of RP Object . . . . .	.13
7.5.2. Domain and PCE Sequence . . . . .	.14
7.6. Relationship with Hierarchical PCE . . . . .	.14
7.7. Parallelism . . . . .	.15
8. Protection . . . . .	.15
8.1. End-to-end Protection . . . . .	.15
8.2. Domain Protection . . . . .	.15
9. Manageability Considerations . . . . .	.16
9.1. Control of Function and Policy . . . . .	.16
9.2. Information and Data Models . . . . .	.16
9.3. Liveness Detection and Monitoring . . . . .	.16
9.4. Verifying Correct Operation . . . . .	.16
9.5. Requirements on Other Protocols and Functional Components.17	
9.6. Impact on Network Operation . . . . .	.17
9.7. Policy Control . . . . .	.17
10. Security Considerations . . . . .	.17
11. IANA Considerations . . . . .	.18
12. Acknowledgements . . . . .	.19
13. References . . . . .	.19
13.1. Normative References . . . . .	.19

Internet-Draft	PCEP P2MP Inter-Domain Procedures	June 2014
13.2. Informative References	. . . . .	.19
14. Contributors' Addresses	. . . . .	.21
15. Authors' Addresses	. . . . .	.21

## 1. Introduction

Multicast services are increasingly in demand for high-capacity applications such as multicast Virtual Private Networks (VPNs), IP-television (IPTV) which may be on-demand or streamed, and content-rich media distribution (for example, software distribution, financial streaming, or database-replication). The ability to compute constrained Traffic Engineering Label Switched Paths (TE LSPs) for point-to-multipoint (P2MP) LSPs in Multiprotocol Label Switching (MPLS) and Generalized MPLS (GMPLS) networks across multiple domains are therefore required.

The applicability of the PCE [RFC4655] for the computation of such paths is discussed in [RFC5671], and the requirements placed on the PCE communications Protocol (PCEP) for this are given in [RFC5862].

This document details the requirements for inter-domain P2MP path computation, it then describes the experimental procedure "core-tree" path computation, developed to address the requirements and objectives for inter-domain P2MP path computation.

When results of implementation and deployment are available, this document will be updated and refined, and then moved from Experimental status to Standards Track.

### 1.2. Scope

The inter-domain P2MP path computation procedures described in this document is experimental. The experiment is intended to enable research for the usage of the PCE to support inter-domain P2MP path computation.

This document is not intended to replace the intra-domain P2MP path computation approach defined by [RFC6006], and will not impact existing PCE procedures and operations.

### 1.3. 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].

## 2. Terminology

Terminology used in this document is consistent with the related MPLS/GMPLS and PCE documents [RFC4461], [RFC4655], [RFC4875], [RFC5376], [RFC5440], [RFC5441], [RFC5671] and [RFC5862].

The additional terms Core-Tree, Leaf Domain, Path Tree, Path Domain Sequence, Path Domain Tree, Root Domain, Sub-Tree and Transit/branch Domain are further defined below.

Core-Tree: a P2MP tree where the root is the ingress Label Switching Router (LSR), and the leaf nodes are the entry BNs of the leaf domains.

Entry BN of domain(n): a Boundary Node (BN) connecting domain(n-1) to domain(n) along a determined sequence of domains.

Exit BN of domain(n): a BN connecting domain(n) to domain(n+1) along a determined sequence of domains.

H-PCE: Hierarchical PCE (as per [RFC6805]).

Leaf Domain: a domain with one or more leaf nodes.

Path Tree: a set of LSRs and TE links that comprise the path of a P2MP TE LSP from the ingress LSR to all egress LSRs (the leaf nodes).

Path Domain Sequence: the known sequence of domains for a path between the root domain and a leaf domain.

Path Domain Tree: the tree formed by the domains that the P2MP path crosses, where the source (ingress) domain is the root domain.

PCE(i): a PCE that performs path computations for domain(i).

Root Domain: the domain that includes the ingress (root) LSR.

Sub-tree: a P2MP tree where the root is the selected entry BN of the leaf domain and the leaf nodes are the destinations (leaves) in that domain. The sub-trees are grafted to the core-tree.

Transit/branch Domain: a domain that has an upstream and one or more downstream neighbor domain.

### 3. Examination of Existing Mechanisms

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

[RFC4875] describes how to set up P2MP TE LSPs for use in MPLS and GMPLS-controlled networks. The PCE is identified as a suitable application for the computation of paths for P2MP TE LSPs [RFC5671].

[RFC5441] specifies a procedure relying on the use of multiple PCEs to compute Point to Point (P2P) inter-domain constrained shortest paths across a predetermined sequence of domains, using a Backward Recursive Path Computation (BRPC) technique. The technique can be combined with the use of Path-Keys [RFC5520] to preserve confidentiality across domains, which is sometimes required when domains are managed by different Service Providers.

PCEP [RFC5440] was extended for point-to-multipoint (P2MP) path computation requests in [RFC6006].

As discussed in [RFC4461], a P2MP tree is the ordered set of LSRs and TE links that comprise the path of a P2MP TE LSP from its ingress LSR to all of its egress LSRs. A P2MP LSP is set up with TE constraints and allows efficient packet or data replication at various branching points in the network. As per [RFC5671] branch point selection is fundamental to the determination of the paths for a P2MP TE LSP. Not only is this selection constrained by the network topology and available network resources, but it is determined by the objective functions (OF) that may be applied to path computation.

Generally, an inter-domain P2MP tree (i.e., a P2MP tree with source and at least one destination residing in different domains) is particularly difficult to compute even for a distributed PCE architecture. For instance, while the BRPC may be well-suited for P2P paths, P2MP path computation involves multiple branching path segments from the source to the multiple destinations. As such, inter-domain P2MP path computation may result in a plurality of per-domain path options that may be difficult to coordinate efficiently and effectively between domains. That is, when one or more domains have multiple ingress and/or egress boundary nodes (i.e., when the domains are multiply inter-connected), existing techniques may be convoluted when used to determine which boundary node of another domain will be utilized for the inter-domain P2MP tree, and no way to limit the computation of the P2MP tree to those utilized boundary nodes.

A trivial solution to the computation of inter-domain P2MP tree would be to compute shortest inter-domain P2P paths from source to each destination and then combine them to generate an inter-domain, shortest-path-to-destination P2MP tree. This solution, however, cannot be used to trade cost to destination for overall tree cost

(i.e., it cannot produce a Minimum Cost Tree (MCT)) and in the context of inter-domain P2MP TE LSPs it cannot be used to reduce the number of domain boundary nodes that are transited. Computing P2P TE LSPs individually does not guarantee the generation of an optimal P2MP tree for every definition of "optimal" in every topology.

Per Domain path computation [RFC5152] may be used to compute P2MP multi-domain paths, but may encounter the issues previously described. Furthermore, this approach may also be considered to have scaling issues during LSP setup. That is, the LSP to each leaf is signaled separately, and each boundary node needs to perform path computation for each leaf.

P2MP Minimum Cost Tree (MCT), i.e. a computation which guarantees the least cost resulting tree, typically is an NP-complete problem. Moreover, adding and/or removing a single destination to/from the tree may result in an entirely different tree. In this case, frequent MCT path computation requests may prove computationally intensive, and the resulting frequent tunnel reconfiguration may even cause network destabilization.

This document presents a solution, procedures and extensions to PCEP to support P2MP inter-domain path computation.

#### 4. Assumptions

Within this document we make the following assumptions:

- o Due to deployment and commercial limitations (e.g., inter-AS (Autonomous System) peering agreements), the path domain tree will be known in advance;
- o Each PCE knows about any leaf LSRs in the domain it serves;

Additional assumptions are documented in [RFC5441] and are not repeated here.

#### 5. Requirements

This section summarizes the requirements specific to computing inter-domain P2MP paths. In these requirements we note that the actual computation time taken by any PCE implementation is outside the scope of this document, but we observe that reducing the complexity of the required computations has a beneficial effect on the computation time regardless of implementation. Additionally, reducing the number of message exchanges and the amount of information exchanged will reduce the overall computation time for the entire P2MP tree. We refer to

It is also important that the solution can preserve confidentiality  
across domains, which is required when domains are managed by  
different Service Providers via Path-Key mechanism [RFC5520].

Other than the requirements specified in [RFC5862], a number of  
requirements specific to inter-domain P2MP are detailed below:

1. The complexity of the computation for each sub-tree within each  
domain SHOULD be dependent only on the topology of the domain and  
it SHOULD be independent of the domain sequence.
2. The number of PCReq (Path Computation Request) and PCRep (Path  
Computation Reply) messages SHOULD be independent of the number  
of multicast destinations in each domain.
3. It SHOULD be possible to specify the domain entry and exit nodes  
in the PCReq.
4. Specifying which nodes are to be used as branch nodes SHOULD be  
supported in the PCReq.
5. Reoptimization of existing sub-trees SHOULD be supported.
6. It SHOULD be possible to compute diverse P2MP paths from existing  
P2MP paths.

## 6. Objective Functions and Constraints

For the computation of a single or a set of P2MP TE LSPs, a request  
to meet specific optimization criteria, called an Objective Function  
(OF), MAY be used. Using an OF to select the "best" candidate path,  
include:

- o The sub-tree within each domain SHOULD be optimized using minimum  
cost tree [RFC5862], or shortest path tree [RFC5862].

In addition to the OFs, the following constraints MAY also be  
beneficial for inter-domain P2MP path computation:

1. The computed P2MP "core-tree" SHOULD be optimal when only  
considering the paths to the leaf domain entry BNs.
2. Grafting and pruning of multicast destinations (sub-tree) within  
a leaf domain SHOULD ensure minimal impact on other domains

3. It SHOULD be possible to choose to optimize the core-tree.
4. It SHOULD be possible to choose optimize the entire tree (P2MP LSP).
5. It SHOULD be possible to combine the aforementioned OFs and constraints for P2MP path computation.

When implementing and operating P2MP LSPs, following needs to be taken into consideration:

- o The complexity of computation.
- o The optimality of the tree (core-tree as well as full P2MP LSP tree).
- o The stability of the core-tree.

The solution SHOULD allow these trade-offs to be made at computation time.

The algorithms used to compute optimal paths using a combination of OFs and multiple constraints is out of scope of this document.

## 7. P2MP Path Computation Procedures

### 7.1. General

A P2MP path computation can be broken down into two steps of core-tree computation and grafting of sub-trees. Breaking the procedure into these specific steps has the following impact:

- o The core-tree and sub-tree are smaller in comparison to the full P2MP Tree and are thus easier to compute.
- o An implementation MAY choose to keep the core-tree fairly static or computed offline (trade-off with optimality).
- o Adding/Pruning of leaves which require changes to sub-tree in leaf-domain only.
- o The PCEP message size is smaller in comparison.

Allowing the core-tree based solution to provide an optimal inter-domain P2MP TE LSP.

The following sub-sections describe the core-tree based mechanism, including procedures and PCEP extensions, that satisfy the requirements and objectives specified in Section 5 and Section 6 of this document.

## 7.2. Core-Trees

A core-tree is defined as a tree that satisfies the following conditions:

- o The root of the core-tree is the ingress LSR in the root domain;
- o The leaves of the core-tree are the entry boundary nodes in the leaf domains.

To support confidentiality these nodes and links MAY be hidden using the path-key mechanism [RFC5520], but they MUST be computed and be a part of core-tree.

For example, consider the Domain Tree in Figure 1 below, representing a domain tree of 6 domains, and part of the resulting core-tree which satisfies the aforementioned conditions.



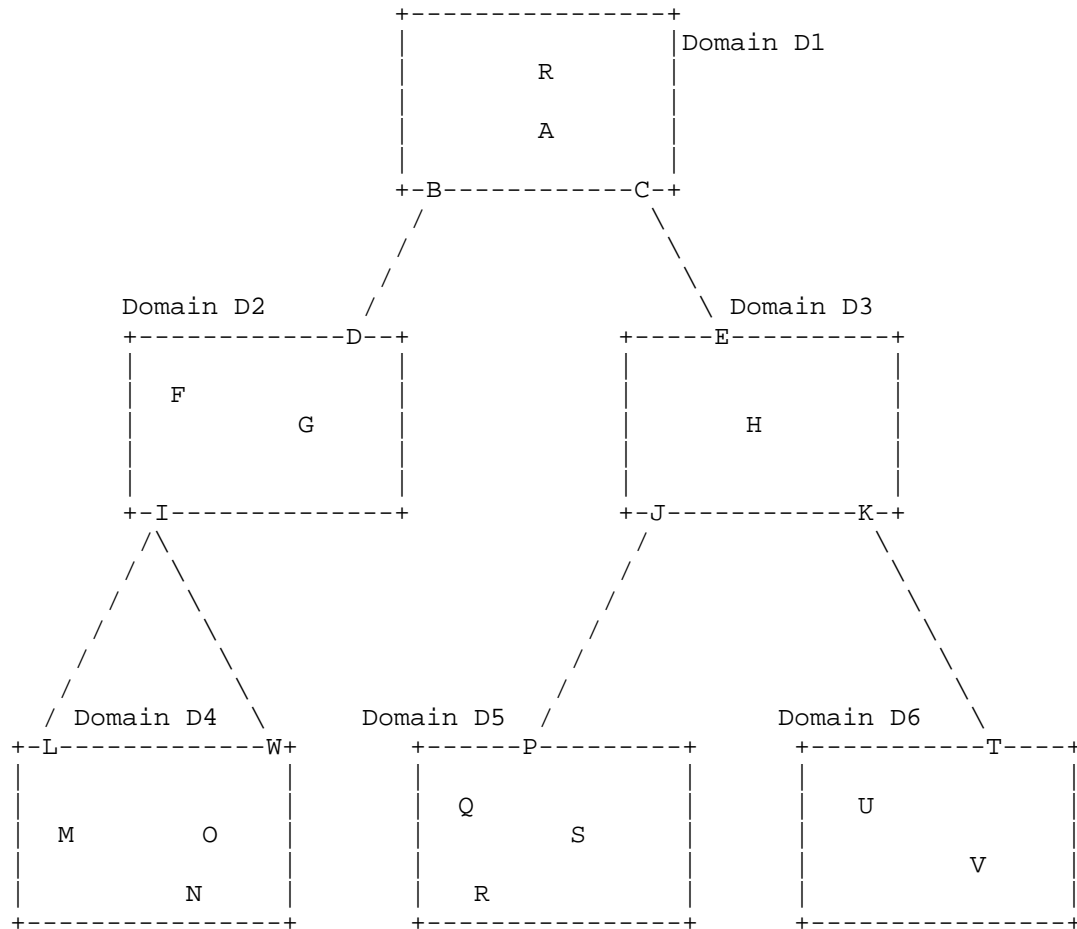


Figure 1: Domain Tree Example

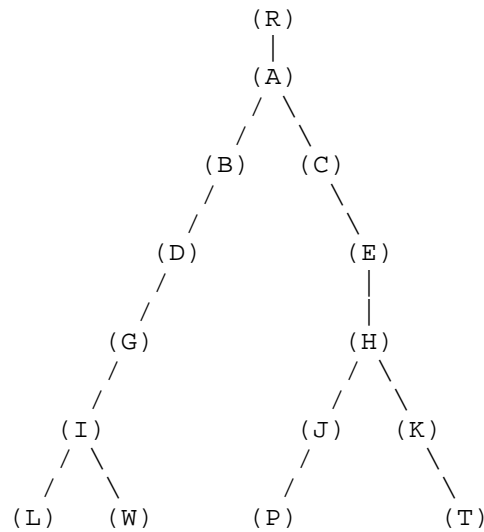


Figure 2: Core-Tree

A core-tree is computed such that root of the tree is R and the leaf node are the entry nodes of the destination domains (L, W, P and T). Path-key mechanism can be used to hide the internal nodes and links (node G and H are hidden via Path-Key PK1 and PK2 respectively) in the final core-tree as shown below for domain D2 and D3.

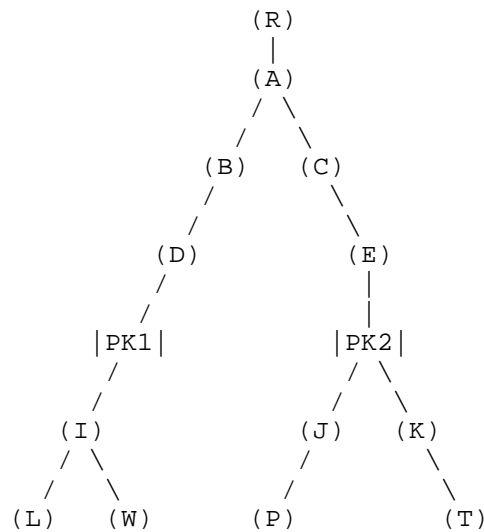


Figure 3: Core-Tree with Path-Key

Applying the core-tree procedure to large groups of domains, such as the Internet, is not considered feasible or desirable, and is out of scope for this document.

The following extended BRPC-based procedure can be used to compute the core-tree. Note that a root PCE MAY further use its own enhanced optimization techniques in future to compute the core-tree.

A BRPC-based core-tree path computation procedure is described below:

1. Using the BRPC procedures to compute the VSPT(i) (Virtual Shortest Path Tree) for each leaf BN(i),  $i=1$  to  $n$ , where  $n$  is the total number of entry nodes for all the leaf domains. In each VSPT(i), there are a number of  $P(i)$  paths.
2. When the root PCE has computed all the VSPT(i),  $i=1$  to  $n$ , take one path from each VSPT and form all possible sets of paths, we call them PathSet(j),  $j=1$  to  $M$ , where  $M=P(1) \times P(2) \dots \times P(n)$ ;
3. For each PathSet(j), there are  $n$  S2L (Source-to-Leaf) BN paths and form these  $n$  paths into a core-tree(j);
4. There will be  $M$  number core-trees computed from step 3. An optimal core-tree is selected based on the OF and constraints.

Note that, since point to point BRPC procedure is used to compute VSPT, the path request and response message format defined in [RFC5440] are used.

Also note that the application of BRPC in the aforementioned procedure differs from the typical one since paths returned from a downstream PCE are not necessarily pruned from the solution set (extended VSPT) by intermediate PCEs. The reason for this is that if the PCE in a downstream domain does the pruning and returns the single optimal sub-path to the upstream PCE, the combination of these single optimal sub-paths into a core-tree is not necessarily optimal even if each S2L (Source-to-Leaf) sub-path is optimal.

Without trimming, the ingress PCE will obtain all the possible S2L sub-paths set for the entry boundary nodes of the leaf domain. The PCE will then, by looking through all the combinations and taking one sub-path from each set to build one tree, can select the optimal core-tree.

A PCE MAY add equal cost paths within the domain while constructing an extended VSPT. This will provide the ingress PCE more candidate paths for an optimal core-tree.

The proposed method may present a scalability problem for the dynamic computation of the core-tree (by iterative checking of all combinations of the solution space), specially with dense/meshed domains. Considering a domain sequence D1, D2, D3, D4, where the Leaf Boundary Node is at domain D4, PCE(4) will return 1 path. PCE(3) will return N paths, where N is  $E(3) \times X(3)$ , where  $E(k) \times X(k)$  denotes the number of entry nodes times the number of exit nodes for that domain. PCE(2) will return M paths, where  $M = E(2) \times X(2) \times N = E(2) \times X(2) \times E(3) \times X(3) \times 1$ , etc. Generally speaking the number of potential paths at the ingress PCE Q =  $\prod E(k) \times X(k)$ .

Consequently, it is expected that the core-tree will be typically computed offline, without precluding the use of dynamic, online mechanisms such as the one presented here, in which case it SHOULD be possible to configure transit PCEs to control the number of paths sent upstream during BRPC (trading trimming for optimality at the point of trimming and downwards).

#### 7.4. Sub-tree Computation Procedures

Once the core-tree is built, the grafting of all the leaf nodes from each domain to the core-tree can be achieved by a number of algorithms. One algorithm for doing this phase is that the root PCE will send the request with C bit set (as defined in section 7.4.1 of this document) for the path computation to the destination(s) directly to the PCE where the destination(s) belong(s) along with the core-tree computed from section 7.2.

This approach requires that the root PCE manage a potentially large number of adjacencies (either in persistent or non-persistent mode), including PCEP adjacencies to PCEs that are not within neighbor domains.

An alternative would involve establishing PCEP adjacencies that correspond to the PCE domain tree. This would require that branch PCEs forward requests and responses from the root PCE towards the leaf PCEs and vice-versa.

Note that the P2MP path request and response format is as per [RFC6006], where Record Route Object (RRO) are used to carry the core-tree paths in the P2MP grafting request.

The algorithms to compute the optimal large sub-tree are outside scope of this document.

#### 7.5. PCEP Protocol Extensions

##### 7.5.1. The Extension of RP Object

This experiment will be carried out by extending the RP (Request Parameters) object (defined in [RFC5440]) used in PCEP requests and responses.

The extended format of the RP object body to include the C bit is as follows:

The C bit is added in the flag bits field of the RP object to signal the receiver of the message that the request/reply is for inter-domain P2MP core-tree or not.

The following flag is added in this draft:

Bit Number	Name Flag
TBA	Core-tree computation (C-bit)

C bit (Core-Tree bit - 1 bit):

- 0: This indicates that this is not for an inter-domain P2MP core-tree.
- 1: This indicates that this is a PCEP request or a response for the computation of a inter-domain core-tree or for the grafting of a sub-tree to a inter-domain core-tree.

#### 7.5.2. Domain and PCE Sequence

The procedure described in this document requires the domain-tree to be known in advance. This information MAY be either administratively predetermined or dynamically discovered by some means such as Hierarchical PCE (H-PCE) [RFC6805] framework, or derived through the IGP/BGP routing information.

Examples of ways to encode the domain path tree include [RFC5886] using PCE-ID Object and [DOMAIN-SEQ].

#### 7.6. Using H-PCE for Scalability

The ingress/root PCE is responsible for the core-tree computation as well as grafting of sub-trees into the multi-domain tree. Therefore, the ingress/root PCE will receive all computed path segments from all the involved domains. When the ingress/root PCE chooses to have a PCEP session with all involved PCEs, this may cause an excessive number of sessions or added complexity in implementations.

The use of the H-PCE framework [RFC6805] may be used to establish a dedicated PCE with the capability (memory and CPU) and knowledge to maintain the necessary PCEP sessions. The parent PCE would be responsible to request intra-domain path computation request to the

### 7.7. Parallelism

In order to minimize latency in path computation in multi-domain networks, intra-domain path segments and intra-domain sub-trees can be computed in parallel when possible. The proposed procedures in this draft present opportunities for parallelism:

1. The BRPC procedure for each leaf boundary node can be launched in parallel by the ingress/root PCE for dynamic computation of core-tree.
2. The grafting of sub-trees can be triggered in parallel once the core-tree is computed.

One of the potential issues of parallelism is that the ingress PCE would require a potentially high number of PCEP adjacencies to "remote" PCEs at the same time and that may not be desirable.

## 8. Protection

It is envisaged that protection may be required when deploying and using inter-domain P2MP TE LSPs. The procedures and mechanisms defined in this document do not prohibit the use of existing and proposed types of protection, including: end-to-end protection [RFC4875] and domain protection schemes.

Segment or facility (link and node) protection is problematic in inter-domain environment due to the limit of Fast-reroute (FRR) [RFC4875] requiring knowledge of its next-hop across domain boundaries whilst maintaining domain confidentiality. Although the FRR protection might be implemented if next-hop information was known in advance.

### 8.1. End-to-end Protection

An end-to-end protection (for nodes and links) principle can be applied for computing backup P2MP TE LSPs. During computation of the core-tree and sub-trees, may also be taken into consideration. A PCE may compute the primary and backup P2MP TE LSP together or sequentially.

### 8.2. Domain Protection

In this protection scheme, backup P2MP Tree can be computed which excludes the transit/branch domain completely. A backup domain path tree is needed with the same source domain and destinations domains

## 9. Manageability Considerations

[RFC5862] describes various manageability requirements in support of P2MP path computation when applying PCEP. This section describes how manageability requirements mentioned in [RFC5862] are supported in the context of PCEP extensions specified in this document.

Note that [RFC5440] describes various manageability considerations in PCEP, and most of manageability requirements mentioned in [RFC6006] are already covered there.

### 9.1. Control of Function and Policy

In addition to PCE configuration parameters listed in [RFC5440] and [RFC6006], the following additional parameters might be required:

- o The ability to enable or disable multi-domain P2MP path computations on the PCE.
- o The PCE may be configured to enable or disable the advertisement of its multi-domain P2MP path computation capability.

### 9.2. Information and Data Models

A number of MIB objects have been defined for general PCEP control and monitoring of P2P computations in [PCEP-MIB]. [RFC5862] specifies that MIB objects will be required to support the control and monitoring of the protocol extensions defined in this document. [PCEP-P2MP-MIB] describes managed objects for modeling of PCEP communications between a PCC and PCE, and PCE to PCE, P2MP path computation requests and responses.

### 9.3. Liveness Detection and Monitoring

No changes are necessary to the liveness detection and monitoring requirements as already embodied in [RFC4657].

It should be noted that multi-domain P2MP computations are likely to take longer than P2P computations, and single domain P2MP computations. The liveness detection and monitoring features of the PCEP SHOULD take this into account.

### 9.4. Verifying Correct Operation

There are no additional requirements beyond those expressed in [RFC4657] for verifying the correct operation of the PCEP. Note that verification of the correct operation of the PCE and its algorithms is out of scope for the protocol requirements, but a PCC MAY send the same request to more than one PCE and compare the results.

#### 9.5. Requirements on Other Protocols and Functional Components

A PCE operates on a topology graph that may be built using information distributed by TE extensions to the routing protocol operating within the network. In order that the PCE can select a suitable path for the signaling protocol to use to install the P2MP TE LSP, the topology graph MUST include information about the P2MP signaling and branching capabilities of each LSR in the network.

Mechanisms for the knowledge of other domains, the discovery of corresponding PCEs and their capabilities SHOULD be provided and that this information MAY be collected by other mechanisms.

Whatever means is used to collect the information to build the topology graph, the graph MUST include the requisite information. If the TE extensions to the routing protocol are used, these SHOULD be as described in [RFC5073].

#### 9.6. Impact on Network Operation

The use of a PCE to compute P2MP paths is not expected to have significant impact on network operations. However, it should be noted that the introduction of P2MP support to a PCE that already provides P2P path computation might change the loading of the PCE significantly, and that might have an impact on the network behavior, especially during recovery periods immediately after a network failure.

The dynamic computation of core-trees might also have an impact on the load of the involved PCEs as well as path computation times.

It should be noted that pre-computing and maintaining domain-trees might be a considerable administration effort on the operator.

#### 9.7. Policy Control

[RFC5394] provides additional details on policy within the PCE architecture and also provides context for the support of PCE Policy. They are also applicable to Inter-domain P2MP Path computation via the core-tree mechanism.

### 10. Security Considerations



As described in [RFC5862], P2MP path computation requests are more CPU-intensive and also utilize more link bandwidth. In the event of an unauthorized P2MP path computation request, or a denial of service attack, the subsequent PCEP requests and processing may be disruptive to the network. Consequently, it is important that implementations conform to the relevant security requirements of [RFC5440] that specifically help to minimize or negate unauthorized P2MP path computation requests and denial of service attacks. These mechanisms include:

- o Securing the PCEP session requests and responses using TCP security techniques (Section 10.2 of [RFC5440]).
- o Authenticating the PCEP requests and responses to ensure the message is intact and sent from an authorized node (Section 10.3 of [RFC5440]).
- o Providing policy control by explicitly defining which PCCs, via IP access-lists, are allowed to send P2MP path requests to the PCE (Section 10.6 of [RFC5440]).

PCEP operates over TCP, so it is also important to secure the PCE and PCC against TCP denial of service attacks. Section 10.7.1 of [RFC5440] outlines a number of mechanisms for minimizing the risk of TCP-based denial of service attacks against PCEs and PCCs.

PCEP implementations SHOULD also consider the additional security provided by the TCP Authentication Option (TCP-AO) [RFC5925].

Finally, any multi-domain operation necessarily involves the exchange of information across domain boundaries. This may represent a significant security and confidentiality risk especially when the domains are controlled by different commercial entities. PCEP allows individual PCEs to maintain confidentiality of their domain path information by using path-keys [RFC5520] and would allow for securing of domain path information when performing core-tree based path computations.

## 11. IANA Considerations

IANA maintains the "Path Computation Element Protocol (PCEP) Numbers" registry with the "RP Object Flag Field" sub-registry.

IANA is requested to allocate a new bit from this registry as follows:

Bit	Description	Reference
-----	-------------	-----------

## 12. Acknowledgements

The authors would like to thank Adrian Farrel, Dan Tappan, Olufemi Komolafe, Oscar Gonzalez de Dios and Julien Meuric for their valuable comments on this document.

## 13. References

### 13.1. 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.
- [RFC5441] Vasseur, JP., Zhang, R., Bitar, N., and JL. Le Roux, "A Backward-Recursive PCE-Based Computation (BRPC) Procedure to Compute Shortest Constrained Inter-Domain Traffic Engineering Label Switched Paths", RFC 5441, April 2009.
- [RFC6006] Zhao, Q., King, D., Verhaeghe, F., Takeda, T., Ali, Z., and J. Meuric, "Extensions to the Path Computation Element Communication Protocol (PCEP) for Point-to-Multipoint Traffic Engineering Label Switched Paths", RFC 6006, September 2010.

### 13.2. Informative References

- [RFC4461] Yasukawa, S., "Signaling Requirements for Point-to-Multipoint Traffic-Engineered MPLS Label Switched Paths (LSPs)", RFC 4461, April 2006.
- [RFC4655] Farrel, A., Vasseur, J., and J. Ash, "A Path Computation Element (PCE)-Based Architecture", RFC 4655, August 2006.
- [RFC4657] Ash, J. and J. Le Roux, "Path Computation Element (PCE) Communication Protocol Generic Requirements", RFC 4657, September 2006.
- [RFC4875] Aggarwal, R., Papadimitriou, D., and S. Yasukawa, "Extensions to Resource Reservation Protocol -

Traffic Engineering (RSVP-TE) for Point-to-Multipoint TE Label Switched Paths (LSPs)", RFC 4875, May 2007.

- [RFC5073] Vasseur, J. and J. Le Roux, "IGP Routing Protocol Extensions for Discovery of Traffic Engineering Node Capabilities", RFC 5073, December 2007.
- [RFC5152] Vasseur, JP., Ayyangar, A., and R. Zhang, "A Per-Domain Path Computation Method for Establishing Inter-Domain Traffic Engineering (TE) Label Switched Paths (LSPs)", RFC 5152, February 2008.
- [RFC5376] Bitar, N., Zhang, R., and K. Kumaki, "Inter-AS Requirements for the Path Computation Element Communication Protocol (PCECP)", RFC 5376, November 2008.
- [RFC5394] Bryskin, I., Papadimitriou, D., Berger, L., and J. Ash, "Policy-Enabled Path Computation Framework", RFC 5394, December 2008.
- [RFC5520] Bradford, R., Vasseur, JP., and A. Farrel, "Preserving Topology Confidentiality in Inter-Domain Path Computation Using a Path-Key-Based Mechanism", RFC 5520, April 2009.
- [RFC5671] Yasukawa, S. and A. Farrel, "Applicability of the Path Computation Element (PCE) to Point-to-Multipoint (P2MP) MPLS and GMPLS Traffic Engineering (TE)", RFC 5671, October 2009.
- [RFC5862] Yasukawa, S. and A. Farrel, "Path Computation Clients (PCC) - Path Computation Element (PCE) Requirements for Point-to-Multipoint MPLS-TE", RFC 5862, June 2010.
- [RFC5886] Vasseur, JP., Le Roux, JL., and Y. Ikejiri, "A Set of Monitoring Tools for Path Computation Element (PCE)-Based Architecture", RFC 5886, June 2010.
- [RFC5925] Touch, J., Mankin, A., and R. Bonica, "The TCP Authentication Option", RFC 5925, June 2010.
- [RFC6805] King, D. and A. Farrel, "The Application of the Path Computation Element Architecture to the Determination of a Sequence of Domains in MPLS and GMPLS", RFC 6805, November 2012.

Internet-Draft	PCEP P2MP Inter-Domain Procedures	June 2014
[PCEP-MIB]	Koushik, K., Stephan, E., Zhao, Q., King, D., and J. Hardwick, "PCE communication protocol (PCEP) Management Information Base (Work in Progress)", April 2014.	
[PCEP-P2MP-MIB]	Zhao, Q., Dhody, D., Palle, U., and D. King, "Management Information Base for the PCE Communications Protocol (PCEP) When Requesting Point-to-Multipoint Services (Work in Progress)", Aug 2012.	
[DOMAIN-SEQ]	Dhody, D., Palle, U., and R. Casellas, "Standard Representation Of Domain Sequence (Work in Progress)", July 2014.	

#### 14. Contributor Addresses

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

EMail: msiva@cisco.com

Tarek Saad  
Cisco Systems, Inc.  
2000 Innovation Drive  
Kanata, Ontario K2K 3E8  
CANADA

EMail: tsaad@cisco.com

#### 15. Authors' Addresses

Quintin Zhao  
Huawei Technology  
125 Nagog Technology Park  
Acton, MA 01719  
US

EMail: quintin.zhao@huawei.com

Dhruv Dhody  
Huawei Technology  
Leela Palace  
Bangalore, Karnataka 560008

EMail: dhruv.dhody@huawei.com

Zafar Ali  
Cisco Systems  
2000 Innovation Drive  
Kanata, Ontario K2K 3E8  
CANADA

EMail: zali@cisco.com

Daniel King  
Old Dog Consulting  
UK

EMail: daniel@olddog.co.uk

Ramon Casellas  
CTTC  
Av. Carl Friedrich Gauss n7  
Castelldefels, Barcelona 08860  
SPAIN

EMail: ramon.casellas@cttc.es



PCE Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: April 27, 2015

A. Koushik  
Brocade Communications Inc.  
E. Stephan  
Orange  
Q. Zhao  
Huawei Technology  
D. King  
Old Dog Consulting  
J. Hardwick  
Metaswitch  
October 24, 2014

Path Computation Element Communications Protocol (PCEP) Management  
Information Base (MIB) Module  
draft-ietf-pce-pcep-mib-11

Abstract

This memo defines a portion of the Management Information Base (MIB) for use with network management protocols in the Internet community. In particular, it describes managed objects for modeling 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.

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 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.

## Table of Contents

1. Introduction . . . . .	2
1.1. Requirements Language . . . . .	3
1.2. Terminology . . . . .	3
2. The Internet-Standard Management Framework . . . . .	3
3. PCEP MIB Module Architecture . . . . .	4
3.1. pcePcepEntityTable . . . . .	4
3.2. pcePcepPeerTable . . . . .	5
3.3. pcePcepSessTable . . . . .	5
3.4. PCEP Notifications . . . . .	6
3.5. Relationship to other MIB modules . . . . .	6
3.6. Illustrative example . . . . .	6
4. Object Definitions . . . . .	7
4.1. PCE-PCEP-MIB . . . . .	7
5. Security Considerations . . . . .	48
6. IANA Considerations . . . . .	49
7. Acknowledgement . . . . .	49
8. References . . . . .	49
8.1. Normative References . . . . .	49
8.2. Informative References . . . . .	50
Appendix A. Contributors . . . . .	51
Appendix B. PCEP MIB Module Example . . . . .	51
B.1. Contents of PCEP MIB module at PCE2 . . . . .	51
B.2. Contents of PCEP MIB module at PCCb . . . . .	59

## 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 memo defines a portion of the Management Information Base (MIB) for use with network management protocols in the Internet community. In particular, it defines a MIB module that can be used to monitor PCEP interactions between a PCC and a PCE, or between two PCEs.

The scope of this document is to provide a MIB module for the PCEP base protocol defined in [RFC5440]. Extensions to the PCEP base protocol are beyond the scope for this document.

### 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 BCP 14, RFC 2119 [RFC2119].

### 1.2. Terminology

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 uses the term "PCEP entity" to refer to a local PCEP speaker, "peer" to refer to a remote PCEP speaker and "PCEP speaker" where it is not necessary to distinguish between local and remote.

## 2. The Internet-Standard Management Framework

For a detailed overview of the documents that describe the current Internet-Standard Management Framework, please refer to section 7 of RFC 3410 [RFC3410].

Managed objects are accessed via a virtual information store, termed the Management Information Base or MIB. MIB objects are generally accessed through the Simple Network Management Protocol (SNMP). Objects in the MIB are defined using the mechanisms defined in the Structure of Management Information (SMI). This memo specifies a MIB module that is compliant to the SMIV2, which is described in STD 58, RFC 2578 [RFC2578], STD 58, RFC 2579 [RFC2579], and STD 58, RFC 2580 [RFC2580].

### 3. PCEP MIB Module Architecture

The PCEP MIB module contains the following information:

- a. PCE and PCC local entity status (see `pcePcepEntityTable`).
- b. PCEP peer information (see `pcePcepPeerTable`).
- c. PCEP session information (see `pcePcepSessTable`).
- d. Notifications to indicate PCEP session changes.

The PCEP MIB module is limited to "read-only" access except for `pcePcepNotificationsMaxRate`, which is used to throttle the rate at which the implementation generates notifications.

#### 3.1. `pcePcepEntityTable`

The PCEP MIB 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, including the following.

- o A physical router, which is partitioned into multiple virtual routers, each with its own PCC.
- o A PCE device which front-ends a cluster of compute resources, each with a different set of capabilities that are accessed via different IP addresses.

The `pcePcepEntityTable` contains one row for each local PCEP entity. Each row is read-only and contains current status information plus the PCEP entity's running configuration.

The pcePcepEntityTable is indexed by pcePcepEntityIndex, which also acts as the primary index for the other tables in this MIB module.

### 3.2. pcePcepPeerTable

The pcePcepPeerTable contains one row for each peer that the local PCEP entity knows about. Each row is read-only and contains information to identify the peer, the running configuration relating to that peer and statistics that track the messages exchanged with that peer and its response times.

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.

The pcePcepPeerTable is indexed first by pcePcepEntityIndex, then by pcePcepPeerAddrType and pcePcepPeerAddr. This indexing structure allows each local PCEP entity to report its own set of peers.

Since PCEP sessions can be ephemeral, the pcePcepPeerTable tracks a peer even when no PCEP session currently exists to that peer. The statistics contained in pcePcepPeerTable 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 a row from the pcePcepPeerTable if and only if no PCEP session exists to the corresponding peer.

### 3.3. pcePcepSessTable

The pcePcepSessTable contains one row for each PCEP session that the PCEP entity (PCE or PCC) is currently participating in. Each row is read-only and contains the running configuration that is applied to the session, plus identifiers and statistics for the session.

The statistics in pcePcepSessTable are semantically different from those in pcePcepPeerTable since the former apply to the current session only, whereas the latter are the aggregate for all sessions that have existed to that peer.

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

The pcePcepSessTable is indexed first by pcePcepEntityIndex, then by pcePcepPeerAddrType and pcePcepPeerAddr, and finally by pcePcepSessInitiator. This indexing structure allows each local PCEP entity to report its own set of active sessions. The pcePcepSessInitiator index allows two rows to exist transiently for a given peer, as discussed above.

### 3.4. PCEP Notifications

The PCEP MIB module contains notifications for the following conditions.

- a. pcePcepSessUp: PCEP Session has gone up.
- b. pcePcepSessDown: PCEP Session has gone down.
- c. pcePcepSessLocalOverload: Local PCEP entity has sent an overload PCNtf on this session.
- d. pcePcepSessLocalOverloadClear: Local PCEP entity has sent an overload-cleared PCNtf on this session.
- e. pcePcepSessPeerOverload: Peer has sent an overload PCNtf on this session.
- f. pcePcepSessPeerOverloadClear: Peer has sent an overload-cleared PCNtf on this session.

### 3.5. Relationship to other MIB modules

The PCEP MIB module imports the following textual conventions from the INET-ADDRESS-MIB defined in RFC 4001 [RFC4001]:

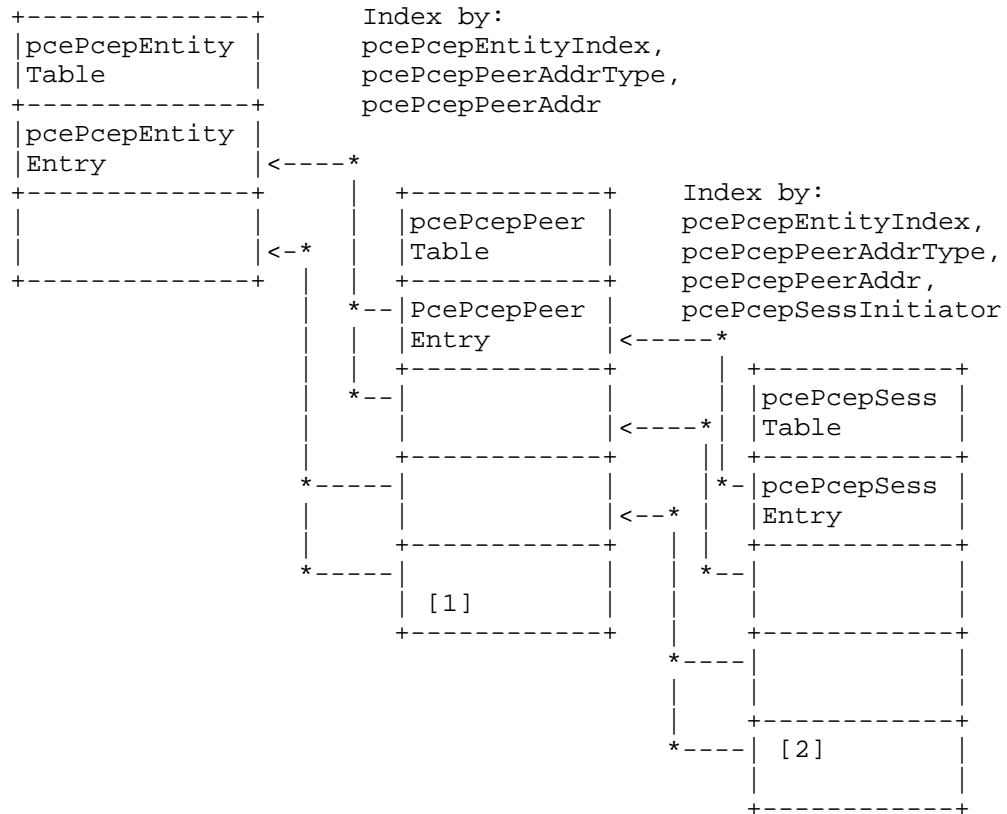
- o InetAddressType
- o InetAddress

PCEP relies on existing protocols which have specialized MIB objects to monitor their own activities. Consequently this document considers that the monitoring of underlying protocols is out of scope of the PCEP MIB module.

### 3.6. Illustrative example

The following diagram illustrates the relationships between the pcePcepEntityTable, pcePcepPeerTable and pcePcepSessTable.

Index by:  
pcePcepEntityIndex



[1]: A peer entry with no current session

[2]: Two sessions exist during a window in session initialization.

#### 4. Object Definitions

##### 4.1. PCE-PCEP-MIB

PCE-PCEP-MIB DEFINITIONS ::= BEGIN

IMPORTS

MODULE-IDENTITY,  
OBJECT-TYPE,  
mib-2,  
NOTIFICATION-TYPE,  
Unsigned32,

```
Counter32
    FROM SNMPv2-SMI                -- RFC 2578
TruthValue,
TimeStamp
    FROM SNMPv2-TC                -- RFC 2579
MODULE-COMPLIANCE,
OBJECT-GROUP,
NOTIFICATION-GROUP
    FROM SNMPv2-CONF              -- RFC 2580
InetAddressType,
InetAddress
    FROM INET-ADDRESS-MIB;        -- RFC 4001

pcePcepMIB MODULE-IDENTITY
    LAST-UPDATED
        "201410241200Z" -- 24 October 2014
    ORGANIZATION
        "IETF Path Computation Element (PCE) Working Group"
    CONTACT-INFO
        "Email: pce@ietf.org
        WG charter:
            http://www.ietf.org/html.charters/pce-charter.html"

    DESCRIPTION
        "This MIB module defines a collection of objects for managing
        Path Computation Element communications Protocol (PCEP).

        Copyright (C) The IETF Trust (2014).  This version of this
        MIB module is part of RFC YYYY; see the RFC itself for full
        legal notices."
-- RFC Ed.: replace YYYY with actual RFC number & remove this note
    REVISION
        "201410241200Z" -- 24 October 2014
    DESCRIPTION
        "Initial version, published as RFC YYYY."
-- RFC Ed.: replace YYYY with actual RFC number & remove this note
        ::= { mib-2 XXX }
-- RFC Ed.: replace XXX with IANA-assigned number & remove this note

pcePcepNotifications OBJECT IDENTIFIER ::= { pcePcepMIB 0 }
pcePcepObjects        OBJECT IDENTIFIER ::= { pcePcepMIB 1 }
pcePcepConformance    OBJECT IDENTIFIER ::= { pcePcepMIB 2 }

--
-- PCEP Entity Objects
--

pcePcepEntityTable OBJECT-TYPE
```

```

SYNTAX          SEQUENCE OF PcePcepEntityEntry
MAX-ACCESS      not-accessible
STATUS          current
DESCRIPTION
    "This table contains information about local PCEP entities.
    The entries in this table are read-only."
 ::= { pcePcepObjects 1 }

```

```

pcePcepEntityEntry OBJECT-TYPE
    SYNTAX          PcePcepEntityEntry
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION
        "This entry represents a local PCEP entity."
    INDEX           { pcePcepEntityIndex }
    ::= { pcePcepEntityTable 1 }

```

```

PcePcepEntityEntry ::= SEQUENCE {
    pcePcepEntityIndex          Unsigned32,
    pcePcepEntityAdminStatus    INTEGER,
    pcePcepEntityOperStatus     INTEGER,
    pcePcepEntityAddrType       InetAddressType,
    pcePcepEntityAddr           InetAddress,
    pcePcepEntityConnectTimer   Unsigned32,
    pcePcepEntityConnectMaxRetry Unsigned32,
    pcePcepEntityInitBackoffTimer Unsigned32,
    pcePcepEntityMaxBackoffTimer Unsigned32,
    pcePcepEntityOpenWaitTimer  Unsigned32,
    pcePcepEntityKeepWaitTimer  Unsigned32,
    pcePcepEntityKeepAliveTimer Unsigned32,
    pcePcepEntityDeadTimer      Unsigned32,
    pcePcepEntityAllowNegotiation TruthValue,
    pcePcepEntityMaxKeepAliveTimer Unsigned32,
    pcePcepEntityMaxDeadTimer   Unsigned32,
    pcePcepEntityMinKeepAliveTimer Unsigned32,
    pcePcepEntityMinDeadTimer   Unsigned32,
    pcePcepEntitySyncTimer      Unsigned32,
    pcePcepEntityRequestTimer   Unsigned32,
    pcePcepEntityMaxSessions    Unsigned32,
    pcePcepEntityMaxUnknownReqs Unsigned32,
    pcePcepEntityMaxUnknownMsgs Unsigned32
}

```

```

pcePcepEntityIndex OBJECT-TYPE
    SYNTAX          Unsigned32
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION

```

"This index is used to uniquely identify the PCEP entity."  
 ::= { pcePcepEntityEntry 1 }

pcePcepEntityAdminStatus OBJECT-TYPE

SYNTAX INTEGER {  
 adminStatusUp(1),  
 adminStatusDown(2)  
 }

MAX-ACCESS read-only

STATUS current

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 pcePcepEntityOperStatus represents the current status of an attempt to reach this desired status."

::= { pcePcepEntityEntry 2 }

pcePcepEntityOperStatus OBJECT-TYPE

SYNTAX INTEGER {  
 operStatusUp(1),  
 operStatusDown(2),  
 operStatusGoingUp(3),  
 operStatusGoingDown(4),  
 operStatusFailed(5),  
 operStatusFailedPerm(6)  
 }

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The operational status of the PCEP entity. Takes one of the following values.

- operStatusUp(1): the PCEP entity is active.
- operStatusDown(2): the PCEP entity is inactive.
- operStatusGoingUp(3): the PCEP entity is activating.
- operStatusGoingDown(4): the PCEP entity is deactivating.
- operStatusFailed(5): the PCEP entity has failed and will recover when possible.
- operStatusFailedPerm(6): the PCEP entity has failed and will not recover without operator intervention."

::= { pcePcepEntityEntry 3 }

pcePcepEntityAddrType OBJECT-TYPE

SYNTAX InetAddressType

MAX-ACCESS read-only

STATUS current

DESCRIPTION



"The type of the PCEP entity's Internet address. This object specifies how the value of the pcePcepEntityAddr object should be interpreted. Only values unknown(0), ipv4(1), or ipv6(2) are supported."  
 ::= { pcePcepEntityEntry 4 }

pcePcepEntityAddr OBJECT-TYPE

SYNTAX InetAddress

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The local Internet address of this PCEP entity. The type is given by pcePcepEntityAddrType.

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."

::= { pcePcepEntityEntry 5 }

pcePcepEntityConnectTimer OBJECT-TYPE

SYNTAX Unsigned32 (1..65535)

UNITS "seconds"

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The time 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."

::= { pcePcepEntityEntry 6 }

pcePcepEntityConnectMaxRetry OBJECT-TYPE

SYNTAX Unsigned32

MAX-ACCESS read-only

STATUS current

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.

When the session transitions to the idle state:

- pcePcepPeerSessionExists transitions to false(2)
- the associated PcePcepSessEntry is deleted

```
        - a backoff timer runs before the session is tried again."
 ::= { pcePcepEntityEntry 7 }

pcePcepEntityInitBackoffTimer OBJECT-TYPE
    SYNTAX      Unsigned32 (1..65535)
    UNITS       "seconds"
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The initial back-off time 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 pcePcepEntityMaxBackoffTimer."
 ::= { pcePcepEntityEntry 8 }

pcePcepEntityMaxBackoffTimer OBJECT-TYPE
    SYNTAX      Unsigned32
    UNITS       "seconds"
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The maximum back-off time 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."
 ::= { pcePcepEntityEntry 9 }

pcePcepEntityOpenWaitTimer OBJECT-TYPE
    SYNTAX      Unsigned32 (1..65535)
    UNITS       "seconds"
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The time 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
        PcePcepSessEntry."
 ::= { pcePcepEntityEntry 10 }

pcePcepEntityKeepWaitTimer OBJECT-TYPE
    SYNTAX      Unsigned32 (1..65535)
    UNITS       "seconds"
```

```
MAX-ACCESS    read-only
STATUS        current
DESCRIPTION
    "The time 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 PcePcepSessEntry."
 ::= { pcePcepEntityEntry 11 }

pcePcepEntityKeepAliveTimer OBJECT-TYPE
    SYNTAX      Unsigned32 (0..255)
    UNITS       "seconds"
    MAX-ACCESS  read-only
    STATUS      current
    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.  For
        the actually negotiated values (per-session), see
        pcePcepSessKeepaliveTimer and
        pcePcepSessPeerKeepaliveTimer."
    ::= { pcePcepEntityEntry 12 }

pcePcepEntityDeadTimer OBJECT-TYPE
    SYNTAX      Unsigned32 (0..255)
    UNITS       "seconds"
    MAX-ACCESS  read-only
    STATUS      current
    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."
    ::= { pcePcepEntityEntry 13 }

pcePcepEntityAllowNegotiation OBJECT-TYPE
    SYNTAX      TruthValue
```

```
MAX-ACCESS    read-only
STATUS        current
DESCRIPTION
    "Whether the PCEP entity will permit negotiation of session
    parameters."
::= { pcePcepEntityEntry 14 }

pcePcepEntityMaxKeepAliveTimer OBJECT-TYPE
SYNTAX        Unsigned32 (0..255)
UNITS         "seconds"
MAX-ACCESS    read-only
STATUS        current
DESCRIPTION
    "In PCEP session parameter negotiation, 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."
::= { pcePcepEntityEntry 15 }

pcePcepEntityMaxDeadTimer OBJECT-TYPE
SYNTAX        Unsigned32 (0..255)
UNITS         "seconds"
MAX-ACCESS    read-only
STATUS        current
DESCRIPTION
    "In PCEP session parameter negotiation, 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."
::= { pcePcepEntityEntry 16 }

pcePcepEntityMinKeepAliveTimer OBJECT-TYPE
SYNTAX        Unsigned32 (0..255)
UNITS         "seconds"
MAX-ACCESS    read-only
STATUS        current
DESCRIPTION
    "In PCEP session parameter negotiation, 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."
::= { pcePcepEntityEntry 17 }

pcePcepEntityMinDeadTimer OBJECT-TYPE
SYNTAX        Unsigned32 (0..255)
UNITS         "seconds"
MAX-ACCESS    read-only
```

```
STATUS      current
DESCRIPTION
    "In PCEP session parameter negotiation, 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."
 ::= { pcePcepEntityEntry 18 }

pcePcepEntitySyncTimer OBJECT-TYPE
SYNTAX      Unsigned32 (0..65535)
UNITS       "seconds"
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "The value of SyncTimer 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).

    A value of zero is returned if and only if the entity does
    not use the SyncTimer."
 ::= { pcePcepEntityEntry 19 }

pcePcepEntityRequestTimer OBJECT-TYPE
SYNTAX      Unsigned32 (1..65535)
UNITS       "seconds"
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "The maximum time that the PCEP entity will wait for a
    response to a PCReq message."
 ::= { pcePcepEntityEntry 20 }

pcePcepEntityMaxSessions OBJECT-TYPE
SYNTAX      Unsigned32
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "Maximum number of sessions involving this PCEP entity"
```

```
        that can exist at any time."
 ::= { pcePcepEntityEntry 21 }

pcePcepEntityMaxUnknownReqs OBJECT-TYPE
    SYNTAX      Unsigned32
    MAX-ACCESS   read-only
    STATUS       current
    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."
 ::= { pcePcepEntityEntry 22 }

pcePcepEntityMaxUnknownMsgs OBJECT-TYPE
    SYNTAX      Unsigned32
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "The maximum number of unknown messages that any session
        on this PCEP entity is willing to accept per minute before
        terminating the session."
 ::= { pcePcepEntityEntry 23 }

--
-- The PCEP Peer Table
--

pcePcepPeerTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF PcePcepPeerEntry
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
        "This table contains information about peers known by
        the local PCEP entity. The entries in this table are
        read-only.

        This table gives peer information that spans PCEP
        sessions. Information about current PCEP sessions can be
        found in the pcePcepSessTable table."
 ::= { pcePcepObjects 2 }
```

```

pcePcepPeerEntry OBJECT-TYPE
    SYNTAX      PcePcepPeerEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "Information about a single peer which spans all PCEP
        sessions to that peer."
    INDEX { pcePcepEntityIndex,
            pcePcepPeerAddrType,
            pcePcepPeerAddr }
    ::= { pcePcepPeerTable 1 }

PcePcepPeerEntry ::= SEQUENCE {
    pcePcepPeerAddrType      InetAddressType,
    pcePcepPeerAddr          InetAddress,
    pcePcepPeerRole          INTEGER,
    pcePcepPeerDiscontinuityTime  TimeStamp,
    pcePcepPeerInitiateSession  TruthValue,
    pcePcepPeerSessionExists    TruthValue,
    pcePcepPeerNumSessSetupOK    Counter32,
    pcePcepPeerNumSessSetupFail  Counter32,
    pcePcepPeerSessionUpTime     TimeStamp,
    pcePcepPeerSessionFailTime   TimeStamp,
    pcePcepPeerSessionFailUpTime TimeStamp,
    pcePcepPeerAvgRspTime        Unsigned32,
    pcePcepPeerLWMRspTime        Unsigned32,
    pcePcepPeerHWMRspTime        Unsigned32,
    pcePcepPeerNumPCReqSent       Counter32,
    pcePcepPeerNumPCReqRcvd       Counter32,
    pcePcepPeerNumPCRepSent       Counter32,
    pcePcepPeerNumPCRepRcvd       Counter32,
    pcePcepPeerNumPCErrSent       Counter32,
    pcePcepPeerNumPCErrRcvd       Counter32,
    pcePcepPeerNumPCNtfSent       Counter32,
    pcePcepPeerNumPCNtfRcvd       Counter32,
    pcePcepPeerNumKeepaliveSent   Counter32,
    pcePcepPeerNumKeepaliveRcvd   Counter32,
    pcePcepPeerNumUnknownRcvd     Counter32,
    pcePcepPeerNumCorruptRcvd     Counter32,
    pcePcepPeerNumReqSent         Counter32,
    pcePcepPeerNumSvecSent        Counter32,
    pcePcepPeerNumSvecReqSent     Counter32,
    pcePcepPeerNumReqSentPendRep  Counter32,
    pcePcepPeerNumReqSentEroRcvd  Counter32,
    pcePcepPeerNumReqSentNoPathRcvd Counter32,
    pcePcepPeerNumReqSentCancelRcvd Counter32,
    pcePcepPeerNumReqSentErrorRcvd Counter32,
    pcePcepPeerNumReqSentTimeout  Counter32,

```

```

    pcePcepPeerNumReqSentCancelSent      Counter32,
    pcePcepPeerNumReqSentClosed          Counter32,
    pcePcepPeerNumReqRcvd                Counter32,
    pcePcepPeerNumSvecRcvd               Counter32,
    pcePcepPeerNumSvecReqRcvd            Counter32,
    pcePcepPeerNumReqRcvdPendRep         Counter32,
    pcePcepPeerNumReqRcvdEroSent         Counter32,
    pcePcepPeerNumReqRcvdNoPathSent      Counter32,
    pcePcepPeerNumReqRcvdCancelSent      Counter32,
    pcePcepPeerNumReqRcvdErrorSent       Counter32,
    pcePcepPeerNumReqRcvdCancelRcvd      Counter32,
    pcePcepPeerNumReqRcvdClosed          Counter32,
    pcePcepPeerNumRepRcvdUnknown         Counter32,
    pcePcepPeerNumReqRcvdUnknown         Counter32
}

pcePcepPeerAddrType OBJECT-TYPE
    SYNTAX      InetAddressType
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "The type of the peer's Internet address.  This object
        specifies how the value of the pcePcepPeerAddr object should
        be interpreted.  Only values unknown(0), ipv4(1), or
        ipv6(2) are supported."
    ::= { pcePcepPeerEntry 1 }

pcePcepPeerAddr OBJECT-TYPE
    SYNTAX      InetAddress
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "The Internet address of the peer.  The type is given by
        pcePcepPeerAddrType."
    ::= { pcePcepPeerEntry 2 }

pcePcepPeerRole OBJECT-TYPE
    SYNTAX      INTEGER {
                    unknown(0),
                    pcc(1),
                    pce(2),
                    pccAndPce(3)
                }
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The role that this peer took the last time a session was
        established.  Takes one of the following values."

```



- unknown(0): this peer's role is not known.
- pcc(1): this peer is a Path Computation Client (PCC).
- pce(2): this peer is a Path Computation Server (PCE).
- pccAndPce(3): this peer is both a PCC and a PCE."

```
::= { pcePcepPeerEntry 3 }
```

pcePcepPeerDiscontinuityTime OBJECT-TYPE

SYNTAX           TimeStamp

MAX-ACCESS   read-only

STATUS       current

DESCRIPTION

"The value of sysUpTime at the time that the information and statistics in this row were last reset."

```
::= { pcePcepPeerEntry 4 }
```

pcePcepPeerInitiateSession OBJECT-TYPE

SYNTAX           TruthValue

MAX-ACCESS   read-only

STATUS       current

DESCRIPTION

"Indicates whether the local PCEP entity initiates sessions to this peer, or waits for the peer to initiate a session."

```
::= { pcePcepPeerEntry 5 }
```

pcePcepPeerSessionExists OBJECT-TYPE

SYNTAX           TruthValue

MAX-ACCESS   read-only

STATUS       current

DESCRIPTION

"Indicates whether a session with this peer currently exists."

```
::= { pcePcepPeerEntry 6 }
```

pcePcepPeerNumSessSetupOK OBJECT-TYPE

SYNTAX           Counter32

MAX-ACCESS   read-only

STATUS       current

DESCRIPTION

"The number of PCEP sessions successfully established with the peer, including any current session. This counter is incremented each time a session with this peer is successfully established."

```
::= { pcePcepPeerEntry 7 }
```

pcePcepPeerNumSessSetupFail OBJECT-TYPE

SYNTAX           Counter32

MAX-ACCESS   read-only

STATUS       current

## 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."

::= { pcePcepPeerEntry 8 }

## pcePcepPeerSessionUpTime OBJECT-TYPE

SYNTAX TimeStamp

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

"The value of sysUpTime the last time a session with this peer was successfully established."

If pcePcepPeerNumSessSetupOK is zero, then this object contains zero."

::= { pcePcepPeerEntry 9 }

## pcePcepPeerSessionFailTime OBJECT-TYPE

SYNTAX TimeStamp

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

"The value of sysUpTime the last time a session with this peer failed to be established."

If pcePcepPeerNumSessSetupFail is zero, then this object contains zero."

::= { pcePcepPeerEntry 10 }

## pcePcepPeerSessionFailUpTime OBJECT-TYPE

SYNTAX TimeStamp

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

"The value of sysUpTime the last time a session with this peer failed from active."

If pcePcepPeerNumSessSetupOK is zero, then this object contains zero."

::= { pcePcepPeerEntry 11 }

## pcePcepPeerAvgRspTime OBJECT-TYPE

SYNTAX Unsigned32

UNITS "milliseconds"

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

"The average response time for this peer.

If an average response time has not been calculated for this peer then this object has the value zero.

If pcePcepPeerRole is pcc then this field is meaningless and is set to zero."

::= { pcePcepPeerEntry 12 }

## pcePcepPeerLWMrspTime OBJECT-TYPE

SYNTAX Unsigned32

UNITS "milliseconds"

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

"The smallest (low-water mark) response time seen from this peer.

If no responses have been received from this peer then this object has the value zero.

If pcePcepPeerRole is pcc then this field is meaningless and is set to zero."

::= { pcePcepPeerEntry 13 }

## pcePcepPeerHWMrspTime OBJECT-TYPE

SYNTAX Unsigned32

UNITS "milliseconds"

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

"The greatest (high-water mark) response time seen from this peer.

If no responses have been received from this peer then this object has the value zero.

If pcePcepPeerRole is pcc then this field is meaningless and is set to zero."

::= { pcePcepPeerEntry 14 }

## pcePcepPeerNumPCReqSent OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

"The number of PCReq messages sent to this peer."

```
 ::= { pcePcepPeerEntry 15 }

pcePcepPeerNumPCReqRcvd OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The number of PCReq messages received from this peer."
    ::= { pcePcepPeerEntry 16 }

pcePcepPeerNumPCRepSent OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The number of PCRep messages sent to this peer."
    ::= { pcePcepPeerEntry 17 }

pcePcepPeerNumPCRepRcvd OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The number of PCRep messages received from this peer."
    ::= { pcePcepPeerEntry 18 }

pcePcepPeerNumPCErrSent OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The number of PCErr messages sent to this peer."
    ::= { pcePcepPeerEntry 19 }

pcePcepPeerNumPCErrRcvd OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The number of PCErr messages received from this peer."
    ::= { pcePcepPeerEntry 20 }

pcePcepPeerNumPCNtfSent OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The number of PCNtf messages sent to this peer."
```

```
 ::= { pcePcepPeerEntry 21 }

pcePcepPeerNumPCNtfRcvd OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "The number of PCNtf messages received from this peer."
    ::= { pcePcepPeerEntry 22 }

pcePcepPeerNumKeepaliveSent OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "The number of Keepalive messages sent to this peer."
    ::= { pcePcepPeerEntry 23 }

pcePcepPeerNumKeepaliveRcvd OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "The number of Keepalive messages received from this peer."
    ::= { pcePcepPeerEntry 24 }

pcePcepPeerNumUnknownRcvd OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "The number of unknown messages received from this peer."
    ::= { pcePcepPeerEntry 25 }

pcePcepPeerNumCorruptRcvd OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "The number of corrupted PCEP message received from this
        peer."
    ::= { pcePcepPeerEntry 26 }

pcePcepPeerNumReqSent OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
```

"The number of requests sent to this peer. A request corresponds 1:1 with an RP object in a PCReq message.

This might be greater than pcePcepPeerNumPCReqSent because multiple requests can be batched into a single PCReq message."

::= { pcePcepPeerEntry 27 }

pcePcepPeerNumSvecSent OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The number of SVEC objects sent to this peer in PCReq messages. An SVEC object represents a set of synchronized requests."

::= { pcePcepPeerEntry 28 }

pcePcepPeerNumSvecReqSent OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The number of requests sent to this peer that appeared in one or more SVEC objects."

::= { pcePcepPeerEntry 29 }

pcePcepPeerNumReqSentPendRep OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The number of requests that have been sent to this peer for which a response is still pending."

::= { pcePcepPeerEntry 30 }

pcePcepPeerNumReqSentEroRcvd OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The number of requests that have been sent to this peer for which a response with an ERO object was received. Such responses indicate that a path was successfully computed by the peer."

::= { pcePcepPeerEntry 31 }

pcePcepPeerNumReqSentNoPathRcvd OBJECT-TYPE

```
SYNTAX      Counter32
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "The number of requests that have been sent to this peer 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."
 ::= { pcePcepPeerEntry 32 }

pcePcepPeerNumReqSentCancelRcvd OBJECT-TYPE
SYNTAX      Counter32
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "The number of requests that were cancelled by the peer with
    a PCNtf message.

    This might be different than pcePcepPeerNumPCNtfRcvd because
    not all PCNtf messages are used to cancel requests, and a
    single PCNtf message can cancel multiple requests."
 ::= { pcePcepPeerEntry 33 }

pcePcepPeerNumReqSentErrorRcvd OBJECT-TYPE
SYNTAX      Counter32
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "The number of requests that were rejected by the peer with a
    PCErr message.

    This might be different than pcePcepPeerNumPCErrRcvd because
    not all PCErr messages are used to reject requests, and a
    single PCErr message can reject multiple requests."
 ::= { pcePcepPeerEntry 34 }

pcePcepPeerNumReqSentTimeout OBJECT-TYPE
SYNTAX      Counter32
MAX-ACCESS  read-only
STATUS      current
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."
 ::= { pcePcepPeerEntry 35 }

pcePcepPeerNumReqSentCancelSent OBJECT-TYPE
SYNTAX      Counter32
```

```
MAX-ACCESS    read-only
STATUS        current
DESCRIPTION
    "The number of requests that were sent to the peer and
    explicitly canceled by the local PCEP entity sending a
    PCNtf."
 ::= { pcePcepPeerEntry 36 }
```

```
pcePcepPeerNumReqSentClosed OBJECT-TYPE
SYNTAX        Counter32
MAX-ACCESS    read-only
STATUS        current
DESCRIPTION
    "The number of requests that were sent to the peer and
    implicitly canceled when the session they were sent over was
    closed."
 ::= { pcePcepPeerEntry 37 }
```

```
pcePcepPeerNumReqRcvd OBJECT-TYPE
SYNTAX        Counter32
MAX-ACCESS    read-only
STATUS        current
DESCRIPTION
    "The number of requests received from this peer.  A request
    corresponds 1:1 with an RP object in a PCReq message.

    This might be greater than pcePcepPeerNumPCReqRcvd because
    multiple requests can be batched into a single PCReq
    message."
 ::= { pcePcepPeerEntry 38 }
```

```
pcePcepPeerNumSvecRcvd OBJECT-TYPE
SYNTAX        Counter32
MAX-ACCESS    read-only
STATUS        current
DESCRIPTION
    "The number of SVEC objects received from this peer in PCReq
    messages.  An SVEC object represents a set of synchronized
    requests."
 ::= { pcePcepPeerEntry 39 }
```

```
pcePcepPeerNumSvecReqRcvd OBJECT-TYPE
SYNTAX        Counter32
MAX-ACCESS    read-only
STATUS        current
DESCRIPTION
    "The number of requests received from this peer that appeared
    in one or more SVEC objects."
```



```
::= { pcePcepPeerEntry 40 }

pcePcepPeerNumReqRcvdPendRep OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The number of requests that have been received from this
        peer for which a response is still pending."
    ::= { pcePcepPeerEntry 41 }

pcePcepPeerNumReqRcvdEroSent OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The number of requests that have been received from this
        peer for which a response with an ERO object was sent.  Such
        responses indicate that a path was successfully computed by
        the local PCEP entity."
    ::= { pcePcepPeerEntry 42 }

pcePcepPeerNumReqRcvdNoPathSent OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The number of requests that have been received from this
        peer 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."
    ::= { pcePcepPeerEntry 43 }

pcePcepPeerNumReqRcvdCancelSent OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The number of requests received from this peer that were
        cancelled by the local PCEP entity sending a PCNtf message.

        This might be different than pcePcepPeerNumPCNtfSent because
        not all PCNtf messages are used to cancel requests, and a
        single PCNtf message can cancel multiple requests."
    ::= { pcePcepPeerEntry 44 }

pcePcepPeerNumReqRcvdErrorSent OBJECT-TYPE
    SYNTAX      Counter32
```

MAX-ACCESS read-only  
STATUS current  
DESCRIPTION

"The number of requests received from this peer that were rejected by the local PCEP entity sending a PCErr message.

This might be different than pcePcepPeerNumPCErrSent because not all PCErr messages are used to reject requests, and a single PCErr message can reject multiple requests."

::= { pcePcepPeerEntry 45 }

pcePcepPeerNumReqRcvdCancelRcvd OBJECT-TYPE

SYNTAX Counter32  
MAX-ACCESS read-only  
STATUS current  
DESCRIPTION

"The number of requests that were received from the peer and explicitly canceled by the peer sending a PCNtf."

::= { pcePcepPeerEntry 46 }

pcePcepPeerNumReqRcvdClosed OBJECT-TYPE

SYNTAX Counter32  
MAX-ACCESS read-only  
STATUS current  
DESCRIPTION

"The number of requests that were received from the peer and implicitly canceled when the session they were received over was closed."

::= { pcePcepPeerEntry 47 }

pcePcepPeerNumRepRcvdUnknown OBJECT-TYPE

SYNTAX Counter32  
MAX-ACCESS read-only  
STATUS current  
DESCRIPTION

"The number of responses to unknown requests received from this peer. 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."

::= { pcePcepPeerEntry 48 }

pcePcepPeerNumReqRcvdUnknown OBJECT-TYPE

SYNTAX Counter32  
MAX-ACCESS read-only  
STATUS current  
DESCRIPTION

"The number of unknown requests that have been received from a peer. An unknown request is a request whose RP object

```

        contains a request ID of zero."
 ::= { pcePcepPeerEntry 49 }

--
-- The PCEP Sessions Table
--

pcePcepSessTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF PcePcepSessEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A table of PCEP sessions that involve the local PCEP
        entity.  Each entry in this table represents a single
        session.  The entries in this table are read-only.

        An entry appears in this table when the corresponding PCEP
        session transitions out of idle state.  If the PCEP session
        transitions back into idle state then the corresponding
        entry in this table is removed."
 ::= { pcePcepObjects 3 }

pcePcepSessEntry OBJECT-TYPE
    SYNTAX      PcePcepSessEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    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, autodiscovery of a peer, or an incoming TCP
        connection."
    INDEX { pcePcepEntityIndex,
            pcePcepPeerAddrType,
            pcePcepPeerAddr,
            pcePcepSessInitiator }
 ::= { pcePcepSessTable 1 }

PcePcepSessEntry ::= SEQUENCE {
    pcePcepSessInitiator          INTEGER,
    pcePcepSessStateLastChange   TimeStamp,
    pcePcepSessState              INTEGER,
    pcePcepSessConnectRetry      Counter32,
    pcePcepSessLocalID           Unsigned32,
    pcePcepSessRemoteID          Unsigned32,
    pcePcepSessKeepaliveTimer    Unsigned32,

```

pcePcepSessPeerKeepaliveTimer	Unsigned32,
pcePcepSessDeadTimer	Unsigned32,
pcePcepSessPeerDeadTimer	Unsigned32,
pcePcepSessKAHoldTimeRem	Unsigned32,
pcePcepSessOverloaded	TruthValue,
pcePcepSessOverloadTime	Unsigned32,
pcePcepSessPeerOverloaded	TruthValue,
pcePcepSessPeerOverloadTime	Unsigned32,
pcePcepSessDiscontinuityTime	TimeStamp,
pcePcepSessAvgRspTime	Unsigned32,
pcePcepSessLWMRspTime	Unsigned32,
pcePcepSessHWMRspTime	Unsigned32,
pcePcepSessNumPCReqSent	Counter32,
pcePcepSessNumPCReqRcvd	Counter32,
pcePcepSessNumPCRepSent	Counter32,
pcePcepSessNumPCRepRcvd	Counter32,
pcePcepSessNumPCErrSent	Counter32,
pcePcepSessNumPCErrRcvd	Counter32,
pcePcepSessNumPCNtfSent	Counter32,
pcePcepSessNumPCNtfRcvd	Counter32,
pcePcepSessNumKeepaliveSent	Counter32,
pcePcepSessNumKeepaliveRcvd	Counter32,
pcePcepSessNumUnknownRcvd	Counter32,
pcePcepSessNumCorruptRcvd	Counter32,
pcePcepSessNumReqSent	Counter32,
pcePcepSessNumSvecSent	Counter32,
pcePcepSessNumSvecReqSent	Counter32,
pcePcepSessNumReqSentPendRep	Counter32,
pcePcepSessNumReqSentEroRcvd	Counter32,
pcePcepSessNumReqSentNoPathRcvd	Counter32,
pcePcepSessNumReqSentCancelRcvd	Counter32,
pcePcepSessNumReqSentErrorRcvd	Counter32,
pcePcepSessNumReqSentTimeout	Counter32,
pcePcepSessNumReqSentCancelSent	Counter32,
pcePcepSessNumReqRcvd	Counter32,
pcePcepSessNumSvecRcvd	Counter32,
pcePcepSessNumSvecReqRcvd	Counter32,
pcePcepSessNumReqRcvdPendRep	Counter32,
pcePcepSessNumReqRcvdEroSent	Counter32,
pcePcepSessNumReqRcvdNoPathSent	Counter32,
pcePcepSessNumReqRcvdCancelSent	Counter32,
pcePcepSessNumReqRcvdErrorSent	Counter32,
pcePcepSessNumReqRcvdCancelRcvd	Counter32,
pcePcepSessNumRepRcvdUnknown	Counter32,
pcePcepSessNumReqRcvdUnknown	Counter32

}

pcePcepSessInitiator OBJECT-TYPE

```
SYNTAX      INTEGER {
                local(1),
                remote(2)
            }
MAX-ACCESS   not-accessible
STATUS       current
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 an
    index for the pcePcepSessTable."
 ::= { pcePcepSessEntry 1 }

pcePcepSessStateLastChange OBJECT-TYPE
    SYNTAX      TimeStamp
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "The value of sysUpTime at the time this session entered its
        current state as denoted by the pcePcepSessState object."
    ::= { pcePcepSessEntry 2 }

pcePcepSessState OBJECT-TYPE
    SYNTAX      INTEGER {
                tcpPending(1),
                openWait(2),
                keepWait(3),
                sessionUp(4)
            }
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "The current state of the session.

        The set of possible states excludes the idle state since
        entries do not exist in this table in the idle state."
    ::= { pcePcepSessEntry 3 }

pcePcepSessConnectRetry OBJECT-TYPE
    SYNTAX      Counter32
```

MAX-ACCESS read-only  
STATUS current  
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 pcePcepEntityConnectMaxRetry."  
 ::= { pcePcepSessEntry 4 }

pcePcepSessLocalID OBJECT-TYPE  
SYNTAX Unsigned32 (0..255)  
MAX-ACCESS read-only  
STATUS current  
DESCRIPTION  
    "The value of the PCEP session ID used by the local PCEP  
    entity in the Open message for this session.  
  
    If pcePcepSessState is tcpPending 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."  
 ::= { pcePcepSessEntry 5 }

pcePcepSessRemoteID OBJECT-TYPE  
SYNTAX Unsigned32 (0..255)  
MAX-ACCESS read-only  
STATUS current  
DESCRIPTION  
    "The value of the PCEP session ID used by the peer in its  
    Open message for this session.  
  
    If pcePcepSessState is tcpPending or openWait then this  
    field is not used and MUST be set to zero."  
 ::= { pcePcepSessEntry 6 }

pcePcepSessKeepaliveTimer OBJECT-TYPE  
SYNTAX Unsigned32 (0..255)  
UNITS "seconds"  
MAX-ACCESS read-only  
STATUS current  
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.  
  
    This field is used if and only if pcePcepSessState is  
    sessionUp. Otherwise, it is not used and MUST be set to  
    zero."

```
::= { pcePcepSessEntry 7 }
```

pcePcepSessPeerKeepaliveTimer OBJECT-TYPE

SYNTAX Unsigned32 (0..255)

UNITS "seconds"

MAX-ACCESS read-only

STATUS current

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.

This field is used if and only if pcePcepSessState is sessionUp. Otherwise, it is not used and MUST be set to zero."

```
::= { pcePcepSessEntry 8 }
```

pcePcepSessDeadTimer OBJECT-TYPE

SYNTAX Unsigned32 (0..255)

UNITS "seconds"

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The DeadTimer interval for this PCEP session."

```
::= { pcePcepSessEntry 9 }
```

pcePcepSessPeerDeadTimer OBJECT-TYPE

SYNTAX Unsigned32 (0..255)

UNITS "seconds"

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The peer's DeadTimer interval for this PCEP session.

If pcePcepSessState is tcpPending or openWait then this field is not used and MUST be set to zero."

```
::= { pcePcepSessEntry 10 }
```

pcePcepSessKAHoldTimeRem OBJECT-TYPE

SYNTAX Unsigned32 (0..255)

UNITS "seconds"

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The keep alive hold time remaining for this session.

If pcePcepSessState is tcpPending or openWait then this field is not used and MUST be set to zero."

```
::= { pcePcepSessEntry 11 }

pcePcepSessOverloaded OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-only
    STATUS      current
    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."
    ::= { pcePcepSessEntry 12 }

pcePcepSessOverloadTime OBJECT-TYPE
    SYNTAX      Unsigned32
    UNITS       "seconds"
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The interval of time that is remaining until the local PCEP
         entity will cease to be overloaded on this session.

         This field is only used if pcePcepSessOverloaded is set to
         true.  Otherwise, it is not used and MUST be set to zero."
    ::= { pcePcepSessEntry 13 }

pcePcepSessPeerOverloaded OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-only
    STATUS      current
    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."
    ::= { pcePcepSessEntry 14 }

pcePcepSessPeerOverloadTime OBJECT-TYPE
    SYNTAX      Unsigned32
    UNITS       "seconds"
    MAX-ACCESS  read-only
    STATUS      current
    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 field is set to
         zero.

         This field is only used if pcePcepSessPeerOverloaded is set
         to true.  Otherwise, it is not used and MUST be set to
```



```
        zero."
 ::= { pcePcepSessEntry 15 }

pcePcepSessDiscontinuityTime OBJECT-TYPE
    SYNTAX      TimeStamp
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "The value of sysUpTime at the time that the statistics in
        this row were last reset."
 ::= { pcePcepSessEntry 16 }

pcePcepSessAvgRspTime OBJECT-TYPE
    SYNTAX      Unsigned32
    UNITS        "milliseconds"
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "The average response time for this peer on this session.

        If an average response time has not been calculated for this
        peer then this object has the value zero."
 ::= { pcePcepSessEntry 17 }

pcePcepSessLWMRspTime OBJECT-TYPE
    SYNTAX      Unsigned32
    UNITS        "milliseconds"
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "The smallest (low-water mark) response time seen from this
        peer on this session.

        If no responses have been received from this peer then this
        object has the value zero."
 ::= { pcePcepSessEntry 18 }

pcePcepSessHWMRspTime OBJECT-TYPE
    SYNTAX      Unsigned32
    UNITS        "milliseconds"
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "The greatest (high-water mark) response time seen from this
        peer on this session.

        If no responses have been received from this peer then this
        object has the value zero."
```

```
 ::= { pcePcepSessEntry 19 }

pcePcepSessNumPCReqSent OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "The number of PCReq messages sent on this session."
    ::= { pcePcepSessEntry 20 }

pcePcepSessNumPCReqRcvd OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "The number of PCReq messages received on this session."
    ::= { pcePcepSessEntry 21 }

pcePcepSessNumPCRepSent OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "The number of PCRep messages sent on this session."
    ::= { pcePcepSessEntry 22 }

pcePcepSessNumPCRepRcvd OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "The number of PCRep messages received on this session."
    ::= { pcePcepSessEntry 23 }

pcePcepSessNumPCErrSent OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "The number of PCErr messages sent on this session."
    ::= { pcePcepSessEntry 24 }

pcePcepSessNumPCErrRcvd OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "The number of PCErr messages received on this session."
```

```
 ::= { pcePcepSessEntry 25 }

pcePcepSessNumPCNtfSent OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "The number of PCNtf messages sent on this session."
    ::= { pcePcepSessEntry 26 }

pcePcepSessNumPCNtfRcvd OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "The number of PCNtf messages received on this session."
    ::= { pcePcepSessEntry 27 }

pcePcepSessNumKeepaliveSent OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "The number of Keepalive messages sent on this session."
    ::= { pcePcepSessEntry 28 }

pcePcepSessNumKeepaliveRcvd OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "The number of Keepalive messages received on this session."
    ::= { pcePcepSessEntry 29 }

pcePcepSessNumUnknownRcvd OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "The number of unknown messages received on this session."
    ::= { pcePcepSessEntry 30 }

pcePcepSessNumCorruptRcvd OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "The number of corrupted PCEP message received on this
```

```
        session."
 ::= { pcePcepSessEntry 31 }

pcePcepSessNumReqSent OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "The number of requests sent on this session.  A request
        corresponds 1:1 with an RP object in a PCReq message.

        This might be greater than pcePcepSessNumPCReqSent because
        multiple requests can be batched into a single PCReq
        message."
 ::= { pcePcepSessEntry 32 }

pcePcepSessNumSvecSent OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "The number of SVEC objects sent on this session in PCReq
        messages.  An SVEC object represents a set of synchronized
        requests."
 ::= { pcePcepSessEntry 33 }

pcePcepSessNumSvecReqSent OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "The number of requests sent on this session that appeared in
        one or more SVEC objects."
 ::= { pcePcepSessEntry 34 }

pcePcepSessNumReqSentPendRep OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "The number of requests that have been sent on this session
        for which a response is still pending."
 ::= { pcePcepSessEntry 35 }

pcePcepSessNumReqSentEroRcvd OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS   read-only
    STATUS       current
```

## DESCRIPTION

"The number of successful responses received on this session. A response corresponds 1:1 with an RP object in a PCRep message. A successful response is a response for which an ERO was successfully computed."

::= { pcePcepSessEntry 36 }

## pcePcepSessNumReqSentNoPathRcvd OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

"The number of unsuccessful responses received on this session. A response corresponds 1:1 with an RP object in a PCRep message. An unsuccessful response is a response with a NO-PATH object."

::= { pcePcepSessEntry 37 }

## pcePcepSessNumReqSentCancelRcvd OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

"The number of requests sent on this session that were cancelled by the peer with a PCNtf message."

This might be different than pcePcepSessNumPCNtfRcvd because not all PCNtf messages are used to cancel requests, and a single PCNtf message can cancel multiple requests."

::= { pcePcepSessEntry 38 }

## pcePcepSessNumReqSentErrorRcvd OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

"The number of requests sent on this session that were rejected by the peer with a PCErr message."

This might be different than pcePcepSessNumPCErrRcvd because not all PCErr messages are used to reject requests, and a single PCErr message can reject multiple requests."

::= { pcePcepSessEntry 39 }

## pcePcepSessNumReqSentTimeout OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

"The number of requests sent on this session that have been sent to a peer and have been abandoned because the peer has taken too long to respond to them."

::= { pcePcepSessEntry 40 }

## pcePcepSessNumReqSentCancelSent OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

"The number of requests sent on this session that were sent to the peer and explicitly canceled by the local PCEP entity sending a PCNtf."

::= { pcePcepSessEntry 41 }

## pcePcepSessNumReqRcvd OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

"The number of requests received on this session. A request corresponds 1:1 with an RP object in a PCReq message."

This might be greater than pcePcepSessNumPCReqRcvd because multiple requests can be batched into a single PCReq message."

::= { pcePcepSessEntry 42 }

## pcePcepSessNumSvecRcvd OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

"The number of SVEC objects received on this session in PCReq messages. An SVEC object represents a set of synchronized requests."

::= { pcePcepSessEntry 43 }

## pcePcepSessNumSvecReqRcvd OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

"The number of requests received on this session that appeared in one or more SVEC objects."

::= { pcePcepSessEntry 44 }

```
pcePcepSessNumReqRcvdPendRep OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "The number of requests that have been received on this
        session for which a response is still pending."
    ::= { pcePcepSessEntry 45 }

pcePcepSessNumReqRcvdEroSent OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "The number of successful responses sent on this session. A
        response corresponds 1:1 with an RP object in a PCRep
        message. A successful response is a response for which an
        ERO was successfully computed."
    ::= { pcePcepSessEntry 46 }

pcePcepSessNumReqRcvdNoPathSent OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "The number of unsuccessful responses sent on this session.
        A response corresponds 1:1 with an RP object in a PCRep
        message. An unsuccessful response is a response with a
        NO-PATH object."
    ::= { pcePcepSessEntry 47 }

pcePcepSessNumReqRcvdCancelSent OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "The number of requests received on this session that were
        cancelled by the local PCEP entity sending a PCNtf message.

        This might be different than pcePcepSessNumPCNtfSent because
        not all PCNtf messages are used to cancel requests, and a
        single PCNtf message can cancel multiple requests."
    ::= { pcePcepSessEntry 48 }

pcePcepSessNumReqRcvdErrorSent OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS   read-only
    STATUS       current
```

## DESCRIPTION

"The number of requests received on this session that were rejected by the local PCEP entity sending a PCErr message.

This might be different than pcePcepSessNumPCErrSent because not all PCErr messages are used to reject requests, and a single PCErr message can reject multiple requests."

::= { pcePcepSessEntry 49 }

## pcePcepSessNumReqRcvdCancelRcvd OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

"The number of requests that were received on this session and explicitly canceled by the peer sending a PCNtf."

::= { pcePcepSessEntry 50 }

## pcePcepSessNumRepRcvdUnknown OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

"The number of responses to unknown requests received on this session. 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."

::= { pcePcepSessEntry 51 }

## pcePcepSessNumReqRcvdUnknown OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

"The number of unknown requests that have been received on this session. An unknown request is a request whose RP object contains a request ID of zero."

::= { pcePcepSessEntry 52 }

---

--- Notifications Configuration

---

## pcePcepNotificationsMaxRate OBJECT-TYPE

SYNTAX Unsigned32

MAX-ACCESS read-write

STATUS current

## 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)."
```

```
::= { pcePcepObjects 4 }
```

```
---
```

```
--- Notifications
```

```
---
```

```
pcePcepSessUp NOTIFICATION-TYPE
  OBJECTS      {
                pcePcepSessState,
                pcePcepSessStateLastChange
              }
  STATUS       current
  DESCRIPTION   "This notification is sent when the value of
                'pcePcepSessState' enters the 'sessionUp' state."
  ::= { pcePcepNotifications 1 }
```

```
pcePcepSessDown NOTIFICATION-TYPE
  OBJECTS      {
                pcePcepSessState,
                pcePcepSessStateLastChange
              }
  STATUS       current
  DESCRIPTION   "This notification is sent when the value of
                'pcePcepSessState' leaves the 'sessionUp' state."
  ::= { pcePcepNotifications 2 }
```

```
pcePcepSessLocalOverload NOTIFICATION-TYPE
  OBJECTS      {
                pcePcepSessOverloaded,
                pcePcepSessOverloadTime
              }
  STATUS       current
  DESCRIPTION   "This notification is sent when the local PCEP entity enters
                overload state for a peer."
  ::= { pcePcepNotifications 3 }
```

```
pcePcepSessLocalOverloadClear NOTIFICATION-TYPE
  OBJECTS      {
```

```

        pcePcepSessOverloaded
    }
    STATUS current
    DESCRIPTION
        "This notification is sent when the local PCEP entity leaves
        overload state for a peer."
    ::= { pcePcepNotifications 4 }

pcePcepSessPeerOverload NOTIFICATION-TYPE
    OBJECTS {
        pcePcepSessPeerOverloaded,
        pcePcepSessPeerOverloadTime
    }
    STATUS current
    DESCRIPTION
        "This notification is sent when a peer enters overload
        state."
    ::= { pcePcepNotifications 5 }

pcePcepSessPeerOverloadClear NOTIFICATION-TYPE
    OBJECTS {
        pcePcepSessPeerOverloaded
    }
    STATUS current
    DESCRIPTION
        "This notification is sent when a peer leaves overload
        state."
    ::= { pcePcepNotifications 6 }

--
-- Module Conformance Statement
--

pcePcepCompliances
    OBJECT IDENTIFIER ::= { pcePcepConformance 1 }

pcePcepGroups
    OBJECT IDENTIFIER ::= { pcePcepConformance 2 }

--
-- Read-Only Compliance
--

pcePcepModuleReadOnlyCompliance MODULE-COMPLIANCE
    STATUS current
    DESCRIPTION
        "The Module is implemented with support for read-only. In
        other words, only monitoring is available by implementing
```

```
        this MODULE-COMPLIANCE."

MODULE -- this module
    MANDATORY-GROUPS {
        pcePcepGeneralGroup,
        pcePcepNotificationsGroup
    }

OBJECT      pcePcepEntityAddrType
SYNTAX      InetAddressType { unknown(0), ipv4(1), ipv6(2) }
DESCRIPTION "Only unknown(0), ipv4(1) and ipv6(2) support
            is required."

OBJECT      pcePcepPeerAddrType
SYNTAX      InetAddressType { unknown(0), ipv4(1), ipv6(2) }
DESCRIPTION "Only unknown(0), ipv4(1) and ipv6(2) support
            is required."

::= { pcePcepCompliances 1 }

-- units of conformance

pcePcepGeneralGroup OBJECT-GROUP
    OBJECTS { pcePcepEntityAdminStatus,
              pcePcepEntityOperStatus,
              pcePcepEntityAddrType,
              pcePcepEntityAddr,
              pcePcepEntityConnectTimer,
              pcePcepEntityConnectMaxRetry,
              pcePcepEntityInitBackoffTimer,
              pcePcepEntityMaxBackoffTimer,
              pcePcepEntityOpenWaitTimer,
              pcePcepEntityKeepWaitTimer,
              pcePcepEntityKeepAliveTimer,
              pcePcepEntityDeadTimer,
              pcePcepEntityAllowNegotiation,
              pcePcepEntityMaxKeepAliveTimer,
              pcePcepEntityMaxDeadTimer,
              pcePcepEntityMinKeepAliveTimer,
              pcePcepEntityMinDeadTimer,
              pcePcepEntitySyncTimer,
              pcePcepEntityRequestTimer,
              pcePcepEntityMaxSessions,
              pcePcepEntityMaxUnknownReqs,
              pcePcepEntityMaxUnknownMsgs,
              pcePcepPeerRole,
              pcePcepPeerDiscontinuityTime,
              pcePcepPeerInitiateSession,
```

pcePcepPeerSessionExists,  
pcePcepPeerNumSessSetupOK,  
pcePcepPeerNumSessSetupFail,  
pcePcepPeerSessionUpTime,  
pcePcepPeerSessionFailTime,  
pcePcepPeerSessionFailUpTime,  
pcePcepPeerAvgRspTime,  
pcePcepPeerLWMRspTime,  
pcePcepPeerHWMRspTime,  
pcePcepPeerNumPCReqSent,  
pcePcepPeerNumPCReqRcvd,  
pcePcepPeerNumPCRepSent,  
pcePcepPeerNumPCRepRcvd,  
pcePcepPeerNumPCErrSent,  
pcePcepPeerNumPCErrRcvd,  
pcePcepPeerNumPCNtfSent,  
pcePcepPeerNumPCNtfRcvd,  
pcePcepPeerNumKeepaliveSent,  
pcePcepPeerNumKeepaliveRcvd,  
pcePcepPeerNumUnknownRcvd,  
pcePcepPeerNumCorruptRcvd,  
pcePcepPeerNumReqSent,  
pcePcepPeerNumSvecSent,  
pcePcepPeerNumSvecReqSent,  
pcePcepPeerNumReqSentPendRep,  
pcePcepPeerNumReqSentEroRcvd,  
pcePcepPeerNumReqSentNoPathRcvd,  
pcePcepPeerNumReqSentCancelRcvd,  
pcePcepPeerNumReqSentErrorRcvd,  
pcePcepPeerNumReqSentTimeout,  
pcePcepPeerNumReqSentCancelSent,  
pcePcepPeerNumReqSentClosed,  
pcePcepPeerNumReqRcvd,  
pcePcepPeerNumSvecRcvd,  
pcePcepPeerNumSvecReqRcvd,  
pcePcepPeerNumReqRcvdPendRep,  
pcePcepPeerNumReqRcvdEroSent,  
pcePcepPeerNumReqRcvdNoPathSent,  
pcePcepPeerNumReqRcvdCancelSent,  
pcePcepPeerNumReqRcvdErrorSent,  
pcePcepPeerNumReqRcvdCancelRcvd,  
pcePcepPeerNumReqRcvdClosed,  
pcePcepPeerNumRepRcvdUnknown,  
pcePcepPeerNumReqRcvdUnknown,  
pcePcepSessStateLastChange,  
pcePcepSessState,  
pcePcepSessConnectRetry,  
pcePcepSessLocalID,

pcePcepSessRemoteID,  
pcePcepSessKeepaliveTimer,  
pcePcepSessPeerKeepaliveTimer,  
pcePcepSessDeadTimer,  
pcePcepSessPeerDeadTimer,  
pcePcepSessKAHoldTimeRem,  
pcePcepSessOverloaded,  
pcePcepSessOverloadTime,  
pcePcepSessPeerOverloaded,  
pcePcepSessPeerOverloadTime,  
pcePcepSessDiscontinuityTime,  
pcePcepSessAvgRspTime,  
pcePcepSessLWMRspTime,  
pcePcepSessHWMRspTime,  
pcePcepSessNumPCReqSent,  
pcePcepSessNumPCReqRcvd,  
pcePcepSessNumPCRepSent,  
pcePcepSessNumPCRepRcvd,  
pcePcepSessNumPCErrSent,  
pcePcepSessNumPCErrRcvd,  
pcePcepSessNumPCNtfSent,  
pcePcepSessNumPCNtfRcvd,  
pcePcepSessNumKeepaliveSent,  
pcePcepSessNumKeepaliveRcvd,  
pcePcepSessNumUnknownRcvd,  
pcePcepSessNumCorruptRcvd,  
pcePcepSessNumReqSent,  
pcePcepSessNumSvecSent,  
pcePcepSessNumSvecReqSent,  
pcePcepSessNumReqSentPendRep,  
pcePcepSessNumReqSentEroRcvd,  
pcePcepSessNumReqSentNoPathRcvd,  
pcePcepSessNumReqSentCancelRcvd,  
pcePcepSessNumReqSentErrorRcvd,  
pcePcepSessNumReqSentTimeout,  
pcePcepSessNumReqSentCancelSent,  
pcePcepSessNumReqRcvd,  
pcePcepSessNumSvecRcvd,  
pcePcepSessNumSvecReqRcvd,  
pcePcepSessNumReqRcvdPendRep,  
pcePcepSessNumReqRcvdEroSent,  
pcePcepSessNumReqRcvdNoPathSent,  
pcePcepSessNumReqRcvdCancelSent,  
pcePcepSessNumReqRcvdErrorSent,  
pcePcepSessNumReqRcvdCancelRcvd,  
pcePcepSessNumRepRcvdUnknown,  
pcePcepSessNumReqRcvdUnknown,  
pcePcepNotificationsMaxRate

```

    }
    STATUS current
    DESCRIPTION
        "Objects that apply to all PCEP MIB module implementations."
    ::= { pcePcepGroups 1 }

pcePcepNotificationsGroup NOTIFICATION-GROUP
    NOTIFICATIONS { pcePcepSessUp,
                    pcePcepSessDown,
                    pcePcepSessLocalOverload,
                    pcePcepSessLocalOverloadClear,
                    pcePcepSessPeerOverload,
                    pcePcepSessPeerOverloadClear
    }
    STATUS current
    DESCRIPTION
        "The notifications for a PCEP MIB module implementation."
    ::= { pcePcepGroups 2 }

END

```

## 5. Security Considerations

The pcePcepNotificationsMaxRate object defined in this MIB module has a MAX-ACCESS clause of read-write. Such objects may be considered sensitive or vulnerable in some network environments. The support for SET operations in a non-secure environment without proper protection can have a negative effect on network operations. In particular, pcePcepNotificationsMaxRate may be used improperly to stop notifications being issued, or to permit a flood of notifications to be sent to the management agent at a high rate.

The readable objects in this MIB module (i.e., objects with a MAX-ACCESS other than not-accessible) may be considered sensitive or vulnerable in some network environments since, collectively, they provide information about the amount and frequency of path computation requests and responses within the network and can reveal some aspects of its configuration. It is thus important to control even GET and/or NOTIFY access to these objects and possibly to even encrypt the values of these objects when sending them over the network via SNMP.

SNMP versions prior to SNMPv3 did not include adequate security. Even if the network itself is secure (for example by using IPsec), there is no control as to who on the secure network is allowed to access and GET/SET (read/change/create/delete) the objects in this MIB module.

Implementations SHOULD provide the security features described by the SNMPv3 framework (see [RFC3410]), and implementations claiming compliance to the SNMPv3 standard MUST include full support for authentication and privacy via the User-based Security Model (USM) [RFC3414] with the AES cipher algorithm [RFC3826]. Implementations MAY also provide support for the Transport Security Model (TSM) [RFC5591] in combination with a secure transport such as SSH [RFC5592] or TLS/DTLS [RFC6353].

Further, deployment of SNMP versions prior to SNMPv3 is NOT RECOMMENDED. Instead, it is RECOMMENDED to deploy SNMPv3 and to enable cryptographic security. It is then a customer/operator responsibility to ensure that the SNMP entity giving access to an instance of this MIB module is properly configured to give access to the objects only to those principals (users) that have legitimate rights to indeed GET or SET (change/create/delete) them.

## 6. IANA Considerations

The MIB module in this document uses the following IANA-assigned OBJECT IDENTIFIER values recorded in the SMI Numbers registry:

Descriptor	OBJECT IDENTIFIER value
-----	-----
pcePcepMIB	{ mib-2 XXX }

Editor's Note (to be removed prior to publication): the IANA is requested to assign a value for "XXX" under the 'mib-2' subtree and to record the assignment in the SMI Numbers registry. When the assignment has been made, the RFC Editor is asked to replace "XXX" (here and in the MIB module) with the assigned value and to remove this note.

## 7. Acknowledgement

The authors would like to thank Santanu Mazumder, Meral Shirazipour and Adrian Farrel for their valuable input.

Funding for the RFC Editor function is currently provided by the Internet Society.

## 8. References

### 8.1. Normative References

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

- [RFC2578] McCloghrie, K., Ed., Perkins, D., Ed., and J. Schoenwaelder, Ed., "Structure of Management Information Version 2 (SMIv2)", STD 58, RFC 2578, April 1999.
- [RFC2579] McCloghrie, K., Ed., Perkins, D., Ed., and J. Schoenwaelder, Ed., "Textual Conventions for SMIv2", STD 58, RFC 2579, April 1999.
- [RFC2580] McCloghrie, K., Perkins, D., and J. Schoenwaelder, "Conformance Statements for SMIv2", STD 58, RFC 2580, April 1999.
- [RFC4001] Daniele, M., Haberman, B., Routhier, S., and J. Schoenwaelder, "Textual Conventions for Internet Network Addresses", RFC 4001, February 2005.
- [RFC5440] Vasseur, JP. and JL. Le Roux, "Path Computation Element (PCE) Communication Protocol (PCEP)", RFC 5440, March 2009.

## 8.2. Informative References

- [RFC3410] Case, J., Mundy, R., Partain, D., and B. Stewart, "Introduction and Applicability Statements for Internet-Standard Management Framework", RFC 3410, December 2002.
- [RFC3414] Blumenthal, U. and B. Wijnen, "User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3)", STD 62, RFC 3414, December 2002.
- [RFC3826] Blumenthal, U., Maino, F., and K. McCloghrie, "The Advanced Encryption Standard (AES) Cipher Algorithm in the SNMP User-based Security Model", RFC 3826, June 2004.
- [RFC4655] Farrel, A., Vasseur, J., and J. Ash, "A Path Computation Element (PCE)-Based Architecture", RFC 4655, August 2006.
- [RFC5591] Harrington, D. and W. Hardaker, "Transport Security Model for the Simple Network Management Protocol (SNMP)", STD 78, RFC 5591, June 2009.
- [RFC5592] Harrington, D., Salowey, J., and W. Hardaker, "Secure Shell Transport Model for the Simple Network Management Protocol (SNMP)", RFC 5592, June 2009.
- [RFC6353] Hardaker, W., "Transport Layer Security (TLS) Transport Model for the Simple Network Management Protocol (SNMP)", STD 78, RFC 6353, July 2011.



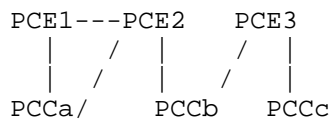
## Appendix A. Contributors

Dhruv Dhody  
 Huawei Technologies  
 Leela Palace  
 Bangalore, Karnataka 560008  
 India

E-Mail: dhruv.ietf@gmail.com

## Appendix B. PCEP MIB Module Example

This example considers the set of PCC / PCE relationships shown in the following figure. The example shows the contents of the PCEP MIB module as read at PCE2 and PCCb.



The IP addresses of the PCE speakers in this diagram are given in the following table.

PCE1	1.1.1.1
PCE2	2.2.2.2
PCE3	3.3.3.3
PCCa	11.11.11.11
PCCb	22.22.22.22
PCCc	33.33.33.33

In this example, the PCEP session between PCCb and PCE3 is currently down.

## B.1. Contents of PCEP MIB module at PCE2

At PCE2, there is a single local PCEP entity which has three peers (PCCa, PCCb and PCE1). There is a session active to all of these peers.

The contents of the PCEP MIB module as read at PCE2 are as follows.

```

In pcePcepEntityTable {
    pcePcepEntityIndex          1,
    pcePcepEntityAdminStatus    adminStatusUp(1),
    pcePcepEntityOperStatus     operStatusUp(1),
    pcePcepEntityAddrType       ipv4(1),
    pcePcepEntityAddr           2.2.2.2, -- PCE2
    pcePcepEntityConnectTimer   60,
    pcePcepEntityConnectMaxRetry 5,
    pcePcepEntityInitBackoffTimer 30,
    pcePcepEntityMaxBackoffTimer 3600,
    pcePcepEntityOpenWaitTimer  60,
    pcePcepEntityKeepWaitTimer  60,
    pcePcepEntityKeepAliveTimer  1,
    pcePcepEntityDeadTimer       4,
    pcePcepEntityAllowNegotiation true(1),
    pcePcepEntityMaxKeepAliveTimer 60,
    pcePcepEntityMaxDeadTimer     240,
    pcePcepEntityMinKeepAliveTimer 1,
    pcePcepEntityMinDeadTimer     4,
    pcePcepEntitySyncTimer        60,
    pcePcepEntityRequestTimer     120,
    pcePcepEntityMaxSessions      999,
    pcePcepEntityMaxUnknownReqs   5,
    pcePcepEntityMaxUnknownMsgs   5
}

In pcePcepPeerTable {
    pcePcepPeerAddrType       ipv4(1), --PCE1
    pcePcepPeerAddr           1.1.1.1,
    pcePcepPeerRole            pccAndPce(3),
    pcePcepPeerDiscontinuityTime TimeStamp,
    pcePcepPeerInitiateSession true(1),
    pcePcepPeerSessionExists   true(1),
    pcePcepPeerNumSessSetupOK  1,
    pcePcepPeerNumSessSetupFail 0,
    pcePcepPeerSessionUpTime   TimeStamp,
    pcePcepPeerSessionFailTime 0,
    pcePcepPeerSessionFailUpTime TimeStamp,
    pcePcepPeerAvgRspTime      0,
    pcePcepPeerLWMRspTime      0,
    pcePcepPeerHWMRspTime      0,
    pcePcepPeerNumPCReqSent     0,
    pcePcepPeerNumPCReqRcvd     0,
    pcePcepPeerNumPCRepSent     0,
    pcePcepPeerNumPCRepRcvd     0,
    pcePcepPeerNumPCErrSent     0,

```

```

pcePcepPeerNumPCErrRcvd          0,
pcePcepPeerNumPCNtfSent          0,
pcePcepPeerNumPCNtfRcvd          0,
pcePcepPeerNumKeepaliveSent      123,
pcePcepPeerNumKeepaliveRcvd      123,
pcePcepPeerNumUnknownRcvd        0,
pcePcepPeerNumCorruptRcvd        0,
pcePcepPeerNumReqSent            0,
pcePcepPeerNumSvecSent           0,
pcePcepPeerNumSvecReqSent        0,
pcePcepPeerNumReqSentPendRep     0,
pcePcepPeerNumReqSentEroRcvd     0,
pcePcepPeerNumReqSentNoPathRcvd  0,
pcePcepPeerNumReqSentCancelRcvd  0,
pcePcepPeerNumReqSentErrorRcvd   0,
pcePcepPeerNumReqSentTimeout     0,
pcePcepPeerNumReqSentCancelSent  0,
pcePcepPeerNumReqSentClosed      0,
pcePcepPeerNumReqRcvd            0,
pcePcepPeerNumSvecRcvd           0,
pcePcepPeerNumSvecReqRcvd        0,
pcePcepPeerNumReqRcvdPendRep     0,
pcePcepPeerNumReqRcvdEroSent     0,
pcePcepPeerNumReqRcvdNoPathSent  0,
pcePcepPeerNumReqRcvdCancelSent  0,
pcePcepPeerNumReqRcvdErrorSent   0,
pcePcepPeerNumReqRcvdCancelRcvd  0,
pcePcepPeerNumReqRcvdClosed      0,
pcePcepPeerNumRepRcvdUnknown     0,
pcePcepPeerNumReqRcvdUnknown     0
},
{
    pcePcepPeerAddrType          ipv4(1),  --PCCa
    pcePcepPeerAddr              11.11.11.11,
    pcePcepPeerRole               pcc(1),
    pcePcepPeerDiscontinuityTime  timeStamp,
    pcePcepPeerInitiateSession   false(0),
    pcePcepPeerSessionExists     true(1),
    pcePcepPeerNumSessSetupOK     1,
    pcePcepPeerNumSessSetupFail   0,
    pcePcepPeerSessionUpTime      timeStamp,
    pcePcepPeerSessionFailTime    0,
    pcePcepPeerSessionFailUpTime  timeStamp,
    pcePcepPeerAvgRspTime         200,
    pcePcepPeerLWMRspTime         100,
    pcePcepPeerHWMRspTime         300,
    pcePcepPeerNumPCReqSent       0,
    pcePcepPeerNumPCReqRcvd       3,

```

```

pcePcepPeerNumPCRepSent          3,
pcePcepPeerNumPCRepRcvd          0,
pcePcepPeerNumPCErrSent          0,
pcePcepPeerNumPCErrRcvd          0,
pcePcepPeerNumPCNtfSent          0,
pcePcepPeerNumPCNtfRcvd          0,
pcePcepPeerNumKeepaliveSent      123,
pcePcepPeerNumKeepaliveRcvd      123,
pcePcepPeerNumUnknownRcvd        0,
pcePcepPeerNumCorruptRcvd        0,
pcePcepPeerNumReqSent            0,
pcePcepPeerNumSvecSent            0,
pcePcepPeerNumSvecReqSent        0,
pcePcepPeerNumReqSentPendRep     0,
pcePcepPeerNumReqSentEroRcvd     0,
pcePcepPeerNumReqSentNoPathRcvd  0,
pcePcepPeerNumReqSentCancelRcvd  0,
pcePcepPeerNumReqSentErrorRcvd   0,
pcePcepPeerNumReqSentTimeout     0,
pcePcepPeerNumReqSentCancelSent  0,
pcePcepPeerNumReqSentClosed      0,
pcePcepPeerNumReqRcvd            3,
pcePcepPeerNumSvecRcvd           0,
pcePcepPeerNumSvecReqRcvd        0,
pcePcepPeerNumReqRcvdPendRep     0,
pcePcepPeerNumReqRcvdEroSent     3,
pcePcepPeerNumReqRcvdNoPathSent  0,
pcePcepPeerNumReqRcvdCancelSent  0,
pcePcepPeerNumReqRcvdErrorSent   0,
pcePcepPeerNumReqRcvdCancelRcvd  0,
pcePcepPeerNumReqRcvdClosed      0,
pcePcepPeerNumRepRcvdUnknown     0,
pcePcepPeerNumReqRcvdUnknown     0
},
{
pcePcepPeerAddrType              ipv4(1), -- PCCb
pcePcepPeerAddr                  22.22.22.22,
pcePcepPeerRole                   pcc(1),
pcePcepPeerDiscontinuityTime      TimeStamp,
pcePcepPeerInitiateSession        true(1),
pcePcepPeerSessionExists          true(1),
pcePcepPeerNumSessSetupOK         1,
pcePcepPeerNumSessSetupFail       0,
pcePcepPeerSessionUpTime          TimeStamp,
pcePcepPeerSessionFailTime        0,
pcePcepPeerSessionFailUpTime      TimeStamp,
pcePcepPeerAvgRspTime             200,
pcePcepPeerLWMRspTime             100,

```

```

pcePcepPeerHWMRspTime          300,
pcePcepPeerNumPCReqSent        0,
pcePcepPeerNumPCReqRcvd        4,
pcePcepPeerNumPCRepSent        4,
pcePcepPeerNumPCRepRcvd        0,
pcePcepPeerNumPCErrSent        0,
pcePcepPeerNumPCErrRcvd        0,
pcePcepPeerNumPCNtfSent        0,
pcePcepPeerNumPCNtfRcvd        0,
pcePcepPeerNumKeepaliveSent    123,
pcePcepPeerNumKeepaliveRcvd    123,
pcePcepPeerNumUnknownRcvd      0,
pcePcepPeerNumCorruptRcvd      0,
pcePcepPeerNumReqSent          0,
pcePcepPeerNumSvecSent         0,
pcePcepPeerNumSvecReqSent      0,
pcePcepPeerNumReqSentPendRep   0,
pcePcepPeerNumReqSentEroRcvd   0,
pcePcepPeerNumReqSentNoPathRcvd 0,
pcePcepPeerNumReqSentCancelRcvd 0,
pcePcepPeerNumReqSentErrorRcvd 0,
pcePcepPeerNumReqSentTimeout   0,
pcePcepPeerNumReqSentCancelSent 0,
pcePcepPeerNumReqSentClosed    0,
pcePcepPeerNumReqRcvd          4,
pcePcepPeerNumSvecRcvd         0,
pcePcepPeerNumSvecReqRcvd      0,
pcePcepPeerNumReqRcvdPendRep   0,
pcePcepPeerNumReqRcvdEroSent    3,
pcePcepPeerNumReqRcvdNoPathSent 1,
pcePcepPeerNumReqRcvdCancelSent 0,
pcePcepPeerNumReqRcvdErrorSent 0,
pcePcepPeerNumReqRcvdCancelRcvd 0,
pcePcepPeerNumReqRcvdClosed    0,
pcePcepPeerNumRepRcvdUnknown   0,
pcePcepPeerNumReqRcvdUnknown   0
}

In pcePcepSessTable {
    pcePcepSessInitiator          local(1), --PCE1
    pcePcepSessStateLastChange    TimeStamp,
    pcePcepSessState              sessionUp(4),
    pcePcepSessConnectRetry       0,
    pcePcepSessLocalID            1,
    pcePcepSessRemoteID           2,
    pcePcepSessKeepaliveTimer     1,
    pcePcepSessPeerKeepaliveTimer 1,
    pcePcepSessDeadTimer          4,

```

```

pcePcepSessPeerDeadTimer          4,
pcePcepSessKAHoldTimeRem          1,
pcePcepSessOverloaded              false(0),
pcePcepSessOverloadTime            0,
pcePcepSessPeerOverloaded          false(0),
pcePcepSessPeerOverloadTime        0,
pcePcepSessDiscontinuityTime       timeStamp,
pcePcepSessAvgRspTime              0,
pcePcepSessLWMRspTime              0,
pcePcepSessHWMRspTime              0,
pcePcepSessNumPCReqSent            0,
pcePcepSessNumPCReqRcvd            0,
pcePcepSessNumPCRepSent            0,
pcePcepSessNumPCRepRcvd            0,
pcePcepSessNumPCErrSent            0,
pcePcepSessNumPCErrRcvd            0,
pcePcepSessNumPCNtfSent            0,
pcePcepSessNumPCNtfRcvd            0,
pcePcepSessNumKeepaliveSent        123,
pcePcepSessNumKeepaliveRcvd        123,
pcePcepSessNumUnknownRcvd          0,
pcePcepSessNumCorruptRcvd          0,
pcePcepSessNumReqSent              0,
pcePcepSessNumSvecSent             0,
pcePcepSessNumSvecReqSent          0,
pcePcepSessNumReqSentPendRep       0,
pcePcepSessNumReqSentEroRcvd       0,
pcePcepSessNumReqSentNoPathRcvd    0,
pcePcepSessNumReqSentCancelRcvd    0,
pcePcepSessNumReqSentErrorRcvd     0,
pcePcepSessNumReqSentTimeout       0,
pcePcepSessNumReqSentCancelSent    0,
pcePcepSessNumReqRcvd              0,
pcePcepSessNumSvecRcvd             0,
pcePcepSessNumSvecReqRcvd          0,
pcePcepSessNumReqRcvdPendRep       0,
pcePcepSessNumReqRcvdEroSent       0,
pcePcepSessNumReqRcvdNoPathSent    0,
pcePcepSessNumReqRcvdCancelSent    0,
pcePcepSessNumReqRcvdErrorSent     0,
pcePcepSessNumReqRcvdCancelRcvd    0,
pcePcepSessNumRepRcvdUnknown        0,
pcePcepSessNumReqRcvdUnknown        0
},
{
    pcePcepSessInitiator              remote(2), --PCCa
    pcePcepSessStateLastChange        timeStamp,
    pcePcepSessState                  sessionUp(4),

```

pcePcepSessConnectRetry	0,
pcePcepSessLocalID	2,
pcePcepSessRemoteID	1,
pcePcepSessKeepaliveTimer	1,
pcePcepSessPeerKeepaliveTimer	1,
pcePcepSessDeadTimer	4,
pcePcepSessPeerDeadTimer	4,
pcePcepSessKAHoldTimeRem	1,
pcePcepSessOverloaded	false(0),
pcePcepSessOverloadTime	0,
pcePcepSessPeerOverloaded	false(0),
pcePcepSessPeerOverloadTime	0,
pcePcepSessDiscontinuityTime	TimeStamp,
pcePcepSessAvgRspTime	200,
pcePcepSessLWMRspTime	100,
pcePcepSessHWMRspTime	300,
pcePcepSessNumPCReqSent	0,
pcePcepSessNumPCReqRcvd	1,
pcePcepSessNumPCRepSent	1,
pcePcepSessNumPCRepRcvd	0,
pcePcepSessNumPCErrSent	0,
pcePcepSessNumPCErrRcvd	0,
pcePcepSessNumPCNtfSent	0,
pcePcepSessNumPCNtfRcvd	0,
pcePcepSessNumKeepaliveSent	123,
pcePcepSessNumKeepaliveRcvd	123,
pcePcepSessNumUnknownRcvd	0,
pcePcepSessNumCorruptRcvd	0,
pcePcepSessNumReqSent	0,
pcePcepSessNumSvecSent	0,
pcePcepSessNumSvecReqSent	0,
pcePcepSessNumReqSentPendRep	0,
pcePcepSessNumReqSentEroRcvd	0,
pcePcepSessNumReqSentNoPathRcvd	0,
pcePcepSessNumReqSentCancelRcvd	0,
pcePcepSessNumReqSentErrorRcvd	0,
pcePcepSessNumReqSentTimeout	0,
pcePcepSessNumReqSentCancelSent	0,
pcePcepSessNumReqRcvd	3,
pcePcepSessNumSvecRcvd	0,
pcePcepSessNumSvecReqRcvd	0,
pcePcepSessNumReqRcvdPendRep	0,
pcePcepSessNumReqRcvdEroSent	3,
pcePcepSessNumReqRcvdNoPathSent	0,
pcePcepSessNumReqRcvdCancelSent	0,
pcePcepSessNumReqRcvdErrorSent	0,
pcePcepSessNumReqRcvdCancelRcvd	0,
pcePcepSessNumRepRcvdUnknown	0,

```

    pcePcepSessNumReqRcvdUnknown      0
  },
  {
    pcePcepSessInitiator               remote(2), --PCCb
    pcePcepSessStateLastChange         TimeStamp,
    pcePcepSessState                   sessionUp(4),
    pcePcepSessConnectRetry            0,
    pcePcepSessLocalID                 2,
    pcePcepSessRemoteID                1,
    pcePcepSessKeepaliveTimer          1,
    pcePcepSessPeerKeepaliveTimer      1,
    pcePcepSessDeadTimer               4,
    pcePcepSessPeerDeadTimer           4,
    pcePcepSessKAHoldTimeRem           1,
    pcePcepSessOverloaded              false(0),
    pcePcepSessOverloadTime            0,
    pcePcepSessPeerOverloaded          false(0),
    pcePcepSessPeerOverloadTime        0,
    pcePcepSessDiscontinuityTime       TimeStamp,
    pcePcepSessAvgRspTime              200,
    pcePcepSessLWMRspTime              100,
    pcePcepSessHWMRspTime              300,
    pcePcepSessNumPCReqSent            0,
    pcePcepSessNumPCReqRcvd            4,
    pcePcepSessNumPCRepSent            4,
    pcePcepSessNumPCRepRcvd            0,
    pcePcepSessNumPCErrSent            0,
    pcePcepSessNumPCErrRcvd            0,
    pcePcepSessNumPCNtfSent            0,
    pcePcepSessNumPCNtfRcvd            0,
    pcePcepSessNumKeepaliveSent        123,
    pcePcepSessNumKeepaliveRcvd        123,
    pcePcepSessNumUnknownRcvd          0,
    pcePcepSessNumCorruptRcvd          0,
    pcePcepSessNumReqSent              0,
    pcePcepSessNumSvecSent             0,
    pcePcepSessNumSvecReqSent          0,
    pcePcepSessNumReqSentPendRep       0,
    pcePcepSessNumReqSentEroRcvd       0,
    pcePcepSessNumReqSentNoPathRcvd    0,
    pcePcepSessNumReqSentCancelRcvd    0,
    pcePcepSessNumReqSentErrorRcvd     0,
    pcePcepSessNumReqSentTimeout       0,
    pcePcepSessNumReqSentCancelSent    0,
    pcePcepSessNumReqRcvd              4,
    pcePcepSessNumSvecRcvd             0,
    pcePcepSessNumSvecReqRcvd          0,
    pcePcepSessNumReqRcvdPendRep       0,

```



```

    pcePcepSessNumReqRcvdEroSent      3,
    pcePcepSessNumReqRcvdNoPathSent   1,
    pcePcepSessNumReqRcvdCancelSent   0,
    pcePcepSessNumReqRcvdErrorSent    0,
    pcePcepSessNumReqRcvdCancelRcvd   0,
    pcePcepSessNumRepRcvdUnknown      0,
    pcePcepSessNumReqRcvdUnknown      0
}

```

## B.2. Contents of PCEP MIB module at PCCb

At PCCb, there is a single local PCEP entity which has two peers (PCE2 and PCE3). There is a session active to PCE2, but the session to PCE3 is currently down.

The contents of the PCEP MIB module as read at PCCb are as follows.

```

In pcePcepEntityTable {
    pcePcepEntityIndex      1,
    pcePcepEntityAdminStatus adminStatusUp(1),
    pcePcepEntityOperStatus operStatusUp(1),
    pcePcepEntityAddrType   ipv4(1),
    pcePcepEntityAddr       22.22.22.22, -- PCCb
    pcePcepEntityConnectTimer 60,
    pcePcepEntityConnectMaxRetry 5,
    pcePcepEntityInitBackoffTimer 30,
    pcePcepEntityMaxBackoffTimer 3600,
    pcePcepEntityOpenWaitTimer 60,
    pcePcepEntityKeepWaitTimer 60,
    pcePcepEntityKeepAliveTimer 1,
    pcePcepEntityDeadTimer 4,
    pcePcepEntityAllowNegotiation true(1),
    pcePcepEntityMaxKeepAliveTimer 60,
    pcePcepEntityMaxDeadTimer 240,
    pcePcepEntityMinKeepAliveTimer 1,
    pcePcepEntityMinDeadTimer 4,
    pcePcepEntitySyncTimer 60,
    pcePcepEntityRequestTimer 120,
    pcePcepEntityMaxSessions 999,
    pcePcepEntityMaxUnknownReqs 5,
    pcePcepEntityMaxUnknownMsgs 5
}

In pcePcepPeerTable {
    pcePcepPeerAddrType   ipv4(1), --PCE2
    pcePcepPeerAddr       2.2.2.2,
    pcePcepPeerRole        pce(2),
    pcePcepPeerDiscontinuityTime TimeStamp,

```

```

pcePcepPeerInitiateSession      true(1),
pcePcepPeerSessionExists        true(1)),
pcePcepPeerNumSessSetupOK       0,
pcePcepPeerNumSessSetupFail     1,
pcePcepPeerSessionUpTime        TimeStamp,
pcePcepPeerSessionFailTime      TimeStamp,
pcePcepPeerSessionFailUpTime    TimeStamp,
pcePcepPeerAvgRspTime           0,
pcePcepPeerLWMRspTime           0,
pcePcepPeerHWMRspTime           0,
pcePcepPeerNumPCReqSent         4,
pcePcepPeerNumPCReqRcvd         0,
pcePcepPeerNumPCRepSent         0,
pcePcepPeerNumPCRepRcvd         4,
pcePcepPeerNumPCErrSent         0,
pcePcepPeerNumPCErrRcvd         0,
pcePcepPeerNumPCNtfSent         0,
pcePcepPeerNumPCNtfRcvd         0,
pcePcepPeerNumKeepaliveSent     0,
pcePcepPeerNumKeepaliveRcvd     0,
pcePcepPeerNumUnknownRcvd       0,
pcePcepPeerNumCorruptRcvd       0,
pcePcepPeerNumReqSent           4,
pcePcepPeerNumSvecSent          0,
pcePcepPeerNumSvecReqSent       0,
pcePcepPeerNumReqSentPendRep    0,
pcePcepPeerNumReqSentEroRcvd    3,
pcePcepPeerNumReqSentNoPathRcvd 1,
pcePcepPeerNumReqSentCancelRcvd 0,
pcePcepPeerNumReqSentErrorRcvd  0,
pcePcepPeerNumReqSentTimeout    0,
pcePcepPeerNumReqSentCancelSent 0,
pcePcepPeerNumReqSentClosed     0,
pcePcepPeerNumReqRcvd           0,
pcePcepPeerNumSvecRcvd          0,
pcePcepPeerNumSvecReqRcvd       0,
pcePcepPeerNumReqRcvdPendRep    0,
pcePcepPeerNumReqRcvdEroSent    0,
pcePcepPeerNumReqRcvdNoPathSent 0,
pcePcepPeerNumReqRcvdCancelSent 0,
pcePcepPeerNumReqRcvdErrorSent  0,
pcePcepPeerNumReqRcvdCancelRcvd 0,
pcePcepPeerNumReqRcvdClosed     0,
pcePcepPeerNumRepRcvdUnknown    0,
pcePcepPeerNumReqRcvdUnknown    0
},
{
    pcePcepPeerAddrType          ipv4(1),  --PCE3

```

pcePcepPeerAddr	3.3.3.3,
pcePcepPeerRole	pce(2),
pcePcepPeerDiscontinuityTime	TimeStamp,
pcePcepPeerInitiateSession	true(1),
pcePcepPeerSessionExists	false(0),
pcePcepPeerNumSessSetupOK	1,
pcePcepPeerNumSessSetupFail	0,
pcePcepPeerSessionUpTime	TimeStamp,
pcePcepPeerSessionFailTime	TimeStamp,
pcePcepPeerSessionFailUpTime	TimeStamp,
pcePcepPeerAvgRspTime	200,
pcePcepPeerLWMRspTime	100,
pcePcepPeerHWMRspTime	300,
pcePcepPeerNumPCReqSent	4,
pcePcepPeerNumPCReqRcvd	0,
pcePcepPeerNumPCRepSent	0,
pcePcepPeerNumPCRepRcvd	3,
pcePcepPeerNumPCErrSent	0,
pcePcepPeerNumPCErrRcvd	0,
pcePcepPeerNumPCNtfSent	0,
pcePcepPeerNumPCNtfRcvd	0,
pcePcepPeerNumKeepaliveSent	123,
pcePcepPeerNumKeepaliveRcvd	123,
pcePcepPeerNumUnknownRcvd	0,
pcePcepPeerNumCorruptRcvd	0,
pcePcepPeerNumReqSent	4,
pcePcepPeerNumSvecSent	0,
pcePcepPeerNumSvecReqSent	0,
pcePcepPeerNumReqSentPendRep	0,
pcePcepPeerNumReqSentEroRcvd	3,
pcePcepPeerNumReqSentNoPathRcvd	0,
pcePcepPeerNumReqSentCancelRcvd	0,
pcePcepPeerNumReqSentErrorRcvd	0,
pcePcepPeerNumReqSentTimeout	0,
pcePcepPeerNumReqSentCancelSent	0,
pcePcepPeerNumReqSentClosed	1,
pcePcepPeerNumReqRcvd	0,
pcePcepPeerNumSvecRcvd	0,
pcePcepPeerNumSvecReqRcvd	0,
pcePcepPeerNumReqRcvdPendRep	0,
pcePcepPeerNumReqRcvdEroSent	0,
pcePcepPeerNumReqRcvdNoPathSent	0,
pcePcepPeerNumReqRcvdCancelSent	0,
pcePcepPeerNumReqRcvdErrorSent	0,
pcePcepPeerNumReqRcvdCancelRcvd	0,
pcePcepPeerNumReqRcvdClosed	0,
pcePcepPeerNumRepRcvdUnknown	0,
pcePcepPeerNumReqRcvdUnknown	0

```

    }

    In pcePcepSessTable {
        pcePcepSessInitiator                local(1), --PCE2
        pcePcepSessStateLastChange          TimeStamp,
        pcePcepSessState                    sessionUp(4),
        pcePcepSessConnectRetry              0,
        pcePcepSessLocalID                   1,
        pcePcepSessRemoteID                  1,
        pcePcepSessKeepaliveTimer            1,
        pcePcepSessPeerKeepaliveTimer        1,
        pcePcepSessDeadTimer                  4,
        pcePcepSessPeerDeadTimer             4,
        pcePcepSessKAHoldTimeRem             1,
        pcePcepSessOverloaded                 false(0),
        pcePcepSessOverloadTime              0,
        pcePcepSessPeerOverloaded            false(0),
        pcePcepSessPeerOverloadTime          0,
        pcePcepSessDiscontinuityTime         TimeStamp,
        pcePcepSessAvgRspTime                 200,
        pcePcepSessLWMRspTime                100,
        pcePcepSessHWMRspTime                300,
        pcePcepSessNumPCReqSent              4,
        pcePcepSessNumPCReqRcvd              0,
        pcePcepSessNumPCRepSent              0,
        pcePcepSessNumPCRepRcvd              4,
        pcePcepSessNumPCErrSent              0,
        pcePcepSessNumPCErrRcvd              0,
        pcePcepSessNumPCNtfSent              0,
        pcePcepSessNumPCNtfRcvd              0,
        pcePcepSessNumKeepaliveSent          123,
        pcePcepSessNumKeepaliveRcvd          123,
        pcePcepSessNumUnknownRcvd            0,
        pcePcepSessNumCorruptRcvd            0,
        pcePcepSessNumReqSent                4,
        pcePcepSessNumSvecSent               0,
        pcePcepSessNumSvecReqSent            0,
        pcePcepSessNumReqSentPendRep         0,
        pcePcepSessNumReqSentEroRcvd         3,
        pcePcepSessNumReqSentNoPathRcvd      1,
        pcePcepSessNumReqSentCancelRcvd      0,
        pcePcepSessNumReqSentErrorRcvd       0,
        pcePcepSessNumReqSentTimeout         0,
        pcePcepSessNumReqSentCancelSent      0,
        pcePcepSessNumReqRcvd                0,
        pcePcepSessNumSvecRcvd               0,
        pcePcepSessNumSvecReqRcvd            0,
        pcePcepSessNumReqRcvdPendRep         0,

```

```
    pcePcepSessNumReqRcvdEroSent      0,  
    pcePcepSessNumReqRcvdNoPathSent   0,  
    pcePcepSessNumReqRcvdCancelSent   0,  
    pcePcepSessNumReqRcvdErrorSent    0,  
    pcePcepSessNumReqRcvdCancelRcvd   0,  
    pcePcepSessNumRepRcvdUnknown      0,  
    pcePcepSessNumReqRcvdUnknown      0  
}
```

-- no session to PCE3

#### Authors' Addresses

Agrahara Kiran Koushik  
Brocade Communications Inc.

EMail: [kkoushik@brocade.com](mailto:kkoushik@brocade.com)

Emile Stephan  
Orange  
2 avenue Pierre Marzin  
Lannion F-22307  
France

EMail: [emile.stephan@orange.com](mailto:emile.stephan@orange.com)

Quintin Zhao  
Huawei Technology  
125 Nagog Technology Park  
Acton, MA 01719  
US

EMail: [qzhao@huawei.com](mailto:qzhao@huawei.com)

Daniel King  
Old Dog Consulting

EMail: [daniel@olddog.co.uk](mailto:daniel@olddog.co.uk)

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: December 21, 2017

E. Crabbe  
Oracle  
I. Minei  
Google, Inc.  
J. Medved  
Cisco Systems, Inc.  
R. Varga  
Pantheon Technologies SRO  
June 19, 2017

PCEP Extensions for Stateful PCE  
draft-ietf-pce-stateful-pce-21

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.

Although PCEP explicitly makes no assumptions regarding the information available to the PCE, it also makes no provisions for PCE control of timing and sequence of path computations within and across PCEP sessions. This document describes a set of extensions to PCEP to enable stateful control of MPLS-TE and GMPLS LSPs via PCEP.

Status of This Memo

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

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

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

This Internet-Draft will expire on December 21, 2017.

Copyright Notice

Copyright (c) 2017 IETF Trust and the persons identified as the document authors. All rights reserved.

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

## Table of Contents

1. Introduction . . . . .	3
1.1. Requirements Language . . . . .	4
2. Terminology . . . . .	4
3. Motivation and Objectives for Stateful PCE . . . . .	5
3.1. Motivation . . . . .	5
3.1.1. Background . . . . .	5
3.1.2. Why a Stateful PCE? . . . . .	6
3.1.3. Protocol vs. Configuration . . . . .	7
3.2. Objectives . . . . .	7
4. New Functions to Support Stateful PCEs . . . . .	8
5. Overview of Protocol Extensions . . . . .	9
5.1. LSP State Ownership . . . . .	9
5.2. New Messages . . . . .	9
5.3. Error Reporting . . . . .	10
5.4. Capability Advertisement . . . . .	10
5.5. IGP Extensions for Stateful PCE Capabilities Advertisement . . . . .	11
5.6. State Synchronization . . . . .	12
5.7. LSP Delegation . . . . .	15
5.7.1. Delegating an LSP . . . . .	15
5.7.2. Revoking a Delegation . . . . .	16
5.7.3. Returning a Delegation . . . . .	18
5.7.4. Redundant Stateful PCEs . . . . .	18
5.7.5. Redefinition on PCE Failure . . . . .	19
5.8. LSP Operations . . . . .	19
5.8.1. Passive Stateful PCE Path Computation Request/Response . . . . .	19
5.8.2. Switching from Passive Stateful to Active Stateful .	21
5.8.3. Active Stateful PCE LSP Update . . . . .	22
5.9. LSP Protection . . . . .	23
5.10. PCEP Sessions . . . . .	23
6. PCEP Messages . . . . .	23
6.1. The PCRpt Message . . . . .	24
6.2. The PCUpd Message . . . . .	26
6.3. The PCErr Message . . . . .	28
6.4. The PCReq Message . . . . .	29



6.5.	The PCRep Message . . . . .	30
7.	Object Formats . . . . .	30
7.1.	OPEN Object . . . . .	30
7.1.1.	Stateful PCE Capability TLV . . . . .	30
7.2.	SRP Object . . . . .	31
7.3.	LSP Object . . . . .	33
7.3.1.	LSP-IDENTIFIERS TLVs . . . . .	35
7.3.2.	Symbolic Path Name TLV . . . . .	38
7.3.3.	LSP Error Code TLV . . . . .	39
7.3.4.	RSVP Error Spec TLV . . . . .	40
8.	IANA Considerations . . . . .	41
8.1.	PCE Capabilities in IGP Advertisements . . . . .	41
8.2.	PCEP Messages . . . . .	41
8.3.	PCEP Objects . . . . .	42
8.4.	LSP Object . . . . .	42
8.5.	PCEP-Error Object . . . . .	43
8.6.	Notification Object . . . . .	43
8.7.	PCEP TLV Type Indicators . . . . .	44
8.8.	STATEFUL-PCE-CAPABILITY TLV . . . . .	44
8.9.	LSP-ERROR-CODE TLV . . . . .	45
9.	Manageability Considerations . . . . .	45
9.1.	Control Function and Policy . . . . .	45
9.2.	Information and Data Models . . . . .	46
9.3.	Liveness Detection and Monitoring . . . . .	47
9.4.	Verifying Correct Operation . . . . .	47
9.5.	Requirements on Other Protocols and Functional Components . . . . .	47
9.6.	Impact on Network Operation . . . . .	47
10.	Security Considerations . . . . .	48
10.1.	Vulnerability . . . . .	48
10.2.	LSP State Snooping . . . . .	48
10.3.	Malicious PCE . . . . .	49
10.4.	Malicious PCC . . . . .	49
11.	Contributing Authors . . . . .	49
12.	Acknowledgements . . . . .	50
13.	References . . . . .	50
13.1.	Normative References . . . . .	50
13.2.	Informative References . . . . .	51
	Authors' Addresses . . . . .	53

## 1. Introduction

[RFC5440] describes the Path Computation Element Communication Protocol (PCEP). PCEP defines the communication between a Path Computation Client (PCC) and a Path Computation Element (PCE), or between PCEs, enabling computation of Multiprotocol Label Switching (MPLS) for Traffic Engineering Label Switched Path (TE LSP) characteristics. Extensions for support of Generalized MPLS (GMPLS) in PCEP are defined in [I-D.ietf-pce-gmpls-pcep-extensions]

This document specifies a set of extensions to PCEP to enable stateful control of LSPs within and across PCEP sessions in compliance with [RFC4657]. It includes mechanisms to effect Label Switched Path (LSP) state synchronization between PCCs and PCEs, delegation of control over LSPs to PCEs, and PCE control of timing and sequence of path computations within and across PCEP sessions.

Extensions to permit the PCE to drive creation of an LSP are defined in [I-D.ietf-pce-pce-initiated-lsp], which specifies PCE-initiated LSP creation.

### 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

This document uses the following terms defined in [RFC5440]: PCC, PCE, PCEP Peer, PCEP Speaker.

This document uses the following terms defined in [RFC4655]: TED.

This document uses the following terms defined in [RFC3031]: LSP.

This document uses the following terms defined in [RFC8051]: Stateful PCE, Passive Stateful PCE, Active Stateful PCE, Delegation, LSP State Database.

The following terms are defined in this document:

**Revocation:** an operation performed by a PCC on a previously delegated LSP. Revocation revokes the rights granted to the PCE in the delegation operation.

**Redelegation Timeout Interval:** the period of time a PCC waits for, when a PCEP session is terminated, before revoking LSP delegation to a PCE and attempting to redelegate LSPs associated with the terminated PCEP session to an alternate PCE. The Redelegation Timeout Interval is a PCC-local value that can be either operator-configured or dynamically computed by the PCC based on local policy.

**State Timeout Interval:** the period of time a PCC waits for, when a PCEP session is terminated, before flushing LSP state associated with that PCEP session and reverting to operator-defined default parameters or behaviors. The State Timeout Interval is a PCC-

local value that can be either operator-configured or dynamically computed by the PCC based on local policy.

LSP State Report: an operation to send LSP state (Operational / Admin Status, LSP attributes configured at the PCC and set by a PCE, etc.) from a PCC to a PCE.

LSP Update Request: an operation where an Active Stateful PCE requests a PCC to update one or more attributes of an LSP and to re-signal the LSP with updated attributes.

SRP-ID-number: a number used to correlate errors and LSP State Reports to LSP Update Requests. It is carried in the SRP (Stateful PCE Request Parameters) Object described in Section 7.2.

Within this document, PCEP communications are described through PCC-PCE relationship. The PCE architecture also supports the PCE-PCE communication, by having the requesting PCE fill the role of a PCC, as usual.

The message formats in this document are specified using Routing Backus-Naur Format (RBNF) encoding as specified in [RFC5511].

### 3. Motivation and Objectives for Stateful PCE

#### 3.1. Motivation

[RFC8051] presents several use cases, demonstrating scenarios that benefit from the deployment of a stateful PCE. The scenarios apply equally to MPLS-TE and GMPLS deployments.

##### 3.1.1. Background

Traffic engineering has been a goal of the MPLS architecture since its inception ([RFC3031], [RFC2702], [RFC3346]). In the traffic engineering system provided by [RFC3630], [RFC5305], and [RFC3209] information about network resources utilization is only available as total reserved capacity by traffic class on a per interface basis; individual LSP state is available only locally on each LER for its own LSPs. In most cases, this makes good sense, as distribution and retention of total LSP state for all LERs within in the network would be prohibitively costly.

Unfortunately, this visibility in terms of global LSP state may result in a number of issues for some demand patterns, particularly within a common setup and hold priority. This issue affects online traffic engineering systems.

A sufficiently over-provisioned system will by definition have no issues routing its demand on the shortest path. However, lowering the degree to which network over-provisioning is required in order to run a healthy, functioning network is a clear and explicit promise of MPLS architecture. In particular, it has been a goal of MPLS to provide mechanisms to alleviate congestion scenarios in which "traffic streams are inefficiently mapped onto available resources; causing subsets of network resources to become over-utilized while others remain underutilized" ([RFC2702]).

### 3.1.2. Why a Stateful PCE?

[RFC4655] defines a stateful PCE to be one in which the PCE maintains "strict synchronization between the PCE and not only the network states (in term of topology and resource information), but also the set of computed paths and reserved resources in use in the network." [RFC4655] also expressed a number of concerns with regard to a stateful PCE, specifically:

- o Any reliable synchronization mechanism would result in significant control plane overhead
- o Out-of-band TED synchronization would be complex and prone to race conditions
- o Path calculations incorporating total network state would be highly complex

In general, stress on the control plane will be directly proportional to the size of the system being controlled and the tightness of the control loop, and indirectly proportional to the amount of over-provisioning in terms of both network capacity and reservation overhead.

Despite these concerns in terms of implementation complexity and scalability, several TE algorithms exist today that have been demonstrated to be extremely effective in large TE systems, providing both rapid convergence and significant benefits in terms of optimality of resource usage [MXMN-TE]. All of these systems share at least two common characteristics: the requirement for both global visibility of a flow (or in this case, a TE LSP) state and for ordered control of path reservations across devices within the system being controlled. While some approaches have been suggested in order to remove the requirements for ordered control (See [MPLS-PC]), these approaches are highly dependent on traffic distribution, and do not allow for multiple simultaneous LSP priorities representing diffserv classes.

The use cases described in [RFC8051] demonstrate a need for visibility into global inter-PCC LSP state in PCE path computations, and for PCE control of sequence and timing in altering LSP path characteristics within and across PCEP sessions.

### 3.1.3. Protocol vs. Configuration

Note that existing configuration tools and protocols can be used to set LSP state, such as a Command Line Interface (CLI) tool. However, this solution has several shortcomings:

- o Scale & Performance: configuration operations often have transactional semantics which are typically heavyweight and often require processing of additional configuration portions beyond the state being directly acted upon, with corresponding cost in CPU cycles, negatively impacting both PCC stability LSP update rate capacity.
- o Security: when a PCC opens a configuration channel allowing a PCE to send configuration, a malicious PCE may take advantage of this ability to take over the PCC. In contrast, the PCEP extensions described in this document only allow a PCE control over a very limited set of LSP attributes.
- o Interoperability: each vendor has a proprietary information model for configuring LSP state, which limits interoperability of a stateful PCE with PCCs from different vendors. The PCEP extensions described in this document allow for a common information model for LSP state for all vendors.
- o Efficient State Synchronization: configuration channels may be heavyweight and unidirectional, therefore efficient state synchronization between a PCC and a PCE may be a problem.

### 3.2. Objectives

The objectives for the protocol extensions to support stateful PCE described in this document are as follows:

- o Allow a single PCC to interact with a mix of stateless and stateful PCEs simultaneously using the same protocol, i.e. PCEP.
- o Support efficient LSP state synchronization between the PCC and one or more active or passive stateful PCEs.
- o Allow a PCC to delegate control of its LSPs to an active stateful PCE such that a given LSP is under the control of a single PCE at any given time.

- \* A PCC may revoke this delegation at any time during the lifetime of the LSP. If LSP delegation is revoked while the PCEP session is up, the PCC MUST notify the PCE about the revocation.
- \* A PCE may return an LSP delegation at any point during the lifetime of the PCEP session. If LSP delegation is returned by the PCE while the PCEP session is up, the PCE MUST notify the PCC about the returned delegation.
- o Allow a PCE to control computation timing and update timing across all LSPs that have been delegated to it.
- o Enable uninterrupted operation of PCC's LSPs in the event of a PCE failure or while control of LSPs is being transferred between PCEs.

#### 4. New Functions to Support Stateful PCEs

Several new functions are required in PCEP to support stateful PCEs. A function can be initiated either from a PCC towards a PCE (C-E) or from a PCE towards a PCC (E-C). The new functions are:

Capability advertisement (E-C,C-E): both the PCC and the PCE must announce during PCEP session establishment that they support PCEP Stateful PCE extensions defined in this document.

LSP state synchronization (C-E): after the session between the PCC and a stateful PCE is initialized, the PCE must learn the state of a PCC's LSPs before it can perform path computations or update LSP attributes in a PCC.

LSP Update Request (E-C): a PCE requests modification of attributes on a PCC's LSP.

LSP State Report (C-E): a PCC sends an LSP state report to a PCE whenever the state of an LSP changes.

LSP control delegation (C-E,E-C): a PCC grants to a PCE the right to update LSP attributes on one or more LSPs; the PCE becomes the authoritative source of the LSP's attributes as long as the delegation is in effect (See Section 5.7); the PCC may withdraw the delegation or the PCE may give up the delegation at any time.

Similarly to [RFC5440], no assumption is made about the discovery method used by a PCC to discover a set of PCEs (e.g., via static configuration or dynamic discovery) and on the algorithm used to select a PCE.

## 5. Overview of Protocol Extensions

### 5.1. LSP State Ownership

In PCEP (defined in [RFC5440]), LSP state and operation are under the control of a PCC (a PCC may be an LSR or a management station). Attributes received from a PCE are subject to PCC's local policy. The PCEP extensions described in this document do not change this behavior.

An active stateful PCE may have control of a PCC's LSPs that were delegated to it, but the LSP state ownership is retained by the PCC. In particular, in addition to specifying values for LSP's attributes, an active stateful PCE also decides when to make LSP modifications.

Retaining LSP state ownership on the PCC allows for:

- o a PCC to interact with both stateless and stateful PCEs at the same time
- o a stateful PCE to only modify a small subset of LSP parameters, i.e. to set only a small subset of the overall LSP state; other parameters may be set by the operator, for example through command line interface (CLI) commands
- o a PCC to revert delegated LSP to an operator-defined default or to delegate the LSPs to a different PCE, if the PCC get disconnected from a PCE with currently delegated LSPs

### 5.2. New Messages

In this document, we define the following new PCEP messages:

Path Computation State Report (PCRpt): a PCEP message sent by a PCC to a PCE to report the status of one or more LSPs. Each LSP State Report in a PCRpt message MAY contain the actual LSP's path, bandwidth, operational and administrative status, etc. An LSP Status Report carried on a PCRpt message is also used in delegation or revocation of control of an LSP to/from a PCE. The PCRpt message is described in Section 6.1.

Path Computation Update Request (PCUpd): a PCEP message sent by a PCE to a PCC to update LSP parameters, on one or more LSPs. Each LSP Update Request on a PCUpd message MUST contain all LSP parameters that a PCE wishes to be set for a given LSP. An LSP Update Request carried on a PCUpd message is also used to return LSP delegations if at any point PCE no longer desires control of an LSP. The PCUpd message is described in Section 6.2.

The new functions defined in Section 4 are mapped onto the new messages as shown in the following table.

Function	Message
Capability Advertisement (E-C,C-E)	Open
State Synchronization (C-E)	PCRpt
LSP State Report (C-E)	PCRpt
LSP Control Delegation (C-E,E-C)	PCRpt, PCUpd
LSP Update Request (E-C)	PCUpd

Table 1: New Function to Message Mapping

### 5.3. Error Reporting

Error reporting is done using the procedures defined in [RFC5440], and reusing the applicable error types and error values of [RFC5440] wherever appropriate. The current document defines new error values for several error types to cover failures specific to stateful PCE.

### 5.4. Capability Advertisement

During PCEP Initialization Phase, PCEP Speakers (PCE or PCC) advertise their support of stateful PCEP extensions. A PCEP Speaker includes the "Stateful PCE Capability" TLV, described in Section 7.1.1, in the OPEN Object to advertise its support for PCEP stateful extensions. The Stateful Capability TLV includes the 'LSP Update' Flag that indicates whether the PCEP Speaker supports LSP parameter updates.

The presence of the Stateful PCE Capability TLV in PCC's OPEN Object indicates that the PCC is willing to send LSP State Reports whenever LSP parameters or operational status changes.

The presence of the Stateful PCE Capability TLV in PCE's OPEN message indicates that the PCE is interested in receiving LSP State Reports whenever LSP parameters or operational status changes.

The PCEP extensions for stateful PCEs MUST NOT be used if one or both PCEP Speakers have not included the Stateful PCE Capability TLV in their respective OPEN message. If the PCEP Speaker on the PCC supports the extensions of this draft but did not advertise this capability, then upon receipt of PCUpd message from the PCE, it MUST generate a PCErr with error-type 19 (Invalid Operation), error-value 2 (Attempted LSP Update Request if the stateful PCE capability was not advertised)(see Section 8.5) and it SHOULD terminate the PCEP



session. If the PCEP Speaker on the PCE supports the extensions of this draft but did not advertise this capability, then upon receipt of a PCRpt message from the PCC, it MUST generate a PCErr with error-type 19 (Invalid Operation), error-value 5 (Attempted LSP State Report if stateful PCE capability was not advertised) (see Section 8.5) and it SHOULD terminate the PCEP session.

LSP delegation and LSP update operations defined in this document may only be used if both PCEP Speakers set the LSP-UPDATE-CAPABILITY Flag in the "Stateful Capability" TLV to 'Updates Allowed (U Flag = 1)'. If this is not the case and LSP delegation or LSP update operations are attempted, then a PCErr with error-type 19 (Invalid Operation) and error-value 1 (Attempted LSP Update Request for a non-delegated LSP) (see Section 8.5) MUST be generated. Note that, even if one of the PCEP speakers does not set the LSP-UPDATE-CAPABILITY flag in its "Stateful Capability" TLV, a PCE can still operate as a passive stateful PCE by accepting LSP State Reports from the PCC in order to build and maintain an up to date view of the state of the PCC's LSPs.

#### 5.5. IGP Extensions for Stateful PCE Capabilities Advertisement

When PCCs are LSRs participating in the IGP (OSPF or IS-IS), and PCEs are either LSRs or servers also participating in the IGP, an effective mechanism for PCE discovery within an IGP routing domain consists of utilizing IGP advertisements. Extensions for the advertisement of PCE Discovery Information are defined for OSPF and for IS-IS in [RFC5088] and [RFC5089] respectively.

The PCE-CAP-FLAGS sub-TLV, defined in [RFC5089], is an optional sub-TLV used to advertise PCE capabilities. It MAY be present within the PCED sub-TLV carried by OSPF or IS-IS. [RFC5088] and [RFC5089] provide the description and processing rules for this sub-TLV when carried within OSPF and IS-IS, respectively.

The format of the PCE-CAP-FLAGS sub-TLV is included below for easy reference:

Type: 5

Length: Multiple of 4.

Value: This contains an array of units of 32 bit flags with the most significant bit as 0. Each bit represents one PCE capability.

PCE capability bits are defined in [RFC5088]. This document defines new capability bits for the stateful PCE as follows:

Bit	Capability
11	Active Stateful PCE capability
12	Passive Stateful PCE capability

Note that while active and passive stateful PCE capabilities may be advertised during discovery, PCEP Speakers that wish to use stateful PCEP MUST negotiate stateful PCEP capabilities during PCEP session setup, as specified in the current document. A PCC MAY initiate stateful PCEP capability negotiation at PCEP session setup even if it did not receive any IGP PCE capability advertisements.

## 5.6. State Synchronization

The purpose of State Synchronization is to provide a checkpoint-in-time state replica of a PCC's LSP state in a PCE. State Synchronization is performed immediately after the Initialization phase ([RFC5440]).

During State Synchronization, a PCC first takes a snapshot of the state of its LSPs state, then sends the snapshot to a PCE in a sequence of LSP State Reports. Each LSP State Report sent during State Synchronization has the SYNC Flag in the LSP Object set to 1. The set of LSPs for which state is synchronized with a PCE is determined by the PCC's local configuration (see more details in Section 9.1) and MAY also be determined by stateful PCEP capabilities defined in other documents, such as [I-D.ietf-pce-stateful-sync-optimizations].

The end of synchronization marker is a PCRpt message with the SYNC Flag set to 0 for an LSP Object with PLSP-ID equal to the reserved value 0 (see Section 7.3). In this case, the LSP Object SHOULD NOT include the SYMBOLIC-PATH-NAME TLV and SHOULD include the LSP-IDENTIFIERS TLV with the special value of all zeroes. The PCRpt message MUST include an empty ERO as its intended path and SHOULD NOT include the optional RRO object for its actual path. If the PCC has no state to synchronize, it SHOULD only send the end of synchronization marker.

A PCE SHOULD NOT send PCUpd messages to a PCC before State Synchronization is complete. A PCC SHOULD NOT send PCReq messages to a PCE before State Synchronization is complete. This is to allow the PCE to get the best possible view of the network before it starts computing new paths.

Either the PCE or the PCC MAY terminate the session using the PCEP session termination procedures during the synchronization phase. If the session is terminated, the PCE MUST clean up state it received from this PCC. The session reestablishment MUST be re-attempted per

the procedures defined in [RFC5440], including use of a back-off timer.

If the PCC encounters a problem which prevents it from completing the LSP state synchronization, it MUST send a PCErr message with error-type 20 (LSP State Synchronization Error) and error-value 5 (indicating an internal PCC error) to the PCE and terminate the session.

The PCE does not send positive acknowledgements for properly received synchronization messages. It MUST respond with a PCErr message with error-type 20 (LSP State Synchronization Error) and error-value 1 (indicating an error in processing the PCRpt) (see Section 8.5) if it encounters a problem with the LSP State Report it received from the PCC and it MUST terminate the session.

A PCE implementing a limit on the resources a single PCC can occupy, MUST send a PCNtf message with Notification Type 4 (Stateful PCE resource limit exceeded) and Notification Value 1 (Entering resource limit exceeded state) in response to the PCRpt message triggering this condition in the synchronization phase and MUST terminate the session.

The successful State Synchronization sequence is shown in Figure 1.

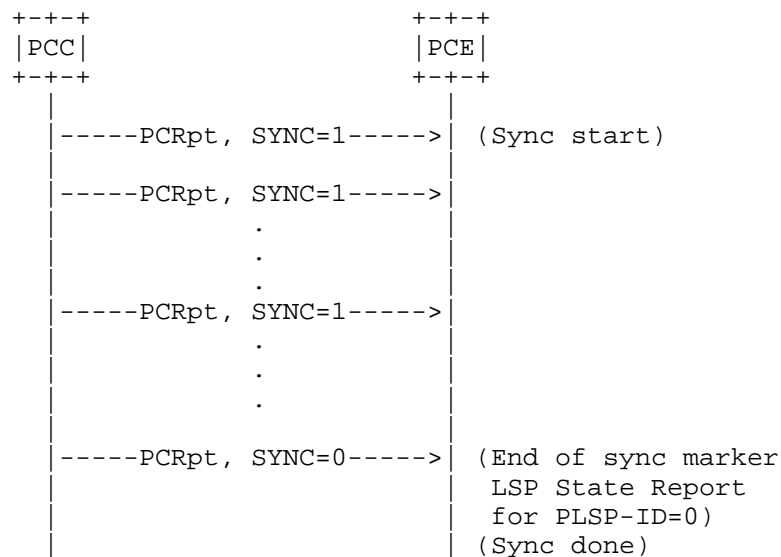


Figure 1: Successful state synchronization

The sequence where the PCE fails during the State Synchronization phase is shown in Figure 2.

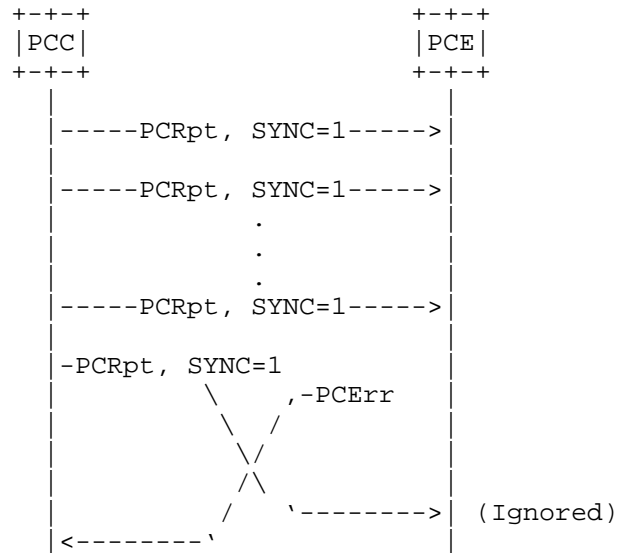


Figure 2: Failed state synchronization (PCE failure)

The sequence where the PCC fails during the State Synchronization phase is shown in Figure 3.

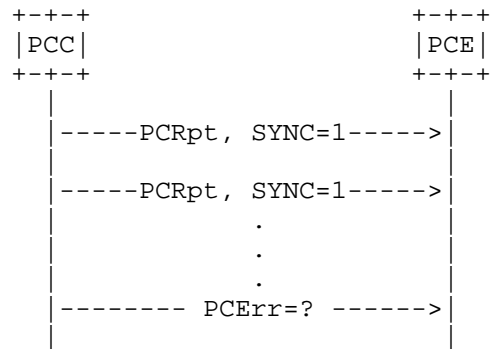


Figure 3: Failed state synchronization (PCC failure)

Optimizations to the synchronization procedures and alternate mechanisms of providing the synchronization function are outside the scope of this document and are discussed elsewhere (see [I-D.ietf-pce-stateful-sync-optimizations]).

### 5.7. LSP Delegation

If during Capability advertisement both the PCE and the PCC have indicated that they support LSP Update, then the PCC may choose to grant the PCE a temporary right to update (a subset of) LSP attributes on one or more LSPs. This is called "LSP Delegation", and it MAY be performed at any time after the Initialization phase, including during the State Synchronization phase.

A PCE MAY return an LSP delegation at any time if it no longer wishes to update the LSP's state. A PCC MAY revoke an LSP delegation at any time. Delegation, Revocation, and Return are done individually for each LSP.

In the event of a delegation being rejected or returned by a PCE, the PCC SHOULD react based on local policy. It can, for example, either retry delegating to the same PCE using an exponentially increasing timer or delegate to an alternate PCE.

#### 5.7.1. Delegating an LSP

A PCC delegates an LSP to a PCE by setting the Delegate flag in LSP State Report to 1. If the PCE does not accept the LSP Delegation, it MUST immediately respond with an empty LSP Update Request which has the Delegate flag set to 0. If the PCE accepts the LSP Delegation, it MUST set the Delegate flag to 1 when it sends an LSP Update Request for the delegated LSP (note that this may occur at a later time). The PCE MAY also immediately acknowledge a delegation by sending an empty LSP Update Request which has the Delegate flag set to 1.

The delegation sequence is shown in Figure 4.

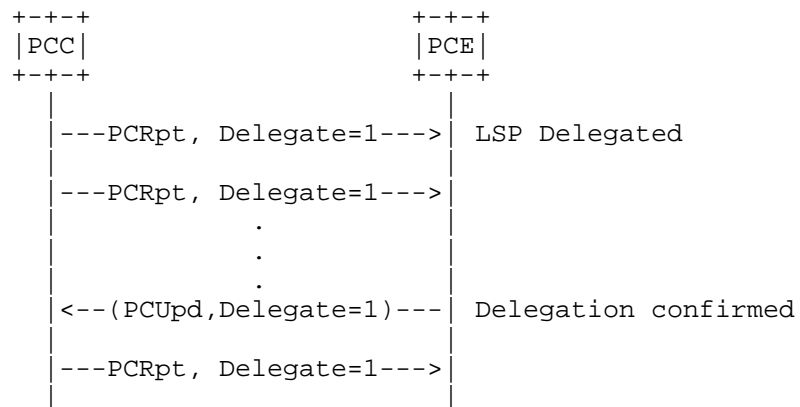


Figure 4: Delegating an LSP

Note that for an LSP to remain delegated to a PCE, the PCC MUST set the Delegate flag to 1 on each LSP State Report sent to the PCE.

#### 5.7.2. Revoking a Delegation

##### 5.7.2.1. Explicit Revocation

When a PCC decides that a PCE is no longer permitted to modify an LSP, it revokes that LSP's delegation to the PCE. A PCC may revoke an LSP delegation at any time during the LSP's life time. A PCC revoking an LSP delegation MAY immediately remove the updated parameters provided by the PCE and revert to the operator-defined parameters, but to avoid traffic loss, it SHOULD do so in a make-before-break fashion. If the PCC has received but not yet acted on PCUpd messages from the PCE for the LSP whose delegation is being revoked, then it SHOULD ignore these PCUpd messages when processing the message queue. All effects of all messages for which processing started before the revocation took place MUST be allowed to complete and the result MUST be given the same treatment as any LSP that had been previously delegated to the PCE (e.g. the state MAY immediately revert to the operator-defined parameters).

If a PCEP session with the PCE to which the LSP is delegated exists in the UP state during the revocation, the PCC MUST notify that PCE by sending an LSP State Report with the Delegate flag set to 0, as shown in Figure 5.

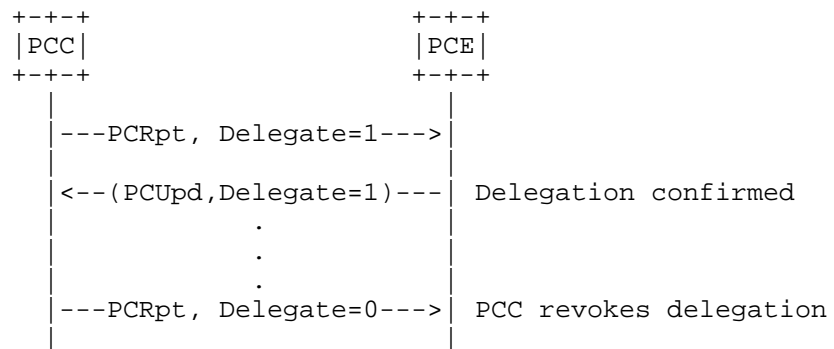


Figure 5: Revoking a Delegation

After an LSP delegation has been revoked, a PCE can no longer update LSP's parameters; an attempt to update parameters of a non-delegated LSP will result in the PCC sending a PCErr message with error-type 19 (Invalid Operation), error-value 1 (attempted LSP Update Request for a non-delegated LSP) (see Section 8.5).

#### 5.7.2.2. Revocation on Redelegating Timeout

When a PCC's PCEP session with a PCE terminates unexpectedly, the PCC MUST wait the time interval specified in Redelegating Timeout Interval before revoking LSP delegations to that PCE and attempting to redelegate LSPs to an alternate PCE. If a PCEP session with the original PCE can be reestablished before the Redelegating Timeout Interval timer expires, LSP delegations to the PCE remain intact.

Likewise, when a PCC's PCEP session with a PCE terminates unexpectedly, and the PCC does not succeed in redelegating its LSPs, the PCC MUST wait for the State Timeout Interval before flushing any LSP state associated with that PCE. Note that the State Timeout Interval timer may expire before the PCC has redelegated the LSPs to another PCE, for example if a PCC is not connected to any active stateful PCE or if no connected active stateful PCE accepts the delegation. In this case, the PCC MUST flush any LSP state set by the PCE upon expiration of the State Timeout Interval and revert to operator-defined default parameters or behaviors. This operation SHOULD be done in a make-before-break fashion.

The State Timeout Interval MUST be greater than or equal to the Redelegating Timeout Interval and MAY be set to infinity (meaning that until the PCC specifically takes action to change the parameters set by the PCE, they will remain intact).

### 5.7.3. Returning a Delegation

In order to keep a delegation, a PCE MUST set the Delegate flag to 1 on each LSP Update Request sent to the PCC. A PCE that no longer wishes to update an LSP's parameters SHOULD return the LSP delegation back to the PCC by sending an empty LSP Update Request which has the Delegate flag set to 0. If a PCC receives an LSP Update Request with the Delegate flag set to 0 (whether the LSP Update Request is empty or not), it MUST treat this as a delegation return.

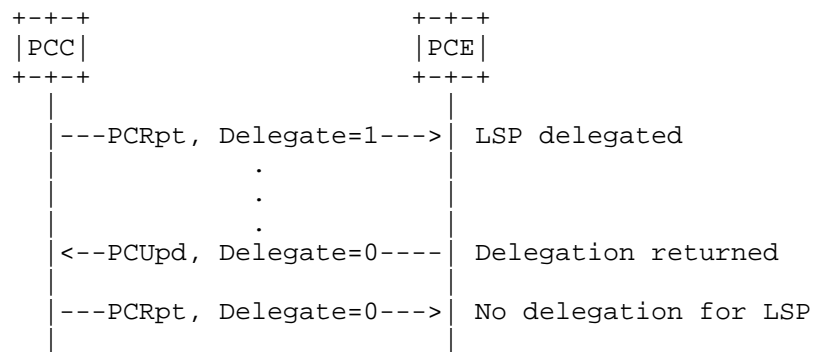


Figure 6: Returning a Delegation

If a PCC cannot delegate an LSP to a PCE (for example, if a PCC is not connected to any active stateful PCE or if no connected active stateful PCE accepts the delegation), the LSP delegation on the PCC will time out within a configurable Redelegating Timeout Interval and the PCC MUST flush any LSP state set by a PCE at the expiration of the State Timeout Interval and revert to operator-defined default parameters or behaviors.

### 5.7.4. Redundant Stateful PCEs

In a redundant configuration where one PCE is backing up another PCE, the backup PCE may have only a subset of the LSPs in the network delegated to it. The backup PCE does not update any LSPs that are not delegated to it. In order to allow the backup to operate in a hot-standby mode and avoid the need for state synchronization in case the primary fails, the backup receives all LSP State Reports from a PCC. When the primary PCE for a given LSP set fails, after expiry of the Redelegating Timeout Interval, the PCC SHOULD delegate to the redundant PCE all LSPs that had been previously delegated to the failed PCE. Assuming that the State Timeout Interval had been configured to be greater than the Redelegating Timeout Interval (as MANDATORY), and assuming that the primary and redundant PCEs take



similar decisions, this delegation change will not cause any changes to the LSP parameters.

#### 5.7.5. Redelegation on PCE Failure

On failure, the goal is to: 1) avoid any traffic loss on the LSPs that were updated by the PCE that crashed 2) minimize the churn in the network in terms of ownership of the LSPs, 3) not leave any "orphan" (undelegated) LSPs and 4) be able to control when the state that was set by the PCE can be changed or purged. The values chosen for the Redelegation Timeout and State Timeout values affect the ability to accomplish these goals.

This section summarizes the behaviour with regards to LSP delegation and LSP state on a PCE failure.

If the PCE crashes but recovers within the Redelegation Timeout, both the delegation state and the LSP state are kept intact.

If the PCE crashes but does not recover within the Redelegation Timeout, the delegation state is returned to the PCC. If the PCC can redelegate the LSPs to another PCE, and that PCE accepts the delegations, there will be no change in LSP state. If the PCC cannot redelegate the LSPs to another PCE, then upon expiration of the State Timeout Interval, the state set by the PCE is removed and the LSP reverts to operator-defined parameters, which may cause a change in the LSP state. Note that an operator may choose to use an infinite State Timeout Interval if he wishes to maintain the PCE state indefinitely. Note also that flushing the state should be implemented using make-before-break to avoid traffic loss.

If there is a standby PCE, the Redelegation Timeout may be set to 0 through policy on the PCC, causing the LSPs to be redelegated immediately to the PCC, which can delegate them immediately to the standby PCE. Assuming that the PCC can redelegate the LSP to the standby PCE within the State Timeout Interval, and assuming the standby PCE takes similar decisions as the failed PCE, the LSP state will be kept intact.

#### 5.8. LSP Operations

##### 5.8.1. Passive Stateful PCE Path Computation Request/Response

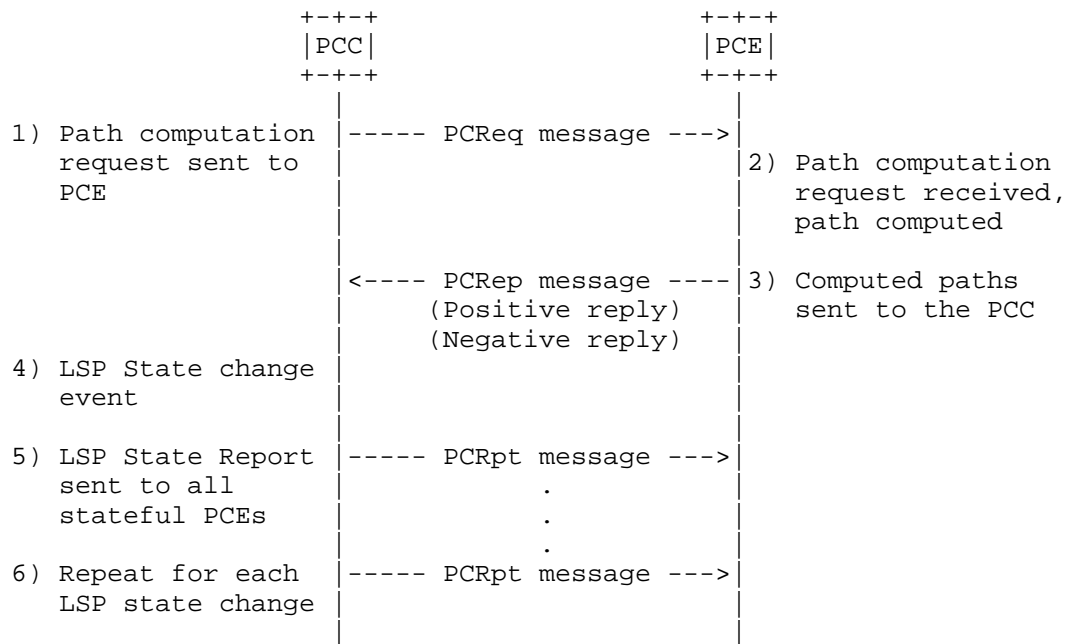


Figure 7: Passive Stateful PCE Path Computation Request/Response

Once a PCC has successfully established a PCEP session with a passive stateful PCE and the PCC's LSP state is synchronized with the PCE (i.e. the PCE knows about all PCC's existing LSPs), if an event is triggered that requires the computation of a set of paths, the PCC sends a path computation request to the PCE ([RFC5440], Section 4.2.3). The PCReq message MAY contain the LSP Object to identify the LSP for which the path computation is requested.

Upon receiving a path computation request from a PCC, the PCE triggers a path computation and returns either a positive or a negative reply to the PCC ([RFC5440], Section 4.2.4).

Upon receiving a positive path computation reply, the PCC receives a set of computed paths and starts to setup the LSPs. For each LSP, it MAY send an LSP State Report carried on a PCRpt message to the PCE, indicating that the LSP's status is "Going-up".

Once an LSP is up or active, the PCC MUST send an LSP State Report carried on a PCRpt message to the PCE, indicating that the LSP's status is 'Up' or 'Active' respectively. If the LSP could not be set up, the PCC MUST send an LSP State Report indicating that the LSP is "Down" and stating the cause of the failure. Note that due to timing constraints, the LSP status may change from 'Going-up' to 'Up' (or

'Down') before the PCC has had a chance to send an LSP State Report indicating that the status is 'Going-up'. In such cases, the PCC MAY choose to only send the PCRpt indicating the latest status ('Active', 'Up' or 'Down').

Upon receiving a negative reply from a PCE, a PCC MAY resend a modified request or take any other appropriate action. For each requested LSP, it SHOULD also send an LSP State Report carried on a PCRpt message to the PCE, indicating that the LSP's status is 'Down'.

There is no direct correlation between PCRep and PCRpt messages. For a given LSP, multiple LSP State Reports will follow a single PCRep message, as a PCC notifies a PCE of the LSP's state changes.

A PCC MUST send each LSP State Report to each stateful PCE that is connected to the PCC.

Note that a single PCRpt message MAY contain multiple LSP State Reports.

The passive stateful model for stateful PCEs is described in [RFC4655], Section 6.8.

#### 5.8.2. Switching from Passive Stateful to Active Stateful

This section deals with the scenario of an LSP transitioning from a passive stateful to an active stateful mode of operation. When the LSP has no working path, prior to delegating the LSP, the PCC MUST first use the procedure defined in Section 5.8.1 to request the initial path from the PCE. This is required because the action of delegating the LSP to a PCE using a PCRpt message is not an explicit request to the PCE to compute a path for the LSP. The only explicit way for a PCC to request a path from PCE is to send a PCReq message. The PCRpt message MUST NOT be used by the PCC to attempt to request a path from the PCE.

When the LSP is delegated after its setup, it may be useful for the PCC to communicate to the PCE the locally configured intended configuration parameters, so that the PCE may reuse them in its computations. Such parameters MAY be acquired through an out of band channel, or MAY be communicated in the PCRpt message delegating the LSPs, by including them as part of the intended-attribute-list as explained in Section 6.1. An implementation MAY allow policies on the PCC to determine the configuration parameters to be sent to the PCE.

## 5.8.3. Active Stateful PCE LSP Update

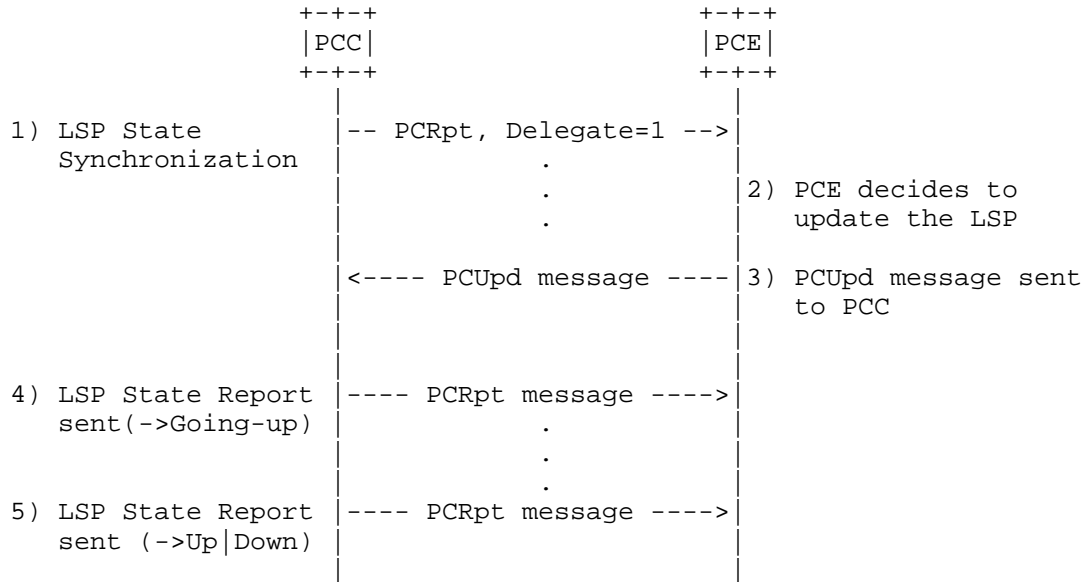


Figure 8: Active Stateful PCE

Once a PCC has successfully established a PCEP session with an active stateful PCE, the PCC's LSP state is synchronized with the PCE (i.e. the PCE knows about all PCC's existing LSPs). After LSPs have been delegated to the PCE, the PCE can modify LSP parameters of delegated LSPs.

To update an LSP, a PCE MUST send the PCC an LSP Update Request using a PCUpd message. The LSP Update Request contains a variety of objects that specify the set of constraints and attributes for the LSP's path. Each LSP Update Request MUST have a unique identifier, the SRP-ID-number, carried in the SRP (Stateful PCE Request Parameters) Object described in Section 7.2. The SRP-ID-number is used to correlate errors and state reports to LSP Update Requests. A single PCUpd message MAY contain multiple LSP Update Requests.

Upon receiving a PCUpd message the PCC starts to setup LSPs specified in LSP Update Requests carried in the message. For each LSP, it MAY send an LSP State Report carried on a PCRpt message to the PCE, indicating that the LSP's status is 'Going-up'. If the PCC decides that the LSP parameters proposed in the PCUpd message are unacceptable, it MUST report this error by including the LSP-ERROR-CODE TLV (Section 7.3.3) with LSP error-value="Unacceptable parameters" in the LSP object in the PCRpt message to the PCE. Based

on local policy, it MAY react further to this error by revoking the delegation. If the PCC receives a PCUpd message for an LSP object identified with a PLSP-ID that does not exist on the PCC, it MUST generate a PCErr with error-type 19 (Invalid Operation), error-value 3, (Attempted LSP Update Request for an LSP identified by an unknown PSP-ID) (see Section 8.5).

Once an LSP is up, the PCC MUST send an LSP State Report (PCRpt message) to the PCE, indicating that the LSP's status is 'Up'. If the LSP could not be set up, the PCC MUST send an LSP State Report indicating that the LSP is 'Down' and stating the cause of the failure. A PCC MAY compress LSP State Reports to only reflect the most up to date state, as discussed in the previous section.

A PCC MUST send each LSP State Report to each stateful PCE that is connected to the PCC.

PCErr and PCRpt messages triggered as a result of a PCUpd message MUST include the SRP-ID-number from the PCUpd. This provides correlation of requests and errors and acknowledgement of state processing. The PCC MAY compress state when processing PCUpd. In this case, receipt of a higher SRP-ID-number implicitly acknowledges processing all the updates with lower SRP-ID-number for the specific LSP (as per Section 7.2).

A PCC MUST NOT send to any PCE a Path Computation Request for a delegated LSP. Should the PCC decide it wants to issue a Path Computation Request on a delegated LSP, it MUST perform Delegation Revocation procedure first.

## 5.9. LSP Protection

LSP protection and interaction with stateful PCE, as well as the extensions necessary to implement this functionality will be discussed in a separate document.

## 5.10. PCEP Sessions

A permanent PCEP session MUST be established between a stateful PCE and the PCC. In the case of session failure, session reestablishment MUST be re-attempted per the procedures defined in [RFC5440].

## 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. For each PCEP message type, a set of rules is defined that specify the set of objects that the message can carry.

### 6.1. The PCRpt Message

A Path Computation LSP State Report message (also referred to as PCRpt message) is a PCEP message sent by a PCC to a PCE to report the current state of an LSP. A PCRpt message can carry more than one LSP State Reports. A PCC can send an LSP State Report either in response to an LSP Update Request from a PCE, or asynchronously when the state of an LSP changes. The Message-Type field of the PCEP common header for the PCRpt message is 10.

The format of the PCRpt message is as follows:

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

Where:

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

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

Where:

```
<path> ::= <intended-path>
           [<actual-attribute-list><actual-path>]
           <intended-attribute-list>
```

```
<actual-attribute-list> ::= [<BANDWIDTH>]
                           [<metric-list>]
```

Where:

```
<intended-path> is represented by the ERO object defined in
section 7.9 of [RFC5440].
<actual-attribute-list> consists of the actual computed and
signaled values of the <BANDWIDTH> and <metric-lists> objects
defined in [RFC5440].
<actual-path> is represented by the RRO object defined in
section 7.10 of [RFC5440].
<intended-attribute-list> is the attribute-list defined in
section 6.5 of [RFC5440] and extended by PCEP extensions.
```

The SRP object (see Section 7.2) is OPTIONAL. If the PCRpt message is not in response to a PCUpd message, the SRP object MAY be omitted. When the PCC does not include the SRP object, the PCE MUST treat this as an SRP object with an SRP-ID-number equal to the reserved value 0x00000000. The reserved value 0x00000000 indicates that the state reported is not as a result of processing a PCUpd message.

If the PCRpt message is in response to a PCUpd message, the SRP object MUST be included and the value of the SRP-ID-number in the SRP Object MUST be the same as that sent in the PCUpd message that triggered the state that is reported. If the PCC compressed several PCUpd messages for the same LSP by only processing the one with the highest number, then it should use the SRP-ID-number of that request. No state compression is allowed for state reporting, e.g. PCRpt messages MUST NOT be pruned from the PCC's egress queue even if subsequent operations on the same LSP have been completed before the PCRpt message has been sent to the TCP stack. The PCC MUST explicitly report state changes (including removal) for paths it manages.

The LSP object (see Section 7.3) is REQUIRED, and it MUST be included in each LSP State Report on the PCRpt message. 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).

If the LSP transitioned to non-operational state, the PCC SHOULD include the LSP-ERROR-TLV (Section 7.3.3) with the relevant LSP Error Code to report the error to the PCE.

The intended path, represented by the ERO object, is REQUIRED. If the ERO object is missing, the receiving PCE MUST send a PCErr message with Error-type=6 (Mandatory Object missing) and Error-value 9 (ERO object missing). The ERO may be empty if the PCE does not have a path for a delegated LSP.

The actual path, represented by the RRO object, SHOULD be included in PCRpt by the PCC when the path is up or active, but MAY be omitted if the path is down due to a signaling error or another failure.

The intended-attribute-list maps to the attribute-list in Section 6.5 of [RFC5440] and is used to convey the requested parameters of the LSP path. This is needed in order to support the switch from passive to active stateful PCE as described in Section 5.8.2. When included as part of the intended-attribute-list, the meaning of the BANDWIDTH object is the requested bandwidth as intended by the operator. In this case, the BANDWIDTH Object-Type of 1 SHOULD be used. Similarly, to indicate a limiting constraint, the METRIC object SHOULD be included as part of the intended-attribute-list with the B flag set and with a specific metric value. To indicate the optimization metric, the METRIC object SHOULD be included as part of the intended-attribute-list with the B flag unset and the metric value set to zero. Note that the intended-attribute-list is optional and thus may be omitted. In this case, the PCE MAY use the values in the actual-attribute-list as the requested parameters for the path.

The actual-attribute-list consists of the actual computed and signaled values of the BANDWIDTH and METRIC objects defined in [RFC5440]. When included as part of the actual-attribute-list, Object-Type 2 ([RFC5440]) SHOULD be used for the BANDWIDTH object and the C flag SHOULD be set in the METRIC object ([RFC5440]).

Note that the ordering of intended-path, actual-attribute-list, actual-path and intended-attribute-list is chosen to retain compatibility with implementations of an earlier version of this standard.

A PCE may choose to implement a limit on the resources a single PCC can occupy. If a PCRpt is received that causes the PCE to exceed this limit, the PCE MUST notify the PCC using a PCNtf message with Notification Type 4 (Stateful PCE resource limit exceeded) and Notification Value 1 (Entering resource limit exceeded state) and MUST terminate the session.

## 6.2. The PCUpd Message

A Path Computation LSP Update Request message (also referred to as PCUpd message) is a PCEP message sent by a PCE to a PCC to update attributes of an LSP. A PCUpd message can carry more than one LSP Update Request. The Message-Type field of the PCEP common header for the PCUpd message is 11.

The format of a PCUpd message is as follows:

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

Where:

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

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

Where:

```
<path> ::= <intended-path><intended-attribute-list>
```

Where:

```
<intended-path> is represented by the ERO object defined in
section 7.9 of [RFC5440].
<intended-attribute-list> is the attribute-list defined in [RFC5440]
and extended by PCEP extensions.
```

There are three mandatory objects that MUST be included within each LSP Update Request in the PCUpd message: the SRP Object (see



Section 7.2), the LSP object (see Section 7.3) and the ERO object (as defined in [RFC5440], which represents the intended path. 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). 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). If the ERO object is missing, the receiving PCC MUST send a PCErr message with Error-type=6 (Mandatory Object missing) and Error-value=9 (ERO object missing).

The ERO in the PCUpd may be empty if the PCE cannot find a valid path for a delegated LSP. One typical situation resulting in this empty ERO carried in the PCUpd message is that a PCE can no longer find a strict SRLG-disjoint path for a delegated LSP after a link failure. The PCC SHOULD implement a local policy to decide the appropriate action to be taken: either tear down the LSP, or revoke the delegation and use a locally computed path, or keep the existing LSP.

A PCC only acts on an LSP Update Request if permitted by the local policy configured by the network manager. Each LSP Update Request that the PCC acts on results in an LSP setup operation. An LSP Update Request MUST contain all LSP parameters that a PCE wishes to be set for the LSP. A PCC MAY set missing parameters from locally configured defaults. If the LSP specified in the Update Request is already up, it will be re-signaled.

The PCC SHOULD minimize the traffic interruption, and MAY use the make-before-break procedures described in [RFC3209] in order to achieve this goal. If the make-before-break procedures are used, two paths will briefly co-exist. The PCC MUST send separate PCRpt messages for each, identified by the LSP-IDENTIFIERS TLV. When the old path is torn down after the head end switches over the traffic, this event MUST be reported by sending a PCRpt message with the LSP-IDENTIFIERS-TLV of the old path and the R bit set. The SRP-ID-number that the PCC associates with this PCRpt MUST be 0x00000000. Thus, a make-before-break operation will typically result in at least two PCRpt messages, one for the new path and one for the removal of the old path (more messages may be possible if intermediate states are reported).

If the path setup fails due to an RSVP signaling error, the error is reported to the PCE. The PCC will not attempt to resignal the path until it is prompted again by the PCE with a subsequent PCUpd message.

A PCC MUST respond with an LSP State Report to each LSP Update Request it processed to indicate the resulting state of the LSP in

the network (even if this processing did not result in changing the state of the LSP). The SRP-ID-number included in the PCRpt MUST match that in the PCUpd. A PCC MAY respond with multiple LSP State Reports to report LSP setup progress of a single LSP. In that case, the SRP-ID-number MUST be included for the first message, for subsequent messages the reserved value 0x00000000 SHOULD be used.

Note that a PCC MUST process all LSP Update Requests - for example, an LSP Update Request is sent when a PCE returns delegation or puts an LSP into non-operational state. The protocol relies on TCP for message-level flow control.

If the rate of PCUpd messages sent to a PCC for the same target LSP exceeds the rate at which the PCC can signal LSPs into the network, the PCC MAY perform state compression on its ingress queue. The compression algorithm is based on the fact that each PCUpd request contains the complete LSP state the PCE wishes to be set and works as follows: when the PCC starts processing a PCUpd message at the head of its ingress queue, it may search the queue forward for more recent PCUpd messages pertaining that particular LSP, prune all but the latest one from the queue and process only the last one as that request contains the most up-to-date desired state for the LSP. The PCC MUST NOT send PCRpt nor PCErr messages for requests which were pruned from the queue in this way. This compression step may be performed only while the LSP is not being signaled, e.g. if two PCUpd arrive for the same LSP in quick succession and the PCC started the signaling of the changes relevant to the first PCUpd, then it MUST wait until the signaling finishes (and report the new state via a PCRpt) before attempting to apply the changes indicated in the second PCUpd.

Note also that it is up to the PCE to handle inter-LSP dependencies; for example, if ordering of LSP set-ups is required, the PCE has to wait for an LSP State Report for a previous LSP before starting the update of the next LSP.

If the PCUpd cannot be satisfied (for example due to unsupported object or TLV), the PCC MUST respond with a PCErr message indicating the failure (see Section 7.3.3).

### 6.3. The PCErr Message

If the stateful PCE capability has been advertised on the PCEP session, the PCErr message MAY include the SRP object. If the error reported is the result of an LSP update request, then the SRP-ID-number MUST be the one from the PCUpd that triggered the error. If the error is unsolicited, the SRP object MAY be omitted. This is

equivalent to including an SRP object with SRP-ID-number equal to the reserved value 0x00000000.

The format of a PCErr message from [RFC5440] is extended as follows:

```

<PCErr Message> ::= <Common Header>
                    ( <error-obj-list> [<Open>] ) | <error>
                    [<error-list>]

<error-obj-list> ::= <PCEP-ERROR> [<error-obj-list>]

<error> ::= [<request-id-list> | <stateful-request-id-list>]
           <error-obj-list>

<request-id-list> ::= <RP> [<request-id-list>]

<stateful-request-id-list> ::= <SRP> [<stateful-request-id-list>]

<error-list> ::= <error> [<error-list>]

```

#### 6.4. The PCReq Message

A PCC MAY include the LSP object in the PCReq message (see Section 7.3) if the stateful PCE capability has been negotiated on a PCEP session between the PCC and a PCE.

The definition of the PCReq message from [RFC5440] is extended to optionally include the LSP object after the END-POINTS object. The encoding from [RFC5440] will become:

```

<PCReq Message> ::= <Common Header>
                    [<svec-list>]
                    <request-list>

```

Where:

```

<svec-list> ::= <SVEC> [<svec-list>]
<request-list> ::= <request> [<request-list>]

<request> ::= <RP>
              <END-POINTS>
              [<LSP>]
              [<LSPA>]
              [<BANDWIDTH>]
              [<metric-list>]
              [<RRO> [<BANDWIDTH>]]
              [<IRO>]
              [<LOAD-BALANCING>]

```

## 6.5. The PCRep Message

A PCE MAY include the LSP object in the PCRep message (see (Section 7.3) if the stateful PCE capability has been negotiated on a PCEP session between the PCC and the PCE and the LSP object was included in the corresponding PCReq message from the PCC.

The definition of the PCRep message from [RFC5440] is extended to optionally include the LSP object after the RP object. The encoding from [RFC5440] will become:

```
<PCRep Message> ::= <Common Header>
                        <response-list>
```

Where:

```
<response-list> ::= <response> [<response-list>]

<response> ::= <RP>
                [<LSP>]
                [<NO-PATH>]
                [<attribute-list>]
                [<path-list>]
```

## 7. Object Formats

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 the current document MUST be set to 0 on transmission and SHOULD be ignored on receipt since the P and I flags are exclusively related to path computation requests.

### 7.1. OPEN Object

This document defines one new optional TLV for use in the OPEN Object.

#### 7.1.1. Stateful PCE Capability TLV

The STATEFUL-PCE-CAPABILITY TLV is an optional TLV for use in the OPEN Object for stateful PCE capability advertisement. Its format is shown in the following figure:



Figure 9: STATEFUL-PCE-CAPABILITY TLV format

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

The value comprises a single field - Flags (32 bits):

U (LSP-UPDATE-CAPABILITY - 1 bit): if set to 1 by a PCC, the U Flag indicates that the PCC allows modification of LSP parameters; if set to 1 by a PCE, the U Flag indicates that the PCE is capable of updating LSP parameters. The LSP-UPDATE-CAPABILITY Flag must be advertised by both a PCC and a PCE for PCUpd messages to be allowed on a PCEP session.

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

A PCEP speaker operating in passive stateful PCE mode advertises the stateful PCE capability with the U flag set to 0. A PCEP speaker operating in active stateful PCE mode advertises the stateful PCE capability with the U Flag set to 1.

Advertisement of the stateful PCE capability implies support of LSPs that are signaled via RSVP, as well as the objects, TLVs and procedures defined in this document.

## 7.2. SRP Object

The SRP (Stateful PCE Request Parameters) object MUST be carried within PCUpd messages and MAY be carried within PCRpt and PCErr messages. The SRP object is used to correlate between update requests sent by the PCE and the error reports and state reports sent by the PCC.

SRP Object-Class is 33.

SRP Object-Type is 1.

The format of the SRP object body is shown in Figure 10:

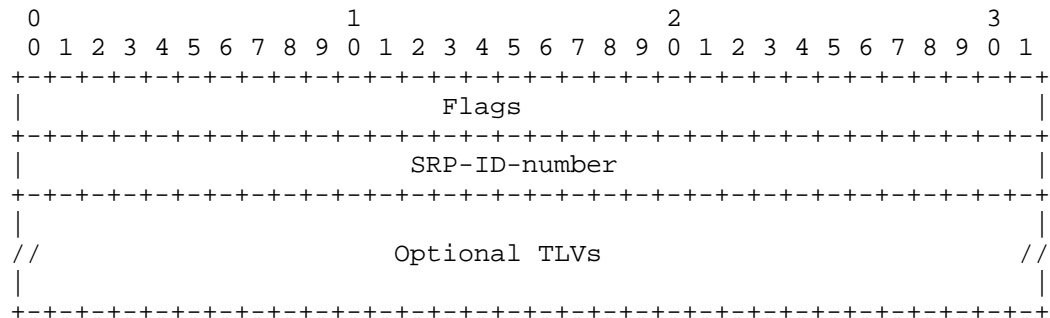


Figure 10: The SRP Object format

The SRP object body has a variable length and may contain additional TLVs.

Flags (32 bits): None defined yet.

SRP-ID-number (32 bits): The SRP-ID-number value in the scope of the current PCEP session uniquely identify the operation that the PCE has requested the PCC to perform on a given LSP. The SRP-ID-number is incremented each time a new request is sent to the PCC, and may wrap around.

The values 0x00000000 and 0xFFFFFFFF are reserved.

Optional TLVs MAY be included within the SRP object body. The specification of such TLVs is outside the scope of this document.

Every request to update an LSP receives a new SRP-ID-number. This number is unique per PCEP session and is incremented each time an operation is requested from the PCE. Thus, for a given LSP there may be more than one SRP-ID-number unacknowledged at a given time. The value of the SRP-ID-number is echoed back by the PCC in PCErr and PCRpt messages to allow for correlation between requests made by the PCE and errors or state reports generated by the PCC. If the error or report were not as a result of a PCE operation (for example in the case of a link down event), the reserved value of 0x00000000 is used for the SRP-ID-number. The absence of the SRP object is equivalent to an SRP object with the reserved value of 0x00000000. An SRP-ID-number is considered unacknowledged and cannot be reused until a PCErr or PCRpt arrives with an SRP-ID-number equal or higher for the same LSP. In case of SRP-ID-number wrapping the last SRP-ID-number before the wrapping MUST be explicitly acknowledged, to avoid a situation where SRP-ID-numbers remain unacknowledged after the wrap.

This means that the PCC may need to issue two PCUpd messages on detecting a wrap.

### 7.3. LSP Object

The LSP object MUST be present within PCRpt and PCUpd messages. The LSP object MAY be carried within PCReq and PCRep messages if the stateful PCE capability has been negotiated on the session. The LSP object contains a set of fields used to specify the target LSP, the operation to be performed on the LSP, and LSP Delegation. It also contains a flag indicating to a PCE that the LSP state synchronization is in progress. This document focuses on LSPs that are signaled with RSVP, many of the TLVs used with the LSP object mirror RSVP state.

LSP Object-Class is 32.

LSP Object-Type is 1.

The format of the LSP object body is shown in Figure 11:

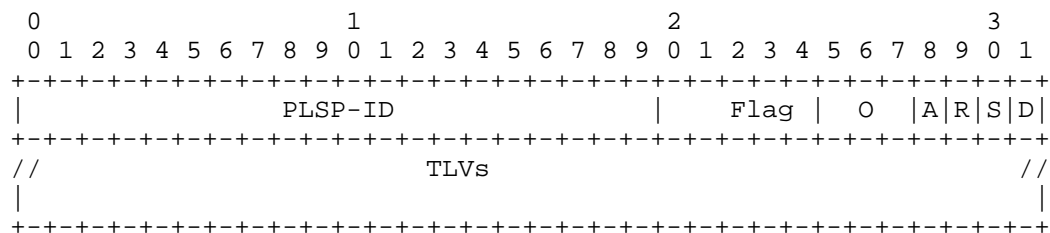


Figure 11: The LSP Object format

**PLSP-ID (20 bits):** 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. The PCC will advertise the same PLSP-ID on all PCEP sessions it maintains at a given time. The mapping of the Symbolic Path Name to PLSP-ID is communicated to the PCE by sending a PCRpt message containing the SYMBOLIC-PATH-NAME TLV. All subsequent PCEP messages then address the LSP by the PLSP-ID. The values of 0 and 0xFFFFF are reserved. Note that the PLSP-ID is a value that is constant for the lifetime of the PCEP session, during which time for an RSVP-signaled LSP there might be a different RSVP identifiers (LSP-id, tunnel-id) allocated to it.

**Flags (12 bits),** starting from the least significant bit:

**D (Delegate - 1 bit):** On a PCRpt message, the D Flag set to 1 indicates that the PCC is delegating the LSP to the PCE. On a

PCUpd message, the D flag set to 1 indicates that the PCE is confirming the LSP Delegation. To keep an LSP delegated to the PCE, the PCC must set the D flag to 1 on each PCRpt message for the duration of the delegation - the first PCRpt with the D flag set to 0 revokes the delegation. To keep the delegation, the PCE must set the D flag to 1 on each PCUpd message for the duration of the delegation - the first PCUpd with the D flag set to 0 returns the delegation.

S (SYNC - 1 bit): The S Flag MUST be set to 1 on each PCRpt sent from a PCC during State Synchronization. The S Flag MUST be set to 0 in other messages sent from the PCC. When sending a PCUpd message, the PCE MUST set the S Flag to 0.

R(Remove - 1 bit): On PCRpt messages the R Flag indicates that the LSP has been removed from the PCC and the PCE SHOULD remove all state from its database. Upon receiving an LSP State Report with the R Flag set to 1 for an RSVP-signaled LSP, the PCE SHOULD remove all state for the path identified by the LSP-IDENTIFIERS TLV from its database. When the all-zeros LSP-IDENTIFIERS TLV is used, the PCE SHOULD remove all state for the PLSP-ID from its database. When sending a PCUpd message, the PCE MUST set the R Flag to 0.

A(Administrative - 1 bit): On PCRpt messages, the A Flag indicates the PCC's target operational status for this LSP. On PCUpd messages, the A Flag indicates the LSP status that the PCE desires for this LSP. In both cases, a value of '1' means that the desired operational state is active, and a value of '0' means that the desired operational state is inactive. A PCC ignores the A flag on a PCUpd message unless the operator's policy allows the PCE to control the corresponding LSP's administrative state.

O(Operational - 3 bits): On PCRpt messages, the O Field represents the operational status of the LSP.

The following values are defined:

0 - DOWN: not active.

1 - UP: signalled.

2 - ACTIVE: up and carrying traffic.

3 - GOING-DOWN: LSP is being torn down, resources are being released.

4 - GOING-UP: LSP is being signalled.



5-7 - Reserved: these values are reserved for future use.

Unassigned bits are considered reserved. They MUST be set to 0 on transmission and MUST be ignored on receipt. When sending a PCUpd message, the PCE MUST set the O Field to 0.

TLVs that may be included in the LSP Object are described in the following sections. Other optional TLVs, that are not defined in this document, MAY also be included within the LSP Object body.

#### 7.3.1. LSP-IDENTIFIERS TLVs

The LSP-IDENTIFIERS TLV MUST be included in the LSP object in PCRpt messages for RSVP-signaled LSPs. If the TLV is missing, the PCE will generate an error with error-type 6 (mandatory object missing) and error-value 11 (LSP-IDENTIFIERS TLV missing) and close the session. The LSP-IDENTIFIERS TLV MAY be included in the LSP object in PCUpd messages for RSVP-signaled LSPs. The special value of all zeros for this TLV is used to refer to all paths pertaining to a particular PLSP-ID. There are two LSP-IDENTIFIERS TLVs, one for IPv4 and one for IPv6.

It is the responsibility of the PCC to send to the PCE the identifiers for each RSVP incarnation of the tunnel. For example, in a make-before-break scenario, the PCC MUST send a separate PCRpt for the old and for the reoptimized paths, and explicitly report removal of any of these paths using the R bit in the LSP object.

The format of the IPV4-LSP-IDENTIFIERS TLV is shown in the following figure:

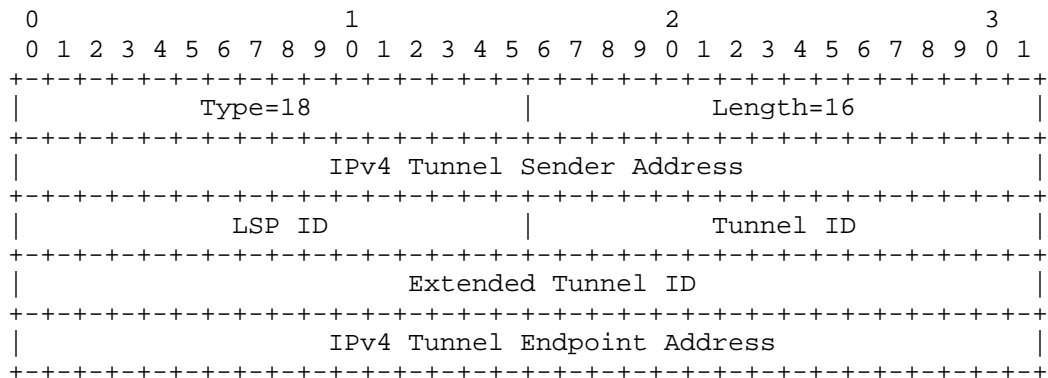


Figure 12: IPV4-LSP-IDENTIFIERS TLV format

The type (16 bits) of the TLV is 18. The length field is 16 bit-long and has a fixed value of 16. The value contains the following fields:

IPv4 Tunnel Sender Address: contains the sender node's IPv4 address, as defined in [RFC3209], Section 4.6.2.1 for the LSP\_TUNNEL\_IPv4 Sender Template Object.

LSP ID: contains the 16-bit 'LSP ID' identifier defined in [RFC3209], Section 4.6.2.1 for the LSP\_TUNNEL\_IPv4 Sender Template Object. A value of 0 MUST be used if the LSP is not yet signaled.

Tunnel ID: contains the 16-bit 'Tunnel ID' identifier defined in [RFC3209], Section 4.6.1.1 for the LSP\_TUNNEL\_IPv4 Session Object.

Extended Tunnel ID: contains the 32-bit 'Extended Tunnel ID' identifier defined in [RFC3209], Section 4.6.1.1 for the LSP\_TUNNEL\_IPv4 Session Object.

IPv4 Tunnel Endpoint Address: contains the egress node's IPv4 address, as defined in [RFC3209], Section 4.6.1.1 for the LSP\_TUNNEL\_IPv4 Sender Template Object.

The format of the IPV6-LSP-IDENTIFIERS TLV is shown in the following figure:

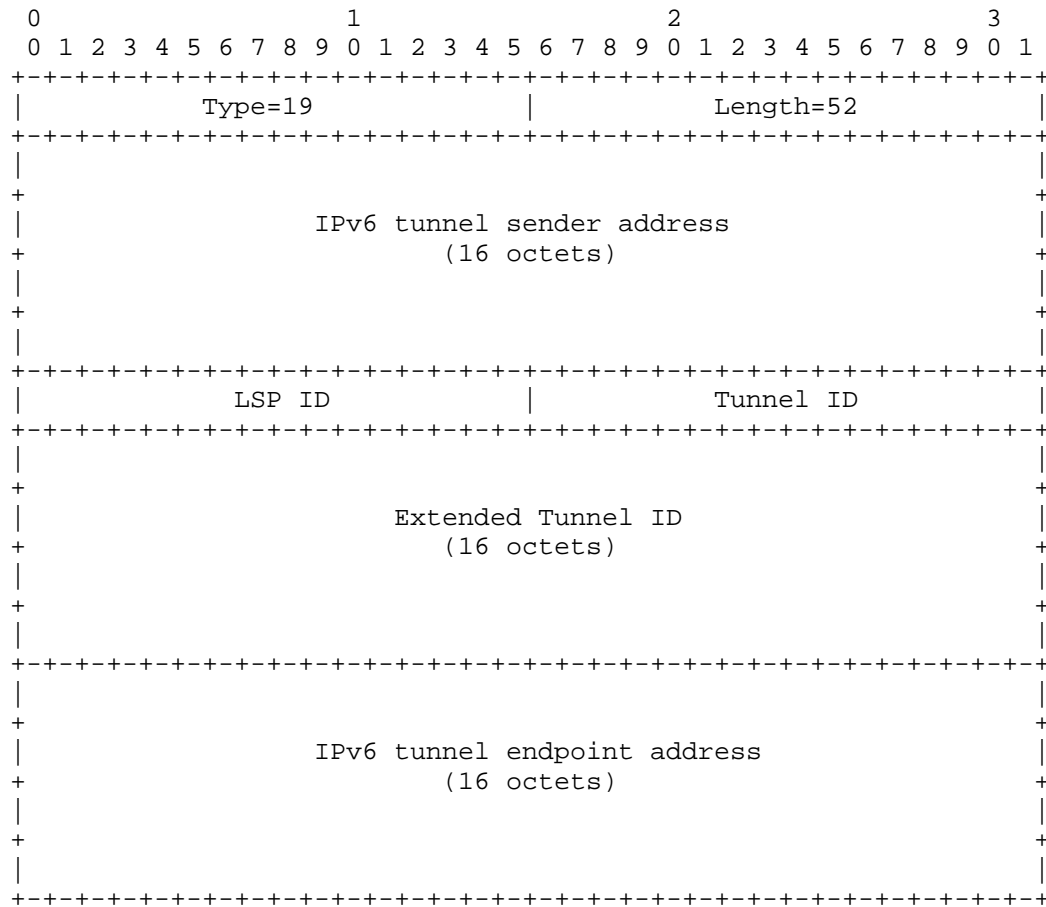


Figure 13: IPV6-LSP-IDENTIFIERS TLV format

The type (16 bits) of the TLV is 19. The length field is 16 bit-long and has a fixed value of 52. The value contains the following fields:

**IPv6 Tunnel Sender Address:** contains the sender node's IPv6 address, as defined in [RFC3209], Section 4.6.2.2 for the LSP\_TUNNEL\_IPv6 Sender Template Object.

**LSP ID:** contains the 16-bit 'LSP ID' identifier defined in [RFC3209], Section 4.6.2.2 for the LSP\_TUNNEL\_IPv6 Sender Template Object. A value of 0 MUST be used if the LSP is not yet signaled.

**Tunnel ID:** contains the 16-bit 'Tunnel ID' identifier defined in [RFC3209], Section 4.6.1.2 for the LSP\_TUNNEL\_IPv6 Session Object.

Extended Tunnel ID: contains the 128-bit 'Extended Tunnel ID' identifier defined in [RFC3209], Section 4.6.1.2 for the LSP\_TUNNEL\_IPv6 Session Object.

IPv6 Tunnel Endpoint Address: contains the egress node's IPv6 address, as defined in [RFC3209], Section 4.6.1.2 for the LSP\_TUNNEL\_IPv6 Session Object.

The Tunnel ID remains constant over the life time of a tunnel.

### 7.3.2. Symbolic Path Name TLV

Each LSP MUST have a symbolic path name that is unique in the PCC. The symbolic path name is a human-readable string that identifies an LSP in the network. The symbolic path name MUST remain constant throughout an LSP's lifetime, which may span across multiple consecutive PCEP sessions and/or PCC restarts. The symbolic path name MAY be specified by an operator in a PCC's configuration. If the operator does not specify a unique symbolic name for an LSP, then the PCC MUST auto-generate one.

The PCE uses the symbolic path name as a stable identifier for the LSP. If the PCEP session restarts, or the PCC restarts, or the PCC re-delegates the LSP to a different PCE, the symbolic path name for the LSP remains constant and can be used to correlate across the PCEP session instances.

The other protocol identifiers for the LSP cannot reliably be used to identify the LSP across multiple PCEP sessions, for the following reasons.

- o The PLSP-ID is unique only within the scope of a single PCEP session.
- o The LSP-IDENTIFIERS TLV is only guaranteed to be present for LSPs that are signalled with RSVP-TE, and may change during the lifetime of the LSP.

The SYMBOLIC-PATH-NAME TLV MUST be included in the LSP object in the LSP State Report (PCRpt) message when during a given PCEP session an LSP is first reported to a PCE. A PCC sends to a PCE the first LSP State Report either during State Synchronization, or when a new LSP is configured at the PCC.

The initial PCRpt creates a binding between the symbolic path name and the PLSP-ID for the LSP which lasts for the duration of the PCEP session. The PCC MAY omit the symbolic path name from subsequent LSP

State Reports for that LSP on that PCEP session, and just use the PLSP-ID.

The format of the SYMBOLIC-PATH-NAME TLV is shown in the following figure:

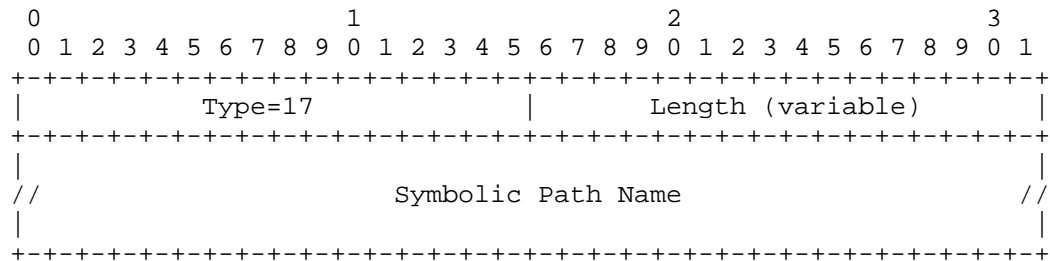


Figure 14: SYMBOLIC-PATH-NAME TLV format

```
Type (16 bits): The type is 17.
```

Length (16 bits): indicates the total length of the TLV in octets and MUST be greater than 0. The TLV MUST be zero-padded so that the TLV is 4-octet aligned.

Symbolic Path Name (variable): symbolic name for the LSP, unique in the PCC. It SHOULD be a string of printable ASCII characters, without a NULL terminator.

### 7.3.3. LSP Error Code TLV

The LSP Error code TLV is an optional TLV for use in the LSP object to convey error information. When an LSP Update Request fails, an LSP State Report **MUST** be sent to report the current state of the LSP, and **SHOULD** contain the LSP-ERROR-CODE TLV indicating the reason for the failure. Similarly, when a PCrpt is sent as a result of an LSP transitioning to non-operational state, the LSP-ERROR-CODE TLV **SHOULD** be included to indicate the reason for the transition.

The format of the LSP-ERROR-CODE TLV is shown in the following figure:

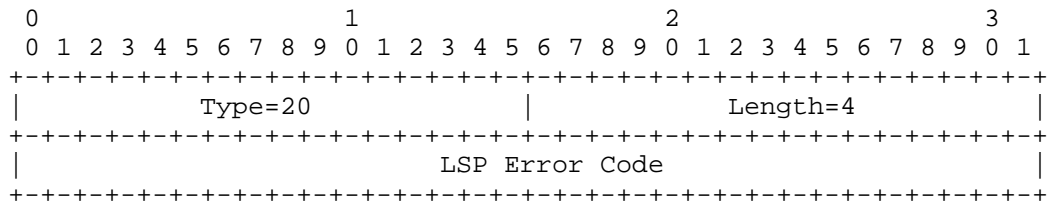


Figure 15: LSP-ERROR-CODE TLV format

The type (16 bits) of the TLV is 20. The length field is 16 bit-long and has a fixed value of 4. The value contains an error code that indicates the cause of the failure.

The following LSP Error Codes are currently defined:

Value	Meaning
1	Unknown reason
2	Limit reached for PCE-controlled LSPs
3	Too many pending LSP update requests
4	Unacceptable parameters
5	Internal error
6	LSP administratively brought down
7	LSP preempted
8	RSVP signaling error

#### 7.3.4. RSVP Error Spec TLV

The RSVP-ERROR-SPEC TLV is an optional TLV for use in the LSP object to carry RSVP error information. It includes the RSVP ERROR\_SPEC or USER\_ERROR\_SPEC Object ([RFC2205] and [RFC5284]) which were returned to the PCC from a downstream node. If the set up of an LSP fails at a downstream node which returned an ERROR\_SPEC to the PCC, the PCC SHOULD include in the PCRpt for this LSP the LSP-ERROR-CODE TLV with LSP Error Code = "RSVP signaling error" and the RSVP-ERROR-SPEC TLV with the relevant RSVP ERROR\_SPEC or USER\_ERROR\_SPEC Object.

The format of the RSVP-ERROR-SPEC TLV is shown in the following figure:

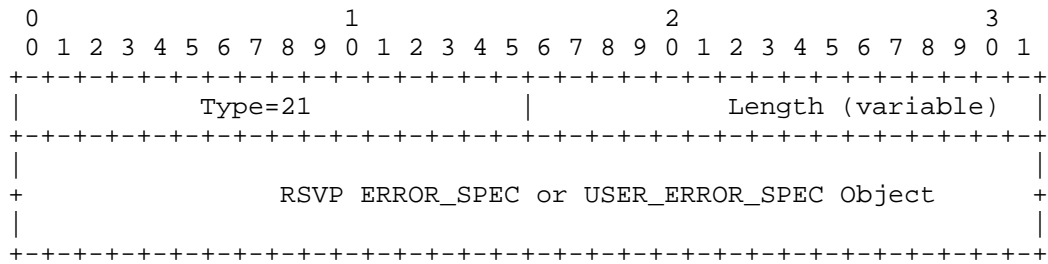


Figure 16: RSVP-ERROR-SPEC TLV format

Type (16 bits): The type is 21.

Length (16 bits): indicates the total length of the TLV in octets. The TLV MUST be zero-padded so that the TLV is 4-octet aligned.

Value (variable): contains the RSVP\_ERROR\_SPEC or USER\_ERROR\_SPEC Object: as specified in [RFC2205] and [RFC5284], including the object header.

## 8. IANA Considerations

This document requests IANA actions to allocate code points for the protocol elements defined in this document.

### 8.1. PCE Capabilities in IGP Advertisements

IANA is requested to confirm the early allocation of the following bits in the OSPF Parameters "PCE Capability Flags" registry, and to update the reference in the registry to point to this document, when it is an RFC:

Bit	Meaning	Reference
11	Active Stateful PCE capability	This document
12	Passive Stateful PCE capability	This document

### 8.2. PCEP Messages

IANA is requested to confirm the early allocation of the following message types within the "PCEP Messages" 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
10	Report	This document
11	Update	This document

### 8.3. PCEP Objects

IANA is requested to confirm the early allocation of the following object-class values and object types within the "PCEP Objects" 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:.

Object-Class Value	Name	Reference
32	LSP Object-Type 1	This document
33	SRP Object-Type 1	This document

### 8.4. LSP Object

This document requests that a new sub-registry, named "LSP Object Flag Field", is created within the "Path Computation Element Protocol (PCEP) Numbers" registry to manage the Flag field of 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
0-4	Reserved	This document
5-7	Operational (3 bits)	This document
8	Administrative	This document
9	Remove	This document
10	SYNC	This document
11	Delegate	This document



### 8.5. PCEP-Error Object

IANA is requested to confirm the early allocation of the following Error Types and Error Values within the "PCEP-ERROR Object Error Types and Values" 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:

Error-Type	Meaning
6	Mandatory Object missing
	Error-value=8: LSP Object missing
	Error-value=9: ERO Object missing
	Error-value=10: SRP Object missing
	Error-value=11: LSP-IDENTIFIERS TLV missing
19	Invalid Operation
	Error-value=1: Attempted LSP Update Request for a non-delegated LSP. The PCEP-ERROR Object is followed by the LSP Object that identifies the LSP.
	Error-value=2: Attempted LSP Update Request if the stateful PCE capability was not advertised.
	Error-value=3: Attempted LSP Update Request for an LSP identified by an unknown PLSP-ID.
	Error-value=5: Attempted LSP State Report if stateful PCE capability was not advertised.
20	LSP State synchronization error.
	Error-value=1: A PCE indicates to a PCC that it can not process (an otherwise valid) LSP State Report. The PCEP-ERROR Object is followed by the LSP Object that identifies the LSP.
	Error-value=5: A PCC indicates to a PCE that it can not complete the state synchronization,

### 8.6. Notification Object

IANA is requested to confirm the early allocation of the following Notification Types and Notification Values within the "Notification Object" 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:

Notification-Type	Meaning
4	Stateful PCE resource limit exceeded

Notification-value=1:	Entering resource limit exceeded state
-----------------------	--

Note to IANA: the early allocation included an additional Notification value 2 for "Exiting resource limit exceeded state". This Notification value is no longer required.

### 8.7. 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
16	STATEFUL-PCE-CAPABILITY	This document
17	SYMBOLIC-PATH-NAME	This document
18	IPV4-LSP-IDENTIFIERS	This document
19	IPV6-LSP-IDENTIFIERS	This document
20	LSP-ERROR-CODE	This document
21	RSVP-ERROR-SPEC	This document

### 8.8. STATEFUL-PCE-CAPABILITY TLV

This document requests that a new sub-registry, named "STATEFUL-PCE-CAPABILITY TLV Flag Field", is created within the "Path Computation Element Protocol (PCEP) Numbers" registry to manage the Flag field in the STATEFUL-PCE-CAPABILITY TLV of the PCEP OPEN object (class = 1). 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	LSP-UPDATE-CAPABILITY	This document

### 8.9. LSP-ERROR-CODE TLV

This document requests that a new sub-registry, named "LSP-ERROR-CODE TLV Error Code Field", is created within the "Path Computation Element Protocol (PCEP) Numbers" registry to manage the LSP Error code field of the LSP-ERROR-CODE TLV. This field specifies the reason for failure to update the LSP.

New values are to be assigned by Standards Action [RFC5226]. Each value should be tracked with the following qualities: value, description and defining RFC. The following values are defined in this document:

Value	Meaning
1	Unknown reason
2	Limit reached for PCE-controlled LSPs
3	Too many pending LSP update requests
4	Unacceptable parameters
5	Internal error
6	LSP administratively brought down
7	LSP preempted
8	RSVP signaling error

## 9. Manageability Considerations

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

### 9.1. Control Function and Policy

In addition to configuring specific PCEP session parameters, as specified in [RFC5440], Section 8.1, a PCE or PCC implementation MUST allow configuring the stateful PCEP capability and the LSP Update capability. A PCC implementation SHOULD allow the operator to specify multiple candidate PCEs for and a delegation preference for each candidate PCE. A PCC SHOULD allow the operator to specify an LSP delegation policy where LSPs are delegated to the most-preferred online PCE. A PCC MAY allow the operator to specify different LSP delegation policies.

A PCC implementation which allows concurrent connections to multiple PCEs SHOULD allow the operator to group the PCEs by administrative domains and it MUST NOT advertise LSP existence and state to a PCE if the LSP is delegated to a PCE in a different group.

A PCC implementation SHOULD allow the operator to specify whether the PCC will advertise LSP existence and state for LSPs that are not

controlled by any PCE (for example, LSPs that are statically configured at the PCC).

A PCC implementation SHOULD allow the operator to specify both the Redelegating Timeout Interval and the State Timeout Interval. The default value of the Redelegating Timeout Interval SHOULD be set to 30 seconds. An operator MAY also configure a policy that will dynamically adjust the Redelegating Timeout Interval, for example setting it to zero when the PCC has an established session to a backup PCE. The default value for the State Timeout Interval SHOULD be set to 60 seconds.

After the expiration of the State Timeout Interval, the LSP reverts to operator-defined default parameters. A PCC implementation MUST allow the operator to specify the default LSP parameters. To achieve a behavior where the LSP retains the parameters set by the PCE until such time that the PCC makes a change to them, a State Timeout Interval of infinity SHOULD be used. Any changes to LSP parameters SHOULD be done in make-before-break fashion.

LSP Delegation is controlled by operator-defined policies on a PCC. LSPs are delegated individually - different LSPs may be delegated to different PCEs. An LSP is delegated to at most one PCE at any given point in time. A PCC implementation SHOULD support the delegation policy, when all PCC's LSPs are delegated to a single PCE at any given time. Conversely, the policy revoking the delegation for all PCC's LSPs SHOULD also be supported.

A PCC implementation SHOULD allow the operator to specify delegation priority for PCEs. This effectively defines the primary PCE and one or more backup PCEs to which primary PCE's LSPs can be delegated when the primary PCE fails.

Policies defined for stateful PCEs and PCCs should eventually fit in the Policy-Enabled Path Computation Framework defined in [RFC5394], and the framework should be extended to support Stateful PCEs.

## 9.2. Information and Data Models

The PCEP YANG module [I-D.ietf-pcep-pcep-yang] should include

- o advertised stateful capabilities and synchronization status per PCEP session
- o the delegation status of each configured LSP.

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

### 9.3. Liveness Detection and Monitoring

PCEP extensions defined in this document do not require any new mechanisms beyond those already defined in [RFC5440], Section 8.3.

### 9.4. Verifying Correct Operation

Mechanisms defined in [RFC5440], Section 8.4 also apply to PCEP extensions defined in this document. In addition to monitoring parameters defined in [RFC5440], a stateful PCC-side PCEP implementation SHOULD provide the following parameters:

- o Total number of LSP updates
- o Number of successful LSP updates
- o Number of dropped LSP updates
- o Number of LSP updates where LSP setup failed

A PCC implementation SHOULD provide a command to show for each LSP whether it is delegated, and if so, to which PCE.

A PCC implementation SHOULD allow the operator to manually revoke LSP delegation.

### 9.5. Requirements on Other Protocols and Functional Components

PCEP extensions defined in this document do not put new requirements on other protocols.

### 9.6. Impact on Network Operation

Mechanisms defined in [RFC5440], Section 8.6 also apply to PCEP extensions defined in this document.

Additionally, a PCEP implementation SHOULD allow a limit to be placed on the number of LSPs delegated to the PCE and on the rate of PCUpd and PCRpt messages sent by a PCEP speaker and processed from a peer. It SHOULD also allow sending a notification when a rate threshold is reached.

A PCC implementation SHOULD allow a limit to be placed on the rate of LSP Updates to the same LSP to avoid signaling overload discussed in Section 10.3.

## 10. Security Considerations

### 10.1. Vulnerability

This document defines extensions to PCEP to enable stateful PCEs. The nature of these extensions and the delegation of path control to PCEs results in more information being available for a hypothetical adversary and a number of additional attack surfaces which must be protected.

The security provisions described in [RFC5440] remain applicable to these extensions. However, because the protocol modifications outlined in this document allow the PCE to control path computation timing and sequence, the PCE defense mechanisms described in [RFC5440] section 7.2 are also now applicable to PCC security.

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) [I-D.ietf-pce-pceps], as per the recommendations and best current practices in [RFC7525].

The following sections identify specific security concerns that may result from the PCEP extensions outlined in this document along with recommended mechanisms to protect PCEP infrastructure against related attacks.

### 10.2. LSP State Snooping

The stateful nature of this extension explicitly requires LSP status updates to be sent from PCC to PCE. While this gives the PCE the ability to provide more optimal computations to the PCC, it also provides an adversary with the opportunity to eavesdrop on decisions made by network systems external to PCE. This is especially true if the PCC delegates LSPs to multiple PCEs simultaneously.

Adversaries may gain access to this information by eavesdropping on unsecured PCEP sessions, and might then use this information in various ways to target or optimize attacks on network infrastructure. For example by flexibly countering anti-DDoS measures being taken to protect the network, or by determining choke points in the network where the greatest harm might be caused.

PCC implementations which allow concurrent connections to multiple PCEs SHOULD allow the operator to group the PCEs by administrative domains and they MUST NOT advertise LSP existence and state to a PCE if the LSP is delegated to a PCE in a different group.

### 10.3. Malicious PCE

The LSP delegation mechanism described in this document allows a PCC to grant effective control of an LSP to the PCE for the duration of a PCEP session. While this enables PCE control of the timing and sequence of path computations within and across PCEP sessions, it also introduces a new attack vector: an attacker may flood the PCC with PCUpd messages at a rate which exceeds either the PCC's ability to process them or the network's ability to signal the changes, either by spoofing messages or by compromising the PCE itself.

A PCC is free to revoke an LSP delegation at any time without needing any justification. A defending PCC can do this by enqueueing the appropriate PCRpt message. As soon as that message is enqueued in the session, the PCC is free to drop any incoming PCUpd messages without additional processing.

### 10.4. Malicious PCC

A stateful session also results in an increased attack surface by placing a requirement for the PCE to keep an LSP state replica for each PCC. It is RECOMMENDED that PCE implementations provide a limit on resources a single PCC can occupy. A PCE implementing such a limit MUST send a PCNtf message with notification-type 4 (Stateful PCE resource limit exceeded) and notification-value 1 (Entering resource limit exceeded state) upon receiving an LSP state report causing it to exceed this threshold.

Delegation of LSPs can create further strain on PCE resources and a PCE implementation MAY preemptively give back delegations if it finds itself lacking the resources needed to effectively manage the delegation. Since the delegation state is ultimately controlled by the PCC, PCE implementations SHOULD provide throttling mechanisms to prevent strain created by flaps of either a PCEP session or an LSP delegation.

## 11. Contributing Authors

Xian Zhang  
Huawei Technology  
F3-5-B R&D Center  
Huawei Industrial Base, Bantian, Longgang District  
Shenzhen, Guangdong 518129  
P.R.China  
EMail: zhang.xian@huawei.com

Dhruv Dhody  
Huawei Technology

Leela Palace  
Bangalore, Karnataka 560008  
INDIA  
EMail: dhruv.dhody@huawei.com

Siva Sivabalan  
Cisco Systems, Inc.  
2000 Innovation Drive  
Kanata, Ontario K2K 3E8  
Canada  
EMail: msiva@cisco.com

## 12. Acknowledgements

We would like to thank Adrian Farrel, Cyril Margaria and Ramon Casellas for their contributions to this document.

We would like to thank Shane Amante, Julien Meuric, Kohei Shiimoto, Paul Schultz and Raveendra Torvi for their comments and suggestions. Thanks also to Jon Hardwick, Oscar Gonzales de Dios, Tomas Janciga, Stefan Kobza, Kexin Tang, Matej Spanik, Jon Parker, Marek Zavodsky, Ambrose Kwong, Ashwin Sampath, Calvin Ying, Mustapha Aissaoui, Stephane Litkowski and Olivier Dugeon for helpful comments and discussions.

## 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>>.
- [RFC2205] Braden, R., Ed., Zhang, L., Berson, S., Herzog, S., and S. Jamin, "Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification", RFC 2205, DOI 10.17487/RFC2205, September 1997, <<http://www.rfc-editor.org/info/rfc2205>>.
- [RFC3209] Awduche, D., Berger, L., Gan, D., Li, T., Srinivasan, V., and G. Swallow, "RSVP-TE: Extensions to RSVP for LSP Tunnels", RFC 3209, DOI 10.17487/RFC3209, December 2001, <<http://www.rfc-editor.org/info/rfc3209>>.
- [RFC5088] Le Roux, JL., Ed., Vasseur, JP., Ed., Ikejiri, Y., and R. Zhang, "OSPF Protocol Extensions for Path Computation Element (PCE) Discovery", RFC 5088, DOI 10.17487/RFC5088, January 2008, <<http://www.rfc-editor.org/info/rfc5088>>.



- [RFC5089] Le Roux, JL., Ed., Vasseur, JP., Ed., Ikejiri, Y., and R. Zhang, "IS-IS Protocol Extensions for Path Computation Element (PCE) Discovery", RFC 5089, DOI 10.17487/RFC5089, January 2008, <<http://www.rfc-editor.org/info/rfc5089>>.
- [RFC5284] Swallow, G. and A. Farrel, "User-Defined Errors for RSVP", RFC 5284, DOI 10.17487/RFC5284, August 2008, <<http://www.rfc-editor.org/info/rfc5284>>.
- [RFC5440] Vasseur, JP., Ed. and JL. Le Roux, Ed., "Path Computation Element (PCE) Communication Protocol (PCEP)", RFC 5440, DOI 10.17487/RFC5440, March 2009, <<http://www.rfc-editor.org/info/rfc5440>>.
- [RFC5511] Farrel, A., "Routing Backus-Naur Form (RBNF): A Syntax Used to Form Encoding Rules in Various Routing Protocol Specifications", RFC 5511, DOI 10.17487/RFC5511, April 2009, <<http://www.rfc-editor.org/info/rfc5511>>.
- [RFC8051] Zhang, X., Ed. and I. Minei, Ed., "Applicability of a Stateful Path Computation Element (PCE)", RFC 8051, DOI 10.17487/RFC8051, January 2017, <<http://www.rfc-editor.org/info/rfc8051>>.

### 13.2. Informative References

- [I-D.ietf-pce-gmpls-pcep-extensions]  
Margarita, C., Dios, O., and F. Zhang, "PCEP extensions for GMPLS", draft-ietf-pce-gmpls-pcep-extensions-11 (work in progress), October 2015.
- [I-D.ietf-pce-pce-initiated-lsp]  
Crabbe, E., Minei, I., Sivabalan, S., and R. Varga, "PCEP Extensions for PCE-initiated LSP Setup in a Stateful PCE Model", draft-ietf-pce-pce-initiated-lsp-09 (work in progress), March 2017.
- [I-D.ietf-pce-pcep-yang]  
Dhody, D., Hardwick, J., Beeram, V., and j. jeffrant@gmail.com, "A YANG Data Model for Path Computation Element Communications Protocol (PCEP)", draft-ietf-pce-pcep-yang-02 (work in progress), March 2017.
- [I-D.ietf-pce-pceps]  
Lopez, D., Dios, O., Wu, Q., and D. Dhody, "Secure Transport for PCEP", draft-ietf-pce-pceps-14 (work in progress), May 2017.

- [I-D.ietf-pce-stateful-sync-optimizations]  
Crabbe, E., Minei, I., Medved, J., Varga, R., Zhang, X.,  
and D. Dhody, "Optimizations of Label Switched Path State  
Synchronization Procedures for a Stateful PCE", draft-  
ietf-pce-stateful-sync-optimizations-10 (work in  
progress), March 2017.
- [MPLS-PC] Chaieb, I., Le Roux, J.L., and B. Cousin, "Improved MPLS-TE  
LSP Path Computation using Preemption", Global  
Information Infrastructure Symposium, July 2007.
- [MXMN-TE] Danna, E., Mandal, S., and A. Singh, "Practical linear  
programming algorithm for balancing the max-min fairness  
and throughput objectives in traffic engineering",  
INFOCOM, 2012 Proceedings IEEE Page(s): 846-854, 2012.
- [RFC2702] Awduche, D., Malcolm, J., Agogbua, J., O'Dell, M., and J.  
McManus, "Requirements for Traffic Engineering Over MPLS",  
RFC 2702, DOI 10.17487/RFC2702, September 1999,  
<<http://www.rfc-editor.org/info/rfc2702>>.
- [RFC3031] Rosen, E., Viswanathan, A., and R. Callon, "Multiprotocol  
Label Switching Architecture", RFC 3031,  
DOI 10.17487/RFC3031, January 2001,  
<<http://www.rfc-editor.org/info/rfc3031>>.
- [RFC3346] Boyle, J., Gill, V., Hannan, A., Cooper, D., Awduche, D.,  
Christian, B., and W. Lai, "Applicability Statement for  
Traffic Engineering with MPLS", RFC 3346,  
DOI 10.17487/RFC3346, August 2002,  
<<http://www.rfc-editor.org/info/rfc3346>>.
- [RFC3630] Katz, D., Kompella, K., and D. Yeung, "Traffic Engineering  
(TE) Extensions to OSPF Version 2", RFC 3630,  
DOI 10.17487/RFC3630, September 2003,  
<<http://www.rfc-editor.org/info/rfc3630>>.
- [RFC4655] Farrel, A., Vasseur, J., and J. Ash, "A Path Computation  
Element (PCE)-Based Architecture", RFC 4655,  
DOI 10.17487/RFC4655, August 2006,  
<<http://www.rfc-editor.org/info/rfc4655>>.
- [RFC4657] Ash, J., Ed. and J. Le Roux, Ed., "Path Computation  
Element (PCE) Communication Protocol Generic  
Requirements", RFC 4657, DOI 10.17487/RFC4657, September  
2006, <<http://www.rfc-editor.org/info/rfc4657>>.

- [RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, DOI 10.17487/RFC5226, May 2008, <<http://www.rfc-editor.org/info/rfc5226>>.
- [RFC5305] Li, T. and H. Smit, "IS-IS Extensions for Traffic Engineering", RFC 5305, DOI 10.17487/RFC5305, October 2008, <<http://www.rfc-editor.org/info/rfc5305>>.
- [RFC5394] Bryskin, I., Papadimitriou, D., Berger, L., and J. Ash, "Policy-Enabled Path Computation Framework", RFC 5394, DOI 10.17487/RFC5394, December 2008, <<http://www.rfc-editor.org/info/rfc5394>>.
- [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>>.
- [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, <<http://www.rfc-editor.org/info/rfc7525>>.

#### Authors' Addresses

Edward Crabbe  
Oracle  
1501 4th Ave, suite 1800  
Seattle, WA 98101  
US  
  
Email: [edward.crabbe@oracle.com](mailto:edward.crabbe@oracle.com)

Ina Minei  
Google, Inc.  
1600 Amphitheatre Parkway  
Mountain View, CA 94043  
US  
  
Email: [inaminei@google.com](mailto:inaminei@google.com)

Jan Medved  
Cisco Systems, Inc.  
170 West Tasman Dr.  
San Jose, CA 95134  
US

Email: [jmedved@cisco.com](mailto:jmedved@cisco.com)

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

Email: [robert.varga@pantheon.tech](mailto:robert.varga@pantheon.tech)

Network Working Group  
Internet Draft

Y. Lee, Ed.  
Huawei Technologies

Intended status: Standard  
Expires: August 2013

R. Casellas, Ed.  
CTTC

February 6, 2013

## PCEP Extension for WSON Routing and Wavelength Assignment

draft-lee-pce-wson-rwa-ext-05.txt

### Abstract

This draft provides the Path Computation Element communication Protocol (PCEP) extensions for the support of Routing and Wavelength Assignment (RWA) in Wavelength Switched Optical Networks (WSON). Lightpath provisioning in WSONs requires a routing and wavelength assignment (RWA) process. From a path computation perspective, wavelength assignment is the process of determining which wavelength can be used on each hop of a path and forms an additional routing constraint to optical light path computation.

### Status of this Memo

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

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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

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

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

This Internet-Draft will expire on August 6, 2013.

#### Copyright Notice

Copyright (c) 2013 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. Terminology.....	3
2. Requirements Language.....	3
3. Introduction.....	3
4. Encoding of a RWA Path Request.....	6
4.1. Wavelength Assignment (WA) Object.....	6
4.2. Wavelength Restriction Constraint TLV.....	8
4.2.1. Link Identifier sub-TLV.....	11
4.2.2. Wavelength Restriction Field sub-TLV.....	12
4.3. Signal processing capability restrictions.....	12
4.3.1. Signal Processing Exclusion XRO Sub-Object.....	13
4.3.2. IRO sub-object: signal processing inclusion.....	14
5. Encoding of a RWA Path Reply.....	14
5.1. Error Indicator.....	15
5.2. NO-PATH Indicator.....	15
6. Manageability Considerations.....	16
6.1. Control of Function and Policy.....	16
6.2. Information and Data Models, e.g. MIB module.....	16
6.3. Liveness Detection and Monitoring.....	16
6.4. Verifying Correct Operation.....	17
6.5. Requirements on Other Protocols and Functional Components.....	17
6.6. Impact on Network Operation.....	17
7. Security Considerations.....	17

8. IANA Considerations.....	17
9. Acknowledgments.....	17
10. References.....	18
10.1. Informative References.....	18
11. Contributors.....	20
Authors' Addresses.....	21
Intellectual Property Statement.....	21
Disclaimer of Validity.....	22

## 1. Terminology

This document uses the terminology defined in [RFC4655], and [RFC5440].

## 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. Introduction

[RFC4655] defines the PCE based Architecture and explains how a Path Computation Element (PCE) may compute Label Switched Paths (LSP) in Multiprotocol Label Switching Traffic Engineering (MPLS-TE) and Generalized MPLS (GMPLS) networks at the request of Path Computation Clients (PCCs). A PCC is said to be any network component that makes such a request and may be, for instance, an Optical Switching Element within a Wavelength Division Multiplexing (WDM) network. The PCE, itself, can be located anywhere within the network, and may be within an optical switching element, a Network Management System (NMS) or Operational Support System (OSS), or may be an independent network server.

The PCE communications Protocol (PCEP) is the communication protocol used between PCC and PCE, and may also be used between cooperating PCEs. [RFC4657] sets out the common protocol requirements for PCEP. Additional application-specific requirements for PCEP are deferred to separate documents.

This document provides the PCEP extensions for the support of Routing and Wavelength Assignment (RWA) in Wavelength Switched

Optical Networks (WSON) based on the requirements specified in [PCE-RWA].

WSON refers to WDM based optical networks in which switching is performed selectively based on the wavelength of an optical signal. In this document, it is assumed that wavelength converters require electrical signal regeneration. Consequently, WSONs can be transparent (A transparent optical network is made up of optical devices that can switch but not convert from one wavelength to another, all within the optical domain) or translucent (3R regenerators are sparsely placed in the network).

A LSC Label Switched Path (LSP) may span one or several transparent segments, which are delimited by 3R regenerators (typically with electronic regenerator and optional wavelength conversion). Each transparent segment or path in WSON is referred to as an optical path. An optical path may span multiple fiber links and the path should be assigned the same wavelength for each link. In such case, the optical path is said to satisfy the wavelength-continuity constraint. Figure 1 illustrates the relationship between a LSC LSP and transparent segments (optical paths).

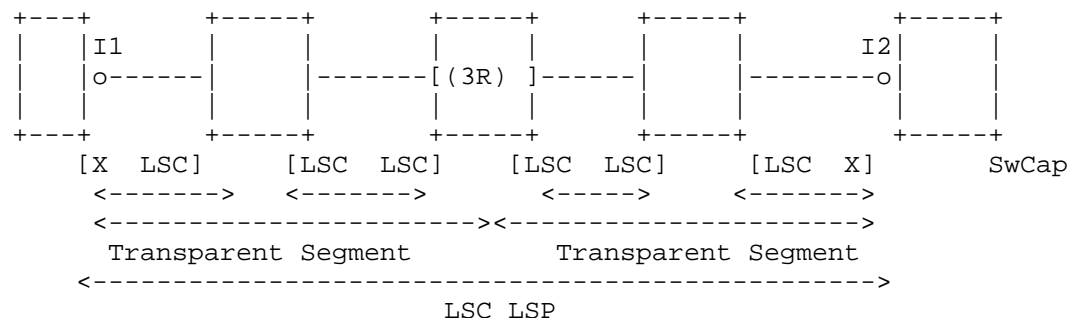


Figure 1 Illustration of a LSC LSP and transparent segments

Note that two optical paths within a WSON LSP need not operate on the same wavelength (due to the wavelength conversion capabilities). Two optical paths that share a common fiber link cannot be assigned the same wavelength. To do otherwise would result in both signals interfering with each other. Note that advanced additional multiplexing techniques such as polarization based multiplexing are not addressed in this document since the physical layer aspects are not currently standardized. Therefore, assigning the proper



wavelength on a lightpath is an essential requirement in the optical path computation process.

When a switching node has the ability to perform wavelength conversion, the wavelength-continuity constraint can be relaxed, and a LSC Label Switched Path (LSP) may use different wavelengths on different links along its route from origin to destination. It is, however, to be noted that wavelength converters may be limited due to their relatively high cost, while the number of WDM channels that can be supported in a fiber is also limited. As a WSON can be composed of network nodes that cannot perform wavelength conversion, nodes with limited wavelength conversion, and nodes with full wavelength conversion abilities, wavelength assignment is an additional routing constraint to be considered in all lightpath computation.

For example, within a translucent WSON, a LSC LSP may be established between interfaces I1 and I2, spanning 2 transparent segments (optical paths) where the wavelength continuity constraint applies (i.e. the same unique wavelength MUST be assigned to the LSP at each TE link of the segment). If the LSC LSP induced a Forwarding Adjacency / TE link, the switching capabilities of the TE link would be [X X] where  $X < \text{LSC (PSC, TDM, ...)}$ .

This document aligns with GMPLS extensions for PCEP [PCEP-GMPLS] for generic property such as label, label-set and label assignment noting that wavelength is a type of label. Wavelength restrictions and constraints are also formulated in terms of labels per [GEN-ENCODE].

The optical modulation properties, which are also referred to as signal compatibility, are already considered in signaling in [RWA-Encode] and [WSON-OSPF]. In order to improve the signal quality and limit some optical effects several advanced modulation processing are used. Those modulation properties contribute not only to optical signal quality checks but also constrain the selection of sender and receiver, as they should have matching signal processing capabilities. This document includes signal compatibility constraint as part of RWA path computation. That is, the signal processing capabilities (e.g., modulation and FEC) must be compatible between the sender and the receiver of the optical path across all optical elements.

This document, however, does not address optical impairments as part of RWA path computation. See [WSON-Imp] and [RSVP-Imp] for more information on optical impairments and GMPLS.

#### 4. Encoding of a RWA Path Request

Figure 2 shows one typical PCE based implementation, which is referred to as Combined Process (R&WA). With this architecture, the two processes of routing and wavelength assignment are accessed via a single PCE. This architecture is the base architecture from which the requirements have been specified in [PCE-RWA] and the PCEP extensions that are going to be specified in this document based on this architecture.

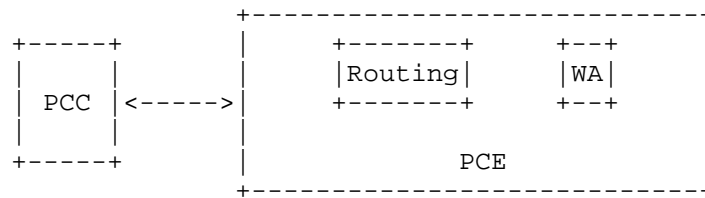


Figure 2 Combined Process (R&WA) architecture

##### 4.1. Wavelength Assignment (WA) Object

The current RP object is used to indicate routing related information in a new path request per [RFC5440]. Since a new RWA path request involves both routing and wavelength assignment, the wavelength assignment related information in the request SHOULD be coupled in the path request.

Wavelength allocation can be performed by the PCE by different means:

- (a) By means of Explicit Label Control, in the sense that one (or two) allocated labels MAY appear after an interface route subobject.
- (b) By means of a Label Set, containing one or more allocated Labels, provided by the PCE.

Option (b) allows distributed label allocation (performed during signaling) to complete wavelength assignment.

Additionally, given a range of potential labels to allocate, the request SHOULD convey the heuristic / mechanism to the allocation.

The format of a PCReq message after incorporating the WA object is as follows:

```
<PCReq Message> ::= <Common Header>
```

```

    [<svec-list>]
    <request-list>

```

Where:

```

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

    <request>::= <RP>

    <ENDPOINTS>

    <WA>

    [other optional objects...]

```

If WA object is present in the request, the WA object MUST be encoded after the ENDPOINTS object.

The format of the Wavelength Assignment (WA) object body 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
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                                     Flags                                     | O | M |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                                     Optional TLVs                                     |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

Figure 3 WA Object

- o Flags (32 bits)

The following new flags SHOULD be set

M (Mode - 1 bit): M bit is used to indicate the mode of wavelength assignment. When M bit is set to 1, this indicates that the label assigned by the PCE must be explicit. That is, the selected way to convey the allocated wavelength is by means of Explicit Label Control (ELC) [RFC4003] for each hop of a computed LSP. Otherwise, the label assigned by the PCE needs not be explicit (i.e., it can be suggested in the form of label set objects in the corresponding response, to allow distributed WA. In such case, the PCE MUST return a Label Set object as described in Section 2.2 of [Gen-Encode] in the response.

O (Order - 3 bits): O bit is used to indicate the wavelength assignment constraint in regard to the order of wavelength assignment to be returned by the PCE. This case is only applied when M bit is set to "explicit." The following indicators should be defined:

000 - Reserved

001 - Random Assignment

010 - First Fit (FF) in descending Order

011 - First Fit (FF) in ascending Order

100 - Last Fit (LF) in ascending Order

101 - Last Fit (LF) in descending Order

110 - Unspecified

111 - Reserved

#### 4.2. Wavelength Restriction Constraint TLV

For any request that contains a wavelength assignment, the requester (PCC) MUST be able to specify a restriction on the wavelengths to be used. This restriction is to be interpreted by the PCE as a constraint on the tuning ability of the origination laser transmitter or on any other maintenance related constraints. Note that if the LSP LSC spans different segments, the PCE MUST have mechanisms to know the tunability restrictions of the involved wavelength converters / regenerators, e.g. by means of the TED either via IGP or NMS. Even if the PCE knows the tunability of the transmitter, the PCC MUST be able to apply additional constraints to the request.

[Ed note: Which PCEP Object will home this TLV is yet to be determined. Since this involves the end-point, The END-POINTS Object might be a good candidate to encode this TLV, which will be provided in a later revision.]

[Ed note: The current encoding assumes that tunability restriction applied to link-level.]

The TLV type is TBD, recommended value is TBD. This TLV MAY appear more than once to be able to specify multiple restrictions.

The TLV data is defined as follows:

```
<Wavelength Restriction Constraint> ::=
```

```
    <Action> <Format> <Reserved>
```

```
    (<Link Identifiers> <Wavelength Restriction>)...
```

Where

```
<Link Identifiers> ::=
```

```
    <Unnumbered IF ID> | <IPv4 Address> | <IPv6 Address>
```

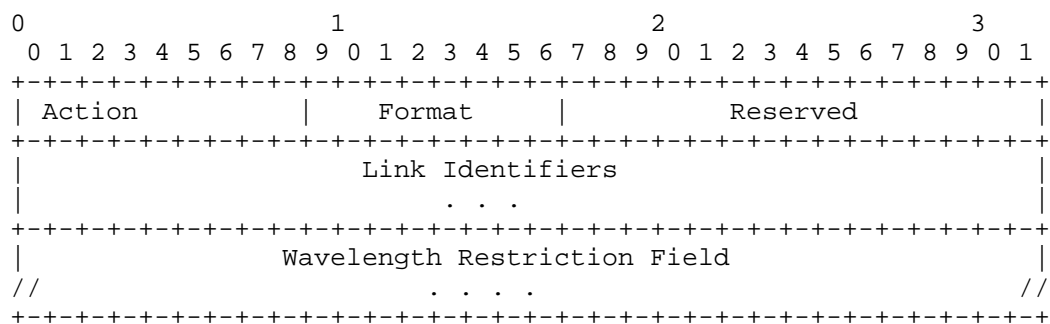


Figure 4 Wavelength Restriction

- o Action: 8 bits

- 0 - Inclusive List indicates that one or more link identifiers are included in the Link Set. Each identifies a separate link that is part of the set.

- 1 - Inclusive Range indicates that the Link Set defines a range of links. It contains two link identifiers. The first identifier indicates the start of the range (inclusive). The second identifier indicates the end of the range (inclusive). All links with numeric values between the bounds are considered to be part of the set. A value of zero in either position indicates that there is no bound on the corresponding portion of the range. Note that the Action field can be set to 0 when unnumbered link identifier is used.

Note that "interfaces" such as those discussed in the Interfaces MIB [RFC2863] are assumed to be bidirectional.

- o Format: The format of the link identifier (8 bits)

- 0 -- Unnumbered Link Identifier
  - 1 -- Local Interface IPv4 Address
  - 2 -- Local Interface IPv6 Address
  - Others TBD.

Note that all link identifiers in the same list must be of the same type.

- o Reserved: Reserved for future use (16 bits)

- o Link Identifiers: Identifies each link ID for which restriction is applied. The length is dependent on the link format. See the following section for Link Identifier encoding.

## 4.2.1. Link Identifier sub-TLV

The link identifier field can be an IPv4, IPv6 or unnumbered interface ID.

<Link Identifier> ::=

<IPv4 Address> | <IPv6 Address> | <Unnumbered IF ID>

The encoding of each case is as follows:

## IPv4 prefix Sub-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	2	3	4	5	6	7	8	9
Type = 1										IPv4 address (4 bytes)																													
IPv4 address (continued)										Prefix Length										Attribute																			

## IPv6 prefix Sub-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	2	3	4	5	6	7	8	9
Type = 2										IPv6 address (16 bytes)																													
IPv6 address (continued)																																							
IPv6 address (continued)																																							
IPv6 address (continued)																																							
IPv6 address (continued)																																							
IPv6 address (continued)										Prefix Length										Attribute																			

## Unnumbered Interface ID Sub-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	2	3	4	5	6	7	8	9

```

+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|  Type = 4      |      Reserved      |      Attribute      |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                                     TE Node ID                                     |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                                     Interface ID                                    |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

#### 4.2.2. Wavelength Restriction Field sub-TLV

The Wavelength Restriction Field of the wavelength restriction TLV is encoded as a Label Set field as specified in [GEN-Encode] section 2.2, as shown below, with base label encoded as a 32 bit LSC label, defined in [RFC6205]. See [RFC6205] for a description of Grid, C.S, Identifier and n, as well as [GEN-Encode] for the details of each action.

```

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

+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| Action|      Num Labels      |      Length      |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|Grid | C.S |      Identifier  |      n      |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|      Additional fields as necessary per action      |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

#### 4.3. Signal processing capability restrictions

Path computation for WSON include the check of signal processing capabilities, those capability MAY be provided by the IGP, however this is not a MUST. Moreover, a PCC should be able to indicate additional restrictions for those signal compatibility, either on the endpoint or any given link.

The supported signal processing capabilities are the one described in [RWA-Info]:

Optical Interface Class List



Bit rate

Client signal

The Bit-rate restriction is already expressed in [PCEP-GMPLS] in the GENERALIZED-BANDWIDTH object.

The client signal information can be expressed using the REQ-ADAP-CAP object from the [PCEP-Layer].

In order to support the Optical Interface Class information a new TLV are introduced as endpoint-restriction in the END-POINTS type Generalized endpoint:

Optical Interface Class List TLV

The END-POINTS type generalized endpoint is extended as follow:

```
<endpoint-restrictions> ::= <LABEL-REQUEST>
                               <label-restriction-list>
                               [<signal-compatibility-restriction>...]
```

Where

```
signal-compatibility-restriction ::=
    <Optical Interface Class List>
```

The encoding for Optical Interface Class List is described in Section 5.2 of [RWA-Encode].

#### 4.3.1. Signal Processing Exclusion XRO Sub-Object

The PCC/PCE should be able to exclude particular types of signal processing along the path in order to handle client restriction or multi-domain path computation.

In order to support the exclusion a new XRO sub-object is defined: the signal processing exclusion:

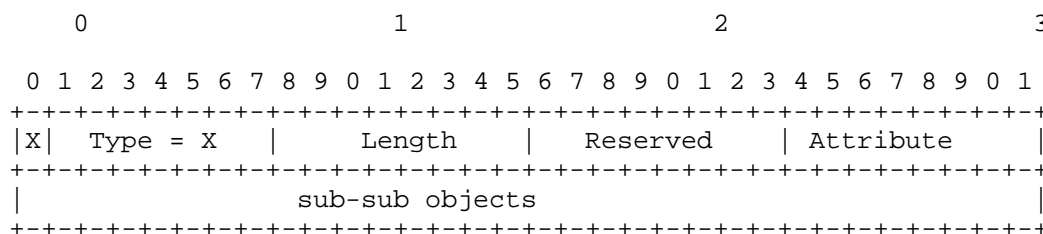


Figure 5 Signaling Processing XRO Sub-Object

The Attribute field indicates how the exclusion sub-object is to be interpreted. The Attribute can only be 0 (Interface) or 1 (Node).

The sub-sub objects are encoded as in RSVP signaling definition [WSON-Sign].

#### 4.3.2. IRO sub-object: signal processing inclusion

Similar to the XRO sub-object the PCC/PCE should be able to include particular types of signal processing along the path in order to handle client restriction or multi-domain path computation.

This is supported by adding the sub-object "processing" defined for ERO in [WSON-Sign] to the PCEP IRO object.

## 5. Encoding of a RWA Path Reply

The ERO is used to encode the path of a TE LSP through the network. The ERO is carried within a given path of a PCEP response, which is in turn carried in a PCRep message to provide the computed TE LSP if the path computation was successful. The preferred way to convey the allocated wavelength is by means of Explicit Label Control (ELC) [RFC4003].

In order to encode wavelength assignment, the Wavelength Assignment (WA) Object needs to be employed to be able to specify wavelength assignment. Since each segment of the computed optical path is associated with wavelength assignment, the WA Object should be aligned with the ERO object.

Encoding details will be provided further revisions and will be aligned as much as possible with [WSON-Sign] and [LSPA-ERO]

### 5.1. Error Indicator

To indicate errors associated with the RWA request, a new Error Type (TDB) and subsequent error-values are defined as follows for inclusion in the PCEP-ERROR Object:

A new Error-Type (TDB) and subsequent error-values are defined as follows:

Error-Type=TBD; Error-value=1: if a PCE receives a RWA request and the PCE is not capable of processing the request due to insufficient memory, the PCE MUST send a PCErr message with a PCEP-ERROR Object (Error-Type=TDB) and an Error-value(Error-value=1). The PCE stops processing the request. The corresponding RWA request MUST be cancelled at the PCC.

Error-Type=TBD; Error-value=2: if a PCE receives a RWA request and the PCE is not capable of RWA computation, the PCE MUST send a PCErr message with a PCEP-ERROR Object (Error-Type=15) and an Error-value (Error-value=2). The PCE stops processing the request. The corresponding RWA computation MUST be cancelled at the PCC.

### 5.2. NO-PATH Indicator

To communicate the reason(s) for not being able to find RWA for the path request, the NO-PATH object can be used in the PCRep message. The format of the NO-PATH object body is defined in [RFC5440]. The object may contain a NO-PATH-VECTOR TLV to provide additional information about why a path computation has failed.

Two new bit flags are defined to be carried in the Flags field in the NO-PATH-VECTOR TLV carried in the NO-PATH Object.

Bit TDB: When set, the PCE indicates no feasible route was found that meets all the constraints associated with RWA.

Bit TDB: When set, the PCE indicates that no wavelength was assigned to at least one hop of the route in the response.

Bit TDB: When set, the PCE indicate that no path was found satisfying the signal compatibility constraints.

## 6. Manageability Considerations

Manageability of WSON Routing and Wavelength Assignment (RWA) with PCE must address the following considerations:

### 6.1. Control of Function and Policy

In addition to the parameters already listed in Section 8.1 of [PCEP], a PCEP implementation SHOULD allow configuring the following PCEP session parameters on a PCC:

The ability to send a WSON RWA request.

In addition to the parameters already listed in Section 8.1 of [PCEP], a PCEP implementation SHOULD allow configuring the following PCEP session parameters on a PCE:

The support for WSON RWA.

A set of WSON RWA specific policies (authorized sender, request rate limiter, etc).

These parameters may be configured as default parameters for any PCEP session the PCEP speaker participates in, or may apply to a specific session with a given PCEP peer or a specific group of sessions with a specific group of PCEP peers.

### 6.2. Information and Data Models, e.g. MIB module

Extensions to the PCEP MIB module defined in [PCEP-MIB] should be defined, so as to cover the WSON RWA information introduced in this document. A future revision of this document will list the information that should be added to the MIB module.

### 6.3. Liveness Detection and Monitoring

Mechanisms defined in this document do not imply any new liveness detection and monitoring requirements in addition to those already listed in section 8.3 of [RFC5440].

#### 6.4. Verifying Correct Operation

Mechanisms defined in this document do not imply any new verification requirements in addition to those already listed in section 8.4 of [RFC5440]

#### 6.5. Requirements on Other Protocols and Functional Components

The PCE Discovery mechanisms ([RFC5089] and [RFC5088]) may be used to advertise WSON RWA path computation capabilities to PCCs.

#### 6.6. Impact on Network Operation

Mechanisms defined in this document do not imply any new network operation requirements in addition to those already listed in section 8.6 of [RFC5440].

### 7. Security Considerations

This document has no requirement for a change to the security models within PCEP [PCEP]. However the additional information distributed in order to address the RWA problem represents a disclosure of network capabilities that an operator may wish to keep private. Consideration should be given to securing this information.

### 8. IANA Considerations

A future revision of this document will present requests to IANA for codepoint allocation.

### 9. Acknowledgments

The authors would like to thank Adrian Farrel for many helpful comments that greatly improved the contents of this draft.

This document was prepared using 2-Word-v2.0.template.dot.

## 10. References

### 10.1. Informative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC3471] Berger, L., "Generalized Multi-Protocol Label Switching (GMPLS) Signaling Functional Description", RFC 3471, January 2003.
- [RFC3473] Berger, L., Ed., "Generalized Multi-Protocol Label Switching (GMPLS) Signaling Resource ReserVation Protocol-Traffic Engineering (RSVP-TE) Extensions", RFC 3473, January 2003.
- [RFC3477] Kompella, K. and Y. Rekhter, "Signalling Unnumbered Links in Resource ReSerVation Protocol - Traffic Engineering (RSVP-TE)", RFC 3477, January 2003.
- [RFC4003] Berger, L., "GMPLS Signaling Procedure for Egress Control", RFC 4003, February 2005.
- [RFC4655] Farrel, A., Vasseur, J., and J. Ash, "A Path Computation Element (PCE)-Based Architecture", RFC 4655, August 2006.
- [RFC4657] Ash, J. and J. Le Roux, "Path Computation Element (PCE) Communication Protocol Generic Requirements", RFC 4657, September 2006.
- [RFC5440] Vasseur, JP., Ed. and JL. Le Roux, Ed., "Path Computation Element (PCE) communication Protocol", RFC 5440, March 2009.
- [PCEP-GMPLS] Margaria, et al., "PCEP extensions for GMPLS", draft-ietf-pce-gmpls-pcep-extensions, work in progress.
- [LSPA-ERO] Margaria, et al., "LSP Attribute in ERO", draft-margaria-ccamp-lsp-attribute-ero, work in progress.
- [PCEP-Layer] Oki, Takeda, Le Roux, and Farrel, "Extensions to the Path Computation Element communication Protocol (PCEP) for Inter-Layer MPLS and GMPLS Traffic Engineering", draft-ietf-pce-inter-layer-ext, work in progress.

- [RFC6163] Lee, Y. and Bernstein, G. (Editors), and W. Imajuku, "Framework for GMPLS and PCE Control of Wavelength Switched Optical Networks", RFC 6163, March 2011.
- [PCE-RWA] Lee, Y., et. al., "PCEP Requirements for WSON Routing and Wavelength Assignment", draft-ietf-pce-wson-routing-wavelength, work in progress.
- [RFC6205] Tomohiro, O. and D. Li, "Generalized Labels for Lambda-Switching Capable Label Switching Routers", RFC 6205, January, 2011.
- [WSON-Sign] Bernstein et al, "Signaling Extensions for Wavelength Switched Optical Networks", draft-ietf-ccamp-wson-signaling, work in progress.
- [WSON-OSPF] Lee and Bernstein, "OSPF Enhancement for Signal and Network Element Compatibility for Wavelength Switched Optical Networks", draft-ietf-ccamp-wson-signal-compatibility-ospf, work in progress.
- [RWA-Info] Bernstein and Lee, "Routing and Wavelength Assignment Information Model for Wavelength Switched Optical Networks", draft-ietf-ccamp-rwa-info, work in progress.
- [RWA-Encode] Bernstein and Lee, "Routing and Wavelength Assignment Information Encoding for Wavelength Switched Optical Networks", draft-ietf-ccamp-rwa-wson-encode, work in progress.
- [GEN-Encode] Bernstein and Lee, "General Network Element Constraint Encoding for GMPLS Controlled Networks", draft-ietf-ccamp-general-constraint-encode, work in progress.
- [WSON-Imp] Y. Lee, G. Bernstein, D. Li, G. Martinelli, "A Framework for the Control of Wavelength Switched Optical Networks (WSON) with Impairments", draft-ietf-ccamp-wson-impairments, work in progress.
- [RSVP-Imp] agraz, "RSVP-TE Extensions in Support of Impairment Aware Routing and Wavelength Assignment in Wavelength Switched Optical Networks (WSONs)", draft-agraz-ccamp-wson-impairment-rsvp, work in progress.
- [OSPF-Imp] Bellagamba, et al., "OSPF Extensions for Wavelength Switched Optical Networks (WSON) with Impairments", draft-eb-ccamp-ospf-wson-impairments, work in progress.

## 11. Contributors



## Authors' Addresses

Young Lee, Editor  
Huawei Technologies  
1700 Alma Drive, Suite 100  
Plano, TX 75075, USA  
Phone: (972) 509-5599 (x2240)  
Email: leeyoung@huawei.com

Ramon Casellas, Editor  
CTTC PMT Ed B4 Av. Carl Friedrich Gauss 7  
08860 Castelldefels (Barcelona)  
Spain  
Phone: (34) 936452916  
Email: ramon.casellas@cttc.es

Fatai Zhang  
Huawei Technologies  
Email: zhangfatai@huawei.com

Cyril Margaria  
Nokia Siemens Networks  
St Martin Strasse 76  
Munich, 81541  
Germany  
Phone: +49 89 5159 16934  
Email: cyril.margaria@nsn.com

Oscar Gonzalez de Dios  
Telefonica Investigacion y Desarrollo  
C/ Emilio Vargas 6  
Madrid, 28043  
Spain  
Phone: +34 91 3374013  
Email: ogondio@tid.es

Greg Bernstein  
Grotto Networking  
Fremont, CA, USA  
Phone: (510) 573-2237  
Email: gregb@grotto-networking.com

## Intellectual Property Statement

The IETF Trust takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in any IETF Document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights.

Copies of Intellectual Property disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at <http://www.ietf.org/ipr>

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement any standard or specification contained in an IETF Document. Please address the information to the IETF at [ietf-ipr@ietf.org](mailto:ietf-ipr@ietf.org).

#### Disclaimer of Validity

All IETF Documents and the information contained therein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY, THE IETF TRUST AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION THEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

#### Acknowledgment

Funding for the RFC Editor function is currently provided by the Internet Society.



Network Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: August 29, 2013

F. Zhang, Ed.  
Q. Zhao  
Huawei  
O. Gonzalez de Dios, Ed.  
Telefonica I+D  
R. Casellas  
CTTC  
D. King  
Old Dog Consulting  
February 25, 2013

Extensions to Path Computation Element Communication Protocol (PCEP) for  
Hierarchical Path Computation Elements (PCE)  
draft-zhang-pce-hierarchy-extensions-03

## Abstract

The Hierarchical Path Computation Element (H-PCE) architecture, defined in the companion framework document [RFC6805], provides a mechanism to allow the optimum sequence of domains to be selected, and the optimum end-to-end path to be derived through the use of a hierarchical relationship between domains.

This document defines the Path Computation Element Protocol (PCEP) extensions for the purpose of implementing Hierarchical PCE procedures which are described in the aforementioned document.

## 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 29, 2013.

## Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the

document authors. All rights reserved.

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

## Table of Contents

1. Introduction . . . . .	4
1.1. Scope . . . . .	5
1.2. Terminology . . . . .	5
1.3. Requirements Language . . . . .	5
2. Requirements for H-PCE . . . . .	5
2.1. PCEP Requests . . . . .	5
2.1.1. Qualification of PCEP Requests . . . . .	6
2.1.2. Multi-domain Objective Functions . . . . .	6
2.1.3. Multi-domain Metrics . . . . .	7
2.2. Parent PCE Capability Discovery . . . . .	7
2.3. PCE Domain and PCE ID Discovery . . . . .	7
3. PCEP Extensions (Encoding) . . . . .	8
3.1. OPEN object . . . . .	8
3.1.1. OF Codes . . . . .	8
3.1.2. OPEN Object Flags . . . . .	8
3.1.3. Domain-ID TLV . . . . .	9
3.1.4. PCE-ID TLV . . . . .	10
3.2. RP object . . . . .	11
3.2.1. RP Object Flags . . . . .	11
3.2.2. Domain-ID TLV . . . . .	11
3.3. Metric Object . . . . .	11
3.4. PCEP-ERROR object . . . . .	11
3.4.1. Hierarchy PCE Error-Type . . . . .	11
3.5. NO-PATH Object . . . . .	12
4. H-PCE Procedures . . . . .	12
4.1. OPEN Procedure between Child PCE and Parent PCE . . . . .	12
4.2. Procedure to obtain Domain Sequence . . . . .	13
5. Error Handling . . . . .	13
6. Manageability Considerations . . . . .	13
7. IANA Considerations . . . . .	14
7.1. Objective Function (OF) codes . . . . .	14
7.2. OPEN Object Flags . . . . .	14
7.3. RP Object Flags . . . . .	14
7.4. PCEP TLVs . . . . .	14
7.5. PCEP PCEP-ERROR types . . . . .	15
7.6. New No-Path Reasons . . . . .	15
8. Security Considerations . . . . .	15
9. Contributing Authors . . . . .	15
10. Acknowledgments . . . . .	15
11. Normative References . . . . .	15
Authors' Addresses . . . . .	16

## 1. Introduction

[RFC6805] describes a Hierarchical PCE (H-PCE) architecture which can be used for computing end-to-end paths for inter-domain MPLS Traffic Engineering (TE) and GMPLS Label Switched Paths (LSPs).

Within the hierarchical PCE architecture, the parent PCE is used to compute a multi-domain path based on the domain connectivity information. A child PCE may be responsible for a single domain or multiple domains, it is used to compute the intra-domain path based on its domain topology information.

The H-PCE end-to-end domain path computation procedure is described below:

- o A path computation client (PCC) sends the inter-domain path computation requests to the child PCE responsible for its domain;
- o The child PCE forwards the request to the parent PCE;
- o The parent PCE computes the likely domain paths from the ingress domain to the egress domain;
- o The parent PCE sends the intra-domain path computation requests (between the domain border nodes) to the child PCEs which are responsible for the domains along the domain path;
- o The child PCEs return the intra-domain paths to the parent PCE;
- o The parent PCE constructs the end-to-end inter-domain path based on the intra-domain paths;
- o The parent PCE returns the inter-domain path to the child PCE;
- o The child PCE forwards the inter-domain path to the PCC;

In addition, the parent PCE may be requested to provide only the sequence of domains to a child PCE so that alternative inter-domain path computation procedures, including Per Domain (PD) [RFC5152] and Backwards Recursive Path Computation (BRPC) [RFC5441] may be used.

This document defines the PCEP extensions for the purpose of implementing Hierarchical PCE procedures, which are described in [RFC6805].

### 1.1. Scope

The following functions are out of scope of this document.

- o Finding end point addresses;
- o Parent Traffic Engineering Database (TED) methods;
- o Domain connectivity;

The document also uses a number of [editor notes] to describe options and alternative solutions. These options and notes will be removed before publication once agreement is reached.

### 1.2. Terminology

This document uses the terminology defined in [RFC4655], [RFC5440] and the additional terms defined in section 1.4 of [RFC6805].

### 1.3. 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. Requirements for H-PCE

This section compiles the set of requirements of the PCEP protocol to support the H-PCE architecture and procedures.

[RFC6805] identifies high-level requirements of PCEP extensions required to support the hierarchical PCE model.

### 2.1. PCEP Requests

The PCReq messages are used by a PCC or PCE to make a path computation request to a PCE. In order to achieve the full functionality of the H-PCE procedures, the PCReq message needs to include:

- o Qualification of PCE Requests.
- o Multi-domain Objective Functions (OF).
- o Multi-domain Metrics.



#### 2.1.1. Qualification of PCEP Requests

As described in section 4.8.1 of [RFC6805], the H-PCE architecture introduces new request qualifications, which are:

- o It MUST be possible for a child PCE to indicate that a request it sends to a parent PCE should be satisfied by a domain sequence only, that is, not by a full end-to-end path. This allows the child PCE to initiate a per-domain (PD) [RFC5152] or a backward recursive path computation (BRPC) [RFC5441].
- o As stated in [RFC6805], section 4.5, if a PCC knows the egress domain, it can supply this information as the path computation request. It SHOULD be possible to specify the destination domain information in a PCEP request, if it is known.

#### 2.1.2. Multi-domain Objective Functions

For inter-domain path computation, there are two new objective functions which are defined in section 1.3.1 and 4.1 of [RFC6805]:

- o Minimize the number of domains crossed. A domain can be either an Autonomous System (AS) or an Internal Gateway Protocol (IGP) area depending on the type of multi-domain network hierarchical PCE is applied to.
- o Disallow domain re-entry.[Editor's note: Disallow domain re-entry may not be an objective function, but an option in the request].

During the PCEP session establishment procedure, the parent PCE needs to be capable of indicating the Objective Functions (OF) capability in the Open message. This capability information may then be announced by child PCEs, and used for selecting the PCE when a PCC wants a path that satisfies one or multiple inter-domain objective functions.

When a PCC requests a PCE to compute an inter-domain path, the PCC needs also to be capable of indicating the new objective functions for inter-domain path. Note that a given child PCE may also act as a parent PCE.

For the reasons described previously, new OF codes need to be defined for the new inter-domain objective functions. Then the PCE can notify its new inter-domain objective functions to the PCC by carrying them in the OF-list TLV which is carried in the OPEN object. The PCC can specify which objective function code to use, which is carried in the OF object when requesting a PCE to compute an inter-domain path.

The proposed solution may need to differentiate between the OF code that is requested at the parent level, and the OF code that is requested at the intra-domain (child domain).

A parent PCE MUST be capable of ensuring homogeneity, across domains, when applying OF codes for strict OF intra-domain requests.

#### 2.1.3. Multi-domain Metrics

For inter-domain path computation, there are several path metrics of interest [Editor's note: Current framework only mentions metric objectives. The metric itself should be also defined]:

- o Domain count (number of domains crossed).
- o Border Node count.

A PCC may be able to limit the number of domains crossed by applying a limit on these metrics.

#### 2.2. Parent PCE Capability Discovery

Parent and child PCE relationships are likely to be configured. However, as mentioned in [RFC6805], it would assist network operators if the child and parent PCE could indicate their H-PCE capabilities.

During the PCEP session establishment procedure, the child PCE needs to be capable of indicating to the parent PCE whether it requests the parent PCE capability or not. Also, during the PCEP session establishment procedure, the parent PCE needs to be capable of indicating whether its parent capability can be provided or not.

#### 2.3. PCE Domain and PCE ID Discovery

A PCE domain is a single domain with an associated PCE. Although it is possible for a PCE to manage multiple domains. The PCE domain may be an IGP area or AS.

The PCE ID is an IPv4 and/or IPv6 address that is used to reach the parent/child PCE. It is RECOMMENDED to use an address that is always reachable if there is any connectivity to the PCE.

The PCE ID information and PCE domain identifiers may be provided during the PCEP session establishment procedure or the domain connectivity information collection procedure.

### 3. PCEP Extensions (Encoding)

#### 3.1. OPEN object

##### 3.1.1. OF Codes

There are two new OF codes defined here for H-PCE:

- o MTD

- \* Name: Minimize the number of Transit Domains.
- \* Objective Function Code: (to be assigned by IANA, recommended 12).
- \* Description: Find a path P such that it passes through the lnumber of transit domains.

- o MBN

- \* Name: Minimize the number of border nodes.
- \* Objective Function Code: (to be assigned by IANA, recommended 13).
- \* Description: Find a path P such that it passes through the least number of border nodes.

- o DDR

- \* Name: Disallow Domain Re-entry (DDR)
- \* Objective Function Code: (to be assigned by IANA, recommended 14)
- \* Description: Find a path P such that does not entry a domain more than once.

##### 3.1.2. OPEN Object Flags

There are two OPEN object flags defined here for H-PCE:

- o Parent PCE Request bit (to be assigned by IANA, recommended bit 0): if set, it would signal that the child PCE wishes to use the peer PCE as a parent PCE.
- o Parent PCE Indication bit (to be assigned by IANA, recommended bit 1): if set, it would signal that the PCE can be used as a parent

PCE by the peer PCE.

### 3.1.3. Domain-ID TLV

The type of Domain-ID TLV is to be assigned by IANA (recommended 7). The length is variable. The format of this TLV is defined below:

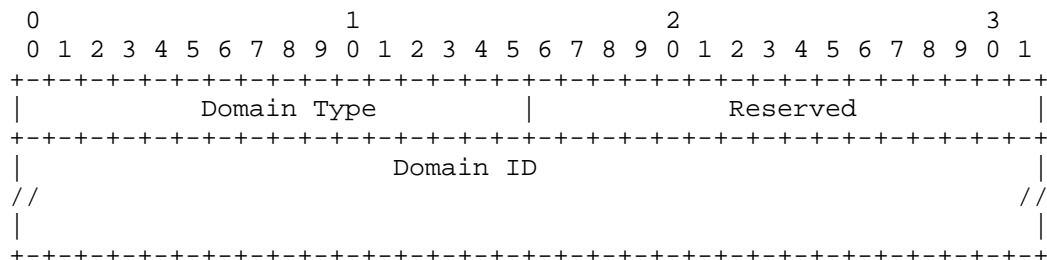


Figure 1: Domain-ID TLV

Domain Type (8 bits): Indicates the domain type. Two types of domain are currently defined:

- o Type=1: the Domain ID field carries an IGP Area ID.
- o Type=2: the Domain ID field carries an AS number.

Domain ID (variable): Indicates an IGP Area ID or AS number. It can be 2 bytes, 4 bytes or 8 bytes long depending on the domain identifier used.

[Editor's note: draft-dhody-pce-pcep-domain-sequence, section 3.2 deals with the encoding of domain sequences, using ERO-subobjects. Work is ongoing to define domain identifiers for OSPF-TE areas, IS-IS area (which are variable sized), 2-byte and 4-byte AS number, and any other domain that may be defined in the future. It uses RSVP-TE subobject discriminators, rather than new type 1/ type 2. A domain sequence may be encoded as a route object. The "VALUE" part of the TLV could follow common RSVP-TE subobject format:

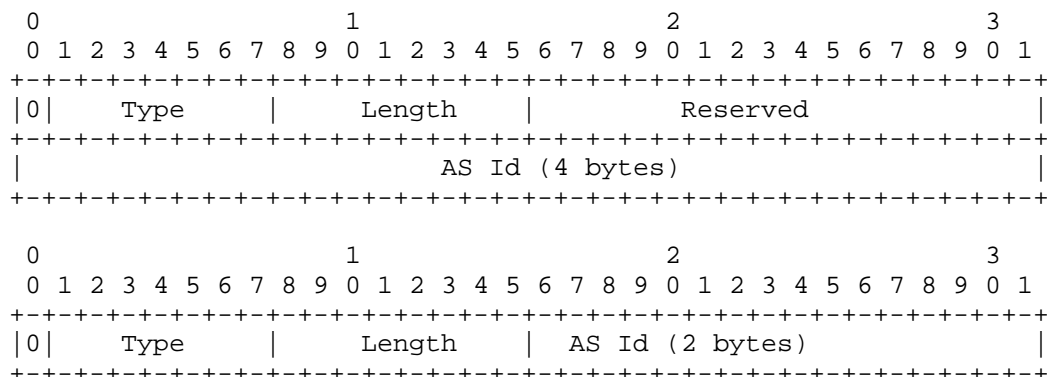


Figure 2: Alternative Domain-ID TLV

## 3.1.4. PCE-ID TLV

The type of PCE-ID TLV is to be assigned by IANA (recommended 8). The length is variable. The format of this TLV is defined below:

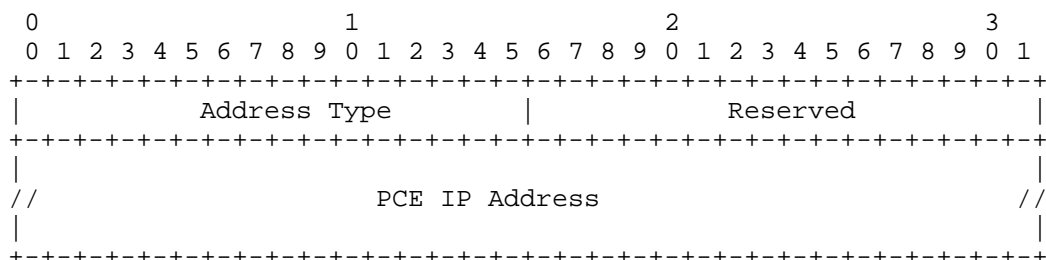


Figure 3: PCE-ID TLV

Address Type (16 bits): Indicates the address type of PCE IP Address. 1 means IPv4 address type, 2 means IPv6 address type.

PCE IP Address: Indicates the reachable address of a PCE.

[Editor's note: [RFC5886] already defines the PCE-ID object. If a semantically equivalent PCE-ID TLV is needed (to avoid modifying message grammars to include the object), it can align with the PCEP object: in any case, the length (4 / 16 bytes) can be used to know whether it is an IPv4 or an IPv6 PCE, the address type is not needed.]

### 3.2. RP object

#### 3.2.1. RP Object Flags

The following flags are defined:

- o Domain Path Request bit (to be assigned by IANA, recommended bit 17): if set, it means the child PCE wishes to get the domain sequence.
- o Destination Domain Query bit (to be assigned by IANA, recommended bit 16): if set, it means the parent PCE wishes to get the destination domain ID.

#### 3.2.2. Domain-ID TLV

The format of this TLV is defined in Section 3.1.3. This TLV can be carried in an OPEN object to indicate a (list of) managed domains, or carried in a RP object to indicate the destination domain ID when a child PCE responds to the parent PCE's destination domain query by a PCRep message.

[Editors note. In some cases, the Parent PCE may need to allocate a node which is not necessarily the destination node.]

### 3.3. Metric Object

There are two new metrics defined in this document for H-PCE:

- o Domain count (number of domains crossed).
- o Border Node Count (number of border nodes crossed).

### 3.4. PCEP-ERROR object

#### 3.4.1. Hierarchy PCE Error-Type

A new PCEP Error-Type is allocated for hierarchy PCE (to be assigned by IANA, recommended 19):

Error-Type	Meaning
19	H-PCE error Error-value=1: parent PCE capability cannot be provided

H-PCE error table

### 3.5. NO-PATH Object

To communicate the reason(s) for not being able to find a multi-domain path or domain sequence, the NO-PATH object can be used in the PCRep message. [RFC5440] defines the format of the NO-PATH object. The object may contain a NO-PATH-VECTOR TLV to provide additional information about why a (domain) path computation has failed.

Three new bit flags are defined to be carried in the Flags field in the NO-PATH-VECTOR TLV carried in the NO-PATH Object.

- o Bit 23(to be assigned by IANA): When set, the parent PCE indicates that destination domain unknown;
- o Bit 22(to be assigned by IANA): When set, the parent PCE indicates un-responsive child PCE(s);
- o Bit 21(to be assigned by IANA): When set, the parent PCE indicates no available resource available in one or more domain(s).

## 4. H-PCE Procedures

### 4.1. OPEN Procedure between Child PCE and Parent PCE

If a child PCE wants to use the peer PCE as a parent, it can set the parent PCE request bit in the OPEN object carried in the Open message during the PCEP session creation procedure. If the peer PCE does not want to provide the parent function to the child PCE, it must send a PCErr message to the child PCE and clear the parent PCE indication bit in the OPEN object.

If the parent PCE can provide the parent function to the peer PCE, it may set the parent PCE indication bit in the OPEN object carried in the Open message during the PCEP session creation procedure.

The PCE may also report its PCE ID and list of domain ID to the peer PCE by specifying them in the PCE-ID TLV and List of Domain-ID TLVs in the OPEN object carried in the Open message during the PCEP session creation procedure.

The OF codes defined in this document can be carried in the OF-list TLV of the OPEN object. If the OF-list TLV carries the OF codes, it means that the PCE is capable of implementing the corresponding objective functions. This information can be used for selecting a proper parent PCE when a child PCE wants to get a path that satisfies a certain objective function.

When a specific child PCE sends a PCReq to a peer PCE that requires parental activity and the peer PCE does not want to act as the parent for it, the peer PCE should send a PCErr message to the child PCE and specify the error-type (IANA) and error-value (1) in the PCEP-ERROR object.

#### 4.2. Procedure to obtain Domain Sequence

If a child PCE only wants to get the domain sequence for a multi-domain path computation from a parent PCE, it can set the Domain Path Request bit in the RP object carried in a PCReq message. The parent PCE which receives the PCReq message tries to compute a domain sequence for it. If the domain path computation succeeds the parent PCE sends a PCRep message which carries the domain sequence in the ERO to the child PCE. The domain sequence is specified as AS or AREA ERO sub-objects (type 32 for AS [RFC3209] or a to-be-defined IGP area type). Otherwise it sends a PCReq message which carries the NO-PATH object to the child PCE.

#### 5. Error Handling

A PCE that is capable of acting as a parent PCE might not be configured or willing to act as the parent for a specific child PCE. This fact could be determined when the child sends a PCReq that requires parental activity (such as querying other child PCEs), and could result in a negative response in a PCEP Error (PCErr) message and indicate the hierarchy PCE error types.

Additionally, the parent PCE may fail to find the multi-domain path or domain sequence due to one or more of the following reasons:

- o A child PCE cannot find a suitable path to the egress;
- o The parent PCE do not hear from a child PCE for a specified time;
- o The objective functions specified in the path request cannot be met.

In this case, the parent PCE MAY need to send a negative path computation reply specifying the reason. This can be achieved by including NO-PATH object in the PCRep message. Extension to NO-PATH object is needed to include the aforementioned reasons.

#### 6. Manageability Considerations

TBD.



## 7. IANA Considerations

As per [RFC5226], IANA is requested to create/update the following registries

### 7.1. Objective Function (OF) codes

Value	Meaning	Reference
12	MBN	This document
13	MTD	This document
14	DDR	This document

### 7.2. OPEN Object Flags

Bit Number	Meaning	Reference
0	Parent PCE Request	This document
1	Parent PCE Indication	This document

### 7.3. RP Object Flags

Bit Number	Meaning	Reference
17	Domain Path Request	This document

### 7.4. PCEP TLVs

Value	Meaning	Reference
x	Interdomain Link TLV	This document (section Section 3.3.2)
x	Interdomain Node TLV	This document (section Section 3.3.3)

## 7.5. PCEP PCEP-ERROR types

Type	Value	Meaning
H-PCE Error 19	1	parent PCE capability cannot be provided
	2	TBD
	3	TBD

## 7.6. New No-Path Reasons

Bit Number	Name	Reference
23	destination domain unknown	This document
22	un-responsive child PCE(s)	This document
21	no resource available in some domains	This document

## 8. Security Considerations

To be added.

## 9. Contributing Authors

Xian Zhang  
Huawei  
zhang.xian@huawei.com

## 10. Acknowledgments

To be added.

## 11. Normative References

- [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.
- [RFC4655] Farrel, A., Vasseur, J., and J. Ash, "A Path Computation Element (PCE)-Based Architecture", RFC 4655, August 2006.

- [RFC5152] Vasseur, JP., Ayyangar, A., and R. Zhang, "A Per-Domain Path Computation Method for Establishing Inter-Domain Traffic Engineering (TE) Label Switched Paths (LSPs)", RFC 5152, February 2008.
- [RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, May 2008.
- [RFC5316] Chen, M., Zhang, R., and X. Duan, "ISIS Extensions in Support of Inter-Autonomous System (AS) MPLS and GMPLS Traffic Engineering", RFC 5316, December 2008.
- [RFC5392] Chen, M., Zhang, R., and X. Duan, "OSPF Extensions in Support of Inter-Autonomous System (AS) MPLS and GMPLS Traffic Engineering", RFC 5392, January 2009.
- [RFC5440] Vasseur, JP. and JL. Le Roux, "Path Computation Element (PCE) Communication Protocol (PCEP)", RFC 5440, March 2009.
- [RFC5441] Vasseur, JP., Zhang, R., Bitar, N., and JL. Le Roux, "A Backward-Recursive PCE-Based Computation (BRPC) Procedure to Compute Shortest Constrained Inter-Domain Traffic Engineering Label Switched Paths", RFC 5441, April 2009.
- [RFC5886] Vasseur, JP., Le Roux, JL., and Y. Ikejiri, "A Set of Monitoring Tools for Path Computation Element (PCE)-Based Architecture", RFC 5886, June 2010.
- [RFC6805] King, D. and A. Farrel, "The Application of the Path Computation Element Architecture to the Determination of a Sequence of Domains in MPLS and GMPLS", RFC 6805, November 2012.

#### Authors' Addresses

Fatai Zhang (editor)  
Huawei  
Huawei Base, Bantian, Longgang District  
Shenzhen, 518129  
China

Phone: +86-755-28972912  
Email: zhangfatai@huawei.com

Quintin Zhao  
Huawei  
125 Nagog Technology Park  
Acton, MA 01719  
US

Phone:  
Email: qzhao@huawei.com

Oscar Gonzalez de Dios (editor)  
Telefonica I+D  
Don Ramon de la Cruz 82-84  
Madrid, 28045  
Spain

Phone: +34913128832  
Email: ogondio@tid.es

Ramon Casellas  
CTTC  
Av. Carl Friedrich Gauss n.7  
Castelldefels, Barcelona  
Spain

Phone: +34 93 645 29 00  
Email: ramon.casellas@cttc.es

Daniel King  
Old Dog Consulting  
UK

Phone:  
Email: daniel@olddog.co.uk



Network Working Group  
Internet-Draft  
Intended status: Experimental  
Expires: January 14, 2014

F. Zhang, Ed.  
Q. Zhao  
Huawei  
O. Gonzalez de Dios, Ed.  
Telefonica I+D  
R. Casellas  
CTTC  
D. King  
Old Dog Consulting  
July 14, 2013

Extensions to Path Computation Element Communication Protocol (PCEP) for  
Hierarchical Path Computation Elements (PCE)  
draft-zhang-pce-hierarchy-extensions-04

## Abstract

The Hierarchical Path Computation Element (H-PCE) architecture, defined in the companion framework document [RFC6805], provides a mechanism to allow the optimum sequence of domains to be selected, and the optimum end-to-end path to be derived through the use of a hierarchical relationship between domains.

This document defines the Path Computation Element Protocol (PCEP) extensions for the purpose of implementing Hierarchical PCE procedures which are described in the aforementioned document. These extensions are experimental and published for examination, discussion, implementation, and evaluation.

## 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 14, 2014.

## Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the

document authors. All rights reserved.

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

## Table of Contents

1. Introduction . . . . .	3
1.1. Scope . . . . .	3
1.2. Terminology . . . . .	4
1.3. Requirements Language . . . . .	4
2. Requirements for H-PCE . . . . .	4
2.1. PCEP Requests . . . . .	4
2.1.1. Qualification of PCEP Requests . . . . .	4
2.1.2. Multi-domain Objective Functions . . . . .	5
2.1.3. Multi-domain Metrics . . . . .	6
2.2. Parent PCE Capability Discovery . . . . .	6
2.3. PCE Domain and PCE ID Discovery . . . . .	6
3. PCEP Extensions (Encoding) . . . . .	6
3.1. OPEN Object . . . . .	6
3.1.1. OF Codes . . . . .	6
3.1.2. OPEN Object Flags . . . . .	7
3.1.3. Domain-ID TLV . . . . .	7
3.1.4. PCE-ID TLV . . . . .	9
3.2. RP object . . . . .	9
3.2.1. RP Object Flags . . . . .	9
3.2.2. Domain-ID TLV . . . . .	9
3.3. Metric Object . . . . .	10
3.4. PCEP-ERROR Object . . . . .	10
3.4.1. Hierarchy PCE Error-Type . . . . .	10
3.5. NO-PATH Object . . . . .	10
4. H-PCE Procedures . . . . .	10
4.1. OPEN Procedure between Child PCE and Parent PCE . . . . .	11
4.2. Procedure to Obtain Domain Sequence . . . . .	11
5. Error Handling . . . . .	11
6. Manageability Considerations . . . . .	12
7. IANA Considerations . . . . .	12
8. Security Considerations . . . . .	12
9. Contributing Authors . . . . .	12
10. Acknowledgments . . . . .	12
11. Normative References . . . . .	13
Authors' Addresses . . . . .	13

## 1. Introduction

[RFC6805] describes a Hierarchical PCE (H-PCE) architecture which can be used for computing end-to-end paths for inter-domain MPLS Traffic Engineering (TE) and GMPLS Label Switched Paths (LSPs).

Within the hierarchical PCE architecture, the parent PCE is used to compute a multi-domain path based on the domain connectivity information. A child PCE may be responsible for a single domain or multiple domains, it is used to compute the intra-domain path based on its domain topology information.

The H-PCE end-to-end domain path computation procedure is described below:

- o A path computation client (PCC) sends the inter-domain path computation requests to the child PCE responsible for its domain;
- o The child PCE forwards the request to the parent PCE;
- o The parent PCE computes the likely domain paths from the ingress domain to the egress domain;
- o The parent PCE sends the intra-domain path computation requests (between the domain border nodes) to the child PCEs which are responsible for the domains along the domain path;
- o The child PCEs return the intra-domain paths to the parent PCE;
- o The parent PCE constructs the end-to-end inter-domain path based on the intra-domain paths;
- o The parent PCE returns the inter-domain path to the child PCE;
- o The child PCE forwards the inter-domain path to the PCC.

In addition, the parent PCE may be requested to provide only the sequence of domains to a child PCE so that alternative inter-domain path computation procedures, including Per Domain (PD) [RFC5152] and Backwards Recursive Path Computation (BRPC) [RFC5441] may be used.

This document defines the PCEP extensions for the purpose of implementing Hierarchical PCE procedures, which are described in [RFC6805].

### 1.1. Scope



The following functions are out of scope of this document.

- o Finding end point addresses;
- o Parent Traffic Engineering Database (TED) methods;
- o Domain connectivity;

The document also uses a number of [editor notes] to describe options and alternative solutions. These options and notes will be removed before publication once agreement is reached.

## 1.2. Terminology

This document uses the terminology defined in [RFC4655], [RFC5440] and the additional terms defined in section 1.4 of [RFC6805].

## 1.3. 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. Requirements for H-PCE

This section compiles the set of requirements of the PCEP protocol to support the H-PCE architecture and procedures.

[RFC6805] identifies high-level requirements of PCEP extensions required to support the hierarchical PCE model.

### 2.1. PCEP Requests

The PCReq messages are used by a PCC or PCE to make a path computation request to a PCE. In order to achieve the full functionality of the H-PCE procedures, the PCReq message needs to include:

- o Qualification of PCE Requests.
- o Multi-domain Objective Functions (OF).
- o Multi-domain Metrics.

#### 2.1.1. Qualification of PCEP Requests

As described in section 4.8.1 of [RFC6805], the H-PCE architecture introduces new request qualifications, which are:

- o It MUST be possible for a child PCE to indicate that a request it sends to a parent PCE should be satisfied by a domain sequence only, that is, not by a full end-to-end path. This allows the child PCE to initiate a per-domain (PD) [RFC5152] or a backward recursive path computation (BRPC) [RFC5441].
- o As stated in [RFC6805], section 4.5, if a PCC knows the egress domain, it can supply this information as the path computation request. It SHOULD be possible to specify the destination domain information in a PCEP request, if it is known.

#### 2.1.2. Multi-domain Objective Functions

For inter-domain path computation, there are two new objective functions which are defined in section 1.3.1 and 4.1 of [RFC6805]:

- o Minimize the number of domains crossed. A domain can be either an Autonomous System (AS) or an Internal Gateway Protocol (IGP) area depending on the type of multi-domain network hierarchical PCE is applied to.
- o Disallow domain re-entry.[Editor's note: Disallow domain re-entry may not be an objective function, but an option in the request].

During the PCEP session establishment procedure, the parent PCE needs to be capable of indicating the Objective Functions (OF) capability in the Open message. This capability information may then be announced by child PCEs, and used for selecting the PCE when a PCC wants a path that satisfies one or multiple inter-domain objective functions.

When a PCC requests a PCE to compute an inter-domain path, the PCC needs also to be capable of indicating the new objective functions for inter-domain path. Note that a given child PCE may also act as a parent PCE.

For the reasons described previously, new OF codes need to be defined for the new inter-domain objective functions. Then the PCE can notify its new inter-domain objective functions to the PCC by carrying them in the OF-list TLV which is carried in the OPEN object. The PCC can specify which objective function code to use, which is carried in the OF object when requesting a PCE to compute an inter-domain path.

The proposed solution may need to differentiate between the OF code that is requested at the parent level, and the OF code that is requested at the intra-domain (child domain).

A parent PCE MUST be capable of ensuring homogeneity, across domains, when applying OF codes for strict OF intra-domain requests.

#### 2.1.3. Multi-domain Metrics

For inter-domain path computation, there are several path metrics of interest [Editor's note: Current framework only mentions metric objectives. The metric itself should be also defined]:

- o Domain count (number of domains crossed).
- o Border Node count.

A PCC may be able to limit the number of domains crossed by applying a limit on these metrics.

#### 2.2. Parent PCE Capability Discovery

Parent and child PCE relationships are likely to be configured. However, as mentioned in [RFC6805], it would assist network operators if the child and parent PCE could indicate their H-PCE capabilities.

During the PCEP session establishment procedure, the child PCE needs to be capable of indicating to the parent PCE whether it requests the parent PCE capability or not. Also, during the PCEP session establishment procedure, the parent PCE needs to be capable of indicating whether its parent capability can be provided or not.

#### 2.3. PCE Domain and PCE ID Discovery

A PCE domain is a single domain with an associated PCE. Although it is possible for a PCE to manage multiple domains. The PCE domain may be an IGP area or AS.

The PCE ID is an IPv4 and/or IPv6 address that is used to reach the parent/child PCE. It is RECOMMENDED to use an address that is always reachable if there is any connectivity to the PCE.

The PCE ID information and PCE domain identifiers may be provided during the PCEP session establishment procedure or the domain connectivity information collection procedure.

### 3. PCEP Extensions (Encoding)

#### 3.1. OPEN object

##### 3.1.1. OF Codes

This H-PCE experiment will be carried out using the following OF codes:

- o MTD
  - \* Name: Minimize the number of Transit Domains.
  - \* Objective Function Code.
  - \* Description: Find a path P such that it passes through the lnumber of transit domains.
- o MBN
  - \* Name: Minimize the number of border nodes.
  - \* Objective Function Code.
  - \* Description: Find a path P such that it passes through the least number of border nodes.
- o DDR
  - \* Name: Disallow Domain Re-entry (DDR)
  - \* Objective Function Code.
  - \* Description: Find a path P such that does not entry a domain more than once.

### 3.1.2. OPEN Object Flags

This H-PCE experiment will also require two OPEN object flags:

- o Parent PCE Request bit (to be assigned by IANA, recommended bit 0): if set, it would signal that the child PCE wishes to use the peer PCE as a parent PCE.
- o Parent PCE Indication bit (to be assigned by IANA, recommended bit 1): if set, it would signal that the PCE can be used as a parent PCE by the peer PCE.

### 3.1.3. Domain-ID TLV

The Domain-ID TLV for this H-PCE experiment is defined below:

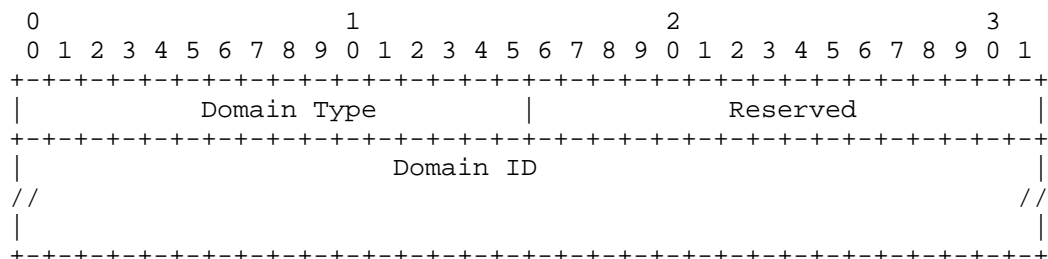


Figure 1: Domain-ID TLV

Domain Type (8 bits): Indicates the domain type. Two types of domain are currently defined:

- o Type=1: the Domain ID field carries an IGP Area ID.
- o Type=2: the Domain ID field carries an AS number.

Domain ID (variable): Indicates an IGP Area ID or AS number. It can be 2 bytes, 4 bytes or 8 bytes long depending on the domain identifier used.

[Editor's note: draft-dhody-pce-pcep-domain-sequence, section 3.2 deals with the encoding of domain sequences, using ERO-subobjects. Work is ongoing to define domain identifiers for OSPF-TE areas, IS-IS area (which are variable sized), 2-byte and 4-byte AS number, and any other domain that may be defined in the future. It uses RSVP-TE subobject discriminators, rather than new type 1/ type 2. A domain sequence may be encoded as a route object. The "VALUE" part of the TLV could follow common RSVP-TE subobject format:

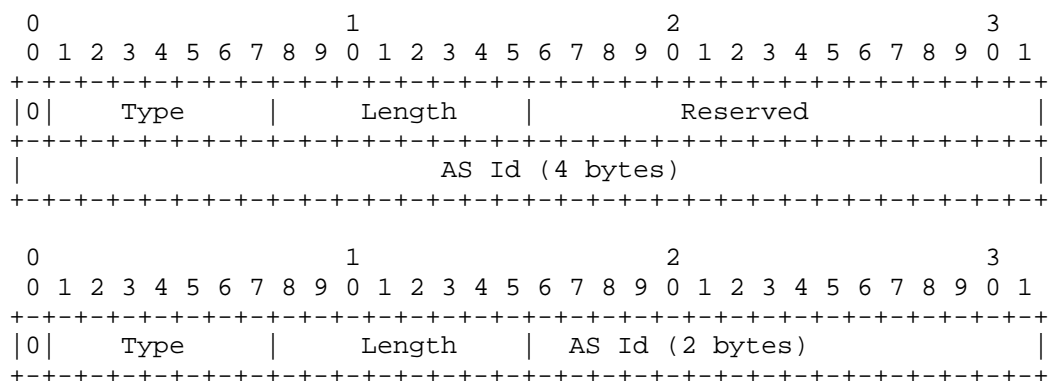


Figure 2: Alternative Domain-ID TLV

### 3.1.4. PCE-ID TLV

The type of PCE-ID TLV for this H-PCE experiment is defined below:

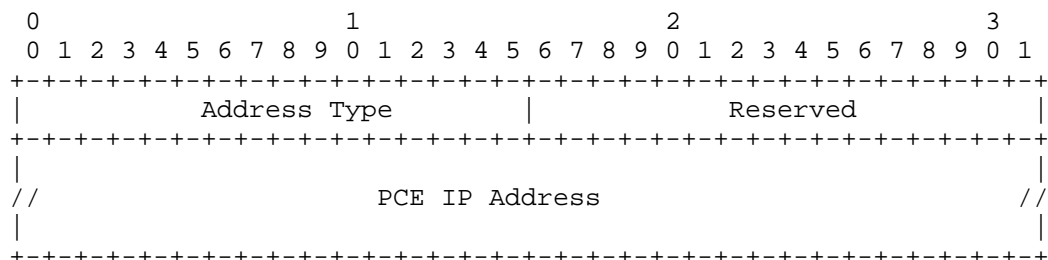


Figure 3: PCE-ID TLV

Address Type (16 bits): Indicates the address type of PCE IP Address. 1 means IPv4 address type, 2 means IPv6 address type.

PCE IP Address: Indicates the reachable address of a PCE.

[Editor's note: [RFC5886] already defines the PCE-ID object. If a semantically equivalent PCE-ID TLV is needed (to avoid modifying message grammars to include the object), it can align with the PCEP object: in any case, the length (4 / 16 bytes) can be used to know whether it is an IPv4 or an IPv6 PCE, the address type is not needed.]

## 3.2. RP object

### 3.2.1. RP Object Flags

The following RP object flags are defined for this H-PCE experiment:

- o Domain Path Request bit: if set, it means the child PCE wishes to get the domain sequence.
- o Destination Domain Query bit: if set, it means the parent PCE wishes to get the destination domain ID.

### 3.2.2. Domain-ID TLV

The format of this TLV is defined in Section 3.1.3. This TLV can be carried in an OPEN object to indicate a (list of) managed domains, or carried in a RP object to indicate the destination domain ID when a child PCE responds to the parent PCE's destination domain query by a PCRep message.

[Editors note. In some cases, the Parent PCE may need to allocate a node which is not necessarily the destination node.]

### 3.3. Metric Object

There are two new metrics defined in this document for H-PCE:

- o Domain count (number of domains crossed).
- o Border Node Count (number of border nodes crossed).

### 3.4. PCEP-ERROR object

#### 3.4.1. Hierarchy PCE Error-Type

A new PCEP Error-Type is used for this H-PCE experiment and is defined below:

Error-Type	Meaning
19	H-PCE error Error-value=1: parent PCE capability cannot be provided

H-PCE error table

### 3.5. NO-PATH Object

To communicate the reason(s) for not being able to find a multi-domain path or domain sequence, the NO-PATH object can be used in the PCRep message. [RFC5440] defines the format of the NO-PATH object. The object may contain a NO-PATH-VECTOR TLV to provide additional information about why a (domain) path computation has failed.

Three new bit flags are defined to be carried in the Flags field in the NO-PATH-VECTOR TLV carried in the NO-PATH Object.

- o Bit 23: When set, the parent PCE indicates that destination domain unknown;
- o Bit 22: When set, the parent PCE indicates unresponsive child PCE(s);
- o Bit 21: When set, the parent PCE indicates no available resource available in one or more domain(s).

## 4. H-PCE Procedures

#### 4.1. OPEN Procedure between Child PCE and Parent PCE

If a child PCE wants to use the peer PCE as a parent, it can set the parent PCE request bit in the OPEN object carried in the Open message during the PCEP session creation procedure. If the peer PCE does not want to provide the parent function to the child PCE, it must send a PCErr message to the child PCE and clear the parent PCE indication bit in the OPEN object.

If the parent PCE can provide the parent function to the peer PCE, it may set the parent PCE indication bit in the OPEN object carried in the Open message during the PCEP session creation procedure.

The PCE may also report its PCE ID and list of domain ID to the peer PCE by specifying them in the PCE-ID TLV and List of Domain-ID TLVs in the OPEN object carried in the Open message during the PCEP session creation procedure.

The OF codes defined in this document can be carried in the OF-list TLV of the OPEN object. If the OF-list TLV carries the OF codes, it means that the PCE is capable of implementing the corresponding objective functions. This information can be used for selecting a proper parent PCE when a child PCE wants to get a path that satisfies a certain objective function.

When a specific child PCE sends a PCReq to a peer PCE that requires parental activity and the peer PCE does not want to act as the parent for it, the peer PCE should send a PCErr message to the child PCE and specify the error-type (IANA) and error-value (1) in the PCEP-ERROR object.

#### 4.2. Procedure to obtain Domain Sequence

If a child PCE only wants to get the domain sequence for a multi-domain path computation from a parent PCE, it can set the Domain Path Request bit in the RP object carried in a PCReq message. The parent PCE which receives the PCReq message tries to compute a domain sequence for it. If the domain path computation succeeds the parent PCE sends a PCRep message which carries the domain sequence in the ERO to the child PCE. The domain sequence is specified as AS or AREA ERO sub-objects (type 32 for AS [RFC3209] or a to-be-defined IGP area type). Otherwise it sends a PCReq message which carries the NO-PATH object to the child PCE.

### 5. Error Handling

A PCE that is capable of acting as a parent PCE might not be configured or willing to act as the parent for a specific child PCE.



This fact could be determined when the child sends a PCReq that requires parental activity (such as querying other child PCEs), and could result in a negative response in a PCEP Error (PCErr) message and indicate the hierarchy PCE error types.

Additionally, the parent PCE may fail to find the multi-domain path or domain sequence due to one or more of the following reasons:

- o A child PCE cannot find a suitable path to the egress;
- o The parent PCE do not hear from a child PCE for a specified time;
- o The objective functions specified in the path request cannot be met.

In this case, the parent PCE MAY need to send a negative path computation reply specifying the reason. This can be achieved by including NO-PATH object in the PCRep message. Extension to NO-PATH object is needed to include the aforementioned reasons.

## 6. Manageability Considerations

TBD.

## 7. IANA Considerations

Due to the experimental nature of this draft no IANA requests are made.

## 8. Security Considerations

To be added.

## 9. Contributing Authors

Xian Zhang  
Huawei  
zhang.xian@huawei.com

## 10. Acknowledgments

To be added.

## 11. Normative References

- [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.
- [RFC4655] Farrel, A., Vasseur, J., and J. Ash, "A Path Computation Element (PCE)-Based Architecture", RFC 4655, August 2006.
- [RFC5152] Vasseur, JP., Ayyangar, A., and R. Zhang, "A Per-Domain Path Computation Method for Establishing Inter-Domain Traffic Engineering (TE) Label Switched Paths (LSPs)", RFC 5152, February 2008.
- [RFC5440] Vasseur, JP. and JL. Le Roux, "Path Computation Element (PCE) Communication Protocol (PCEP)", RFC 5440, March 2009.
- [RFC5441] Vasseur, JP., Zhang, R., Bitar, N., and JL. Le Roux, "A Backward-Recursive PCE-Based Computation (BRPC) Procedure to Compute Shortest Constrained Inter-Domain Traffic Engineering Label Switched Paths", RFC 5441, April 2009.
- [RFC5886] Vasseur, JP., Le Roux, JL., and Y. Ikejiri, "A Set of Monitoring Tools for Path Computation Element (PCE)-Based Architecture", RFC 5886, June 2010.
- [RFC6805] King, D. and A. Farrel, "The Application of the Path Computation Element Architecture to the Determination of a Sequence of Domains in MPLS and GMPLS", RFC 6805, November 2012.

## Authors' Addresses

Fatai Zhang (editor)  
Huawei  
Huawei Base, Bantian, Longgang District  
Shenzhen, 518129  
China

Phone: +86-755-28972912  
Email: zhangfatai@huawei.com

Quintin Zhao  
Huawei  
125 Nagog Technology Park  
Acton, MA 01719  
US

Phone:  
Email: qzhao@huawei.com

Oscar Gonzalez de Dios (editor)  
Telefonica I+D  
Don Ramon de la Cruz 82-84  
Madrid, 28045  
Spain

Phone: +34913128832  
Email: ogondio@tid.es

Ramon Casellas  
CTTC  
Av. Carl Friedrich Gauss n.7  
Castelldefels, Barcelona  
Spain

Phone: +34 93 645 29 00  
Email: ramon.casellas@cttc.es

Daniel King  
Old Dog Consulting  
UK

Phone:  
Email: daniel@olddog.co.uk



Network Working Group  
Internet-Draft  
Intended status: Standards Track

Xian Zhang  
Young Lee  
Fatai Zhang  
Huawei  
Ramon Casellas  
CTTC  
Oscar Gonzalez de Dios  
Telefonica I+D

Expires: August 21, 2013

February 22, 2013

Path Computation Element (PCE) Protocol Extension for Stateful PCE  
Usage in GMPLS Networks

draft-zhang-pce-pcep-stateful-pce-gmpls-02.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/lid-abstracts.txt>.

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

This Internet-Draft will expire on August 21, 2013.

## Abstract

The Path Computation Element (PCE) facilitates Traffic Engineering (TE) based path calculation in large, multi-domain, multi-region, or multi-layer networks. [Stateful-PCE] provides the fundamental PCEP extensions needed to support stateful PCE functions, without specifying the technology-specific extensions. This memo provides extensions required for PCE communication protocol (PCEP) so as to enable the usage of a stateful PCE capability in GMPLS networks.

## Conventions used in this document

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

## Table of Contents

Table of Contents .....	2
1. Introduction .....	3
2. PCEP Extensions .....	3
2.1. Overview of Requirements.....	3
2.2. Stateful PCE Capability Advertisement and Negotiation....	4
2.2.1. PCE Capability Negotiation/Advertisement in Multi-layer Networks .....	4
2.3. LSP Delegation in GMPLS Networks .....	5
2.4. LSP Synchronization in GMPLS networks .....	6
2.5. Modification of Existing PCEP Messages and Procedures....	8
2.5.1. Use cases .....	8
2.5.2. Modification for LSP Re-optimization .....	9
2.5.3. Modification for Route Exclusion .....	9
2.6. Additional Error Type and Error Values Defined.....	10
3. IANA Considerations .....	10
4. Manageability Considerations.....	10
4.1. Requirements on Other Protocols and Functional Components	11
5. Security Considerations.....	11
6. Acknowledgement .....	11
7. References .....	11
7.1. Normative References.....	11
7.2. Informative References.....	12
8. Contributors' Address.....	12
Authors' Addresses .....	13

## 1. Introduction

[RFC 4655] presents the architecture of a Path Computation Element (PCE)-based model for computing Multiprotocol Label Switching (MPLS) and Generalized MPLS (GMPLS) Traffic Engineering Label Switched Paths (TE LSPs). To perform such a constrained computation, a PCE stores the network topology (i.e., TE links and nodes) and resource information (i.e., TE attributes) in its TE Database (TED). To request path computation services to a PCE, [RFC 5440] defines the PCE Communication Protocol (PCEP) for communications between a Path Computation Client (PCC) and a PCE, or between two PCEs. PCEP protocol specified in [RFC 5440] mainly focuses on MPLS networks and the PCEP extensions needed for GMPLS-controlled networks are provided in [PCEP-GMPLS].

Stateful PCEs are shown to be helpful in many application scenarios, in both MPLS and GMPLS networks, as illustrated in [Stateful-APP]. In order for these applications to be able to exploit the capability of stateful PCEs, extensions to the PCE communication protocol (i.e., PCEP) are required.

[Stateful-PCE] provides the fundamental extensions needed for stateful PCE to support general functionality, but leaves out the specification for technology-specific objects/TLVs. Complementarily, this document focuses on the extensions that are necessary in order for its deployment in GMPLS-controlled networks.

## 2. PCEP Extensions

### 2.1. Overview of Requirements

This section notes the main functional requirements for PCEP extensions to support stateful PCE for use in GMPLS networks, based on the description in [Stateful-APP]. Many requirements are common across a variety of network types (e.g., MPLS-TE networks and GMPLS networks) and the protocol extensions to meet the requirements are already described in [Stateful-PCE]. This document does not repeat the description of those protocol extensions. Other requirements that are also common across a variety of network types do not currently have protocol extensions defined in [Stateful-PCE]. In these cases, this document presents protocol extensions for discussion by the PCE working group and potential inclusion in [Stateful-PCE]. In addition, this document presents protocol extensions for a set of requirements which are specific to the use of a stateful PCE in a GMPLS-controlled network.

The basic requirements are as follows:

- o Advertisement and negotiation of the stateful PCE capability. This generic requirement is covered in Section 7.1.1 of [Stateful-PCE]. Section 2.2 of this document discusses other potential extensions for this functionality.
- o LSP delegation is already covered in Section 5.5 of [Stateful-PCE]. Section 2.3 of this document provides extension for its application in GMPLS-controlled networks. Moreover, further discussion of some generic details that may need additional consideration is provided.
- o LSP synchronization (see [Stateful-APP] Section 2.2). This is a generic requirement already covered in Section 5.4 of [Stateful-PCE]. However, there are further extensions required specifically for GMPLS networks and discussed in Section 2.4.o Reference to LSPs by identifiers is discussed in Section 7.2 of [Stateful-PCE]. This feature can be applied to reduce the data carried in PCEP messages. Use cases and additional Error Codes are necessary, as described in Section 2.5 and 2.6.

## 2.2. Stateful PCE Capability Advertisement and Negotiation

Whether a PCE has stateful capability or not can be negotiated during the PCEP session establishment process. It can also be advertised through routing protocols as described in [RFC5088]. In either case, the following additional aspects should also be considered.

### 2.2.1. PCE Capability Negotiation/Advertisement in Multi-layer Networks

In multi-layer network scenarios, such as an IP-over-optical network, if there are dedicated PCEs responsible for each layer, then the PCCs should be informed of which PCEs they should synchronize their LSP states with, as well as send path computation requests to. The Layer-Cap TLV defined in [INTER-LAYER] can be used to indicate which layer a PCE is in charge of. This TLV is optional and MAY be carried in the OPEN object. It is RECOMMENDED that a PCC synchronizes its LSP states with the same PCEs that it can use for path computation in a multi-layer network. In a single layer, this TLV MAY not be used. However, if the PCE capability discovery depends on IGP and if an IGP instance spans across multiple layers, this TLV is still needed.

Alternatively, the extension to current OSPF PCED TLV is needed. A new domain-type denoting the layer information can be defined:

domain-type: T.B.D.



When it is carried in PCE-DOMAIN sub-TLV, it denotes the layer for which a PCE is responsible for path computation as well as LSP state synchronization. When carried in the PCE-NEIG-DOMAIN sub-TLV, it denotes its adjacent layers for which a PCE can compute paths and synchronize the LSP states. The DOMAIN-ID information can be represented using the following format, to denote the layer information:

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	2	3	4	5	6	7	8	9	
LSP Enc. Type										Switching Type										Reserved																				

### 2.3. LSP Delegation in GMPLS Networks

To enable the PCE to control an LSP, the PCUpd message is defined in [Stateful-PCE]. However, technology-specific specification is not covered. The following defines the <path> descriptor that should be used in GMPLS networks:

$$\langle \text{path} \rangle ::= \langle \text{ERO} \rangle \langle \text{attribute-list} \rangle$$

Where:

```

<attribute-list> ::= [ <LSPA> ]
                    [ <BANDWIDTH> ]
                    [ <GENERALIZED-BANDWIDTH>... ]
                    [ <metric-list> ]

<metric-list> ::= <METRIC>[ <metric-list> ]

```

The O bit in the <GENERALIZED-BANDWIDTH> object has no meaning for LSP state synchronization and MUST be set to 0. Furthermore, this object MAY appear twice, one with R set to 1 and the other with R set to 0. This is to denote the asymmetric bandwidth property of the updated bi-directional LSP.

As explained in [stateful-APP], LSP state synchronization and/or LSP parameter change controlled by a stateful PCE in a multi-domain network is complex and requires well-defined operational procedures as well as protocol design.

[TBD: protocol extensions]

## 2.4. LSP Synchronization in GMPLS networks

For LSP state synchronization of stateful PCEs in GMPLS networks, the LSP attributes, such as its bandwidth, associated route as well as protection information etc, should be updated by PCCs to PCE LSP database (LSP-DB). Note the LSP state synchronization described in this document denotes both the bulk LSP report at the initialization phase as well as the LSP state report afterwards described in [Stateful-PCE].

As per [Stateful-PCE], it does not cover technology-specific specification for state synchronization. Therefore, extensions of PCEP protocol for stateful PCE usage in GMPLS networks are required. For LSP state synchronization, the objects/TLVs that should be used for stateful PCE in GMPLS networks are defined in [PCEP-GMPLS] and are briefly summarized as below:

- o GENERALIZED BANDWIDTH
- o GENERALIZED ENDPOINTS
- o PROTECTION ATTRIBUTE
- o Use of IF\_ID\_ERROR\_SPEC. [Stateful-PCE] section 7.2.2 only considers RSVP\_ERROR\_SPEC TLVs. GMPLS extends this to also support IF\_ID\_ERROR\_SPEC, for example, to report about failed unnumbered interfaces.
- o Extended Objects to support the inclusion of the label and unnumbered links.

Per [Stateful-PCE], the PCRpt message is defined for LSP state synchronization purpose. PCRpt is used by a PCC to report one or more of its LSPs to a stateful PCE. However, the <path> descriptor is technology-specific and left undefined.

For LSP state synchronization in GMPLS networks, the encoding of the <path> descriptor is defined as follows:

```
<path> ::= <ERO> <attribute-list>
```

Where:

```
<attribute-list> ::= [ <LSPA> ]
                        [ <BANDWIDTH> ]
```

[<GENERALIZED-BANDWIDTH>...]

[<RRO>]

[<IRO>]

[<XRO>]

[<metric-list>]

<metric-list>::= <METRIC>[<metric-list>]

The objects included in the <path> descriptor can be found in [RFC5440], [PCE-GMPLS] and [RFC5521].

For all the objects presented in this section, the P and I bit MUST be set to 0 since they are only used by a PCC to report its LSP information.

In GMPLS networks, the <ERO> object may include a list of the label sub-object for SDH/SONET, OTN and DWDM networks. It may also include a list of unnumbered interface IDs to denote the allocated resource. The <RRO>, <IRO> and <XRO> objects MAY include unnumbered interface IDs and labels for networks such as OTN and WDM networks.

If the LSP being reported is a protecting LSP, the <PROTECTION-ATTRIBUTE> TLV MUST be included in the <LSPA> object to denote its attributes and restrictions. Moreover, if the status of the protecting LSP changes from non-operational to operational, this should be synchronized to the stateful PCE. For example, in 1:1 protection, the combination of S=0, P=1 and O=0 denotes the protecting path is set up already but not used for carrying traffic. Upon the working path failure, the operational status of the aforementioned protecting LSP changes to in-use (i.e., O=1). This information should be synchronized with a stateful PCE through a PCRpt message.

The O bit in the <GENERALIZED-BANDWIDTH> object has no meaning for LSP state synchronization and MUST be set to 0. Furthermore, this object MAY appear twice, one with R set to 1 and the other with R set to 0. This is to denote the asymmetric bandwidth property of the updated bi-directional LSP.

## 2.5. Modification of Existing PCEP Messages and Procedures

One of the advantages mentioned in [Stateful-APP] is that the stateful nature of a PCE simplifies the information conveyed in PCEP messages, notably between PCC and PCE, since it is possible to refer to PCE managed state for active LSPs. To be more specific, with a stateful PCE, it is possible to refer to a LSP with a unique identifier in the scope of the PCC-PCEP session and thus use such identifier to refer to that LSP.

### 2.5.1. Use cases

Use Case 1: Assuming a stateful PCE's LSP-DB is up-to-date, a PCC (e.g. NMS) requesting for a re-optimization of one or several LSPs can send the request with 'R' bit set and only provides the relevant LSP unique identifiers.

Upon receiving the PCReq message, PCE should be able to correlate with one or multiple LSPs with their detailed state information and carry out optimization accordingly.

The handling of RP object specified in [RFC5440] is stated as following:

'The absence of an RRO in the PCReq message for a non-zero-bandwidth TE LSP (when the R bit of the RP object is set) MUST trigger the sending of a PCErr message with Error-Type="Required Object Missing" and Error-value="RRO Object missing for re-optimization."

If a PCE has stateful capabilities, and such capabilities have been negotiated and advertised, specific rules given in [RFC5440] may need to be relaxed. In particular, the re-optimization case: if the re-optimization request refers to a given LSP state, and the RRO information is available, the PCE can proceed.

Use Case 2: in order to set up a LSP which has a constraint that its route should not use resources used by one or more existing LSPs, a PCC can send a PCReq with the identifiers of these LSPs. A stateful PCE should be able to find the corresponding route and resource information so as to meet the constraints set by the requesting PCC. Hence, the LSP identifier TLV defined in [Stateful-PCE] can be used in XRO object for this purpose. Note that if the PCC is a node in the network, the constraint LSP ID information will be confined to the LSPs initiated by itself.

## 2.5.2. Modification for LSP Re-optimization

For re-optimization, upon receiving a path computation request and the 'R' bit is set, the stateful PCE SHOULD still perform the re-optimization in the following two cases:

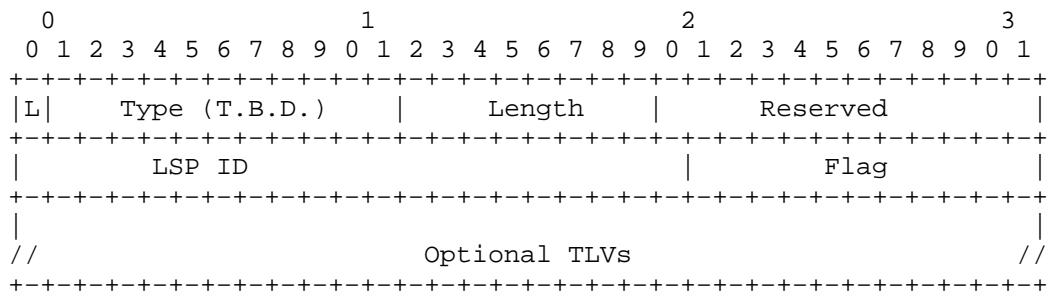
Case 1: the existing bandwidth and route information of the to-be-optimized LSP is provided in the path computation request. This information should be provided via <BANDWIDTH>, <GENERALIZED-BANDWIDTH>, <ERO> objects.

Case 2: the existing bandwidth and route information can be found locally in its LSP-DB. In this case, the PCRep and PCReq messages need to be modified to carry LSP identifiers. The stateful PCE can find this information using the per-node LSP ID together with the PCC's address.

If no LSP state information is available to carry out re-optimization, the stateful PCE should report the error 'LSP state information unavailable for the LSP re-optimization' (Error Type = T.B.D., Error value= T.B.D.).

## 2.5.3. Modification for Route Exclusion

A LSP identifier sub-object is defined and its format as follows:



L bit:

The L bit SHOULD NOT be set, so that the subobject represents a strict hop in the explicit route.

Type:

Subobject Type for a per-node LSP identifier.

Length:

The Length contains the total length of the subobject in bytes, including the Type and Length fields.

**LSP ID:**

This is the identifier given to a LSP and it is unique on a node basis. It is defined in [Stateful-PCE].

**Flags:**

This field is defined in [Stateful-PCE]. It is not used in this sub-object and should be ignored upon receipt.

**Optional TLVs:**

Additional TLVs can be defined in the future to provide further information to identify a LSP. In this document, no TLVs are defined.

One or multiple of these sub-objects can be present in the XRO object. When a stateful PCE receives a path computation request carrying this sub-object, it should find relevant information of these LSPs and preclude the resource during the path computation process. If a stateful PCE cannot recognize one or more of the received LSP identifiers, it should reply PCErr saying "the LSP state information for route exclusion purpose cannot be found" (Error-type = T.B.D., Error-value= T.B.D.). Optionally, it may provide with the unrecognized identifier information to the requesting PCC.

## 2.6. Additional Error Type and Error Values Defined

**Error Type Meaning**

21(TBD)      LSP state information missing

                 Error-value 1: LSP state information unavailable for the LSP re-optimization

## 3. Error-value 2: the LSP state information for route exclusion purpose cannot be found IANA Considerations

IANA is requested to allocate new Types for the TLV/Object defined in this document.

T.B.D.

## 4. Manageability Considerations

The description and functionality specifications presented related to stateful PCEs should also comply with the manageability specifications covered in Section 8 of [RFC4655]. Furthermore, a further list of manageability issues presented in [Stateful-PCE] should also be considered.

Additional considerations are presented in the next sections.

#### 4.1. Requirements on Other Protocols and Functional Components

When the detailed route information is included for LSP state synchronization (either at the initial stage or during LSP state report process), this require the ingress node of an LSP carry the RRO object in order to enable the collection of such information.

#### 5. Security Considerations

The security issues presented in [RFC5440] and [Stateful-PCE] apply to this document.

#### 6. Acknowledgement

We would like to thank Adrian Farrel and Cyril Margaria for the useful comments and discussions.

#### 7. References

##### 7.1. 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.
- [RFC5088] Le Roux, JL., Vasseur, J.-P., Ikejiri, Y., Zhang, R., ''OSPF Protocol Extensions for Path Computation Element (PCE) Discovery'', RFC 5088, January 2008.
- [INTER-LAYER] Oki, E., Takeda, Tomonori, Le Roux, JL., Farrel, A., Zhang, F., ''Extensions to the Path Computation Element communication Protocol(PCEP) for Inter-Layer MPLS and GMPLS Traffic Engineering'', draft-ietf-pce-inter-layer-ext-08.txt, XX 2013.

## 7.2. Informative References

- [Stateful-APP] Zhang, F., Zhang, X., Lee, Y., Casellas, R., Gonzalez de Dios, O., "Applicability of Stateful Path Computation Element (PCE) ", draft-zhang-pce-stateful-pce-app-03, work in progress.
- [Stateful-PCE] Crabbe, E., Medved, J., Varga, R., Minei, I., ''PCEP Extensions for Stateful PCE'', draft-ietf-pce-stateful-pce, work in progress.
- [PCE-IA-WSO] Lee, Y., Bernstein G., Takeda, T., Tsuritani, T., ''PCEP Extensions for WSON Impairments'', draft-lee-pce-wson-impairments, work in progress.
- [PCEP-GMPLS] Margaria, C., Gonzalez de Dios, O., Zhang, F., ''PCEP extensions for GMPLS'', draft-ietf-pce-gmpls-pcep-extensions, work in progress.

## 8. Contributors' Address

Dhruv Dhody  
Huawei Technology  
Leela Palace  
Bangalore, Karnataka 560008  
INDIA

EMail: dhruvd@huawei.com

Yi Lin  
Huawei Technologies  
F3-5-B R&D Center, Huawei Base  
Bantian, Longgang District  
Shenzhen 518129 P.R.China

Phone: +86-755-28972914  
Email: yi.lin@huawei.com



Authors' Addresses

Xian Zhang  
Huawei Technologies  
F3-5-B R&D Center, Huawei Base  
Bantian, Longgang District  
Shenzhen 518129 P.R.China

Phone: +86-755-28972913  
Email: zhang.xian@huawei.com

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

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

Fatai Zhang  
Huawei  
F3-5-B R&D Center, Huawei Base  
Bantian, Longgang District  
P.R. China

Phone: +86-755-28972912  
Email: zhangfatai@huawei.com

Ramon Casellas  
CTTC  
Av. Carl Friedrich Gauss n7  
Castelldefels, Barcelona 08860  
Spain

Phone:  
Email: ramon.casellas@cttc.es

Oscar Gonzalez de Dios  
Telefonica Investigacion y Desarrollo  
Emilio Vargas 6  
Madrid, 28045  
Spain

Phone: +34 913374013  
Email: ogondio@tid.es

## Intellectual Property

The IETF Trust takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in any IETF Document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights.

Copies of Intellectual Property disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at <http://www.ietf.org/ipr>

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement any standard or specification contained in an IETF Document. Please address the information to the IETF at [ietf-ipr@ietf.org](mailto:ietf-ipr@ietf.org).

The definitive version of an IETF Document is that published by, or under the auspices of, the IETF. Versions of IETF Documents that are published by third parties, including those that are translated into other languages, should not be considered to be definitive versions of IETF Documents. The definitive version of these Legal Provisions is that published by, or under the auspices of, the IETF. Versions of these Legal Provisions that are published by third parties, including those that are translated into other languages, should not be considered to be definitive versions of these Legal Provisions.

For the avoidance of doubt, each Contributor to the IETF Standards Process licenses each Contribution that he or she makes as part of the IETF Standards Process to the IETF Trust pursuant to the provisions of RFC 5378. No language to the contrary, or terms, conditions or rights that differ from or are inconsistent with the rights and licenses granted under RFC 5378, shall have any effect and shall be null and void, whether published or posted by such Contributor, or included with or in such Contribution.

#### Disclaimer of Validity

All IETF Documents and the information contained therein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY, THE IETF TRUST AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION THEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

#### Full Copyright Statement

Copyright (c) 2013 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.



Network Working Group  
Internet-Draft  
Intended status: Standards Track

Xian Zhang  
Young Lee  
Fatai Zhang  
Huawei  
Ramon Casellas  
CTTC  
Oscar Gonzalez de Dios  
Telefonica I+D  
Zafar Ali  
Cisco Systems

Expires: April 21, 2014

October 21, 2013

Path Computation Element (PCE) Protocol Extensions for Stateful PCE  
Usage in GMPLS-controlled Networks

draft-zhang-pce-pcep-stateful-pce-gmpls-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/lid-abstracts.txt>.

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

This Internet-Draft will expire on April 21, 2014.

## Abstract

The Path Computation Element (PCE) facilitates Traffic Engineering (TE) based path calculation in large, multi-domain, multi-region, or multi-layer networks. [Stateful-PCE] provides the fundamental PCE communication Protocol (PCEP) extensions needed to support stateful PCE functions, without specifying the technology-specific extensions. This memo provides extensions required for PCEP so as to enable the usage of a stateful PCE capability in GMPLS-controlled networks.

## Conventions used in this document

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

## Table of Contents

Table of Contents .....	2
1. Introduction .....	3
2. PCEP Extensions .....	3
2.1. Overview of Requirements.....	3
2.2. Stateful PCE Capability Advertisement .....	4
2.2.1. PCE Capability Advertisement in Multi-layer Networks	4
2.3. LSP Delegation in GMPLS-controlled Networks .....	5
2.4. LSP Synchronization in GMPLS-controlled networks.....	6
2.5. Modification of Existing PCEP Messages and Procedures....	7
2.5.1. Use cases .....	8
2.5.2. Modification for LSP Re-optimization .....	8
2.5.3. Modification for Route Exclusion .....	9
2.6. Additional Error Type and Error Values Defined.....	10
3. IANA Considerations .....	10
4. Manageability Considerations .....	10
4.1. Requirements on Other Protocols and Functional Components	10
5. Security Considerations.....	11
6. Acknowledgement .....	11
7. References .....	11
7.1. Normative References.....	11
7.2. Informative References.....	11
8. Contributors' Address.....	12
Authors' Addresses .....	13

## 1. Introduction

[RFC 4655] presents the architecture of a Path Computation Element (PCE)-based model for computing Multiprotocol Label Switching (MPLS) and Generalized MPLS (GMPLS) Traffic Engineering Label Switched Paths (TE LSPs). To perform such a constrained computation, a PCE stores the network topology (i.e., TE links and nodes) and resource information (i.e., TE attributes) in its TE Database (TED). To request path computation services to a PCE, [RFC 5440] defines the PCE communication Protocol (PCEP) for interaction between a Path Computation Client (PCC) and a PCE, or between two PCEs. PCEP as specified in [RFC 5440] mainly focuses on MPLS networks and the PCEP extensions needed for GMPLS-controlled networks are provided in [PCEP-GMPLS].

Stateful PCEs are shown to be helpful in many application scenarios, in both MPLS and GMPLS networks, as illustrated in [Stateful-APP]. In order for these applications to be able to exploit the capability of stateful PCEs, extensions to the PCE communication protocol (i.e., PCEP) are required.

[Stateful-PCE] provides the fundamental extensions needed for stateful PCE to support general functionality, but leaves out the specification for technology-specific objects/TLVs. Complementarily, this document focuses on the extensions that are necessary in order for the deployment of stateful PCEs in GMPLS-controlled networks.

## 2. PCEP Extensions

### 2.1. Overview of Requirements

This section notes the main functional requirements for PCEP extensions to support stateful PCE for use in GMPLS-controlled networks, based on the description in [Stateful-APP]. Many requirements are common across a variety of network types (e.g., MPLS-TE networks and GMPLS networks) and the protocol extensions to meet the requirements are already described in [Stateful-PCE]. This document does not repeat the description of those protocol extensions. Other requirements that are also common across a variety of network types do not currently have protocol extensions defined in [Stateful-PCE]. In these cases, this document presents protocol extensions for discussion by the PCE working group and potential inclusion in [Stateful-PCE]. In addition, this document presents protocol extensions for a set of requirements which are specific to the use of a stateful PCE in a GMPLS-controlled network.

The basic requirements are as follows:

- o Advertisement of the stateful PCE capability. This generic requirement is covered in Section 7.1.1 of [Stateful-PCE]. Section 2.2 of this document discusses other potential extensions for this functionality.
- o LSP delegation is already covered in Section 5.5 of [Stateful-PCE]. Section 2.3 of this document provides extension for its application in GMPLS-controlled networks. Moreover, further discussion of some generic details that may need additional consideration is provided.
- o LSP state synchronization. This is a generic requirement already covered in Section 5.4 of [Stateful-PCE]. However, there are further extensions required specifically for GMPLS-controlled networks and discussed in Section 2.4. Reference to LSPs by identifiers is discussed in Section 7.2 of [Stateful-PCE]. This feature can be applied to reduce the data carried in PCEP messages. Use cases and additional Error Codes are necessary, as described in Section 2.5 and 2.6.

## 2.2. Stateful PCE Capability Advertisement

Whether a PCE has stateful capability or not can be advertised during the PCEP session establishment process. It can also be advertised through routing protocols as described in [RFC5088]. In either case, the following additional aspects should also be considered.

### 2.2.1. PCE Capability Advertisement in Multi-layer Networks

In multi-layer network scenarios, such as an IP-over-optical network, if there are dedicated PCEs responsible for each layer, then the PCCs should be informed of which PCEs they should synchronize their LSP states with, as well as send path computation requests to. The Layer-Cap TLV defined in [INTER-LAYER] can be used to indicate which layer a PCE is in charge of. (Editor's note: this change is currently not included in the current version of the [INTER-LAYER] draft. It is expected that it will be included in its next version.) This TLV is optional and MAY be carried in the OPEN object. It is RECOMMENDED that a PCC synchronizes its LSP states with the same PCEs that it can use for path computation in a multi-layer network. In a single layer, this TLV MAY not be used. However, if the PCE capability discovery depends on IGP and if an IGP instance spans across multiple layers, this TLV is still needed.

Alternatively, the extension to current OSPF PCED TLV is needed. A new domain-type denoting the layer information can be defined:



domain-type: T.B.D.

When it is carried in PCE-DOMAIN sub-TLV, it denotes the layer for which a PCE is responsible for path computation as well as LSP state synchronization. When carried in the PCE-NEIG-DOMAIN sub-TLV, it denotes its adjacent layers for which a PCE can compute paths and synchronize the LSP states. The DOMAIN-ID information can be represented using the following format, to denote the layer information:

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								
LSP Enc. Type										Switching Type										Reserved																			

### 2.3. LSP Delegation in GMPLS-controlled Networks

To enable the PCE to control an LSP, the PCUpd message is defined in [Stateful-PCE]. However, the specification of technology specific extensions is not covered. The following defines the <path> descriptor, present in the PCUpd message, that should be used in GMPLS-controlled networks:

<path> ::= <ERO> <attribute-list>

Where:

```

<attribute-list> ::= [ <LSPA> ]
                    [ <BANDWIDTH> ]
                    [ <GENERALIZED-BANDWIDTH>... ]
                    [ <metric-list> ]

<metric-list> ::= <METRIC> [ <metric-list> ]

```

As explained in [stateful-APP], LSP parameter update controlled by a stateful PCE in a multi-domain network is complex and requires well-defined operational procedures as well as protocol design.

[TBD: protocol extensions]

## 2.4. LSP Synchronization in GMPLS-controlled networks

For LSP state synchronization of stateful PCEs in GMPLS networks, the LSP attributes, such as its bandwidth, associated route as well as protection information etc, should be updated by PCCs to PCE LSP database (LSP-DB). Note the LSP state synchronization described in this document denotes both the bulk LSP report at the initialization phase as well as the LSP state report afterwards described in [Stateful-PCE].

As per [Stateful-PCE], it does not cover technology-specific specification for state synchronization. Therefore, extensions of PCEP for stateful PCE usage in GMPLS networks are required. For LSP state synchronization, the objects/TLVs that should be used for stateful PCE in GMPLS networks are defined in [PCEP-GMPLS] and are briefly summarized as below:

- o GENERALIZED BANDWIDTH
- o GENERALIZED ENDPOINTS
- o PROTECTION ATTRIBUTE
- o Use of IF\_ID\_ERROR\_SPEC. [Stateful-PCE] section 7.2.2 only considers RSVP\_ERROR\_SPEC TLVs. GMPLS extends this to also support IF\_ID\_ERROR\_SPEC, for example, to report about failed unnumbered interfaces.
- o Extended objects to support the inclusion of the label and unnumbered links.

Per [Stateful-PCE], the PCRpt message is defined for LSP state synchronization purposes. PCRpt is used by a PCC to report one or more of its LSPs to a stateful PCE. However, the <path> descriptor is technology-specific and left undefined.

For LSP state synchronization in GMPLS-controlled networks, the encoding of the <path> descriptor is defined as follows:

```
<path> ::= <ERO> <attribute-list>
```

Where:

```
<attribute-list> ::= [<LSPA>]
                    [<BANDWIDTH>]
                    [<GENERALIZED-BANDWIDTH>...]
```

[<IRO>]

[<XRO>]

[<metric-list>]

<metric-list>::= <METRIC>[<metric-list>]

The objects included in the <path> descriptor can be found in [RFC5440], [PCE-GMPLS] and [RFC5521].

For all the objects presented in this section, the P and I bit MUST be set to 0 since they are only used by a PCC to report its LSP information.

In GMPLS-controlled networks, the <ERO> object may include a list of the label sub-object for SDH/SONET, OTN and DWDM networks. It may also include a list of unnumbered interface IDs to denote the allocated resource. The <RRO>, <IRO> and <XRO> objects MAY include unnumbered interface IDs and labels for networks such as OTN and WDM networks.

If the LSP being reported is a protecting LSP, the <PROTECTION-ATTRIBUTE> TLV MUST be included in the <LSPA> object to denote its attributes and restrictions. Moreover, if the status of the protecting LSP changes from non-operational to operational, this should be synchronized to the stateful PCE. For example, in 1:1 protection, the combination of S=0, P=1 and O=0 denotes the protecting path is set up already but not used for carrying traffic. Upon the working path failure, the operational status of the aforementioned protecting LSP changes to in-use (i.e., O=1). This information should be synchronized with a stateful PCE through a PCRpt message.

The O bit in the <GENERALIZED-BANDWIDTH> object has no meaning for LSP state synchronization and MUST be set to 0. Furthermore, this object MAY appear twice, one with R set to 1 and the other with R set to 0. This is to denote the asymmetric bandwidth property of the updated bi-directional LSP.

## 2.5. Modification of Existing PCEP Messages and Procedures

One of the advantages mentioned in [Stateful-APP] is that the stateful nature of a PCE simplifies the information conveyed in PCEP messages, notably between PCC and PCE, since it is possible to refer to PCE managed state for active LSPs. To be more specific, with a

stateful PCE, it is possible to refer to a LSP with a unique identifier in the scope of the PCC-PCEP session and thus use such identifier to refer to that LSP.

#### 2.5.1. Use cases

Use Case 1: Assuming a stateful PCE's LSP-DB is up-to-date, a PCC (e.g. NMS) requesting for a re-optimization of one or several LSPs can send the request with "R" bit set and only provides the relevant LSP unique identifiers.

Upon receiving the PCReq message, PCE should be able to correlate with one or multiple LSPs with their detailed state information and carry out optimization accordingly.

The handling of RP object specified in [RFC5440] is stated as following:

"The absence of an RRO in the PCReq message for a non-zero-bandwidth TE LSP (when the R bit of the RP object is set) MUST trigger the sending of a PCErr message with Error-Type="Required Object Missing" and Error-value="RRO Object missing for re-optimization."

If a PCE has stateful capabilities, and such capabilities have been negotiated and advertised, specific rules given in [RFC5440] may need to be relaxed. In particular, the re-optimization case: if the re-optimization request refers to a given LSP state, and the RRO information is available, the PCE can proceed.

Use Case 2: in order to set up a LSP which has a constraint that its route should not use resources used by one or more existing LSPs, a PCC can send a PCReq with the identifiers of these LSPs. A stateful PCE should be able to find the corresponding route and resource information so as to meet the constraints set by the requesting PCC. Hence, the LSP identifier TLV defined in [Stateful-PCE] can be used in XRO object for this purpose. Note that if the PCC is a node in the network, the constraint LSP ID information will be confined to the LSPs initiated by itself.

#### 2.5.2. Modification for LSP Re-optimization

For re-optimization, upon receiving a path computation request and the "R" bit is set, the stateful PCE SHOULD still perform the re-optimization in the following two cases:

Case 1: the existing bandwidth and route information of the to-be-optimized LSP is provided in the path computation request. This

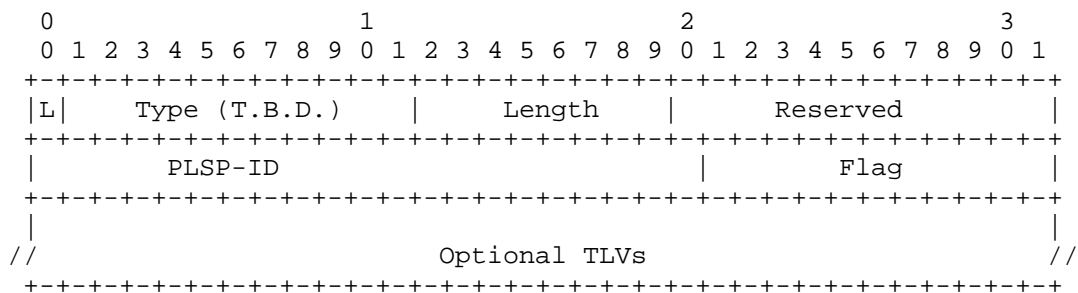
information should be provided via <BANDWIDTH>, <GENERALIZED-BANDWIDTH>, <ERO> objects.

Case 2: the existing bandwidth and route information can be found locally in its LSP-DB. In this case, the PCRep and PCReq messages need to be modified to carry LSP identifiers. The stateful PCE can find this information using the per-node LSP ID together with the PCC's address.

If no LSP state information is available to carry out re-optimization, the stateful PCE should report the error "LSP state information unavailable for the LSP re-optimization" (Error Type = T.B.D., Error value= T.B.D.).

### 2.5.3. Modification for Route Exclusion

A LSP identifier sub-object is defined and its format as follows:



L bit:

The L bit SHOULD NOT be set, so that the subobject represents a strict hop in the explicit route.

Type:

Subobject Type for a per-node LSP identifier.

Length:

The Length contains the total length of the subobject in bytes, including the Type and Length fields.

PLSP-ID:

This is the identifier given to a LSP and it is unique on a node basis. It is defined in [Stateful-PCE].

Flags:

This field is defined in [Stateful-PCE]. It is not used in this sub-object and should be ignored upon receipt.

#### Optional TLVs:

Additional TLVs can be defined in the future to provide further information to identify a LSP. In this document, no TLVs are defined.

One or multiple of these sub-objects can be present in the XRO object. When a stateful PCE receives a path computation request carrying this sub-object, it should find relevant information of these LSPs and preclude the resource during the path computation process. If a stateful PCE cannot recognize one or more of the received LSP identifiers, it should reply PCErr saying "the LSP state information for route exclusion purpose cannot be found" (Error-type = T.B.D., Error-value= T.B.D.). Optionally, it may provide with the unrecognized identifier information to the requesting PCC.

### 2.6. Additional Error Type and Error Values Defined

#### Error Type Meaning

21(TBD)      LSP state information missing

Error-value 1: LSP state information unavailable for the LSP re-optimization

Error-value 2: the LSP state information for route exclusion purpose cannot be found

### 3. IANA Considerations

IANA is requested to allocate new Types for the TLV/Object defined in this document.T.B.D.

### 4. Manageability Considerations

The description and functionality specifications presented related to stateful PCEs should also comply with the manageability specifications covered in Section 8 of [RFC4655]. Furthermore, a further list of manageability issues presented in [Stateful-PCE] should also be considered.

Additional considerations are presented in the next sections.

#### 4.1. Requirements on Other Protocols and Functional Components

When the detailed route information is included for LSP state synchronization (either at the initial stage or during LSP state

report process), this require the ingress node of an LSP carry the RRO object in order to enable the collection of such information.

## 5. Security Considerations

The security issues presented in [RFC5440] and [Stateful-PCE] apply to this document.

## 6. Acknowledgement

We would like to thank Adrian Farrel and Cyril Margaria for the useful comments and discussions.

## 7. References

### 7.1. 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.
- [RFC5088] Le Roux, JL., Vasseur, J.-P., Ikejiri, Y., Zhang, R., "OSPF Protocol Extensions for Path Computation Element (PCE) Discovery", RFC 5088, January 2008.
- [INTER-LAYER] Oki, E., Takeda, Tomonori, Le Roux, JL., Farrel, A., Zhang, F., "Extensions to the Path Computation Element communication Protocol (PCEP) for Inter-Layer MPLS and GMPLS Traffic Engineering", draft-ietf-pce-inter-layer-ext, work in progress.

### 7.2. Informative References

- [Stateful-APP] Zhang, X., Minei, I., et al "Applicability of Stateful Path Computation Element (PCE) ", draft-ietf-pce-stateful-pce-app, , work in progress.
- [Stateful-PCE] Crabbe, E., Medved, J., Varga, R., Minei, I., "PCEP Extensions for Stateful PCE", draft-ietf-pce-stateful-pce, work in progress.

[PCE-IA-WSO] Lee, Y., Bernstein G., Takeda, T., Tsuritani, T.,  
"PCEP Extensions for WSO Impairments", draft-lee-pce-  
wso-impairments, work in progress.

[PCEP-GMPLS] Margaria, C., Gonzalez de Dios, O., Zhang, F., "PCEP  
extensions for GMPLS", draft-ietf-pce-gmpls-pcep-  
extensions, work in progress.

#### 8. Contributors' Address

Dhruv Dhody  
Huawei Technology  
Leela Palace  
Bangalore, Karnataka 560008  
INDIA

EMail: dhruvd@huawei.com

Yi Lin  
Huawei Technologies  
F3-5-B R&D Center, Huawei Base  
Bantian, Longgang District  
Shenzhen 518129 P.R.China

Phone: +86-755-28972914  
Email: yi.lin@huawei.com



Authors' Addresses

Xian Zhang  
Huawei Technologies  
F3-5-B R&D Center, Huawei Base  
Bantian, Longgang District  
Shenzhen 518129 P.R.China

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

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

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

Fatai Zhang  
Huawei  
F3-5-B R&D Center, Huawei Base  
Bantian, Longgang District  
P.R. China

Phone: +86-755-28972912  
Email: zhangfatai@huawei.com

Ramon Casellas  
CTTC  
Av. Carl Friedrich Gauss n7  
Castelldefels, Barcelona 08860  
Spain

Phone:  
Email: ramon.casellas@cttc.es

Oscar Gonzalez de Dios  
Telefonica Investigacion y Desarrollo  
Emilio Vargas 6  
Madrid, 28045  
Spain

Phone: +34 913374013  
Email: ogondio@tid.es

Zafar Ali  
Cisco Systems  
Email: zali@cisco.com

## Intellectual Property

The IETF Trust takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in any IETF Document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights.

Copies of Intellectual Property disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at <http://www.ietf.org/ipr>

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement any standard or specification contained in an IETF Document. Please address the information to the IETF at [ietf-ipr@ietf.org](mailto:ietf-ipr@ietf.org).

The definitive version of an IETF Document is that published by, or under the auspices of, the IETF. Versions of IETF Documents that are published by third parties, including those that are translated into other languages, should not be considered to be definitive versions of IETF Documents. The definitive version of these Legal Provisions is that published by, or under the auspices of, the IETF. Versions of these Legal Provisions that are published by third parties, including those that are translated into other languages, should not be considered to be definitive versions of these Legal Provisions.

For the avoidance of doubt, each Contributor to the IETF Standards Process licenses each Contribution that he or she makes as part of the IETF Standards Process to the IETF Trust pursuant to the provisions of RFC 5378. No language to the contrary, or terms,

conditions or rights that differ from or are inconsistent with the rights and licenses granted under RFC 5378, shall have any effect and shall be null and void, whether published or posted by such Contributor, or included with or in such Contribution.

#### Disclaimer of Validity

All IETF Documents and the information contained therein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY, THE IETF TRUST AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION THEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

#### Full Copyright Statement

Copyright (c) 2013 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.



Network Working Group  
Internet-Draft  
Intended status: Informational  
Expires: August 26, 2013

X. Zhang, Ed.  
Huawei Technologies  
I. Minei, Ed.  
Juniper Networks, Inc.  
February 22, 2013

Applicability of PCEP Extensions for Stateful PCE  
draft-zhang-pce-stateful-pce-app-03

## Abstract

A Stateful PCE maintains information about LSP characteristics and resource usage within the network in order to provide traffic engineering calculations for its associated PCCs. This document describes general considerations for stateful PCEP and examines its applicability and benefits through a number of use cases. PCEP extensions required for stateful PCE usage are covered in separate documents.

## 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 26, 2013.

## Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

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

include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	3
2. Terminology . . . . .	3
3. Overview of stateful PCE . . . . .	4
4. Deployment considerations . . . . .	5
4.1. Multi-PCE deployments . . . . .	5
4.2. LSP State Synchronization . . . . .	5
4.3. PCE Survivability . . . . .	5
5. Application scenarios . . . . .	5
5.1. Optimization of LSP placement . . . . .	6
5.1.1. Throughput Maximization and Bin Packing . . . . .	6
5.1.2. Deadlock . . . . .	8
5.1.3. Minimum Perturbation . . . . .	9
5.1.4. Predictability . . . . .	10
5.2. Stateful PCE in SDN . . . . .	11
5.2.1. Smart Bandwidth Adjustment . . . . .	11
5.2.2. Bandwidth Scheduling . . . . .	12
5.3. Recovery . . . . .	12
5.3.1. Protection . . . . .	12
5.3.2. Restoration . . . . .	13
5.3.3. SRLG Diversity . . . . .	14
5.4. Maintenance of Virtual Network Topology (VNT) . . . . .	15
5.5. LSP Re-optimization . . . . .	15
5.6. Resource defragmentation . . . . .	16
5.7. Future applications . . . . .	16
5.7.1. Impairment-Aware Routing and Wavelength Assignment (IA-RWA) . . . . .	16
6. Security Considerations . . . . .	17
7. Contributing authors . . . . .	18
8. Acknowledgements . . . . .	19
9. References . . . . .	19
9.1. Normative References . . . . .	19
9.2. Informative References . . . . .	20
Appendix A. Editorial notes and open issues . . . . .	22
Authors' Addresses . . . . .	23

## 1. Introduction

[RFC5440] describes the Path Computation Element Protocol (PCEP). PCEP defines the communication between a Path Computation Client (PCC) and a Path Control Element (PCE), or between PCE and PCE, enabling computation of Multiprotocol Label Switching (MPLS) for Traffic Engineering Label Switched Path (TE LSP) characteristics. Extensions for support of GMPLS in PCEP are defined in [I-D.ietf-pce-gmpls-pcep-extensions].

As per [RFC4655], a PCE can be either stateful or stateless. Compared to a stateless PCE, a stateful PCE has access to not only the network states, but also to the set of active paths and their reserved resources in use in the network. In other words, the state in a stateful PCE is determined not only by the TED but also by the set of active LSPs and their corresponding reserved resources. Furthermore, a stateful PCE might also retain the information of LSPs under construction in order to reduce resource contention. Such augmented state allows the PCE to compute constrained paths while considering individual LSPs and their interaction.

This document describes how stateful PCE can solve various problems for MPLS-TE and GMPLS deployment use cases, and the benefits it brings to such deployments.

## 2. Terminology

This document uses the following terms defined in [RFC5440]: PCC, PCE, PCEP Peer.

This document uses the following terms defined in [I-D.ietf-pce-stateful-pce]: Passive Stateful PCE, Active Stateful PCE, Delegation, Revocation, Delegation Timeout Interval, LSP State Report, LSP Update Request, LSP State Database.

This document defines the following terms:

**Minimum Cut Set:** the minimum set of links for a specific source destination pair which, when removed from the network, result in a specific source being completely isolated from specific destination. The summed capacity of these links is equivalent to the maximum capacity from the source to the destination by the max-flow min-cut theorem.

### 3. Overview of stateful PCE

This section is included for the convenience of the reader, please refer to the specification documents for details of the operation.

[I-D.ietf-pce-stateful-pce] specifies a set of extensions to PCEP to enable stateful control of tunnels between and across PCEP sessions in compliance with [RFC4657]. It includes mechanisms to effect tunnel state synchronization between PCCs and PCEs, delegation of control over tunnels to PCEs, and PCE control of timing and sequence of path computations within and across PCEP sessions.

[I-D.ietf-pce-stateful-pce] applies equally to MPLS-TE and GMPLS LSPs.

Several new functions were added in PCEP to support stateful PCEs and are described in [I-D.ietf-pce-stateful-pce]. A function can be initiated either from a PCC towards a PCE (C-E) or from a PCE towards a PCC (E-C). The new functions are:

Capability negotiation (E-C,C-E): both the PCC and the PCE must announce during PCEP session establishment that they support PCEP Stateful PCE extensions.

LSP state synchronization (C-E): after the session between the PCC and a stateful PCE is initialized, the PCE must learn the state of a PCC's LSPs before it can perform path computations or update LSP attributes in a PCC.

LSP Update Request (E-C): A PCE requests modification of attributes on a PCC's LSP.

LSP State Report (C-E): a PCC sends an LSP state report to a PCE whenever the state of an LSP changes.

LSP control delegation (C-E,E-C): a PCC grants to a PCE the right to update LSP attributes on one or more LSPs; the PCE becomes the authoritative source of the LSP's attributes as long as the delegation is in effect; the PCC may withdraw the delegation or the PCE may give up the delegation.

[I-D.sivabalan-pce-disco-stateful] defines the extensions needed to support autodiscovery of stateful PCEs when using the IGPs for PCE discovery.



## 4. Deployment considerations

### 4.1. Multi-PCE deployments

Stateless and stateful PCEs can co-exist in the same network and be in charge of path computation of different types. To solve the problem of distinguishing between the two types of PCEs, either discovery or configuration can be used. The capability negotiation in [I-D.ietf-pce-stateful-pce] ensures correct operation when the PCE address is configured on the PCC.

### 4.2. LSP State Synchronization

A stateful PCE maintains two databases for path computation. The first database is the Traffic Engineering Database (TED) which includes the topology and resource state in the network. This information can be obtained by a stateful PCE using the same mechanisms as a stateless PCE (see [RFC4655]). The second database is the LSP state Database (LSP-DB), in which a PCE stores attributes of all active LSPs in the network, such as their path through the network, bandwidth/resource usage, switching types, and LSP constraints etc. The stateful PCE extensions support population of this database using information received from the network nodes via LSP Report messages. Population of the LSP database via other means is not precluded.

### 4.3. PCE Survivability

For a stateful PCE, an important issue is to get the LSP state information resynchronized after a restart. [I-D.ietf-pce-stateful-pce] includes support of a synchronization function, allowing the PCC to synchronize its LSP state with the PCE. This can be applied equally to an LER client or another PCE, allowing for support of multiple ways of re-acquiring the LSP database on a restart. For example, the state can be retrieved from the network nodes, or from another stateful PCE. Because synchronization may also be skipped, if a PCE implementation has the means to retrieve its database in a different way (for example from a backup copy stored locally), the state can be restored without further overhead in the network.

## 5. Application scenarios

In the following sections, several use cases are described, showcasing scenarios that benefit from the deployment of a stateful PCE.

### 5.1. Optimization of LSP placement

The following use cases demonstrate a need for visibility into global inter-PCC LSP state in PCE path computations, and for a PCE control of sequence and timing in altering LSP path characteristics within and across PCEP sessions. Reference topologies for the use cases described later in this section are shown in Figures 1 and 2.

Although the use cases are for MPLS-TE deployments, they are equally applicable to GMPLS. Unless otherwise cited, use cases assume that all LSPs listed exist at the same LSP priority.

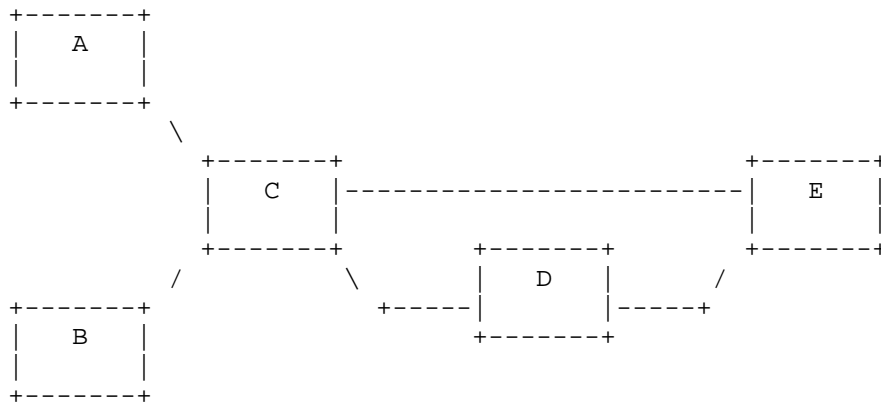


Figure 1: Reference topology 1

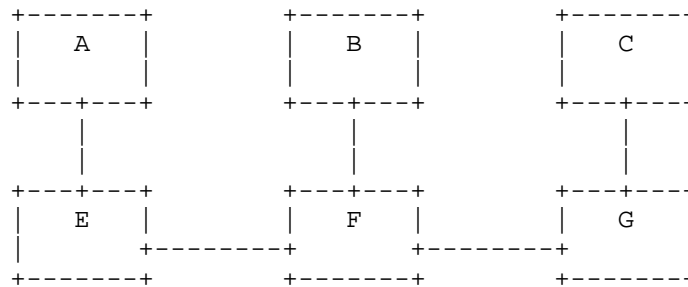


Figure 2: Reference topology 2

#### 5.1.1. Throughput Maximization and Bin Packing

Because LSP attribute changes in [RFC5440] are driven by PCReq messages under control of a PCC's local timers, the sequence of RSVP reservation arrivals occurring in the network will be randomized.

This, coupled with a lack of global LSP state visibility on the part of a stateless PCE may result in suboptimal throughput in a given network topology.

Reference topology 2 in Figure 2 and Tables 1 and 2 show an example in which throughput is at 50% of optimal as a result of lack of visibility and synchronized control across PCC's. In this scenario, the decision must be made as to whether to route any portion of the E-G demand, as any demand routed for this source and destination will decrease system throughput.

Link	Metric	Capacity
A-E	1	10
B-F	1	10
C-G	1	10
E-F	1	10
F-G	1	10

Table 1: Link parameters for Throughput use case

Time	LSP	Src	Dst	Demand	Routable	Path
1	1	E	G	10	Yes	E-F-G
2	2	A	B	10	No	---
3	1	F	C	10	No	---

Table 2: Throughput use case demand time series

In many cases throughput maximization becomes a bin packing problem. While bin packing itself is an NP-hard problem, a number of common heuristics which run in polynomial time can provide significant improvements in throughput over random reservation event distribution, especially when traversing links which are members of the minimum cut set for a large subset of source destination pairs.

Tables 3 and 4 show a simple use case using Reference Topology 1 in Figure 1, where LSP state visibility and control of reservation order across PCCs would result in significant improvement in total throughput.

Link	Metric	Capacity
A-C	1	10
B-C	1	10
C-E	10	5
C-D	1	10
D-E	1	10

Table 3: Link parameters for Bin Packing use case

Time	LSP	Src	Dst	Demand	Routable	Path
1	1	A	E	5	Yes	A-C-D-E
2	2	B	E	10	No	---

Table 4: Bin Packing use case demand time series

#### 5.1.2. Deadlock

Most existing RSVP-TE implementations will not tear down established LSPs in the event of the failure of the bandwidth increase procedure detailed in [RFC3209]. This behavior is directly implied to be correct in [RFC3209] and is often desirable from an operator's perspective, because either a) the destination prefixes are not reachable via any means other than MPLS or b) this would result in significant packet loss as demand is shifted to other LSPs in the overlay mesh.

In addition, there are currently few implementations offering ingress admission control at the LSP level. Again, having ingress admission control on a per LSP basis is not necessarily desirable from an operational perspective, as a) one must over-provision tunnels significantly in order to avoid deleterious effects resulting from stacked transport and flow control systems and b) there is currently no efficient commonly available northbound interface for dynamic configuration of per LSP ingress admission control (such an interface could easily be defined using the extensions present in this spec, but it beyond the scope of the current document).

Lack of ingress admission control coupled with the behavior in [RFC3209] effectively results in mis-signaled LSPs during periods of contention for network capacity between LSPs in a given LSP priority. This in turn causes information loss in the TED with regard to actual network state, resulting in LSPs sharing common network interfaces

with mis-signalized LSPs operating in a degraded state for significant periods of time, even when unused network capacity may potentially be available.

Reference Topology 1 in Figure 1 and Tables 5 and 6 show a use case that demonstrates this behavior. Two LSPs, LSP 1 and LSP 2 are signaled with demand 2 and routed along paths A-C-D-E and B-C-D-E respectively. At a later time, the demand of LSP 1 increases to 20. Under such a demand, the LSP cannot be resignalized. However, the existing LSP will not be torn down. In the absence of ingress policing, traffic on LSP 1 will cause degradation for traffic of LSP 2 (due to oversubscription on the links C-D and D-E), as well as information loss in the TED with regard to the actual network state.

The problem could be easily ameliorated by global visibility of LSP state coupled with PCC- external demand measurements and placement of two LSPs on disjoint links. Note that while the demand of 20 for LSP 1 could never be satisfied in the given topology, what could be achieved would be isolation from the ill-effects of the (unsatisfiable) increased demand.

Link	Metric	Capacity
A-C	1	10
B-C	1	10
C-E	10	5
C-D	1	10
D-E	1	10

Table 5: Link parameters for the 'Deadlock' example

Time	LSP	Src	Dst	Demand	Routable	Path
1	1	A	E	2	Yes	A-C-D-E
2	2	B	E	2	Yes	B-C-D-E
3	1	A	E	20	No	---

Table 6: Deadlock LSP and demand time series

### 5.1.3. Minimum Perturbation

As a result of both the lack of visibility into global LSP state and the lack of control over event ordering across PCE sessions, unnecessary perturbations may be introduced into the network by a

stateless PCE. Tables 7 and 8 show an example of an unnecessary network perturbation using Reference Topology 1 in Figure 1. In this case an unimportant (high LSP priority value) LSP (LSP1) is first set up along the shortest path. At time 2, which is assumed to be relatively close to time 1, a second more important (lower LSP-priority value) LSP is established, preempting LSP 1 and shifting it to the longer A-C-E path.

Link	Metric	Capacity
A-C	1	10
B-C	1	10
C-E	10	10
C-D	1	10
D-E	1	10

Table 7: Link parameters for the 'Minimum-Perturbation' example

Time	LSP	Src	Dst	Demand	LSP Prio	Routable	Path
1	1	A	E	7	7	Yes	A-C-D-E
2	2	B	E	7	0	Yes	B-C-D-E
3	1	A	E	7	7	Yes	A-C-E

Table 8: Minimum-Perturbation LSP and demand time series

#### 5.1.4. Predictability

Randomization of reservation events caused by lack of control over event ordering across PCE sessions results in poor predictability in LSP routing. An offline system applying a consistent optimization method will produce predictable results to within either the boundary of forecast error when reservations are over-provisioned by reasonable margins or to the variability of the signal and the forecast error when applying some hysteresis in order to minimize churn.

Reference Topology 1 and Tables 9, 10 and 11 show the impact of event ordering and predictability of LSP routing.

Link	Metric	Capacity
A-C	1	10
B-C	1	10
C-E	1	10
C-D	1	10
D-E	1	10

Table 9: Link parameters for the 'Predictability' example

Time	LSP	Src	Dst	Demand	Routable	Path
1	1	A	E	7	Yes	A-C-E
2	2	B	E	7	Yes	B-C-D-E

Table 10: Predictability LSP and demand time series 1

Time	LSP	Src	Dst	Demand	Routable	Path
1	2	B	E	7	Yes	B-C-E
2	1	A	E	7	Yes	A-C-D-E

Table 11: Predictability LSP and demand time series 2

## 5.2. Stateful PCE in SDN

SDN promises to incorporate more intelligence into the network by using smart centralized controllers. The use cases below show the integration between a stateful PCE and such a controller. Note that although from an implementation point of view, the SDN controller and stateful PCE could be combined, in the discussion below they are separate to show how stateful PCE enables the control-loop feedback central to SDN.

### 5.2.1. Smart Bandwidth Adjustment

The bandwidth requirement of LSPs often change over time, requiring resizing the LSP. Currently router software performs this function by monitoring the actual bandwidth usage, triggering a recomputation and ressignaling when a threshold is reached. A central controller can use additional information (such as historical trending data, information from specific applications or policy information) in

order to make the determination of when and along which path an LSP should be resized. The controller can rely on a stateful PCE to perform the central function.

#### 5.2.2. Bandwidth Scheduling

Bandwidth Scheduling allows network operators to reserve resources in advance upon request from the customers to transmit large bulk of data with specified starting time and duration, such as in support of scheduled data transmission between data centers.

Traditionally, this can be supported by NMS operation through path pre-establishment and activation on the agreed starting time. However, this does not provide efficient network usage since the established paths exclude the possibility of being used by other services even when they are not used for undertaking any service. It can also be accomplished through GMPLS protocol extensions by carrying the related request information (e.g., starting time and duration) across the network. Nevertheless, this method inevitably increases the complexity of signaling and routing process.

A stateful PCE can support this application with better efficiency since it can alleviate the burden of processing on network elements as well as enable the flexibility of resources usage by only excluding the time slot(s) reserved for bandwidth scheduling requests. The details of organizing bandwidth scheduling related information as well as its impact on LSP-DB is subject to network providers policy and administrative consideration and thus outside of the scope of this document.

### 5.3. Recovery

#### 5.3.1. Protection

For protection purposes, a PCC may send a request to a PCE for computing a set of paths for a given LSP. Alternatively, the PCC can send multiple requests to the PCE, asking for working and backup LSPs separately. Either way, the resources bound to backup paths can be shared by different LSPs to improve the overall network efficiency, such as m:n protection or pre-configured shared mesh recovery techniques as specified in [RFC4427]. If resource sharing is supported for LSP protection, the information relating to existing LSPs is required to avoid allocation of shared protection resources to two LSPs that might fail together and cause protection contention issues. Stateful PCEs can easily accommodate this need using the information stored in its LSP-DB.



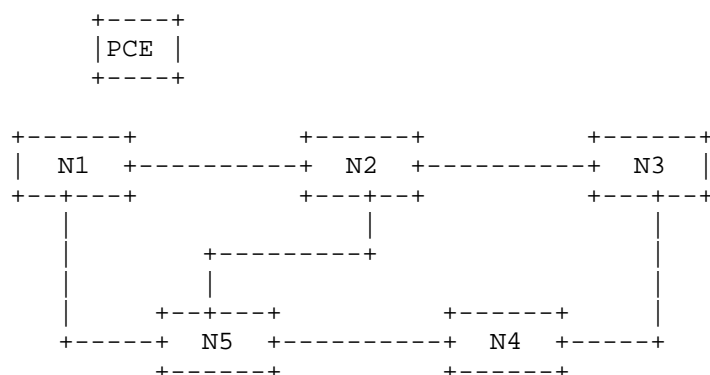


Figure 3: Reference topology 3

For example, in the network depicted in 3, suppose there exists LSP1 (N1->N5) with backup route following N1->N2->N5. A request arrives asking for a working and backup path pair to be computed for a request from N2 to N5. If the PCE decides N2->N1->N5 to be the best working route, then the backup path should not use the same protection resource with LSP1 since the new LSP shares part of its resource with LSP1 (i.e., these two LSPs are in the same shared risk group). Alternatively, there is no such constraint if N2->N3->N4->N5 is chosen to be the right candidate for undertaking the request.

### 5.3.2. Restoration

In case of a link failure, such as fiber cut, multiple LSPs may fail at the same time. Thus, the source nodes of the affected LSPs will be informed of the failure by the nodes detecting the failure. These source nodes will send requests to a PCE for rerouting. In order to reuse the resource taken by an existing LSP, the source node can send a PCReq message including the XRO object with F bit set, together with RRO object, as specified in [RFC5521].

If a stateless PCE is exploited, it might respond to the rerouting requests separately if they arrive at different times. Thus, it might result in sub-optimal resource usage. Even worse, it might unnecessarily block some of the rerouting requests due to insufficient resources for later-arrived rerouting messages. If a stateful PCE is used to fulfill this task, it can re-compute the affected LSPs concurrently while reusing part of the existing LSPs resources when it is informed of the failed link identifier provided by the first request. This is made possible since the stateful PCE can check what other LSPs are affected by the failed link and their route information by inspecting its LSP-DB. As a result, a better

performance, such as better resource usage, minimal probability of blocking upcoming new rerouting requests sent as a result of the link failure, can be achieved.

In order to further reduce the amount of LSP rerouting messages flow in the network, the notification can be performed at the node(s) which detect the link failure. For example, suppose there are two LSPs in the network as shown in Figure 3: (i) LSP1: N1->N5->N4->N3; (ii) LSP2: N2->N5->N4. They traverse the failed link between N5-N4. When N4 detects the failure, it can send a notification message to a stateful PCE. Note that the stateful PCE stores the path information of the LSPs that are affected by the link failure, so it does not need to acquire this information from N4. Moreover, it can make use of the bandwidth resources occupied by the affected LSPs when performing path recalculation. After N4 receives the new paths from the PCE, it notifies the ingress nodes of the LSPs, i.e., N1 and N2, and specifies the new paths which should be used as the rerouting paths. To support this, it would require extensions to existing signaling protocol.

Alternatively, if the target is to avoid resource contention within the time-window of high LSP requests, a stateful PCE can retain the under-construction LSP resource usage information for a given time and exclude it from being used for forthcoming LSPs request. In this way, it can ensure that the resource will not be double-booked and thus the issue of resource contention and computation crank-backs can be resolved.

### 5.3.3. SRLG Diversity

An alternative way to achieve efficient recovery is to maintain SRLG disjointness between LSPs. This can be achieved at provisioning time, if the routes of all the LSPs are requested together, using a synchronized computation of the different LSPs with SRLG disjointness constraint. If the LSPs need to be provisioned at different times (more general, the routes are requested at different times, e.g. in the case of a restoration), the PCC can specify, as constraints to the path computation a set of Shared Risk Link Groups (SRLGs) using the Explicit route Object [RFC5521]. However, for the latter to be effective, it is needed that the entity that requests the route to the PCE maintains updated SRLG information of all the LSPs to which it must maintain the disjointness.

Using a stateful PCE allows the maintenance of the updated SRLG information of the established LSPs in a centralized manner. Having such information in the PCE facilitates the PCC to specify, as constraint to the path computation, the SRLG disjointness of a set of already established LSPs by only providing the LSP identifiers.

#### 5.4. Maintenance of Virtual Network Topology (VNT)

In Multi-Layer Networks (MLN), a Virtual Network Topology (VNT) [RFC5212] consists of a set of one or more TE LSPs in the lower layer which provides TE links to the upper layer. In [RFC5623], the PCE-based architecture is proposed to support path computation in MLN networks in order to achieve inter-layer TE.

The establishment/teardown of a TE link in VNT needs to take into consideration the state of existing LSPs and/or new LSP request(s) in the higher layer. As specified in [RFC5623], a VNT manager (VNTM) is in charge of the topology in the upper layer by connections in the lower layer. Hence, when a stateless PCE is requested to compute a new TE link, it will need interaction with VNTM for detailed TE link information. To be more specific, without detailed LSP information, this process would be inefficient or even infeasible for stateless PCE(s), unless with cooperation with VNTM. On the other hand, a stateful PCE seems more suitable to make the decision of when and how to modify the VNT either to accommodate new LSP requests or to re-optimize resource use across layers irrespective of PCE models. As described in Section 2.2, path computation for a VNT change can be performed by the PCE if a single PCE model is adopted. On the other hand, if a per-layer PCE model is more appropriate, coordination between PCEs is required.

#### 5.5. LSP Re-optimization

In order to make efficient usage of network resource, re-optimization of one or more LSPs dynamically through online planning is desirable. In case of a stateless PCE, in order to optimize network resource usage dynamically through online planning, PCC (e.g., NMS) should send a request to PCE together with detailed path/bandwidth information of the LSPs that need to be concurrently optimized. This would require a PCC (e.g., NMS) to determine when and which LSPs should be optimized. Given all of the existing LSP state information kept at a stateful PCE, it allows automation of this process without the PCC (e.g., NMS) to supply give the re-optimization commands and the existing LSP state information. Moreover, since a stateful PCE can maintain the information regarding to all LSPs that are currently under signaling, it makes the optimization procedures be performed more intelligently and effectively.

A special case of LSP re-optimization is Global Concurrent Optimization (GCO) [RFC5557]. Global control of LSP operation sequence in [RFC5557] is predicated on the use of what is effectively a stateful (or semi-stateful) NMS. The NMS can be either not local to the switch, in which case another northbound interface is required for LSP attribute changes, or local/collocated, in which case there

are significant issues with efficiency in resource usage. Stateful PCE adds a few features that:

- o Roll the NMS visibility into the PCE and remove the requirement for an additional northbound interface
- o Allow the PCE to determine when re-optimization is needed, with which level (GCO or a more incremental optimization)
- o Allow the PCE to determine which LSPs should be re-optimized
- o Allow a PCE to control the sequence of events across multiple PCCs, allowing for bulk (and truly global) optimization, LSP shuffling etc.

#### 5.6. Resource defragmentation

In networks with link bundles, if LSPs are dynamically allocated and released over time, the resource becomes fragmented. The overall available resource on a (bundle) link might be sufficient for a new LSP request. But if the available resource is not continuous, the request would be rejected. In order to perform the defragmentation procedure, stateful PCEs can be used, since existing TE LSPs information is required to accurately assess spectrum resources on the LSPs, and perform de-fragmentation while ensuring a minimal disruption of the network, e.g., based on active LSP priorities .

A case of particular interest to GMPLS-based transport networks is the frequency defragmentation in flexible grid. In Flexible grid networks [I-D.ogrcetal-ccamp-flexi-grid-fwk], LSPs with different slot widths (such as 12.5G, 25G etc.) can co-exist so as to accommodate the services with different bandwidth requests. Therefore, even if the overall spectrum can meet the service request, it may not be usable if they are not contiguous. Thus, with the help of existing LSP state information, stateful PCE can make the resource grouped together to be usable. Moreover, stateful PCE can proactively choose routes for upcoming path requests to reduce the chance of spectrum defragmentation.

#### 5.7. Future applications

##### 5.7.1. Impairment-Aware Routing and Wavelength Assignment (IA-RWA)

In WSON networks [RFC6163], a wavelength-switched LSP traverses one or more fiber links. The bit rates of the client signals carried by the wavelength LSPs may be the same or different. Hence, a fiber link may transmit a number of wavelength LSPs with equal or mixed bit rate signals. For example, a fiber link may multiplex the

wavelengths with only 10G signals, mixed 10G and 40G signals, or mixed 40G and 100G signals.

IA-RWA in WSONs refers to the RWA process (i.e., lightpath computation) that takes into account the optical layer/transmission imperfections by considering as additional (i.e., physical layer) constraints. To be more specific, linear and non-linear effects associated with the optical network elements should be incorporated into the route and wavelength assignment procedure. For example, the physical imperfection can result in the interference of two adjacent lightpaths. Thus, a guard band should be reserved between them to alleviate these effects. The width of the guard band between two adjacent wavelengths depends on their characteristics, such as modulation formats and bit rates. Two adjacent wavelengths with different characteristics (e.g., different bit rates) may need a wider guard band and with same characteristics may need a narrower guard band. For example, 50GHz spacing may be acceptable for two adjacent wavelengths with 40G signals. But for two adjacent wavelengths with different bit rates (e.g., 10G and 40G), a larger spacing such as 300GHz spacing may be needed. Hence, the characteristics (states) of the existing wavelength LSPs should be considered for a new RWA request in WSON.

In summary, when stateful PCEs are used to perform the IA-RWA procedure, they need to know the characteristics of the existing wavelength LSPs. The impairment information relating to existing and to-be-established LSPs can be obtained by nodes in WSON networks via external configuration or other means such as monitoring or estimation based on a vendor-specific impair model. However, WSON related routing protocols, i.e., [I-D.ietf-ccamp-wson-signal-compatibility-ospf] and [I-D.ietf-ccamp-gmpls-general-constraints-ospf-te], only advertise limited information (i.e., availability) of the existing wavelengths, without defining the supported client bit rates. It will incur substantial amount of control plane overhead if routing protocols are extended to support dissemination of the new information relevant for the IA-RWA process. In this scenario, stateful PCE(s) would be a more appropriate mechanism to solve this problem. Stateful PCE(s) can exploit impairment information of LSPs stored in LSP-DB to provide accurate RWA calculation.

## 6. Security Considerations

This document does not introduce any new security considerations.

## 7. Contributing authors

The following people all contributed significantly to this document and are listed below in alphabetical order:

Ramon Casellas  
CTTC - Centre Tecnologic de Telecomunicacions de Catalunya  
Av. Carl Friedrich Gauss n7  
Castelldefels, Barcelona 08860  
Spain  
Email: ramon.casellas@cttc.es

Edward Crabbe  
Google, Inc.  
1600 Amphitheatre Parkway  
Mountain View, CA 94043  
US  
Email: edc@google.com

Dhruv Dhody  
Huawei Technology  
Leela Palace  
Bangalore, Karnataka 560008  
INDIA  
Email: dhruvd@huawei.com

Oscar Gonzalez de Dios  
Telefonica Investigacion y Desarrollo  
Emilio Vargas 6  
Madrid, 28045  
Spain  
Phone: +34 913374013  
Email: ogondio@tid.es

Young Lee  
Huawei  
1700 Alma Drive, Suite 100  
Plano, TX 75075  
US  
Phone: +1 972 509 5599 x2240  
Fax: +1 469 229 5397  
Email: ylee@huawei.com

Jan Medved  
Cisco Systems, Inc.  
170 West Tasman Dr.  
San Jose, CA 95134  
US

Email: jmedved@cisco.com

Robert Varga  
Pantheon Technologies LLC  
Mlynske Nivy 56  
Bratislava 821 05  
Slovakia  
Email: robert.varga@pantheon.sk

Fatai Zhang  
Huawei Technologies  
F3-5-B R&D Center, Huawei Base  
Bantian, Longgang District  
Shenzhen 518129 P.R.China  
Phone: +86-755-28972912  
Email: zhangfatai@huawei.com

Xiaobing Zi  
Huawei Technologies  
F3-5-B R&D Center, Huawei Base  
Bantian, Longgang District  
Shenzhen 518129 P.R.China  
Phone: +86-755-28973229  
Email: zixiaobing@huawei.com

## 8. Acknowledgements

We would like to thank Cyril Margaria for the useful comments and discussions.

## 9. References

### 9.1. Normative References

- [RFC2205] Braden, B., Zhang, L., Berson, S., Herzog, S., and S. Jamin, "Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification", RFC 2205, September 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.
- [RFC4090] Pan, P., Swallow, G., and A. Atlas, "Fast Reroute Extensions to RSVP-TE for LSP Tunnels", RFC 4090, May 2005.

- [RFC4427] Mannie, E. and D. Papadimitriou, "Recovery (Protection and Restoration) Terminology for Generalized Multi-Protocol Label Switching (GMPLS)", RFC 4427, March 2006.
- [RFC5088] Le Roux, JL., Vasseur, JP., Ikejiri, Y., and R. Zhang, "OSPF Protocol Extensions for Path Computation Element (PCE) Discovery", RFC 5088, January 2008.
- [RFC5089] Le Roux, JL., Vasseur, JP., Ikejiri, Y., and R. Zhang, "IS-IS Protocol Extensions for Path Computation Element (PCE) Discovery", RFC 5089, January 2008.
- [RFC5212] Shiomoto, K., Papadimitriou, D., Le Roux, JL., Vigoureux, M., and D. Brungard, "Requirements for GMPLS-Based Multi-Region and Multi-Layer Networks (MRN/MLN)", RFC 5212, July 2008.
- [RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, May 2008.
- [RFC5440] Vasseur, JP. and JL. Le Roux, "Path Computation Element (PCE) Communication Protocol (PCEP)", RFC 5440, March 2009.
- [RFC5511] Farrel, A., "Routing Backus-Naur Form (RBNF): A Syntax Used to Form Encoding Rules in Various Routing Protocol Specifications", RFC 5511, April 2009.
- [RFC5521] Oki, E., Takeda, T., and A. Farrel, "Extensions to the Path Computation Element Communication Protocol (PCEP) for Route Exclusions", RFC 5521, April 2009.
- [RFC5623] Oki, E., Takeda, T., Le Roux, JL., and A. Farrel, "Framework for PCE-Based Inter-Layer MPLS and GMPLS Traffic Engineering", RFC 5623, September 2009.
- [RFC6163] Lee, Y., Bernstein, G., and W. Imajuku, "Framework for GMPLS and Path Computation Element (PCE) Control of Wavelength Switched Optical Networks (WSOs)", RFC 6163, April 2011.

## 9.2. Informative References

- [I-D.crabbe-pce-stateful-pce-mpls-te]  
Crabbe, E., Medved, J., Minei, I., and R. Varga, "Stateful PCE extensions for MPLS-TE LSPs",  
draft-crabbe-pce-stateful-pce-mpls-te-00 (work in



progress), October 2012.

- [I-D.ietf-ccamp-gmpls-general-constraints-ospf-te]  
Zhang, F., Lee, Y., Han, J., Bernstein, G., and Y. Xu,  
"OSPF-TE Extensions for General Network Element  
Constraints",  
draft-ietf-ccamp-gmpls-general-constraints-ospf-te-04  
(work in progress), July 2012.
- [I-D.ietf-ccamp-wson-signal-compatibility-ospf]  
Lee, Y. and G. Bernstein, "GMPLS OSPF Enhancement for  
Signal and Network Element Compatibility for Wavelength  
Switched Optical Networks",  
draft-ietf-ccamp-wson-signal-compatibility-ospf-11 (work  
in progress), February 2013.
- [I-D.ietf-pce-gmpls-pcep-extensions]  
Margaria, C., Dios, O., and F. Zhang, "PCEP extensions for  
GMPLS", draft-ietf-pce-gmpls-pcep-extensions-07 (work in  
progress), October 2012.
- [I-D.ietf-pce-stateful-pce]  
Crabbe, E., Medved, J., Minei, I., and R. Varga, "PCEP  
Extensions for Stateful PCE",  
draft-ietf-pce-stateful-pce-02 (work in progress),  
October 2012.
- [I-D.ogrcetal-ccamp-flexi-grid-fwk]  
Dios, O., Casellas, R., Zhang, F., Fu, X., Ceccarelli, D.,  
and I. Hussain, "Framework for GMPLS based control of  
Flexi-grid DWDM networks",  
draft-ogrcetal-ccamp-flexi-grid-fwk-01 (work in progress),  
October 2012.
- [I-D.sivabalan-pce-disco-stateful]  
Sivabalan, S. and J. Medved, "IGP Extensions for Stateful  
PCE Discovery", draft-sivabalan-pce-disco-stateful-00  
(work in progress), January 2013.
- [MPLS-PC] Chaieb, I., Le Roux, J.L., and B. Cousin, "Improved MPLS-TE  
LSP Path Computation using Preemption", Global  
Information Infrastructure Symposium, July 2007.
- [MXMN-TE] Danna, E., Mandal, S., and A. Singh, "Practical linear  
programming algorithm for balancing the max-min fairness  
and throughput objectives in traffic engineering", pre-  
print, 2011.

- [NET-REC] Vasseur, JP., Pickavet, M., and P. Demeester, "Network Recovery: Protection and Restoration of Optical, SONET-SDH, IP, and MPLS", The Morgan Kaufmann Series in Networking, June 2004.
- [RFC2702] Awduche, D., Malcolm, J., Agogbua, J., O'Dell, M., and J. McManus, "Requirements for Traffic Engineering Over MPLS", RFC 2702, September 1999.
- [RFC3031] Rosen, E., Viswanathan, A., and R. Callon, "Multiprotocol Label Switching Architecture", RFC 3031, January 2001.
- [RFC3346] Boyle, J., Gill, V., Hannan, A., Cooper, D., Awduche, D., Christian, B., and W. Lai, "Applicability Statement for Traffic Engineering with MPLS", RFC 3346, August 2002.
- [RFC3630] Katz, D., Kompella, K., and D. Yeung, "Traffic Engineering (TE) Extensions to OSPF Version 2", RFC 3630, September 2003.
- [RFC4655] Farrel, A., Vasseur, J., and J. Ash, "A Path Computation Element (PCE)-Based Architecture", RFC 4655, August 2006.
- [RFC4657] Ash, J. and J. Le Roux, "Path Computation Element (PCE) Communication Protocol Generic Requirements", RFC 4657, September 2006.
- [RFC5305] Li, T. and H. Smit, "IS-IS Extensions for Traffic Engineering", RFC 5305, October 2008.
- [RFC5394] Bryskin, I., Papadimitriou, D., Berger, L., and J. Ash, "Policy-Enabled Path Computation Framework", RFC 5394, December 2008.
- [RFC5557] Lee, Y., Le Roux, JL., King, D., and E. Oki, "Path Computation Element Communication Protocol (PCEP) Requirements and Protocol Extensions in Support of Global Concurrent Optimization", RFC 5557, July 2009.

#### Appendix A. Editorial notes and open issues

This section will be removed prior to publication.

The following open issues remain:

Use cases from draft-ietf-pce-stateful-pce To avoid loss of information, the use cases will be removed from [I-D.ietf-pce-stateful-pce] only after this document becomes a working group document.

This document WILL NOT repeat terminology defined in other documents or attempt to place any additional requirements on stateful PCE.

#### Authors' Addresses

Xian Zhang (editor)  
Huawei Technologies  
F3-5-B R&D Center, Huawei Base Bantian, Longgang District  
Shenzhen, Guangdong 518129  
P.R.China

Email: zhang.xian@huawei.com

Ina Minei (editor)  
Juniper Networks, Inc.  
1194 N. Mathilda Ave.  
Sunnyvale, CA 94089  
US

Email: ina@juniper.net



Network Working Group  
Internet-Draft  
Intended status: Informational  
Expires: November 25, 2013

X. Zhang, Ed.  
Huawei Technologies  
I. Minei, Ed.  
Juniper Networks, Inc.  
May 24, 2013

Applicability of Stateful Path Computation Element (PCE)  
draft-zhang-pce-stateful-pce-app-04

Abstract

A stateful Path Computation Element (PCE) maintains information about Label Switched Path (LSP) characteristics and resource usage within a network in order to provide traffic engineering calculations for its associated Path Computation Clients (PCCs). This document describes general considerations for a stateful PCE deployment and examines its applicability and benefits through a number of use cases. Path Computation Element Protocol (PCEP) extensions required for stateful PCE usage are covered in separate documents.

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 November 25, 2013.

Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect

to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	3
2. Terminology . . . . .	3
3. Overview of stateful PCE . . . . .	4
4. Deployment considerations . . . . .	5
4.1. Multi-PCE deployments . . . . .	5
4.2. LSP State Synchronization . . . . .	5
4.3. PCE Survivability . . . . .	5
5. Application scenarios . . . . .	6
5.1. Optimization of LSP placement . . . . .	6
5.1.1. Throughput Maximization and Bin Packing . . . . .	7
5.1.2. Deadlock . . . . .	8
5.1.3. Minimum Perturbation . . . . .	10
5.1.4. Predictability . . . . .	11
5.2. Auto-bandwidth Adjustment . . . . .	12
5.3. Bandwidth Scheduling . . . . .	13
5.4. Recovery . . . . .	13
5.4.1. Protection . . . . .	13
5.4.2. Restoration . . . . .	15
5.4.3. SRLG Diversity . . . . .	16
5.5. Maintenance of Virtual Network Topology (VNT) . . . . .	16
5.6. LSP Re-optimization . . . . .	17
5.7. Resource Defragmentation . . . . .	17
5.8. Impairment-Aware Routing and Wavelength Assignment (IA-RWA) . . . . .	18
6. Security Considerations . . . . .	19
7. Contributing Authors . . . . .	19
8. Acknowledgements . . . . .	21
9. References . . . . .	21
9.1. Normative References . . . . .	21
9.2. Informative References . . . . .	21
Appendix A. Editorial notes and open issues . . . . .	23
Authors' Addresses . . . . .	23

## 1. Introduction

[RFC4655] defines the architecture for a Path Computation Element (PCE)-based model for the computation of Multiprotocol Label Switching (MPLS) and Generalized MPLS (GMPLS) Traffic Engineering Label Switched Paths (TE LSPs). To perform such a constrained computation, a PCE stores the network topology (i.e., TE links and nodes) and resource information (i.e., TE attributes) in its TE Database (TED). [RFC5440] describes the Path Computation Element Protocol (PCEP). PCEP defines the communication between a Path Computation Client (PCC) and a Path Control Element (PCE), or between two PCEs, enabling computation of Multiprotocol Label Switching (MPLS) for Traffic Engineering Label Switched Path (TE LSP) characteristics. Extensions for support of GMPLS in PCEP are defined in [I-D.ietf-pce-gmpls-pcep-extensions].

As per [RFC4655], a PCE can be either stateful or stateless. Stateless PCEs have been shown to be useful in many scenarios, including constraint-based path computation in multi-domain/multi-layer networks. Compared to a stateless PCE, a stateful PCE has access to not only the network state, but also to the set of active paths and their reserved resources. Furthermore, a stateful PCE might also retain information regarding LSPs under construction in order to reduce churn and resource contention. This state allows the PCE to compute constrained paths while considering individual LSPs and their interactions. Note that this requires reliable state synchronization mechanisms between the PCE and the network, PCE and PCC, and between cooperating PCEs, with potentially significant control plane overhead and maintenance of a large amount of state data, as explained in [RFC4655].

This document describes how a stateful PCE can be used to solve various problems for MPLS-TE and GMPLS networks, and the benefits it brings to such deployments. Note that alternative solutions relying on stateless PCEs may also be possible for some of these use cases, and will be mentioned for completeness where appropriate.

## 2. Terminology

This document uses the following terms defined in [RFC5440]: PCC, PCE, PCEP Peer.

This document uses the following terms defined in [I-D.ietf-pce-stateful-pce]: Passive Stateful PCE, Active Stateful PCE, Delegation, Revocation, Delegation Timeout Interval, LSP State Report, LSP Update Request, LSP State Database.

This document defines the following term:

**Minimum Cut Set:** the minimum set of links for a specific source destination pair which, when removed from the network, result in a specific source being completely isolated from specific destination. The summed capacity of these links is equivalent to the maximum capacity from the source to the destination by the max-flow min-cut theorem.

### 3. Overview of stateful PCE

This section is included for the convenience of the reader, please refer to the referenced documents for details of the operation.

[I-D.ietf-pce-stateful-pce] specifies a set of extensions to PCEP to enable stateful control of tunnels within and across PCEP sessions in compliance with [RFC4657]. It includes mechanisms to effect tunnel state synchronization between PCCs and PCEs, delegation of control over tunnels to PCEs, and PCE control of timing and sequence of path computations within and across PCEP sessions.

[I-D.ietf-pce-stateful-pce] applies equally to MPLS-TE and GMPLS LSPs.

Several new functions were added in PCEP to support stateful PCEs and are described in [I-D.ietf-pce-stateful-pce]. A function can be initiated either from a PCC towards a PCE (C-E) or from a PCE towards a PCC (E-C). The new functions are:

**Capability negotiation (E-C,C-E):** both the PCC and the PCE must announce during PCEP session establishment that they support PCEP Stateful PCE extensions.

**LSP state synchronization (C-E):** after the session between the PCC and a stateful PCE is initialized, the PCE must learn the state of a PCC's LSPs before it can perform path computations or update LSP attributes in a PCC.

**LSP Update Request (E-C):** A PCE requests modification of attributes on a PCC's LSP.

**LSP State Report (C-E):** a PCC sends an LSP State Report to a PCE whenever the state of an LSP changes.



LSP control delegation (C-E,E-C): a PCC grants to a PCE the right to update LSP attributes on one or more LSPs; the PCE becomes the authoritative source of the LSP's attributes as long as the delegation is in effect; the PCC may withdraw the delegation or the PCE may give up the delegation.

[I-D.sivabalan-pce-disco-stateful] defines the extensions needed to support autodiscovery of stateful PCEs when using the IGPs for PCE discovery.

#### 4. Deployment considerations

This section discusses generic issues with Stateful PCE deployments, and how specific protocol mechanisms can be used to address them.

##### 4.1. Multi-PCE deployments

Stateless and stateful PCEs can co-exist in the same network and be in charge of path computation of different types. To solve the problem of distinguishing between the two types of PCEs, either discovery or configuration may be used. The capability negotiation in [I-D.ietf-pce-stateful-pce] ensures correct operation when the PCE address is configured on the PCC.

##### 4.2. LSP State Synchronization

A stateful PCE maintains two sets of information for use in path computation. The first is the Traffic Engineering Database (TED) which includes the topology and resource state in the network. This information can be obtained by a stateful PCE using the same mechanisms as a stateless PCE (see [RFC4655]). The second is the LSP State Database (LSP-DB), in which a PCE stores attributes of all active LSPs in the network, such as their paths through the network, bandwidth/resource usage, switching types and LSP constraints. The stateful PCE extensions defined in [I-D.ietf-pce-stateful-pce] support population of this database using information received from the network nodes via LSP State Report messages. Population of the LSP database via other means is not precluded.

##### 4.3. PCE Survivability

For a stateful PCE, an important issue is to get the LSP state information resynchronized after a restart. [I-D.ietf-pce-stateful-pce] includes support of a synchronization function, allowing the PCC to synchronize its LSP state with the PCE. This can be applied equally to an Label Edge Router (LER) client or another PCE, allowing for support of multiple ways of re-acquiring

the LSP database on a restart. For example, the state can be retrieved from the network nodes, or from another stateful PCE. Because synchronization may also be skipped, if a PCE implementation has the means to retrieve its database in a different way (for example from a backup copy stored locally), the state can be restored without further overhead in the network. Note that locally recovering the state would still require some degree of resynchronization to ensure that the recovered state is indeed up-to-date.

## 5. Application scenarios

In the following sections, several use cases are described, showcasing scenarios that benefit from the deployment of a stateful PCE.

### 5.1. Optimization of LSP placement

The following use cases demonstrate a need for visibility into global inter-PCC LSP state in PCE path computations, and for a PCE control of sequence and timing in altering LSP path characteristics within and across PCEP sessions. Reference topologies for the use cases described later in this section are shown in Figures 1 and 2.

Some of the use cases below are focused on MPLS-TE deployments, but may also apply to GMPLS. Unless otherwise cited, use cases assume that all LSPs listed exist at the same LSP priority.

The main benefit in the cases below comes from moving away from an asynchronous PCC-driven mode of operation to a model that allows for central control over LSP computations and setup, and focuses specifically on the active stateful PCE model of operation.

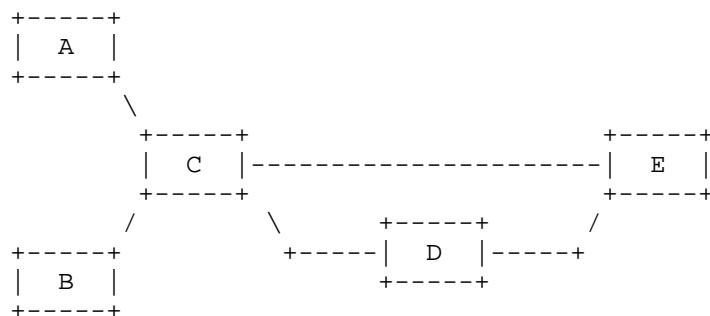


Figure 1: Reference topology 1

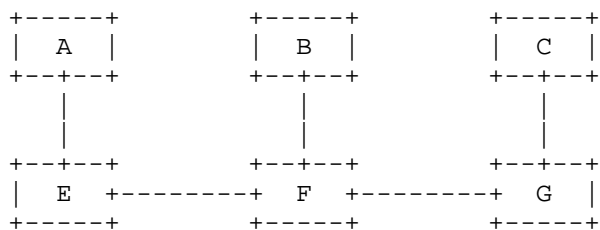


Figure 2: Reference topology 2

#### 5.1.1. Throughput Maximization and Bin Packing

Because LSP attribute changes in [RFC5440] are driven by PCReq messages under control of a PCC's local timers, the sequence of RSVP reservation arrivals occurring in the network will be randomized. This, coupled with a lack of global LSP state visibility on the part of a stateless PCE may result in suboptimal throughput in a given network topology, as will be shown in the example below.

Reference topology 2 in Figure 2 and Tables 1 and 2 show an example in which throughput is at 50% of optimal as a result of lack of visibility and synchronized control across PCC's. In this scenario, the decision must be made as to whether to route any portion of the E-G demand, as any demand routed for this source and destination will decrease system throughput.

Link	Metric	Capacity
A-E	1	10
B-F	1	10
C-G	1	10
E-F	1	10
F-G	1	10

Table 1: Link parameters for Throughput use case

Time	LSP	Src	Dst	Demand	Routable	Path
1	1	E	G	10	Yes	E-F-G
2	2	A	B	10	No	---
3	1	F	C	10	No	---

Table 2: Throughput use case demand time series

In many cases throughput maximization becomes a bin packing problem. While bin packing itself is an NP-hard problem, a number of common heuristics which run in polynomial time can provide significant improvements in throughput over random reservation event distribution, especially when traversing links which are members of the minimum cut set for a large subset of source destination pairs.

Tables 3 and 4 show a simple use case using Reference Topology 1 in Figure 1, where LSP state visibility and control of reservation order across PCCs would result in significant improvement in total throughput.

Link	Metric	Capacity
A-C	1	10
B-C	1	10
C-E	10	5
C-D	1	10
D-E	1	10

Table 3: Link parameters for Bin Packing use case

Time	LSP	Src	Dst	Demand	Routable	Path
1	1	A	E	5	Yes	A-C-D-E
2	2	B	E	10	No	---

Table 4: Bin Packing use case demand time series

#### 5.1.2. Deadlock

This section discusses a use case of cross-LSP impact under degraded operation. Most existing RSVP-TE implementations will not tear down established LSPs in the event of the failure of the bandwidth

increase procedure detailed in [RFC3209]. This behavior is directly implied to be correct in [RFC3209] and is often desirable from an operator's perspective, because either a) the destination prefixes are not reachable via any means other than MPLS or b) this would result in significant packet loss as demand is shifted to other LSPs in the overlay mesh.

In addition, there are currently few implementations offering dynamic ingress admission control (policing of the traffic volume mapped onto an LSP) at the LER. Having ingress admission control on a per LSP basis is not necessarily desirable from an operational perspective, as a) one must over-provision tunnels significantly in order to avoid deleterious effects resulting from stacked transport and flow control systems and b) there is currently no efficient commonly available northbound interface for dynamic configuration of per LSP ingress admission control (such an interface could easily be defined using the extensions for stateful PCE, but has not been yet at the time of this writing).

Lack of ingress admission control coupled with the behavior in [RFC3209] may result in LSPs operating out of profile for significant periods of time. It is reasonable to expect that these out-of-profile LSPs will be operating in a degraded state and experience traffic loss, but because they end up sharing common network interfaces with other LSPs operating within their bandwidth reservations, they will end up impacting the operation of the in-profile LSPs, even when there is unused network capacity elsewhere in the network. Furthermore, this behavior will cause information loss in the TED with regards to the actual available bandwidth on the links used by the out-of-profile LSPs, as the reservations on the links no longer reflect the capacity used.

Reference Topology 1 in Figure 1 and Tables 5 and 6 show a use case that demonstrates this behavior. Two LSPs, LSP 1 and LSP 2 are signaled with demand 2 and routed along paths A-C-D-E and B-C-D-E respectively. At a later time, the demand of LSP 1 increases to 20. Under such a demand, the LSP cannot be resigaled. However, the existing LSP will not be torn down. In the absence of ingress policing, traffic on LSP 1 will cause degradation for traffic of LSP 2 (due to oversubscription on the links C-D and D-E), as well as information loss in the TED with regard to the actual network state.

The problem could be easily ameliorated by global visibility of LSP state coupled with PCC-external demand measurements and placement of two LSPs on disjoint links. Note that while the demand of 20 for LSP 1 could never be satisfied in the given topology, what could be achieved would be isolation from the ill-effects of the (unsatisfiable) increased demand.

Link	Metric	Capacity
A-C	1	10
B-C	1	10
C-E	10	5
C-D	1	10
D-E	1	10

Table 5: Link parameters for the 'Degraded operation' example

Time	LSP	Src	Dst	Demand	Routable	Path
1	1	A	E	2	Yes	A-C-D-E
2	2	B	E	2	Yes	B-C-D-E
3	1	A	E	20	No	---

Table 6: Degraded operation demand time series

#### 5.1.3. Minimum Perturbation

As a result of both the lack of visibility into global LSP state and the lack of control over event ordering across PCE sessions, unnecessary perturbations may be introduced into the network by a stateless PCE. Tables 7 and 8 show an example of an unnecessary network perturbation using Reference Topology 1 in Figure 1. In this case an unimportant (high LSP priority value) LSP (LSP1) is first set up along the shortest path. At time 2, which is assumed to be relatively close to time 1, a second more important (lower LSP-priority value) LSP (LSP2) is established, preempting LSP1, potentially causing traffic loss. LSP1 is then reestablished on the longer A-C-E path.

Link	Metric	Capacity
A-C	1	10
B-C	1	10
C-E	10	10
C-D	1	10
D-E	1	10

Table 7: Link parameters for the 'Minimum-Perturbation' example

Time	LSP	Src	Dst	Demand	LSP Prio	Routable	Path
1	1	A	E	7	7	Yes	A-C-D-E
2	2	B	E	7	0	Yes	B-C-D-E
3	1	A	E	7	7	Yes	A-C-E

Table 8: Minimum-Perturbation LSP and demand time series

A stateful PCE can help in this scenario by evaluating both requests at the same time (due to their proximity in time). This will ensure placement of the more important LSP along the shortest path, avoiding the preemption of the lower priority LSP.

#### 5.1.4. Predictability

Randomization of reservation events caused by lack of control over event ordering across PCE sessions results in poor predictability in LSP routing. An offline system applying a consistent optimization method will produce predictable results to within either the boundary of forecast error when reservations are over-provisioned by reasonable margins or to the variability of the signal and the forecast error when applying some hysteresis in order to minimize churn. Predictable results are valuable for being able to simulate the network and reliably test it under various scenarios, especially under various failure modes and planned maintenances when predictable path characteristics are desired under contention for network resources.

Reference Topology 1 and Tables 9, 10 and 11 show the impact of event ordering and predictability of LSP routing.

Link	Metric	Capacity
A-C	1	10
B-C	1	10
C-E	1	10
C-D	1	10
D-E	1	10

Table 9: Link parameters for the 'Predictability' example

Time	LSP	Src	Dst	Demand	Routable	Path
1	1	A	E	7	Yes	A-C-E
2	2	B	E	7	Yes	B-C-D-E

Table 10: Predictability LSP and demand time series 1

Time	LSP	Src	Dst	Demand	Routable	Path
1	2	B	E	7	Yes	B-C-E
2	1	A	E	7	Yes	A-C-D-E

Table 11: Predictability LSP and demand time series 2

As can be shown in the example, both LSPs were routed in both cases, but along very different paths. This would be a challenge if reliable simulation of the network was attempted. A stateful PCE can solve this through control over LSP ordering.

## 5.2. Auto-bandwidth Adjustment

The bandwidth requirement of LSPs often change over time, requiring resizing the LSP. Currently the head-end node performs this function by monitoring the actual bandwidth usage, triggering a recomputation and resignaling when a threshold is reached. This operation is referred as auto-bandwidth adjustment. The head-end node either recomputes the path locally, or it requests a recomputation from a PCE by sending a PCReq message. In the latter case, the PCE computes a new path and provides the new route suggestion. Upon receiving the reply from the PCE, the PCC re-signals the LSP in Shared-Explicit (SE) mode along the newly computed path. If a passive stateful PCE is used, only the new bandwidth information is needed to trigger a path re-computation since the LSP information is already known to the PCE. Note that in this scenario, the head-end node is the one that drives the LSP resizing based on local information, and that the difference between using a stateless and a passive stateful PCE is in the level of optimization of the LSP placement as discussed in the previous section.

A more interesting smart bandwidth adjustment case is one where the LSP resizing decision is done by an external entity, with access to additional information such as historical trending data, application-specific information about expected demands or policy information, as well as knowledge of the actual desired flow volumes. In this case



an active stateful PCE provides an advantage in both the computation with knowledge of all LSPs in the domain and in the ability to trigger bandwidth modification of the LSP.

### 5.3. Bandwidth Scheduling

Bandwidth scheduling allows network operators to reserve resources in advance according to the agreements with their customers, and allow them to transmit data with specified starting time and duration, for example for a scheduled bulk data replication between data centers.

Traditionally, this can be supported by NMS operation through path pre-establishment and activation on the agreed starting time. However, this does not provide efficient network usage since the established paths exclude the possibility of being used by other services even when they are not used for undertaking any service. It can also be accomplished through GMPLS protocol extensions by carrying the related request information (e.g., starting time and duration) across the network. Nevertheless, this method inevitably increases the complexity of signaling and routing process.

A passive stateful PCE can support this application with better efficiency since it can alleviate the burden of processing on network elements. This requires the PCE to maintain the scheduled LSPs and their associated resource usage, as well as the ability of head-ends to trigger signaling for LSP setup/deletion at the correct time. This approach requires coarse time synchronization between PCEs and PCCs. If an active stateful PCE is available, the PCE can trigger the setup/deletion of scheduled requests in a centralized manner, without modification of existing head-end behaviors.

### 5.4. Recovery

The recovery use cases discussed in the following sections show how leveraging a stateful PCE can simplify the computation of recovery path(s). In particular, two characteristics of a stateful PCE are used: 1) using information stored in the LSP-DB for determining shared protection resources and 2) performing computations with knowledge of all LSPs in a domain.

#### 5.4.1. Protection

For protection purposes, a PCC may send a request to a PCE for computing a set of paths for a given LSP. Alternatively, the PCC can send multiple requests to the PCE, asking for working and backup LSPs separately. Either way, the resources bound to backup paths can be shared by different LSPs to improve the overall network efficiency, such as m:n protection or pre-configured shared mesh recovery

techniques as specified in [RFC4427]. If resource sharing is supported for LSP protection, the information relating to existing LSPs is required to avoid allocation of shared protection resources to two LSPs that might fail together and cause protection contention issues. A stateless PCE can accommodate this use case by having the PCC pass in this information as a constraint to the path computation request. A stateful PCE can more easily accommodate this need using the information stored in its LSP-DB.

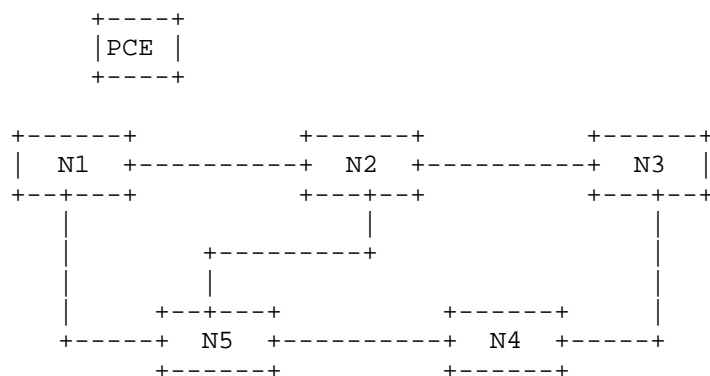


Figure 3: Reference topology 3

For example, in the network depicted in Figure 3, suppose there exists LSP1 with working path LSP1\_working following N1->N5 and with backup path LSP1\_backup following N1->N2->N5. A request arrives asking for a working and backup path pair to be computed for LSP2, for a request from N2 to N5. If the PCE decides LSP2\_working follows N2->N1->N5, then the backup path LSP2\_backup should not use the same protection resource with LSP1 since LSP2 shares part of its resource (specifically N1->N5) with LSP1 (i.e., these two LSPs are in the same shared risk group). Alternatively, there is no such constraint if N2->N3->N4->N5 is chosen for LSP2\_working.

If a stateless PCE is used, the head node N2 needs to be aware of the existence of LSPs which share the route of LSP2\_working and of the details of their protection resources. N2 must pass this information to the PCE as a constraint so as to request a path with SRLG diversity. On the other hand, a stateful PCE can get the LSPs information by itself and can achieve the goal of finding SRLG-diversified protection paths for both LSPs. This is made possible by comparing the LSP resource usage exploiting the LSP DB accessible by the stateful PCE.

#### 5.4.2. Restoration

In case of a link failure, such as fiber cut, multiple LSPs may fail at the same time. Thus, the source nodes of the affected LSPs will be informed of the failure by the nodes detecting the failure. These source nodes will send requests to a PCE for rerouting. In order to reuse the resource taken by an existing LSP, the source node can send a PCReq message including the XRO object with F bit set, together with RRO object, as specified in [RFC5521].

If a stateless PCE is exploited, it might respond to the rerouting requests separately if they arrive at different times. Thus, it might result in sub-optimal resource usage. Even worse, it might unnecessarily block some of the rerouting requests due to insufficient resources for later-arrived rerouting messages. If a stateful PCE is used to fulfill this task, it can re-compute the affected LSPs concurrently while reusing part of the existing LSPs resources when it is informed of the failed link identifier provided by the first request. This is made possible since the stateful PCE can check what other LSPs are affected by the failed link and their route information by inspecting its LSP-DB. As a result, a better performance, such as better resource usage, minimal probability of blocking upcoming new rerouting requests sent as a result of the link failure, can be achieved.

In order to further reduce the amount of LSP rerouting messages flow in the network, the notification can be performed at the node(s) which detect the link failure. For example, suppose there are two LSPs in the network as shown in Figure 3: (i) LSP1: N1->N5->N4->N3; (ii) LSP2: N2->N5->N4. They traverse the failed link between N5-N4. When N4 detects the failure, it can send a notification message to a stateful PCE. Note that the stateful PCE stores the path information of the LSPs that are affected by the link failure, so it does not need to acquire this information from N4. Moreover, it can make use of the bandwidth resources occupied by the affected LSPs when performing path recalculation. After N4 receives the new paths from the PCE, it notifies the ingress nodes of the LSPs, i.e., N1 and N2, and specifies the new paths which should be used as the rerouting paths. To support this, it would require extensions to the existing signaling protocols.

Alternatively, if the target is to avoid resource contention within the time-window of high LSP requests, a stateful PCE can retain the under-construction LSP resource usage information for a given time and exclude it from being used for forthcoming LSPs request. In this way, it can ensure that the resource will not be double-booked and thus the issue of resource contention and computation crank-backs can be resolved.

#### 5.4.3. SRLG Diversity

An alternative way to achieve efficient resilience is to maintain SRLG disjointness between LSPs, irrespective of whether these LSPs share the source and destination nodes or not. This can be achieved at provisioning time, if the routes of all the LSPs are requested together, using a synchronized computation of the different LSPs with SRLG disjointness constraint. If the LSPs need to be provisioned at different times (more general, the routes are requested at different times, e.g. in the case of a restoration), the PCC can specify, as constraints to the path computation a set of Shared Risk Link Groups (SRLGs) using the Explicit Route Object [RFC5521]. However, for the latter to be effective, it is needed that the entity that requests the route to the PCE maintains updated SRLG information of all the LSPs to which it must maintain the disjointness. A stateless PCE can compute an SRLG-disjoint path by inspecting the TED and precluding the links with the same SRLG values specified in the PCReq message sent by a PCC.

A stateful PCE maintains the updated SRLG information of the established LSPs in a centralized manner. Therefore, the PCC can specify as constraints to the path computation the SRLG disjointness of a set of already established LSPs by only providing the LSP identifiers.

#### 5.5. Maintenance of Virtual Network Topology (VNT)

In Multi-Layer Networks (MLN), a Virtual Network Topology (VNT) [RFC5212] consists of a set of one or more TE LSPs in the lower layer which provides TE links to the upper layer. In [RFC5623], the PCE-based architecture is proposed to support path computation in MLN networks in order to achieve inter-layer TE.

The establishment/teardown of a TE link in VNT needs to take into consideration the state of existing LSPs and/or new LSP request(s) in the higher layer. As specified in [RFC5623], a VNT manager (VNTM) is in charge of setting up connections in the lower layer to provide TE links for upper layer. Hence, when a stateless PCE cannot find the route for a request based on the upper layer topology information, it needs to interact with the VNTM and rely on the VNTM to decide whether to set up or remove a TE link or not. On the other hand, a stateful PCE can make the decision of when and how to modify the VNT either to accommodate new LSP requests or to re-optimize resource usage across layers irrespective of the PCE models as described in [RFC5623].

## 5.6. LSP Re-optimization

In order to make efficient usage of network resources, it is sometimes desirable to re-optimize one or more LSPs dynamically. In the case of a stateless PCE, in order to optimize network resource usage dynamically through online planning, a PCC must send a request to the PCE together with detailed path/bandwidth information of the LSPs that need to be concurrently optimized. This means the PCC must be able to determine when and which LSPs should be optimized. In the case of a stateful PCE, given the LSP state information in the LSP database, the process of dynamic optimization of network resources can be automated without requiring the PCC to supply LSP state information or to trigger the request. Moreover, since a stateful PCE can maintain information for all LSPs that are in the process of being set up and since it may have the ability to control timing and sequence of LSP setup/deletion, the optimization procedures can be performed more intelligently and effectively.

A special case of LSP re-optimization is Global Concurrent Optimization (GCO) [RFC5557]. Global control of LSP operation sequence in [RFC5557] is predicated on the use of what is effectively a stateful (or semi-stateful) NMS. The NMS can be either not local to the switch, in which case another northbound interface is required for LSP attribute changes, or local/collocated, in which case there are significant issues with efficiency in resource usage. A stateful PCE adds a few features that:

- o Roll the NMS visibility into the PCE and remove the requirement for an additional northbound interface
- o Allow the PCE to determine when re-optimization is needed, with which level (GCO or a more incremental optimization)
- o Allow the PCE to determine which LSPs should be re-optimized
- o Allow a PCE to control the sequence of events across multiple PCCs, allowing for bulk (and truly global) optimization, LSP shuffling etc.

## 5.7. Resource Defragmentation

In networks with link bundles, if LSPs are dynamically allocated and released over time, the resource becomes fragmented. The overall available resource on a (bundle) link might be sufficient for a new LSP request, but if the available resource is not continuous, the request is rejected. In order to perform the defragmentation procedure, stateful PCEs can be used, since global visibility of LSPs in the network is required to accurately assess resources on the

LSPs, and perform de-fragmentation while ensuring a minimal disruption of the network. This use case cannot be accommodated by a stateless PCE since it does not possess the detailed information of existing LSPs in the network.

A case of particular interest to GMPLS-based transport networks is the frequency defragmentation in flexible grid. In Flexible grid networks [I-D.ogrcetal-ccamp-flexi-grid-fwk], LSPs with different slot widths (such as 12.5G, 25G etc.) can co-exist so as to accommodate the services with different bandwidth requests. Therefore, even if the overall spectrum can meet the service request, it may not be usable if it is not contiguous. Thus, with the help of existing LSP state information, stateful PCE can make the resource grouped together to be usable. Moreover, stateful PCE can proactively choose routes for upcoming path requests to reduce the chance of spectrum fragmentation.

#### 5.8. Impairment-Aware Routing and Wavelength Assignment (IA-RWA)

In WSONs [RFC6163], a wavelength-switched LSP traverses one or more fiber links. The bit rates of the client signals carried by the wavelength LSPs may be the same or different. Hence, a fiber link may transmit a number of wavelength LSPs with equal or mixed bit rate signals. For example, a fiber link may multiplex the wavelengths with only 10G signals, mixed 10G and 40G signals, or mixed 40G and 100G signals.

IA-RWA in WSONs refers to the RWA process (i.e., lightpath computation) that takes into account the optical layer/transmission imperfections by considering as additional (i.e., physical layer) constraints. To be more specific, linear and non-linear effects associated with the optical network elements should be incorporated into the route and wavelength assignment procedure. For example, the physical imperfection can result in the interference of two adjacent lightpaths. Thus, a guard band should be reserved between them to alleviate these effects. The width of the guard band between two adjacent wavelengths depends on their characteristics, such as modulation formats and bit rates. Two adjacent wavelengths with different characteristics (e.g., different bit rates) may need a wider guard band and with same characteristics may need a narrower guard band. For example, 50GHz spacing may be acceptable for two adjacent wavelengths with 40G signals. But for two adjacent wavelengths with different bit rates (e.g., 10G and 40G), a larger spacing such as 300GHz spacing may be needed. Hence, the characteristics (states) of the existing wavelength LSPs should be considered for a new RWA request in WSON.

In summary, when stateful PCEs are used to perform the IA-RWA

procedure, they need to know the characteristics of the existing wavelength LSPs. The impairment information relating to existing and to-be-established LSPs can be obtained by nodes in WSON networks via external configuration or other means such as monitoring or estimation based on a vendor-specific impair model. However, WSON related routing protocols, i.e., [I-D.ietf-ccamp-wson-signal-compatibility-ospf] and [I-D.ietf-ccamp-gmpls-general-constraints-ospf-te], only advertise limited information (i.e., availability) of the existing wavelengths, without defining the supported client bit rates. It will incur substantial amount of control plane overhead if routing protocols are extended to support dissemination of the new information relevant for the IA-RWA process. In this scenario, stateful PCE(s) would be a more appropriate mechanism to solve this problem. Stateful PCE(s) can exploit impairment information of LSPs stored in LSP-DB to provide accurate RWA calculation.

## 6. Security Considerations

This document does not introduce any new security considerations beyond those discussed in [I-D.ietf-pce-stateful-pce].

The following topics will be discussed in a future version of this document: whether use of a stateful PCE makes the network more or less secure, and security use cases if any.

## 7. Contributing Authors

The following people all contributed significantly to this document and are listed below in alphabetical order:

Ramon Casellas  
CTTC - Centre Tecnologic de Telecomunicacions de Catalunya  
Av. Carl Friedrich Gauss n7  
Castelldefels, Barcelona 08860  
Spain  
Email: ramon.casellas@cttc.es

Edward Crabbe  
Google, Inc.  
1600 Amphitheatre Parkway  
Mountain View, CA 94043  
US  
Email: edc@google.com

Dhruv Dhody

Huawei Technology  
Leela Palace  
Bangalore, Karnataka 560008  
INDIA  
EMail: dhruvd@huawei.com

Oscar Gonzalez de Dios  
Telefonica Investigacion y Desarrollo  
Emilio Vargas 6  
Madrid, 28045  
Spain  
Phone: +34 913374013  
Email: ogondio@tid.es

Young Lee  
Huawei  
1700 Alma Drive, Suite 100  
Plano, TX 75075  
US  
Phone: +1 972 509 5599 x2240  
Fax: +1 469 229 5397  
EMail: ylee@huawei.com

Jan Medved  
Cisco Systems, Inc.  
170 West Tasman Dr.  
San Jose, CA 95134  
US  
Email: jmedved@cisco.com

Robert Varga  
Pantheon Technologies LLC  
Mlynske Nivy 56  
Bratislava 821 05  
Slovakia  
Email: robert.varga@pantheon.sk

Fatai Zhang  
Huawei Technologies  
F3-5-B R&D Center, Huawei Base  
Bantian, Longgang District  
Shenzhen 518129 P.R.China  
Phone: +86-755-28972912  
Email: zhangfatai@huawei.com

Xiaobing Zi  
Email: unknown



## 8. Acknowledgements

We would like to thank Cyril Margaria, Adrian Farrel and JP Vasseur for the useful comments and discussions.

## 9. References

### 9.1. Normative References

- [I-D.ietf-pce-stateful-pce]  
Crabbe, E., Medved, J., Minei, I., and R. Varga, "PCEP Extensions for Stateful PCE",  
draft-ietf-pce-stateful-pce-04 (work in progress),  
May 2013.
- [RFC4655] Farrel, A., Vasseur, J., and J. Ash, "A Path Computation Element (PCE)-Based Architecture", RFC 4655, August 2006.
- [RFC5440] Vasseur, JP. and JL. Le Roux, "Path Computation Element (PCE) Communication Protocol (PCEP)", RFC 5440, March 2009.

### 9.2. Informative References

- [I-D.crabbe-pce-stateful-pce-mpls-te]  
Crabbe, E., Medved, J., Minei, I., and R. Varga, "Stateful PCE extensions for MPLS-TE LSPs",  
draft-crabbe-pce-stateful-pce-mpls-te-01 (work in progress), May 2013.
- [I-D.ietf-ccamp-gmpls-general-constraints-ospf-te]  
Zhang, F., Lee, Y., Han, J., Bernstein, G., and Y. Xu, "OSPF-TE Extensions for General Network Element Constraints",  
draft-ietf-ccamp-gmpls-general-constraints-ospf-te-04 (work in progress), July 2012.
- [I-D.ietf-ccamp-wson-signal-compatibility-ospf]  
Lee, Y. and G. Bernstein, "GMPLS OSPF Enhancement for Signal and Network Element Compatibility for Wavelength Switched Optical Networks",  
draft-ietf-ccamp-wson-signal-compatibility-ospf-11 (work in progress), February 2013.
- [I-D.ietf-pce-gmpls-pcep-extensions]  
Margaria, C., Dios, O., and F. Zhang, "PCEP extensions for GMPLS", draft-ietf-pce-gmpls-pcep-extensions-07 (work in progress), May 2013.

progress), October 2012.

- [I-D.ogrcetal-ccamp-flexi-grid-fwk]  
Dios, O., Casellas, R., Zhang, F., Fu, X., Ceccarelli, D.,  
and I. Hussain, "Framework and Requirements for GMPLS  
based control of Flexi-grid DWDM networks",  
draft-ogrcetal-ccamp-flexi-grid-fwk-02 (work in progress),  
February 2013.
- [I-D.sivabalan-pce-disco-stateful]  
Sivabalan, S., Medved, J., and X. Zhang, "IGP Extensions  
for Stateful PCE Discovery",  
draft-sivabalan-pce-disco-stateful-01 (work in progress),  
April 2013.
- [MPLS-PC] Chaieb, I., Le Roux, JL., and B. Cousin, "Improved MPLS-TE  
LSP Path Computation using Preemption", Global  
Information Infrastructure Symposium, July 2007.
- [MXMN-TE] Danna, E., Mandal, S., and A. Singh, "Practical linear  
programming algorithm for balancing the max-min fairness  
and throughput objectives in traffic engineering", pre-  
print, 2011.
- [NET-REC] Vasseur, JP., Pickavet, M., and P. Demeester, "Network  
Recovery: Protection and Restoration of Optical, SONET-  
SDH, IP, and MPLS", The Morgan Kaufmann Series in  
Networking, June 2004.
- [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.
- [RFC4427] Mannie, E. and D. Papadimitriou, "Recovery (Protection and  
Restoration) Terminology for Generalized Multi-Protocol  
Label Switching (GMPLS)", RFC 4427, March 2006.
- [RFC4657] Ash, J. and J. Le Roux, "Path Computation Element (PCE)  
Communication Protocol Generic Requirements", RFC 4657,  
September 2006.
- [RFC5212] Shiimoto, K., Papadimitriou, D., Le Roux, JL., Vigoureux,  
M., and D. Brungard, "Requirements for GMPLS-Based Multi-  
Region and Multi-Layer Networks (MRN/MLN)", RFC 5212,  
July 2008.
- [RFC5394] Bryskin, I., Papadimitriou, D., Berger, L., and J. Ash,  
"Policy-Enabled Path Computation Framework", RFC 5394,

December 2008.

- [RFC5521] Oki, E., Takeda, T., and A. Farrel, "Extensions to the Path Computation Element Communication Protocol (PCEP) for Route Exclusions", RFC 5521, April 2009.
- [RFC5557] Lee, Y., Le Roux, JL., King, D., and E. Oki, "Path Computation Element Communication Protocol (PCEP) Requirements and Protocol Extensions in Support of Global Concurrent Optimization", RFC 5557, July 2009.
- [RFC5623] Oki, E., Takeda, T., Le Roux, JL., and A. Farrel, "Framework for PCE-Based Inter-Layer MPLS and GMPLS Traffic Engineering", RFC 5623, September 2009.
- [RFC6163] Lee, Y., Bernstein, G., and W. Imajuku, "Framework for GMPLS and Path Computation Element (PCE) Control of Wavelength Switched Optical Networks (WSOs)", RFC 6163, April 2011.

#### Appendix A. Editorial notes and open issues

This section will be removed prior to publication.

The following open issues remain:

Use cases from draft-ietf-pce-stateful-pce To avoid loss of information, the use cases will be removed from [I-D.ietf-pce-stateful-pce] only after this document becomes a working group document.

This document WILL NOT repeat terminology defined in other documents or attempt to place any additional requirements on stateful PCE.

#### Authors' Addresses

Xian Zhang (editor)  
Huawei Technologies  
F3-5-B R&D Center, Huawei Base Bantian, Longgang District  
Shenzhen, Guangdong 518129  
P.R.China

Email: zhang.xian@huawei.com

Ina Minei (editor)  
Juniper Networks, Inc.  
1194 N. Mathilda Ave.  
Sunnyvale, CA 94089  
US

Email: [ina@juniper.net](mailto:ina@juniper.net)

