

PCE Working Group
Internet-Draft
Intended status: Standards Track
Expires: December 23, 2020

Q. Zhao
Z. Li
Huawei Technologies
M. Negi
RtBrick India
S. Peng
Huawei Technologies
C. Zhou
Cisco Systems
June 21, 2020

**PCEP Procedures and Protocol Extensions for Using PCE as a Central
Controller (PCECC) of LSPs
draft-ietf-pce-pcep-extension-for-pce-controller-05**

Abstract

The Path Computation Element (PCE) is a core component of Software-Defined Networking (SDN) systems. It can compute optimal paths for traffic across a network and can also update the paths to reflect changes in the network or traffic demands.

PCE was developed to derive paths for MPLS Label Switched Paths (LSPs), which are supplied to the head end of the LSP using the Path Computation Element Communication Protocol (PCEP). But SDN has a broader applicability than signaled (G)MPLS traffic-engineered (TE) networks, and the PCE may be used to determine paths in a range of use cases. PCEP has been proposed as a control protocol for use in these environments to allow the PCE to be fully enabled as a central controller.

A PCE-based central controller (PCECC) can simplify the processing of a distributed control plane by blending it with elements of SDN and without necessarily completely replacing it. Thus, the LSP can be calculated/setup/initiated and the label forwarding entries can also be downloaded through a centralized PCE server to each network devices along the path while leveraging the existing PCE technologies as much as possible.

This document specifies the procedures and PCEP protocol extensions for using the PCE as the central controller.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

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

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

This Internet-Draft will expire on December 23, 2020.

Copyright Notice

Copyright (c) 2020 IETF Trust and the persons identified as the document authors. All rights reserved.

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

Table of Contents

1.	Introduction	3
1.1.	Requirements Language	5
2.	Terminology	5
3.	Basic PCECC Mode	5
4.	PCEP Requirements	5
5.	Procedures for Using the PCE as the Central Controller (PCECC)	6
5.1.	Stateful PCE Model	6
5.2.	New LSP Functions	6
5.3.	New PCEP Object	7
5.4.	PCECC Capability Advertisement	7
5.5.	LSP Operations	8
5.5.1.	Basic PCECC LSP Setup	8
5.5.2.	Central Control Instructions	12
5.5.2.1.	Label Download CCI	12
5.5.2.2.	Label Cleanup CCI	12
5.5.3.	PCE Initiated PCECC LSP	13
5.5.4.	PCECC LSP Update	15
5.5.5.	Re Delegation and Cleanup	17

5.5.6.	Synchronization of Central Controllers Instructions .	17
5.5.7.	PCECC LSP State Report	17
5.5.8.	PCC Based Allocations	18
5.5.9.	Binding Label	18
6.	PCEP messages	19
6.1.	The PCInitiate message	19
6.2.	The PCRpt message	21
7.	PCEP Objects	21
7.1.	OPEN Object	22
7.1.1.	PCECC Capability sub-TLV	22
7.2.	PATH-SETUP-TYPE TLV	22
7.3.	CCI Object	23
7.3.1.	Address TLVs	24
8.	Implementation Status	25
8.1.	Huawei's Proof of Concept based on ONOS	26
9.	Security Considerations	26
9.1.	Malicious PCE	26
10.	Manageability Considerations	27
10.1.	Control of Function and Policy	27
10.2.	Information and Data Models	27
10.3.	Liveness Detection and Monitoring	27
10.4.	Verify Correct Operations	27
10.5.	Requirements On Other Protocols	27
10.6.	Impact On Network Operations	27
11.	IANA Considerations	27
11.1.	PCEP TLV Type Indicators	27
11.2.	PATH-SETUP-TYPE-CAPABILITY Sub-TLV Type Indicators	28
11.3.	New Path Setup Type Registry	28
11.4.	PCEP Object	28
11.5.	CCI Object Flag Field	28
11.6.	PCEP-Error Object	29
12.	Acknowledgments	29
13.	References	30
13.1.	Normative References	30
13.2.	Informative References	31
Appendix A.	Contributor Addresses	33
	Authors' Addresses	34

[1.](#) Introduction

The Path Computation Element (PCE) [[RFC4655](#)] was developed to offload path computation function from routers in an MPLS traffic-engineered network. Since then, the role and function of the PCE has grown to cover a number of other uses (such as GMPLS [[RFC7025](#)]) and to allow delegated control [[RFC8231](#)] and PCE-initiated use of network resources [[RFC8281](#)].

According to [\[RFC7399\]](#), Software-Defined Networking (SDN) refers to a separation between the control elements and the forwarding components so that software running in a centralized system, called a controller, can act to program the devices in the network to behave in specific ways. A required element in an SDN architecture is a component that plans how the network resources will be used and how the devices will be programmed. It is possible to view this component as performing specific computations to place traffic flows within the network given knowledge of the availability of network resources, how other forwarding devices are programmed, and the way that other flows are routed. This is the function and purpose of a PCE, and the way that a PCE integrates into a wider network control system (including an SDN system) is presented in [\[RFC7491\]](#).

In early PCE implementations, where the PCE was used to derive paths for MPLS Label Switched Paths (LSPs), paths were requested by network elements (known as Path Computation Clients (PCCs)), and the results of the path computations were supplied to network elements using the Path Computation Element Communication Protocol (PCEP) [\[RFC5440\]](#). This protocol was later extended to allow a PCE to send unsolicited requests to the network for LSP establishment [\[RFC8281\]](#).

[\[RFC8283\]](#) introduces the architecture for PCE as a central controller as an extension of the architecture described in [\[RFC4655\]](#) and assumes the continued use of PCEP as the protocol used between PCE and PCC. [\[RFC8283\]](#) further examines the motivations and applicability for PCEP as a Southbound Interface (SBI), and introduces the implications for the protocol. [\[I-D.ietf-teas-pcecc-use-cases\]](#) describes the use cases for the PCECC architecture.

A PCE-based central controller (PCECC) can simplify the processing of a distributed control plane by blending it with elements of SDN and without necessarily completely replacing it. Thus, the LSP can be calculated/setup/initiated and the label forwarding entries can also be downloaded through a centralized PCE server to each network devices along the path while leveraging the existing PCE technologies as much as possible.

This draft specify the procedures and PCEP protocol extensions for using the PCE as the central controller for static LSPs, where LSPs can be provisioned as explicit label instructions at each hop on the end-to-end path. Each router along the path must be told what label-forwarding instructions to program and what resources to reserve. The PCE-based controller keeps a view of the network and determines the paths of the end-to-end LSPs, and the controller uses PCEP to communicate with each router along the path of the end-to-end LSP.

The extension for PCECC in Segment Routing (SR) is specified in a separate draft [[I-D.zhao-pce-pcep-extension-pce-controller-sr](#)].

1.1. Requirements Language

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

2. Terminology

Terminologies used in this document is same as described in the draft [[RFC8283](#)].

3. Basic PCECC Mode

In this mode LSPs are provisioned as explicit label instructions at each hop on the end-to-end path. Each router along the path must be told what label forwarding instructions to program and what resources to reserve. The controller uses PCEP to communicate with each router along the path of the end-to-end LSP.

Note that the PCE-based controller will take responsibility for managing some part of the MPLS label space for each of the routers that it controls, and may take wider responsibility for partitioning the label space for each router and allocating different parts for different uses. This is also described in [section 3.1.2. of](#) [\[RFC8283\]](#). For the purpose of this document, it is assumed that label range to be used by a PCE is known and set on both PCEP peers. A future extension could add this capability to advertise the range via possible PCEP extensions as well (see [[I-D.li-pce-controlled-id-space](#)]). The rest of processing is similar to the existing stateful PCE mechanism.

This document also allow a case where the label space is maintained by PCC itself, and the labels are allocated by the PCC, in this case, the PCE should request the allocation from PCC as described in [Section 5.5.8](#).

4. PCEP Requirements

Following key requirements associated PCECC should be considered when designing the PCECC based solution:

1. PCEP speaker supporting this draft needs to have the capability to advertise its PCECC capability to its peers.

2. PCEP speaker needs a means to identify PCECC based LSP in the PCEP messages.
3. PCEP procedures needs to allow for PCC based label allocations.
4. PCEP procedures needs to provide a means to update (or cleanup) the label-download entry to the PCC.
5. PCEP procedures needs to provide a means to synchronize the labels between PCE to PCC in PCEP messages.

5. Procedures for Using the PCE as the Central Controller (PCECC)

5.1. Stateful PCE Model

Active stateful PCE is described in [[RFC8231](#)]. PCE as a central controller (PCECC) reuses existing Active stateful PCE mechanism as much as possible to control the LSP.

5.2. New LSP Functions

Several new functions are required in PCEP to support PCECC. This document extends the existing messages to support the new functions required by PCECC:

(PCInitiate): a PCEP message described in [[RFC8281](#)]. PCInitiate message is used to setup PCE-Initiated LSP based on PCECC mechanism. It is also extended for Central Controller's Instructions (CCI) (download or cleanup the Label forwarding instructions in the context of this document) on all nodes along the path.

(PCRpt): a PCEP message described in [[RFC8231](#)]. PCRpt message is used to send PCECC LSP Reports. It is also extended to report the set of Central Controller's Instructions (CCI) (label forwarding instructions in the context of this document) received from the PCE. See [Section 5.5.6](#) for more details.

(PCUpd): a PCEP message described in [[RFC8231](#)]. PCUpd message is used to send PCECC LSP Update.

The new functions defined in this document are mapped onto the PCEP messages as shown in the following table.

Function	Message
PCECC Capability advertisement	Open
Label entry Add	PCInitiate
Label entry Cleanup	PCInitiate
PCECC Initiated LSP	PCInitiate
PCECC LSP Update	PCUpd
PCECC LSP State Report	PCRpt
PCECC LSP Delegation	PCRpt
PCECC Label Report	PCRpt

5.3. New PCEP Object

This document specifies a new PCEP object called CCI (see [Section 7.3](#)) for the encoding of central controller's instructions. In the scope of this document this is limited to Label forwarding instructions. Future documents can create new CCI object-type for other types of central control instructions. The CC-ID is the unique identifier for the central controller's instructions in PCEP. The PCEP messages are extended in this document to handle the PCECC operations.

5.4. PCECC Capability Advertisement

During PCEP Initialization Phase, PCEP Speakers (PCE or PCC) advertise their support of PCECC extensions.

This document defines a new Path Setup Type (PST) [[RFC8408](#)] for PCECC, as follows:

- o PST = TBD1: Path is setup via PCECC mode.

A PCEP speaker MUST indicate its support of the function described in this document by sending a PATH-SETUP-TYPE-CAPABILITY TLV in the OPEN object with this new PST included in the PST list.

This document also defines the PCECC Capability sub-TLV [Section 7.1.1](#). PCEP speakers use this sub-TLV to exchange information about their PCECC capability. If a PCEP speaker includes PST=TBD1 in the PST List of the PATH-SETUP-TYPE-CAPABILITY TLV then it MUST also include the PCECC Capability sub-TLV inside the PATH-SETUP-TYPE-CAPABILITY TLV. If the sub-TLV is absent, then the PCEP speaker MUST send a PCErr message with Error-Type 10 (Reception of an invalid object) and Error-Value TBD2 (Missing PCECC Capability sub-TLV) and MUST then close the PCEP session. If a PCEP speaker receives a PATH-SETUP-TYPE-CAPABILITY TLV with a PCECC-CAPABILITY

sub-TLV, but the PST list does not contain PST=TBD1, then the PCEP speaker MUST ignore the PCECC-CAPABILITY sub-TLV.

The presence of the PST=TBD1 and PCECC Capability sub-TLV in PCC's OPEN Object indicates that the PCC is willing to function as a PCECC client. The presence of the PST=TBD1 and PCECC Capability sub-TLV in PCE's OPEN message indicates that the PCE is interested in function as a PCECC server.

The PCEP protocol extensions for PCECC MUST NOT be used if one or both PCEP Speakers have not included the PST=TBD1 or the PCECC Capability sub-TLV in their respective OPEN message. If the PCEP Speakers support the extensions of this draft but did not advertise this capability attempts a PCECC operation then a PCErr message with Error-Type=19(Invalid Operation) and Error-Value=TBD3 (Attempted PCECC operations when PCECC capability was not advertised) will be generated and the PCEP session will be terminated.

A PCC or a PCE MUST include both PCECC-CAPABILITY sub-TLV and STATEFUL-PCE-CAPABILITY TLV ([[RFC8231](#)]) (with I flag set [[RFC8281](#)]) in OPEN Object to support the extensions defined in this document. If PCECC-CAPABILITY sub-TLV is advertised and STATEFUL-PCE-CAPABILITY TLV is not advertised in OPEN Object, it MUST send a PCErr message with Error-Type=19 (Invalid Operation) and Error-value=TBD4 (stateful PCE capability was not advertised) and terminate the session. This error is also triggered if PCECC-CAPABILITY sub-TLV is advertised and I flag in the STATEFUL-PCE-CAPABILITY TLV is not set.

5.5. LSP Operations

The PCEP messages pertaining to PCECC MUST include PATH-SETUP-TYPE TLV [[RFC8408](#)] in the SRP object to clearly identify the PCECC LSP is intended.

5.5.1. Basic PCECC LSP Setup

In order to setup a LSP based on PCECC mechanism, a PCC MUST delegate the LSP by sending a PCRpt message with PST set for PCECC (see [Section 7.2](#)) and D (Delegate) flag (see [[RFC8231](#)]) set in the LSP object.

LSP-IDENTIFIER TLV MUST be included for PCECC LSP, the tuple uniquely identifies the LSP in the network. The LSP object is included in central controller's instructions (label download) to identify the PCECC LSP for this instruction. The PLSP-ID is the original identifier used by the ingress PCC, so the transit LSR could have multiple central controller instructions that have the same PLSP-ID. The PLSP-ID in combination with the source (in LSP-IDENTIFIER TLV)

MUST be unique. The PLSP-ID is included for maintainability reasons to ease debugging. As per [\[RFC8281\]](#), the LSP object could include SPEAKER-ENTITY-ID TLV to identify the PCE that initiated these instructions. Also the CC-ID is unique on the PCEP session as described in [Section 7.3](#).

When a PCE receives PCRpt message with D flags and PST Type set, it calculates the path and assigns labels along the path; and set up the path by sending PCInitiate message to each node along the path of the LSP. The PCC generates a Path Computation State Report (PCRpt) and include the central controller's instruction (CCI) and the identified LSP. The CC-ID is uniquely identify the central controller's instruction within a PCEP session. The PCC further responds with the PCRpt messages including the CCI and LSP objects.

The Ingress node would receive one CCI object with 0 bit (out-label) set. The transit node(s) would receive two CCI object with the in-label CCI without 0 bit set and the out-label CCI with 0 bit set. The egress node would receive one CCI object without 0 bit set. A node can determine its role based on the setting of the 0 bit in the CCI object(s).

Once the central controller's instructions (label operations) are completed, the PCE MUST send the PCUpd message to the Ingress PCC. This PCUpd message is as per [\[RFC8231\]](#) SHOULD include the path information as calculated by the PCE.

Note that the PCECC LSPs MUST be delegated to a PCE at all times.

LSP deletion operation for PCECC LSP is same as defined in [\[RFC8231\]](#). If the PCE receives PCRpt message for LSP deletion then it does Label cleanup operation as described in [Section 5.5.2.2](#) for the corresponding LSP.

The Basic PCECC LSP setup sequence is as shown below.

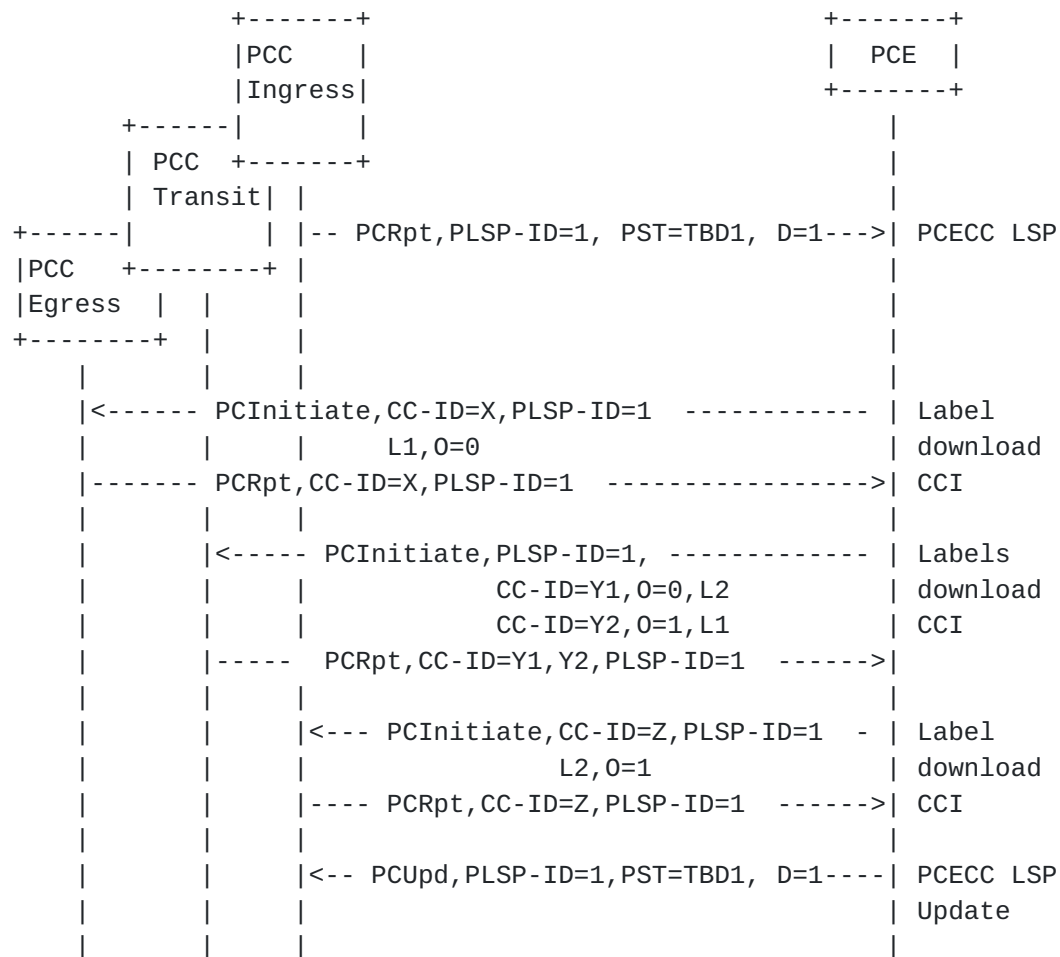
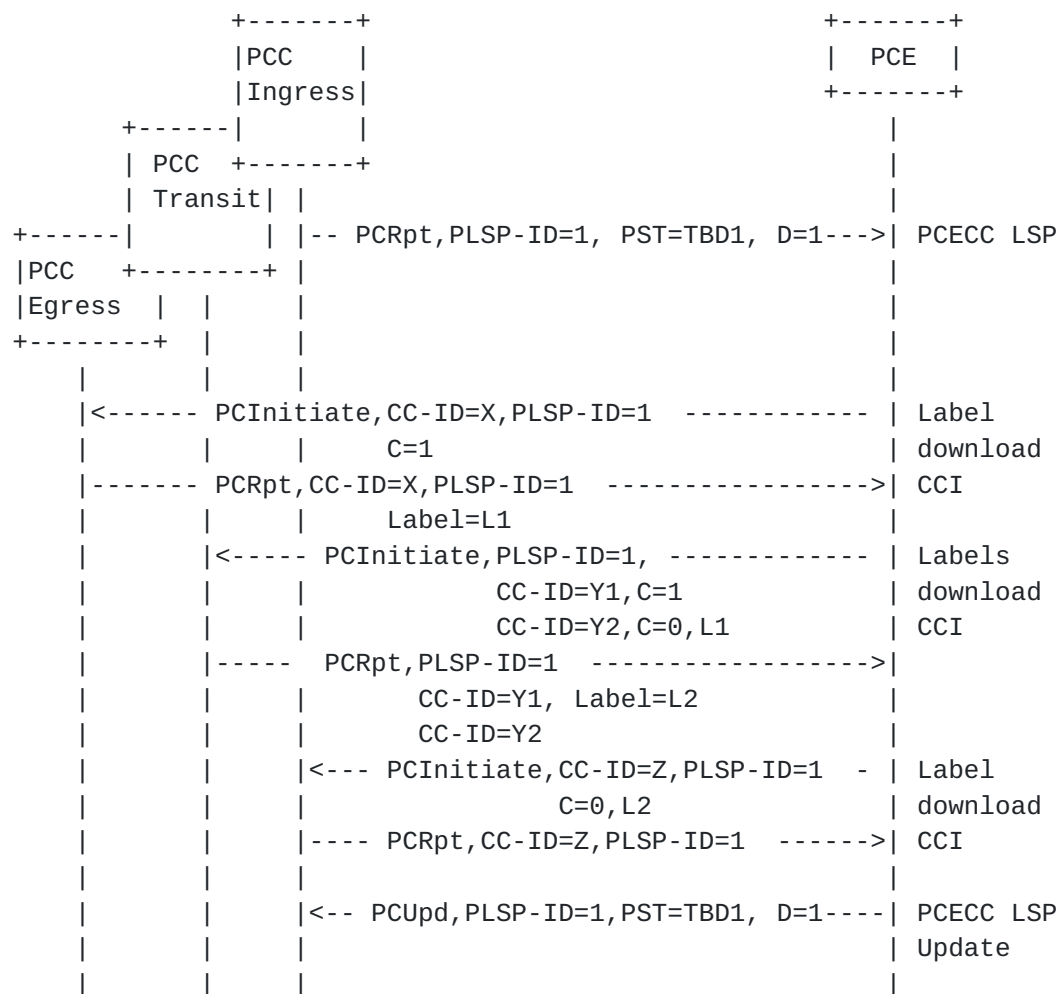


Figure 2: Basic PCECC LSP setup

The PCECC LSP are considered to be 'up' by default (on receipt of PCUpd message from PCE). The Ingress MAY further choose to deploy a data plane check mechanism and report the status back to the PCE via PCRpt message.

In case where the label allocation are made by the PCC itself (see [Section 5.5.8](#)), the PCE could request an allocation to be made by the PCC, and where the PCC would send a PCRpt with the allocated label encoded in the CC-ID object as shown below -



- The 0 bit is set as before (and thus not included)

Figure 3: Basic PCECC LSP setup (PCC allocation)

It should be noted that in this example, the request is made to the egress node with C bit set in the CCI object to indicate that the label allocation needs to be done by the egress and it responds with the allocated label to the PCE. The PCE would further inform the transit PCC without setting the C bit in the CCI object for out-label but the C-bit is set for in-label so the transit node make the label allocation (for the in-label) and report to the PCE. Similarly C bit is unset towards the ingress to complete all the label allocation for the PCECC LSP.

5.5.2. Central Control Instructions

The new central controller's instructions (CCI) for the label operations in PCEP is done via the PCInitiate message, by defining a new PCEP Objects for CCI operations. Local label range of each PCC is assumed to be known at both the PCC and the PCE.

5.5.2.1. Label Download CCI

In order to setup an LSP based on PCECC, the PCE sends a PCInitiate message to each node along the path to download the Label instruction as described in [Section 5.5.1](#).

The CCI object MUST be included, along with the LSP object in the PCInitiate message. The LSP-IDENTIFIER TLV MUST be included in LSP object. The SPEAKER-ENTITY-ID TLV SHOULD be included in LSP object.

If a node (PCC) receives a PCInitiate message which includes a Label to download as part of CCI, that is out of the range set aside for the PCE, it MUST send a PCErr message with Error-type=TBD5 (PCECC failure) and Error-value=TBD6 (Label out of range) and MUST include the SRP object to specify the error is for the corresponding label update via PCInitiate message. If a PCC receives a PCInitiate message but failed to download the Label entry, it MUST send a PCErr message with Error-type=TBD5 (PCECC failure) and Error-value=TBD7 (instruction failed) and MUST include the SRP object to specify the error is for the corresponding label update via PCInitiate message.

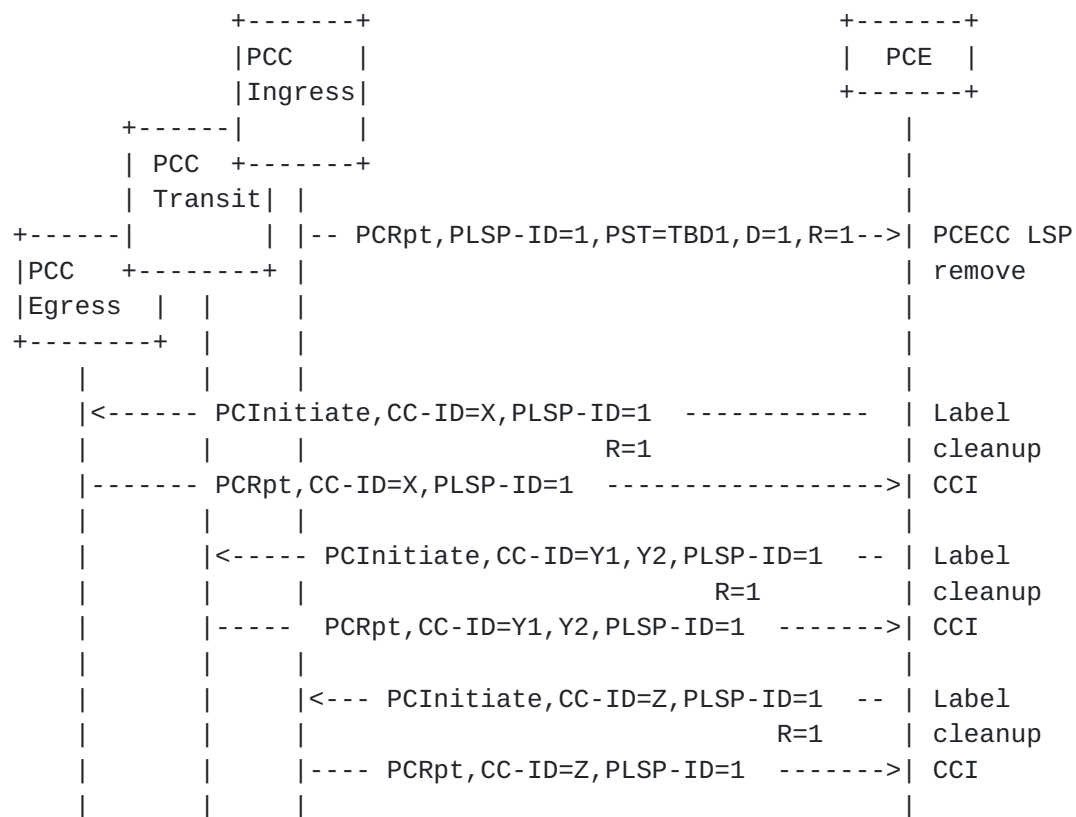
New PCEP object for central control instructions (CCI) is defined in [Section 7.3](#).

5.5.2.2. Label Cleanup CCI

In order to delete an LSP based on PCECC, the PCE sends a central controller instructions via a PCInitiate message to each node along the path of the LSP to cleanup the Label forwarding instruction.

If the PCC receives a PCInitiate message but does not recognize the label in the CCI, the PCC MUST generate a PCErr message with Error-Type 19(Invalid operation) and Error-Value=TBD8, "Unknown Label" and MUST include the SRP object to specify the error is for the corresponding label cleanup (via PCInitiate message).

The R flag in the SRP object defined in [[RFC8281](#)] specifies the deletion of Label Entry in the PCInitiate message.



As per [RFC8281], following the removal of the Label forwarding instruction, the PCC MUST send a PCRpt message. The SRP object in the PCRpt MUST include the SRP-ID-number from the PCInitiate message that triggered the removal. The R flag in the SRP object MUST be set.

In case where the label allocation are made by the PCC itself (see [Section 5.5.8](#)), the removal procedure remains the same.

5.5.3. PCE Initiated PCECC LSP

The LSP Instantiation operation is same as defined in [RFC8281].

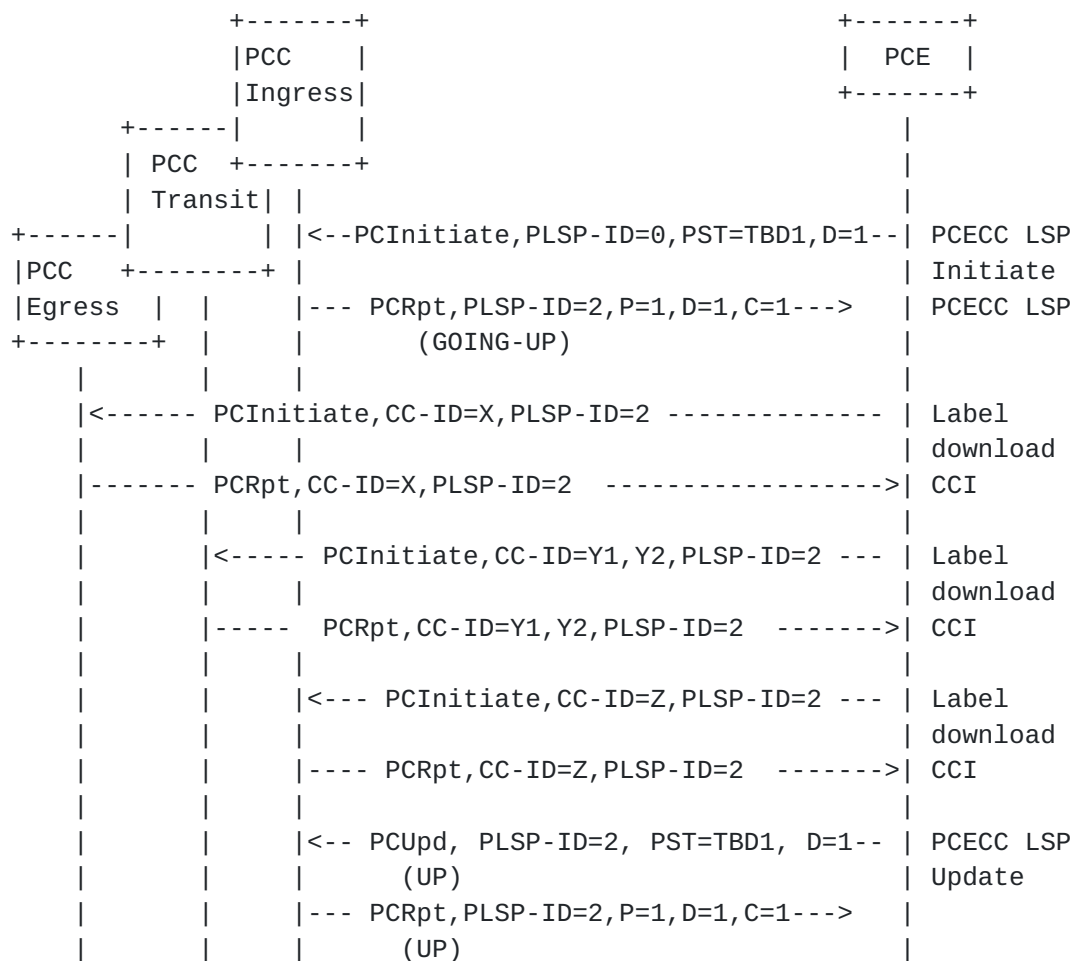
In order to setup a PCE Initiated LSP based on the PCECC mechanism, a PCE sends PCInitiate message with Path Setup Type set for PCECC (see [Section 7.2](#)) to the Ingress PCC.

The Ingress PCC MUST also set D (Delegate) flag (see [RFC8231]) and C (Create) flag (see [RFC8281]) in LSP object of PCRpt message. The PCC responds with first PCRpt message with the status as "GOING-UP" and assigned PLSP-ID.

Note that the label forwarding instructions from PCECC are send after the initial PCInitiate and PCRpt exchange. This is done so that the PLSP-ID and other LSP identifiers can be obtained from the ingress and can be included in the label forwarding instruction in the next PCInitiate message. The rest of the PCECC LSP setup operations are same as those described in [Section 5.5.1](#).

The LSP deletion operation for PCE Initiated PCECC LSP is same as defined in [\[RFC8281\]](#). The PCE should further perform Label entry cleanup operation as described in [Section 5.5.2.2](#) for the corresponding LSP.

The PCE Initiated PCECC LSP setup sequence is shown below -



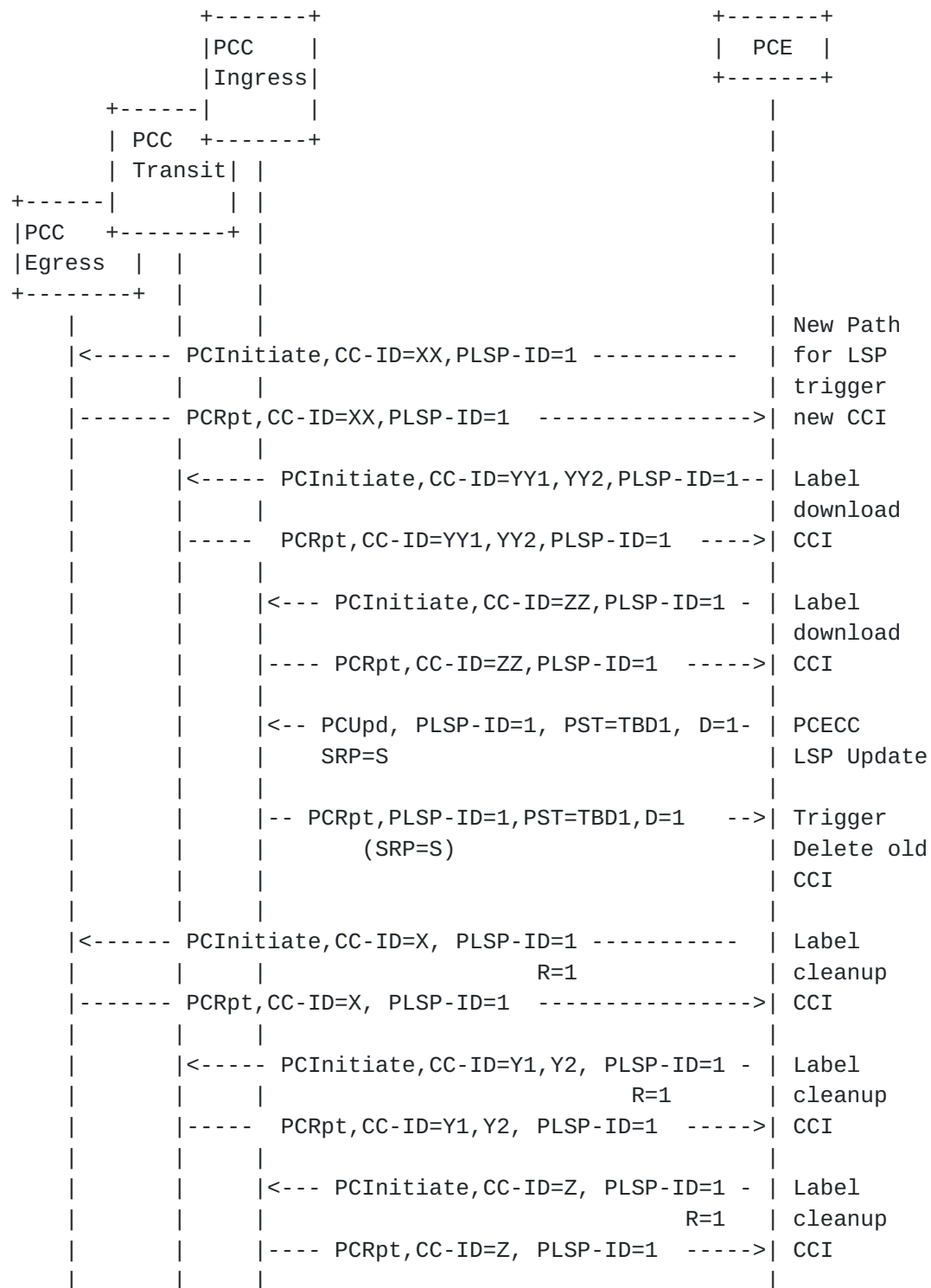
Once the label operations are completed, the PCE SHOULD send the PCUpd message to the Ingress PCC. The PCUpd message is as per [\[RFC8231\]](#).

In case where the label allocation are made by the PCC itself (see [Section 5.5.8](#)), the procedure remains similar.

5.5.4. PCECC LSP Update

In case of a modification of PCECC LSP with a new path, a PCE sends a PCUpd message to the Ingress PCC. But to follow the make-before-break procedures, the PCECC first update new instructions based on the updated LSP and then update to ingress to switch traffic, before cleaning up the old instructions. A new CC-ID is used to identify the updated instruction, the existing identifiers in the LSP object identify the existing LSP. Once new instructions are downloaded, the PCE further updates the new path at the ingress which triggers the traffic switch on the updated path. The Ingress PCC acknowledges with a PCRpt message, on receipt of PCRpt message, the PCE does cleanup operation for the old LSP as described in [Section 5.5.2.2](#).

The PCECC LSP Update sequence is shown below -



The modified PCECC LSP are considered to be 'up' by default. The Ingress MAY further choose to deploy a data plane check mechanism and report the status back to the PCE via PCRpt message.

In case where the label allocation are made by the PCC itself (see [Section 5.5.8](#)), the procedure remains similar.

5.5.5. Re Delegation and Cleanup

As described in [\[RFC8281\]](#), a new PCE can gain control over the orphaned LSP. In case of PCECC LSP, the new PCE MUST also gain control over the central controllers instructions in the same way by sending a PCInitiate message that includes the SRP, LSP and CCI objects and carries the CC-ID and PLSP-ID identifying the instruction, it wants to take control of.

Further, as described in [\[RFC8281\]](#), the State Timeout Interval timer ensures that a PCE crash does not result in automatic and immediate disruption for the services using PCE-initiated LSPs. Similarly the central controller instructions are not removed immediately upon PCE failure. Instead, they are cleaned up on the expiration of this timer. This allows for network cleanup without manual intervention. The PCC MUST support removal of CCI as one of the behaviors applied on expiration of the State Timeout Interval timer.

5.5.6. Synchronization of Central Controllers Instructions

The purpose of Central Controllers Instructions synchronization (labels in the context of this document) is to make sure that the PCE's view of CCI (Labels) matches with the PCC's Label allocation. This synchronization is performed as part of the LSP state synchronization as described in [\[RFC8231\]](#) and [\[RFC8233\]](#).

As per LSP State Synchronization [\[RFC8231\]](#), a PCC reports the state of its LSPs to the PCE using PCRpt messages and as per [\[RFC8281\]](#), PCE would initiate any missing LSPs and/or remove any LSPs that are not wanted. The same PCEP messages and procedure is also used for the Central Controllers Instructions synchronization. The PCRpt message includes the CCI and the LSP object to report the label forwarding instructions. The PCE would further remove any unwanted instructions or initiate any missing instructions.

5.5.7. PCECC LSP State Report

As mentioned before, an Ingress PCC MAY choose to apply any OAM mechanism to check the status of LSP in the Data plane and MAY further send its status in PCRpt message to the PCE.

5.5.8. PCC Based Allocations

The PCE can request the PCC to allocate the label using the PCInitiate message. The C flag in the CCI object is set to 1 to indicate that the allocation needs to be done by the PCC. The PCC would allocate the Label and would report to the PCE using the PCRpt message.

If the value of the Label is 0 and the C flag is set, it indicates that the PCE is requesting the allocation to be done by the PCC. If the Label is 'n' and the C flag is set in the CCI object, it indicates that the PCE requests a specific value 'n' for the Label. If the allocation is successful, the PCC should report via PCRpt message with the CCI object. Else, it MUST send a PCErr message with Error-Type = TBD5 ("PCECC failure") and Error Value = TBD9 ("Invalid CCI"). If the value of the the Label in the CCI object is valid, but the PCC is unable to allocate it, it MUST send a PCErr message with Error-Type = TBD5 ("PCECC failure") and Error Value = TBD10 ("Unable to allocate the specified CCI").

If the PCC wishes to withdraw or modify the previously assigned label, it MUST send a PCRpt message without any Label or with the Label containing the new value respectively in the CCI object. The PCE would further trigger the removal of the central controller instruction as per this document.

5.5.9. Binding Label

As per [[I-D.sivabalan-pce-binding-label-sid](#)], when a stateful PCE is deployed for setting up TE paths, it may be desirable to report the binding label to the stateful PCE for the purpose of enforcing end-to-end TE. In case of PCECC, the binding label may be allocated by the PCE itself as described in this section. This procedure is thus applicable for all path setup types including PCECC.

A P flag in LSP object is introduced in [[I-D.li-pce-sr-path-segment](#)] to indicate the allocation needs to be made by the PCE. This flag is used to indicate that the allocation needs to be made by the PCE. A PCC would set this bit to 1 (and carry the TE-PATH-BINDING TLV [[I-D.sivabalan-pce-binding-label-sid](#)] in LSP object) to request for allocation of the binding label by the PCE in the PCReq or PCRpt message. A PCE would also set this bit to 1 to indicate that the binding label is allocated by PCE and encoded in the PCRep, PCUpd or PCInitiate message (the TE-PATH-BINDING TLV is present in LSP object). Further, a PCE would set this bit to 0 to indicate that the allocation is done by the PCC instead.

The ingress PCC could request the binding label to be allocated by the PCE via PCRpt message as per [RFC8231]. The delegate flag (D-flag) MUST also be set for this LSP. The TE-PATH-BINDING TLV MUST be included with no Binding Value. The PCECC would allocate the binding label and further respond to Ingress PCC with PCUpd message as per [RFC8231] and MUST include the TE-PATH-BINDING TLV in a LSP object. The P flag in the LSP object would be set to 1 to indicate that the allocation is made by the PCE.

The PCE could allocate the binding label on its own accord for a PCE-Initiated (or delegated LSP). The allocated binding label needs to be informed to the PCC. The PCE would use the PCInitiate message [RFC8281] or PCUpd message [RFC8231] towards the PCC and MUST include the TE-PATH-BINDING TLV in the LSP object. The P flag in the LSP object would be set to 1 to indicate that the allocation is made by the PCE.

The PCECC capability MUST be exchanged on the PCEP session, before PCE could allocate binding label. Note that the CCI object is not used for binding allocation; this is done to maintain consistency with the rest of the binding label/SID procedures as per [I-D.sivabalan-pce-binding-label-sid].

6. PCEP messages

As defined in [RFC5440], a PCEP message consists of a common header followed by a variable-length body made of a set of objects that can be either mandatory or optional. An object is said to be mandatory in a PCEP message when the object must be included for the message to be considered valid. For each PCEP message type, a set of rules is defined that specify the set of objects that the message can carry. An implementation MUST form the PCEP messages using the object ordering specified in this document.

LSP-IDENTIFIERS TLV MUST be included in the LSP object for PCECC LSP.

6.1. The PCInitiate message

The PCInitiate message [RFC8281] can be used to download or remove the labels, the message has been extended as shown below -


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

Where:

<Common Header> is defined in [[RFC5440](#)]

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

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

```
<PCE-initiated-lsp-central-control> ::= <SRP>
                                         <LSP>
                                         <cci-list>
```

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

Where:

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

The LSP and SRP object is defined in [[RFC8231](#)].

When PCInitiate message is used for central controller's instructions (labels), the SRP, LSP and CCI objects MUST be present. The SRP object is defined in [[RFC8231](#)] and if the SRP object is missing, the receiving PCC MUST send a PCErr message with Error-type=6 (Mandatory Object missing) and Error-value=10 (SRP object missing). The LSP object is defined in [[RFC8231](#)] and if the LSP object is missing, the receiving PCC MUST send a PCErr message with Error-type=6 (Mandatory Object missing) and Error-value=8 (LSP object missing). The CCI object is defined in [Section 7.3](#) and if the CCI object is missing, the receiving PCC MUST send a PCErr message with Error-type=6 (Mandatory Object missing) and Error-value=TBD11 (CCI object missing). More than one CCI object MAY be included in the PCInitiate message for the transit LSR.

To cleanup the SRP object must set the R (remove) bit and include the LSP and the CCI object.

At max two instances of CCI object would be included in case of transit LSR to encode both in-coming and out-going label forwarding instructions. Other instances MUST be ignored.

6.2. The PCRpt message

The PCRpt message can be used to report the labels that were allocated by the PCE, to be used during the state synchronization phase.

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

Where:

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

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

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

```
<central-control-report> ::= [<SRP>]
                             <LSP>
                             <cci-list>
```

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

Where:

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

When PCRpt message is used to report the central controller's instructions (labels), the LSP and CCI objects MUST be present. The LSP object is defined in [\[RFC8231\]](#) and if the LSP object is missing, the receiving PCE MUST send a PCErr message with Error-type=6 (Mandatory Object missing) and Error-value=8 (LSP object missing). The CCI object is defined in [Section 7.3](#) and if the CCI object is missing, the receiving PCC MUST send a PCErr message with Error-type=6 (Mandatory Object missing) and Error-value=TBD11 (CCI object missing). Two CCI object can be included in the PCRpt message for the transit LSR.

7. PCEP Objects

The PCEP objects defined in this document are compliant with the PCEP object format defined in [\[RFC5440\]](#).

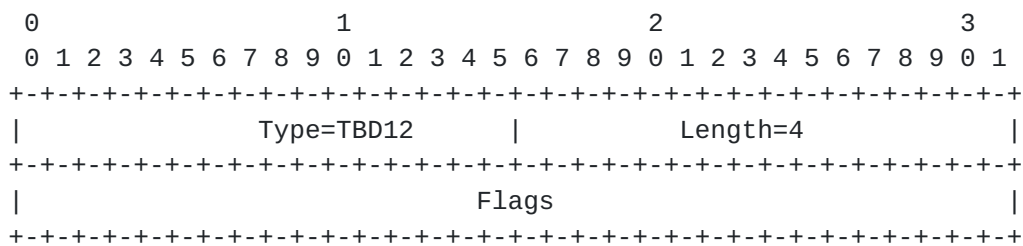
7.1. OPEN Object

This document defines a new optional TLVs for use in the OPEN Object.

7.1.1. PCECC Capability sub-TLV

The PCECC-CAPABILITY sub-TLV is an optional TLV for use in the OPEN Object for PCECC capability advertisement in PATH-SETUP-TYPE-CAPABILITY TLV. Advertisement of the PCECC capability implies support of LSPs that are setup through PCECC as per PCEP extensions defined in this document.

Its format is shown in the following figure:



The type of the TLV is TBD12 and it has a fixed length of 4 octets.

The value comprises a single field - Flags (32 bits).

No flags are assigned right now.

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

7.2. PATH-SETUP-TYPE TLV

The PATH-SETUP-TYPE TLV is defined in [[RFC8408](#)]; this document defines a new PST value:

- o PST = TBD1: Path is setup via PCECC mode.

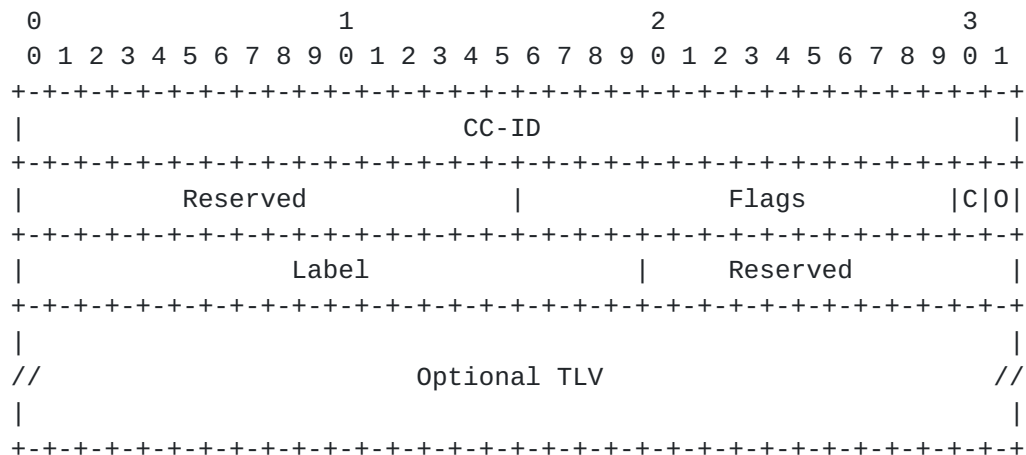
On a PCRpt/PCUpd/PCInitiate message, the PST=TBD1 in PATH-SETUP-TYPE TLV in SRP object indicates that this LSP was setup via a PCECC based mechanism.

7.3. CCI Object

The Central Control Instructions (CCI) Object is used by the PCE to specify the forwarding instructions (Label information in the context of this document) to the PCC, and MAY be carried within PCInitiate or PCRpt message for label download.

CCI Object-Class is TBD13.

CCI Object-Type is 1 for the MPLS Label.



The fields in the CCI object are as follows:

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

Flags: is used to carry any additional information pertaining to the CCI. Currently, the following flag bit is defined:

- * 0 bit(Out-label) : If the bit is set, it specifies the label is the OUT label and it is mandatory to encode the next-hop information (via IPV4-ADDRESS TLV or IPV6-ADDRESS TLV or UNNUMBERED-IPV4-ID-ADDRESS TLV in the CCI object). If the bit is not set, it specifies the label is the IN label and it is optional to encode the local interface information (via IPV4-ADDRESS TLV or IPV6-ADDRESS TLV or UNNUMBERED-IPV4-ID-ADDRESS TLV in the CCI object).

- * C-Bit (PCC Allocation): If the bit is set to 1, it indicates that the allocation needs to be done by the PCC for this central controller instruction. A PCE set this bit to request the PCC to make an allocation from its label space. A PCC would set this bit to indicate that it has allocated the CC-ID and report it to the PCE.

Label (20-bit): The Label information.

Reserved (12 bit): Set to zero while sending, ignored on receive.

[7.3.1.](#) Address TLVs

This document defines the following TLVs for the CCI object to associate the next-hop information in case of an outgoing label and local interface information in case of an incoming label.

IPV4-ADDRESS TLV:

```

0                               1                               2                               3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                               Type=TBD14                               | Length = 4
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                               IPv4 address                               |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

IPV6-ADDRESS TLV:

```

0                               1                               2                               3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                               Type=TBD15                               | Length = 16
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                               |                                         |
//                               IPv6 address (16 bytes)                               //
|                               |                                         |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

UNNUMBERED-IPV4-ID-ADDRESS TLV:

```

0                               1                               2                               3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                               Type=TBD16                               | Length = 8
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                               Node-ID                               |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```



```

+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                                     Interface ID                                     |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

LINKLOCAL-IPV6-ID-ADDRESS TLV:

      0               1               2               3
      0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                                     Type=TBD17                                     | Length = 40 |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
//                                     Local IPv6 address (16 octets)                                     //
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                                     Local Interface ID                                     |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
//                                     Remote IPv6 address (16 octets)                                     //
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                                     Remote Interface ID                                     |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

The address TLVs are as follows:

IPV4-ADDRESS TLV: an IPv4 address.

IPV6-ADDRESS TLV: an IPv6 address.

UNNUMBERED-IPV4-ID-ADDRESS TLV: a Node ID / Interface ID tuple.

LINKLOCAL-IPV6-ID-ADDRESS TLV: a pair of (global IPv6 address,
interface ID) tuples.

8. Implementation Status

[Note to the RFC Editor - remove this section before publication, as well as remove the reference to [RFC 7942](#).]

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

features. Readers are advised to note that other implementations may exist.

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

8.1. Huawei's Proof of Concept based on ONOS

The PCE function was developed in the ONOS open source platform. This extension was implemented on a private version as a proof of concept for PCECC.

- o Organization: Huawei
- o Implementation: Huawei's PoC based on ONOS
- o Description: PCEP as a southbound plugin was added to ONOS. To support PCECC, an earlier version of this I-D was implemented. Refer <https://wiki.onosproject.org/display/ONOS/PCEP+Protocol>
- o Maturity Level: Prototype
- o Coverage: Partial
- o Contact: satishk@huawei.com

9. Security Considerations

The security considerations described in [RFC8231] and [RFC8281] apply to the extensions described in this document. Additional considerations related to a malicious PCE are introduced.

9.1. Malicious PCE

PCE has complete control over PCC to update the labels and can cause the LSP's to behave inappropriate and cause major impact to the network. As a general precaution, it is RECOMMENDED that these PCEP extensions only be activated on authenticated and encrypted sessions across PCEs and PCCs belonging to the same administrative authority, using Transport Layer Security (TLS) [RFC8253], as per the recommendations and best current practices in [RFC7525].

10. Manageability Considerations

10.1. Control of Function and Policy

A PCE or PCC implementation SHOULD allow to configure to enable/disable PCECC capability as a global configuration.

10.2. Information and Data Models

[RFC7420] describes the PCEP MIB, this MIB can be extended to get the PCECC capability status.

The PCEP YANG module [[I-D.ietf-pce-pcep-yang](#)] could be extended to enable/disable PCECC capability.

10.3. Liveness Detection and Monitoring

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

10.4. Verify Correct Operations

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

10.5. Requirements On Other Protocols

PCEP extensions defined in this document do not put new requirements on other protocols.

10.6. Impact On Network Operations

PCEP extensions defined in this document do not put new requirements on network operations.

11. IANA Considerations

11.1. PCEP TLV Type Indicators

IANA is requested to allocate the following TLV Type Indicator values within the "PCEP TLV Type Indicators" sub- registry of the PCEP Numbers registry:

Value	Meaning	Reference
TBD14	IPV4-ADDRESS TLV	This document
TBD15	IPV6-ADDRESS TLV	This document
TBD16	UNNUMBERED-IPV4-ID-ADDRESS TLV	This document
TBD17	LINKLOCAL-IPV6-ID-ADDRESS TLV	This document

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

[RFC8408] requested creation of "PATH-SETUP- TYPE-CAPABILITY Sub-TLV Type Indicators" sub-registry. Further IANA is requested to allocate the following code-point:

Value	Meaning	Reference
TBD12	PCECC-CAPABILITY	This document

11.3. New Path Setup Type Registry

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

Value	Description	Reference
TBD1	Traffic engineering path is setup using PCECC mode	This document

11.4. PCEP Object

IANA is requested to allocate new code-point in the "PCEP Objects" sub-registry for the CCI object as follows:

Object-Class	Value	Name	Reference
TBD13		CCI Object-Type	This document
	0		Reserved
	1		MPLS Label

11.5. CCI Object Flag Field

IANA is requested to create a new sub-registry to manage the Flag field of the CCI object called "CCI Object Flag Field". New values are to be assigned by Standards Action [[RFC8126](#)]. Each bit should be tracked with the following qualities:

- o Bit number (counting from bit 0 as the most significant bit)
- o Capability description
- o Defining RFC

Two bits to be defined for the CCI Object flag field in this document as follows:

Bit	Description	Reference
0-13	Unassigned	This document
14	C Bit - PCC allocation	This document
15	O Bit - Specifies label is out-label	This document

[11.6.](#) PCEP-Error Object

IANA is requested to allocate new error types and error values within the "PCEP-ERROR Object Error Types and Values" sub-registry of the PCEP Numbers registry for the following errors:

Error-Type	Meaning	
-----	-----	
10	Reception of an invalid object.	
	Error-value = TBD2 :	Missing PCECC Capability sub-TLV
19	Invalid operation.	
	Error-value = TBD3 :	Attempted PCECC operations when PCECC capability was not advertised
	Error-value = TBD4 :	Stateful PCE capability was not advertised
	Error-value = TBD8 :	Unknown Label
6	Mandatory Object missing.	
	Error-value = TBD11 :	CCI object missing
TBD5	PCECC failure.	
	Error-value = TBD6 :	Label out of range.
	Error-value = TBD7 :	Instruction failed.
	Error-value = TBD9 :	Invalid CCI.
	Error-value = TBD10 :	Unable to allocate the specified CCI.

[12.](#) Acknowledgments

We would like to thank Robert Tao, Changjing Yan, Tieying Huang and Avantika for their useful comments and suggestions.

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, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC5440] Vasseur, JP., Ed. and JL. Le Roux, Ed., "Path Computation Element (PCE) Communication Protocol (PCEP)", [RFC 5440](#), DOI 10.17487/RFC5440, March 2009, <<https://www.rfc-editor.org/info/rfc5440>>.
- [RFC7420] Koushik, A., Stephan, E., Zhao, Q., King, D., and J. Hardwick, "Path Computation Element Communication Protocol (PCEP) Management Information Base (MIB) Module", [RFC 7420](#), DOI 10.17487/RFC7420, December 2014, <<https://www.rfc-editor.org/info/rfc7420>>.
- [RFC7525] Sheffer, Y., Holz, R., and P. Saint-Andre, "Recommendations for Secure Use of Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS)", [BCP 195](#), [RFC 7525](#), DOI 10.17487/RFC7525, May 2015, <<https://www.rfc-editor.org/info/rfc7525>>.
- [RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", [BCP 26](#), [RFC 8126](#), DOI 10.17487/RFC8126, June 2017, <<https://www.rfc-editor.org/info/rfc8126>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in [RFC 2119](#) Key Words", [BCP 14](#), [RFC 8174](#), DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [RFC8231] Crabbe, E., Minei, I., Medved, J., and R. Varga, "Path Computation Element Communication Protocol (PCEP) Extensions for Stateful PCE", [RFC 8231](#), DOI 10.17487/RFC8231, September 2017, <<https://www.rfc-editor.org/info/rfc8231>>.
- [RFC8233] Dhody, D., Wu, Q., Manral, V., Ali, Z., and K. Kumaki, "Extensions to the Path Computation Element Communication Protocol (PCEP) to Compute Service-Aware Label Switched Paths (LSPs)", [RFC 8233](#), DOI 10.17487/RFC8233, September 2017, <<https://www.rfc-editor.org/info/rfc8233>>.

- [RFC8281] Crabbe, E., Minei, I., Sivabalan, S., and R. Varga, "Path Computation Element Communication Protocol (PCEP) Extensions for PCE-Initiated LSP Setup in a Stateful PCE Model", [RFC 8281](#), DOI 10.17487/RFC8281, December 2017, <<https://www.rfc-editor.org/info/rfc8281>>.
- [RFC8408] Sivabalan, S., Tantsura, J., Minei, I., Varga, R., and J. Hardwick, "Conveying Path Setup Type in PCE Communication Protocol (PCEP) Messages", [RFC 8408](#), DOI 10.17487/RFC8408, July 2018, <<https://www.rfc-editor.org/info/rfc8408>>.

13.2. Informative References

- [RFC4655] Farrel, A., Vasseur, J., and J. Ash, "A Path Computation Element (PCE)-Based Architecture", [RFC 4655](#), DOI 10.17487/RFC4655, August 2006, <<https://www.rfc-editor.org/info/rfc4655>>.
- [RFC7025] Otani, T., Ogaki, K., Caviglia, D., Zhang, F., and C. Margaria, "Requirements for GMPLS Applications of PCE", [RFC 7025](#), DOI 10.17487/RFC7025, September 2013, <<https://www.rfc-editor.org/info/rfc7025>>.
- [RFC7399] Farrel, A. and D. King, "Unanswered Questions in the Path Computation Element Architecture", [RFC 7399](#), DOI 10.17487/RFC7399, October 2014, <<https://www.rfc-editor.org/info/rfc7399>>.
- [RFC7491] King, D. and A. Farrel, "A PCE-Based Architecture for Application-Based Network Operations", [RFC 7491](#), DOI 10.17487/RFC7491, March 2015, <<https://www.rfc-editor.org/info/rfc7491>>.
- [RFC7942] Sheffer, Y. and A. Farrel, "Improving Awareness of Running Code: The Implementation Status Section", [BCP 205](#), [RFC 7942](#), DOI 10.17487/RFC7942, July 2016, <<https://www.rfc-editor.org/info/rfc7942>>.
- [RFC8253] Lopez, D., Gonzalez de Dios, O., Wu, Q., and D. Dhody, "PCEPS: Usage of TLS to Provide a Secure Transport for the Path Computation Element Communication Protocol (PCEP)", [RFC 8253](#), DOI 10.17487/RFC8253, October 2017, <<https://www.rfc-editor.org/info/rfc8253>>.

- [RFC8283] Farrel, A., Ed., Zhao, Q., Ed., Li, Z., and C. Zhou, "An Architecture for Use of PCE and the PCE Communication Protocol (PCEP) in a Network with Central Control", [RFC 8283](#), DOI 10.17487/RFC8283, December 2017, <<https://www.rfc-editor.org/info/rfc8283>>.
- [I-D.ietf-teas-pcecc-use-cases]
Zhao, Q., Li, Z., Khasanov, B., Dhody, D., Ke, Z., Fang, L., Zhou, C., Communications, T., Rachitskiy, A., and A. Gulida, "The Use Cases for Path Computation Element (PCE) as a Central Controller (PCECC).", [draft-ietf-teas-pcecc-use-cases-05](#) (work in progress), March 2020.
- [I-D.ietf-pce-pcep-yang]
Dhody, D., Hardwick, J., Beeram, V., and J. Tantsura, "A YANG Data Model for Path Computation Element Communications Protocol (PCEP)", [draft-ietf-pce-pcep-yang-13](#) (work in progress), October 2019.
- [I-D.zhao-pce-pcep-extension-pce-controller-sr]
Zhao, Q., Li, Z., Negi, M., Peng, S., and C. Zhou, "PCEP Procedures and Protocol Extensions for Using PCE as a Central Controller (PCECC) of SR-LSPs", [draft-zhao-pce-pcep-extension-pce-controller-sr-06](#) (work in progress), March 2020.
- [I-D.li-pce-controlled-id-space]
Li, C., Chen, M., Dong, J., Li, Z., Wang, A., Cheng, W., and C. Zhou, "PCE Controlled ID Space", [draft-li-pce-controlled-id-space-04](#) (work in progress), January 2020.
- [I-D.sivabalan-pce-binding-label-sid]
Sivabalan, S., Filsfils, C., Tantsura, J., Hardwick, J., Previdi, S., and C. Li, "Carrying Binding Label/Segment-ID in PCE-based Networks.", [draft-sivabalan-pce-binding-label-sid-07](#) (work in progress), July 2019.
- [I-D.li-pce-sr-path-segment]
Li, C., Chen, M., Cheng, W., Dong, J., Li, Z., Gandhi, R., and Q. Xiong, "Path Computation Element Communication Protocol (PCEP) Extension for Path Segment in Segment Routing (SR)", [draft-li-pce-sr-path-segment-08](#) (work in progress), August 2019.

Appendix A. Contributor Addresses

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

EMail: dhruv.ietf@gmail.com

Satish Karunanithi
Huawei Technologies
Divyashree Techno Park, Whitefield
Bangalore, Karnataka 560066
India

EMail: satishk@huawei.com

Adrian Farrel
Juniper Networks, Inc
UK

EMail: adrian@olddog.co.uk

Xuesong Geng
Huawei Technologies
China

Email: gengxuesong@huawei.com

Udayasree Palle

EMail: udayasreereddy@gmail.com

Katherine Zhao
Futurewei Technologies

EMail: katherine.zhao@futurewei.com

Boris Zhang
Telus Ltd.
Toronto
Canada

EMail: boris.zhang@telus.com

Alex Tokar
Cisco Systems

Slovak Republic

EMail: atokar@cisco.com

Authors' Addresses

Quintin Zhao
Huawei Technologies
125 Nagog Technology Park
Acton, MA 01719
USA

EMail: quintin.zhao@huawei.com

Zhenbin Li
Huawei Technologies
Huawei Bld., No.156 Beiqing Rd.
Beijing 100095
China

EMail: lizhenbin@huawei.com

Mahendra Singh Negi
RtBrick India
N-17L, Floor-1, 18th Cross Rd, HSR Layout Sector-3
Bangalore, Karnataka 560102
India

EMail: mahend.ietf@gmail.com

Shuping Peng
Huawei Technologies
Huawei Bld., No.156 Beiqing Rd.
Beijing 100095
China

EMail: pengshuping@huawei.com

Chao Zhou
Cisco Systems

EMail: chao.zhou@cisco.com

