

---

Stream: Internet Engineering Task Force (IETF)  
RFC: [9050](#)  
Category: Standards Track  
Published: July 2021  
ISSN: 2070-1721

Authors:

Z. Li	S. Peng	M. Negi	Q. Zhao	C. Zhou
<i>Huawei Technologies</i>	<i>Huawei Technologies</i>	<i>RtBrick Inc</i>	<i>Etheric Networks</i>	<i>HPE</i>

## RFC 9050

# Path Computation Element Communication Protocol (PCEP) Procedures and Extensions for Using the PCE as a Central Controller (PCECC) of LSPs

---

### Abstract

The Path Computation Element (PCE) is a core component of Software-Defined Networking (SDN) systems.

A PCE as a 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 Label Switched Path (LSP) can be calculated/set up/initiated and the label-forwarding entries can also be downloaded through a centralized PCE server to each network device along the path while leveraging the existing PCE technologies as much as possible.

This document specifies the procedures and Path Computation Element Communication Protocol (PCEP) extensions for using the PCE as the central controller for provisioning labels along the path of the static LSP.

### Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in Section 2 of RFC 7841.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at <https://www.rfc-editor.org/info/rfc9050>.

## Copyright Notice

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

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

## Table of Contents

1. Introduction
2. Terminology
  - 2.1. Requirements Language
3. Basic PCECC Mode
4. PCEP Requirements
5. Procedures for Using the PCE as a Central Controller (PCECC)
  - 5.1. Stateful PCE Model
  - 5.2. New LSP Functions
  - 5.3. New PCEP Object
  - 5.4. PCECC Capability Advertisement
  - 5.5. LSP Operations
    - 5.5.1. PCE-Initiated PCECC LSP
    - 5.5.2. PCC-Initiated PCECC LSP
    - 5.5.3. Central Controller Instructions
      - 5.5.3.1. Label Download CCI
      - 5.5.3.2. Label Cleanup CCI
    - 5.5.4. PCECC LSP Update
    - 5.5.5. Re-delegation and Cleanup
    - 5.5.6. Synchronization of Central Controller Instructions
    - 5.5.7. PCECC LSP State Report

### 5.5.8. PCC-Based Allocations

## 6. PCEP Messages

### 6.1. The PCInitiate Message

### 6.2. The PCRpt Message

## 7. PCEP Objects

### 7.1. OPEN Object

#### 7.1.1. PCECC Capability Sub-TLV

### 7.2. PATH-SETUP-TYPE TLV

### 7.3. CCI Object

#### 7.3.1. Address TLVs

## 8. Security Considerations

### 8.1. Malicious PCE

### 8.2. Malicious PCC

## 9. Manageability Considerations

### 9.1. Control of Function and Policy

### 9.2. Information and Data Models

### 9.3. Liveness Detection and Monitoring

### 9.4. Verify Correct Operations

### 9.5. Requirements on Other Protocols

### 9.6. Impact on Network Operations

## 10. IANA Considerations

### 10.1. PATH-SETUP-TYPE-CAPABILITY Sub-TLV Type Indicators

### 10.2. PCECC-CAPABILITY Sub-TLV's Flag Field

### 10.3. PCEP Path Setup Type Registry

### 10.4. PCEP Object

### 10.5. CCI Object Flag Field

### 10.6. PCEP-Error Object

## 11. References

### 11.1. Normative References

### 11.2. Informative References

[Acknowledgments](#)

[Contributors](#)

[Authors' Addresses](#)

## 1. Introduction

The Path Computation Element (PCE) [[RFC4655](#)] was developed to offload the path computation function from routers in an MPLS traffic-engineered (TE) network. 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. Since then, the role and function of the PCE have 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](#)].

The PCE was developed to derive paths for MPLS LSPs, which are supplied to the head end of the LSP using the PCEP. But SDN has a broader applicability than signaled MPLS and GMPLS 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.

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

A 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/set up/initiated and the label-forwarding entries can also be downloaded through a centralized PCE server to each network device along the path while leveraging the existing PCE technologies as much as possible.

This document specifies the procedures and PCEP 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.

While this document is focused on the procedures for the static LSPs (referred to as the basic PCECC mode in [Section 3](#)), the mechanisms and protocol encodings are specified in such a way that extensions for other use cases are easy to achieve. For example, the extensions for the PCECC for Segment Routing (SR) are specified in [\[PCECC-SR\]](#) and [\[PCECC-SRv6\]](#).

## 2. Terminology

The terminology used in this document is the same as that described in the [\[RFC8283\]](#).

### 2.1. 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.

## 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.

[\[RFC8283\]](#) examines the motivations and applicability for the PCECC and use of PCEP as an SBI. [Section 3.1.2](#) of [\[RFC8283\]](#) highlights the use of the PCECC for label allocation along the static LSPs, and it simplifies the processing of a distributed control plane by blending it with elements of SDN and without necessarily completely replacing it. This allows the operator to introduce the advantages of SDN (such as programmability) into the network. Further, [Section 3.3](#) of [\[PCECC\]](#) describes some of the scenarios where the PCECC technique could be useful. [Section 4](#) of [\[RFC8283\]](#) also describes the implications on the protocol when used as an SDN SBI. The operator needs to evaluate the advantages offered by the PCECC against the operational and scalability needs of the PCECC.

As per [Section 3.1.2](#) of [\[RFC8283\]](#), 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. The PCC **MUST NOT** make allocations from the label space set aside for the PCE to avoid overlap and collisions of label allocations. It is **RECOMMENDED** that the PCE makes allocations (from the label space set aside for the PCE) for all nodes along the path. For the purpose of this document, it is assumed that the exclusive label range to be used by a PCE is known and set on both PCEP peers. A future extension could add the capability to advertise this range via a possible PCEP extension as well (see [\[PCE-ID\]](#)). The rest of the processing is similar to the existing stateful PCE mechanism.

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

## 4. PCEP Requirements

The following key requirements should be considered when designing the PCECC-based solution:

1. A PCEP speaker supporting this document needs to have the capability to advertise its PCECC capability to its peers.
2. A PCEP speaker needs means to identify PCECC-based LSPs in the PCEP messages.
3. PCEP procedures need to allow for PCC-based label allocations.
4. PCEP procedures need to provide a means to update (or clean up) label entries downloaded to the PCC.
5. PCEP procedures need to provide a means to synchronize the labels between the PCE and the PCC via PCEP messages.

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

### 5.1. Stateful PCE Model

Active stateful PCE is described in [\[RFC8231\]](#). A PCE as a Central Controller (PCECC) reuses the existing active stateful PCE mechanism as much as possible to control LSPs.

### 5.2. New LSP Functions

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

PCInitiate:

A PCEP message described in [RFC8281]. A PCInitiate message is used to set up a PCE-initiated LSP based on a PCECC mechanism. It is also extended for Central Controller Instructions (CCI) (download or clean up the label-forwarding instructions in the context of this document) on all nodes along the path, as described in Section 6.1.

PCRpt: A PCEP message described in [RFC8231]. A PCRpt message is used to send the PCECC LSP Reports. It is also extended to report the set of CCI (label-forwarding instructions in the context of this document) received from the PCE, as described in Section 6.2. Section 5.5.6 describes the use of a PCRpt message during synchronization.

PCUpd: A PCEP message described in [RFC8231]. A PCUpd message is used to send the PCECC LSP Updates.

The new functions defined in this document are mapped onto the PCEP messages, as shown in Table 1.

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

Table 1: Functions Mapped to the PCEP Messages

### 5.3. New PCEP Object

This document defines a new PCEP object called CCI (Section 7.3) to specify the Central Controller Instructions. In the scope of this document, this is limited to label-forwarding instructions. Future documents can create new CCI object-types for other types of Central Controller Instructions. The CC-ID is the unique identifier for the CCI in PCEP. The PCEP messages are extended in this document to handle the PCECC operations.

## 5.4. PCECC Capability Advertisement

During the PCEP initialization phase, PCEP speakers (PCE or PCC) advertise their support of and willingness to use PCEP extensions for the PCECC using these elements in the OPEN message:

- a new Path Setup Type (PST) ([Section 7.2](#)) in the PATH-SETUP-TYPE-CAPABILITY TLV to indicate support for PCEP extensions for the PCECC - 2 (Traffic engineering path is set up using PCECC mode)
- a new PCECC-CAPABILITY sub-TLV ([Section 7.1.1](#)) with the L bit set to '1' inside the PATH-SETUP-TYPE-CAPABILITY TLV to indicate a willingness to use PCEP extensions for the PCECC-based Central Controller Instructions for label download
- the STATEFUL-PCE-CAPABILITY TLV [[RFC8231](#)] (with the I flag set [[RFC8281](#)])

The new PST is to be listed in the PATH-SETUP-TYPE-CAPABILITY TLV by all PCEP speakers that support the PCEP extensions for the PCECC in this document.

The new PCECC-CAPABILITY sub-TLV is included in the PATH-SETUP-TYPE-CAPABILITY TLV in the OPEN object to indicate a willingness to use the PCEP extensions for the PCECC during the established PCEP session. Using the L bit in this TLV, the PCE shows the intention to function as a PCECC server, and the PCC shows a willingness to act as a PCECC client for label download instructions (see [Section 7.1.1](#)).

If the PCECC-CAPABILITY sub-TLV is advertised and the STATEFUL-PCE-CAPABILITY TLV is not advertised, or is advertised without the I flag set, in the OPEN object, the receiver **MUST**:

- send a PCErr message with Error-Type=19 (Invalid Operation) and Error-value=17 (Stateful PCE capability was not advertised) and
- terminate the session.

If a PCEP speaker receives the PATH-SETUP-TYPE-CAPABILITY TLV with the PCECC PST but without the PCECC-CAPABILITY sub-TLV, it **MUST**:

- send a PCErr message with Error-Type=10 (Reception of an invalid object) and Error-value=33 (Missing PCECC Capability sub-TLV) and
- terminate the PCEP session.

The PCECC-CAPABILITY sub-TLV **MUST NOT** be used without the corresponding PST being listed in the PATH-SETUP-TYPE-CAPABILITY TLV. If it is present without the corresponding PST listed in the PATH-SETUP-TYPE-CAPABILITY TLV, it **MUST** be ignored.



If one or both speakers (PCE and PCC) have not indicated support and willingness to use the PCEP extensions for the PCECC, the PCEP extensions for the PCECC **MUST NOT** be used. If a PCECC operation is attempted when both speakers have not agreed in the OPEN messages, the receiver of the message **MUST**:

- send a PCErr message with Error-Type=19 (Invalid Operation) and Error-value=16 (Attempted PCECC operations when PCECC capability was not advertised) and
- terminate the PCEP session.

A legacy PCEP speaker (that does not recognize the PCECC Capability sub-TLV) will ignore the sub-TLV in accordance with [RFC8408] and [RFC5440]. As per [RFC8408], the legacy PCEP speaker, on receipt of an unsupported PST in a Request Parameter (RP) / Stateful PCE Request Parameter (SRP) object, will:

- send a PCErr message with Error-Type=21 (Invalid traffic engineering path setup type) and Error-value=1 (Unsupported path setup type) and
- terminate the PCEP session.

## 5.5. LSP Operations

The PCEP messages pertaining to a PCECC **MUST** include the PATH-SETUP-TYPE TLV [RFC8408] in the SRP object [RFC8231] with the PST set to '2' to clearly identify that the PCECC LSP is intended.

### 5.5.1. PCE-Initiated PCECC LSP

The LSP instantiation operation is defined in [RFC8281]. In order to set up a PCE-initiated LSP based on the PCECC mechanism, a PCE sends a PCInitiate message with the PST set to '2' for the PCECC (see Section 7.2) to the ingress PCC.

The label-forwarding instructions (see Section 5.5.3) from the PCECC are sent after the initial PCInitiate and PCRpt message exchange with the ingress PCC, as per [RFC8281] (see Figure 1). This is done so that the PCEP-specific identifier for the LSP (PLSP-ID) and other LSP identifiers can be obtained from the ingress and can be included in the label-forwarding instruction in the next set of PCInitiate messages along the path, as described below.

An LSP-IDENTIFIERS TLV [RFC8231] **MUST** be included for the PCECC LSPs; it uniquely identifies the LSP in the network. Note that the fields in the LSP-IDENTIFIERS TLV are described for the RSVP-signaled LSPs but are applicable to the PCECC LSP as well. The LSP object is included in the CCI (label download Section 7.3) to identify the PCECC LSP for this instruction. The PLSP-ID is the original identifier used by the ingress PCC, so a transit/egress Label Switching Router (LSR) could have multiple Central Controller Instructions that have the same PLSP-ID. The PLSP-ID in combination with the source (in the LSP-IDENTIFIERS TLV) **MUST** be unique. The PLSP-ID is included for maintainability reasons to ease debugging. As per [RFC8281], the LSP object could also include the SPEAKER-ENTITY-ID TLV to identify the PCE that initiated these instructions. Also, the CC-ID is unique in each PCEP session, as described in Section 7.3.

On receipt of a PCInitiate message for the PCECC LSP, the PCC responds with a PCRpt message with the status set to 'Going-up' and carrying the assigned PLSP-ID (see [Figure 1](#)). The ingress PCC also sets the D (Delegate) flag (see [\[RFC8231\]](#)) and C (Create) flag (see [\[RFC8281\]](#)) in the LSP object. When the PCE receives this PCRpt message with the PLSP-ID, it assigns labels along the path and sets up the path by sending a PCInitiate message to each node along the path of the LSP, as per the PCECC technique. The CC-ID uniquely identifies the Central Controller Instructions within a PCEP session. Each node along the path (PCC) responds with a PCRpt message to acknowledge the CCI with the PCRpt messages including the CCI and LSP objects.

The ingress node would receive one CCI object with the O bit (out-label) set. The transit node(s) would receive two CCI objects with the in-label CCI without the O bit set and the out-label CCI with the O bit set. The egress node would receive one CCI object without the O bit set (see [Figure 1](#)). A node can determine its role based on the setting of the O bit in the CCI object(s) and the LSP-IDENTIFIERS TLV in the LSP object.

The LSP deletion operation for the PCE-initiated PCECC LSP is the same as defined in [\[RFC8281\]](#). The PCE should further perform the label entry cleanup operation, as described in [Section 5.5.3.2](#), for the corresponding LSP.

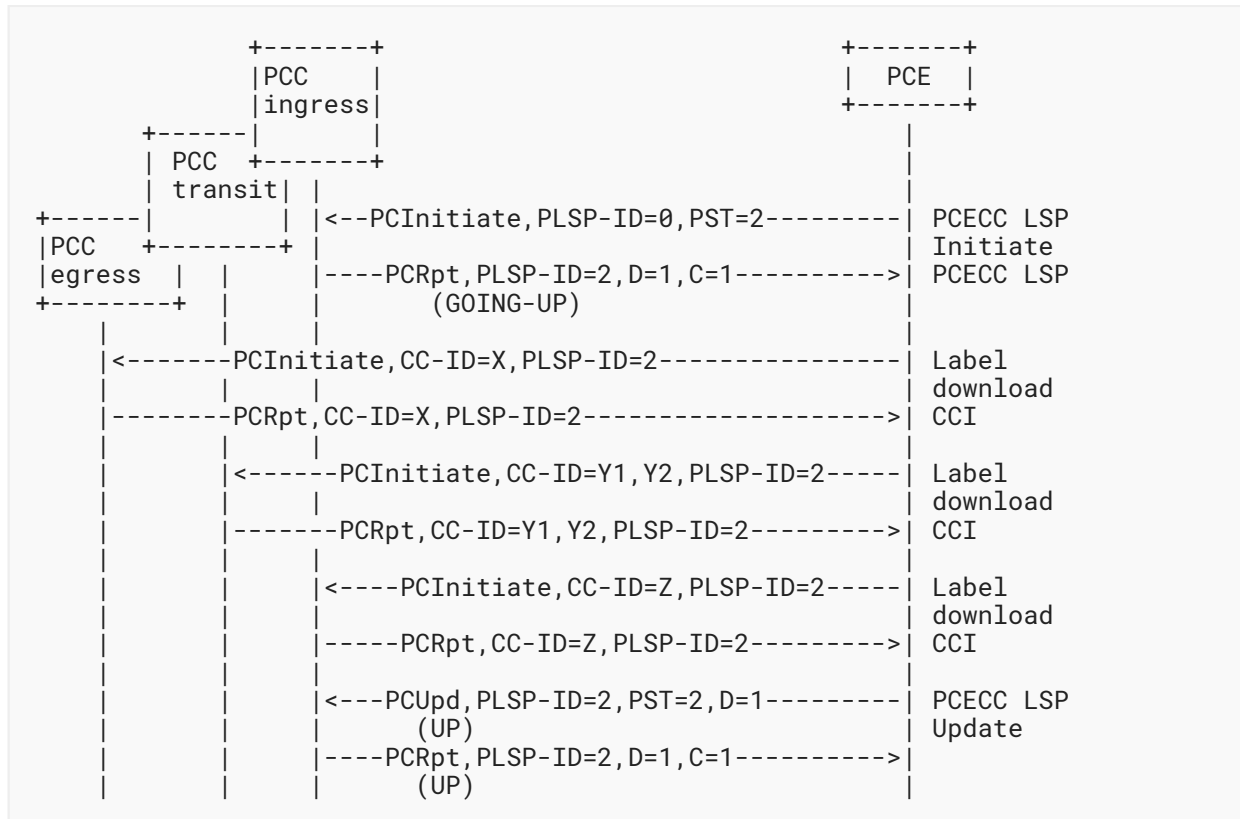


Figure 1: PCE-Initiated PCECC LSP

Once the label operations are completed, the PCE **MUST** send a PCUpd message to the ingress PCC. As per [\[RFC8231\]](#), the PCUpd message is with the D flag set.

The PCECC LSPs are considered to be 'up' by default (on receipt of a PCUpd message from the PCE). The ingress could further choose to deploy a data-plane check mechanism and report the status back to the PCE via a PCRpt message to make sure that the correct label instructions are made along the path of the PCECC LSP (and it is ready to carry traffic). The exact mechanism is out of scope of this document.

In the case where the label allocations are made by the PCC itself (see [Section 5.5.8](#)), the PCE could request an allocation to be made by the PCC; then, the PCC would send a PCRpt message with the allocated label encoded in the CC-ID object (as shown in [Figure 2](#)) in the configuration sequence from the egress towards the ingress along the path.

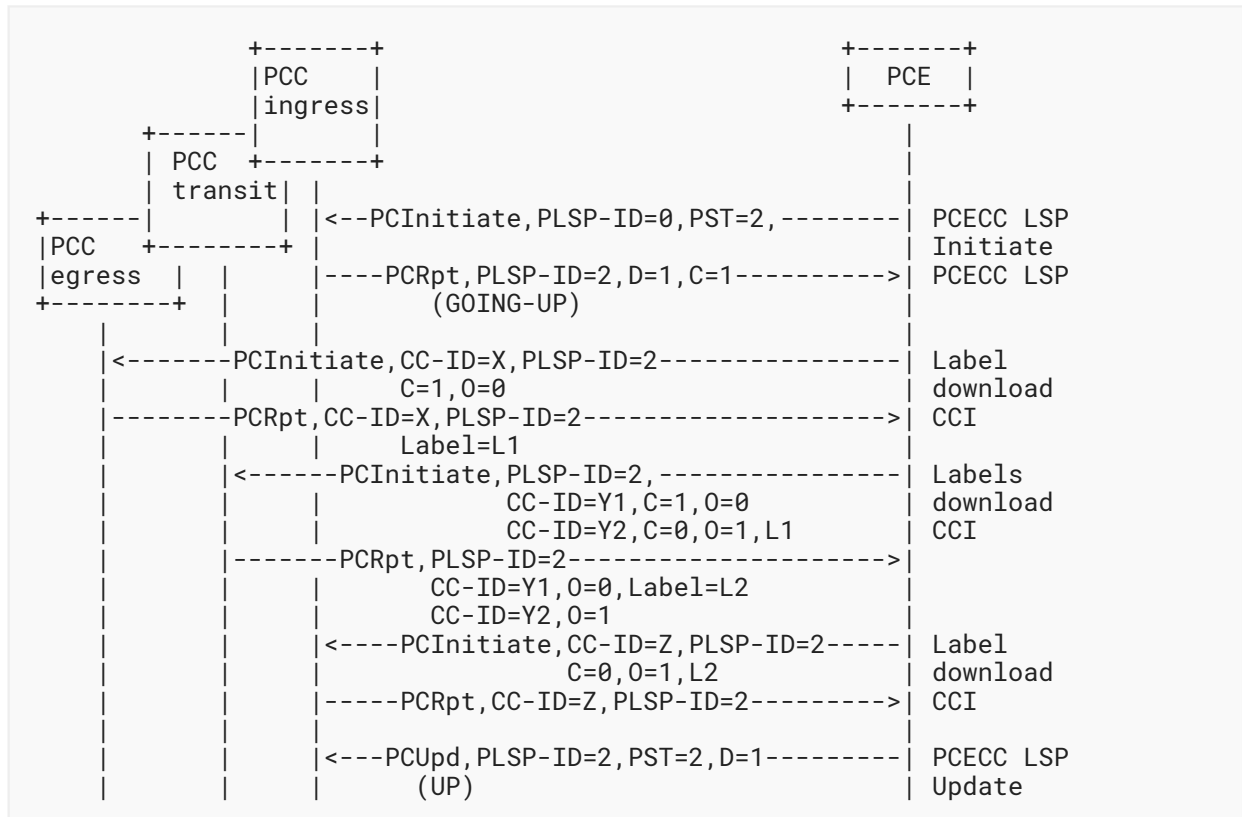


Figure 2: PCE-Initiated PCECC LSP (PCC Allocation)

In this example, it should be noted that the request is made to the egress node with the C bit set in the CCI object to indicate that the label allocation needs to be done by the egress, and the egress responds with the allocated label to the PCE. The PCE further informs the transit PCC without setting the C bit to '1' in the CCI object for the out-label, but the C bit is set to '1' for the in-label, so the transit node makes the label allocation (for the in-label) and reports to the PCE. Similarly, the C bit is unset towards the ingress to complete all the label allocations for the PCECC LSP.

### 5.5.2. PCC-Initiated PCECC LSP

In order to set up an LSP based on the PCECC mechanism where the LSP is configured at the PCC, a PCC **MUST** delegate the LSP by sending a PCRpt message with the PST set for the PCECC (see [Section 7.2](#)) and D (Delegate) flag (see [\[RFC8231\]](#)) set in the LSP object (see [Figure 3](#)).

When a PCE receives the initial PCRpt message with the D flag and PST set to '2', it **SHOULD** calculate the path and assign labels along the path in addition to setting up the path by sending a PCInitiate message to each node along the path of the LSP, as per the PCECC technique (see [Figure 3](#)). The CC-ID uniquely identifies the CCI within a PCEP session. Each PCC further responds with the PCRpt messages, including the CCI and LSP objects.

Once the CCI (label operations) are completed, the PCE **MUST** send the PCUpd message to the ingress PCC. As per [\[RFC8231\]](#), this PCUpd message should include the path information calculated by the PCE.

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

The LSP deletion operation for the PCECC LSPs is the same as defined in [\[RFC8231\]](#). If the PCE receives a PCRpt message for LSP deletion, then it does label the cleanup operation, as described in [Section 5.5.3.2](#), for the corresponding LSP.

The basic PCECC LSP setup sequence is as shown in [Figure 3](#).

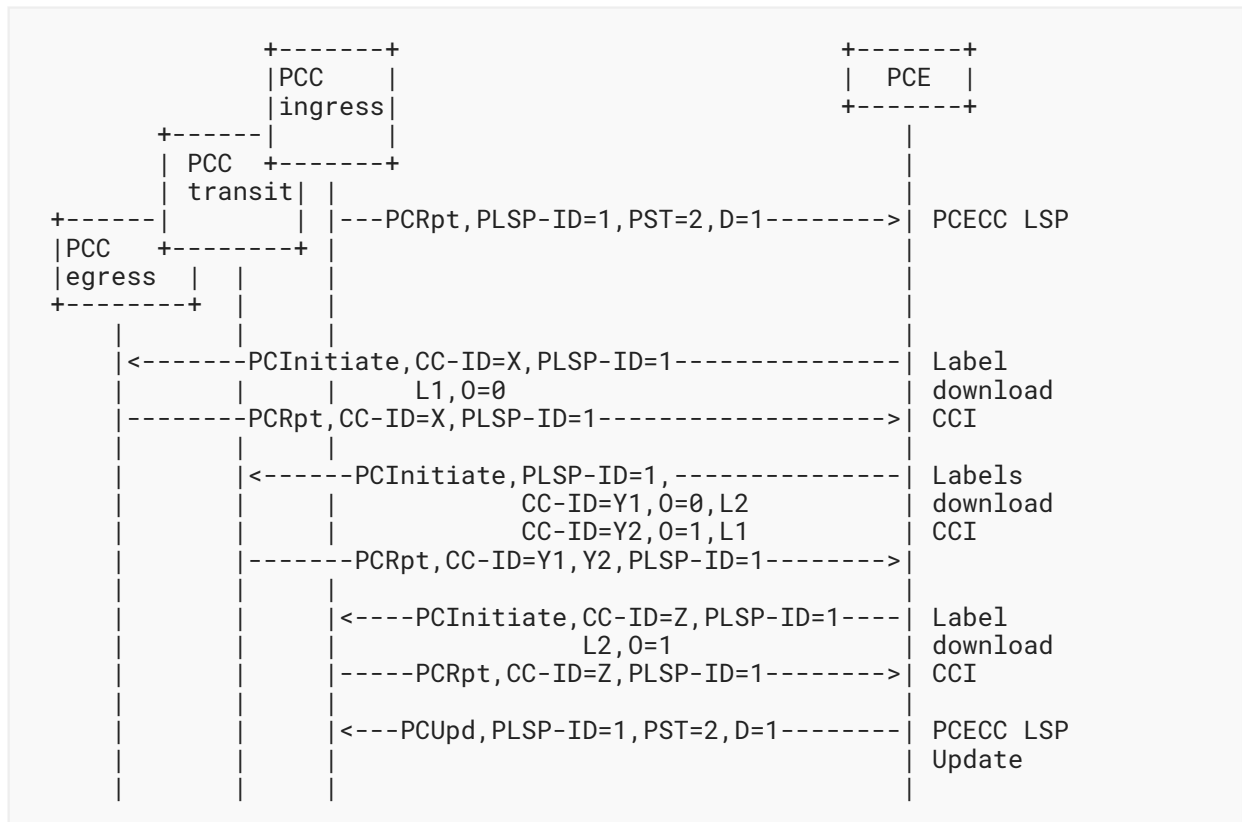


Figure 3: PCC-Initiated PCECC LSP

In the case where the label allocations are made by the PCC itself (see [Section 5.5.8](#)), the PCE could request an allocation to be made by the PCC; then, the PCC would send a PCRpt message with the allocated label encoded in the CC-ID object, as shown in [Figure 4](#).

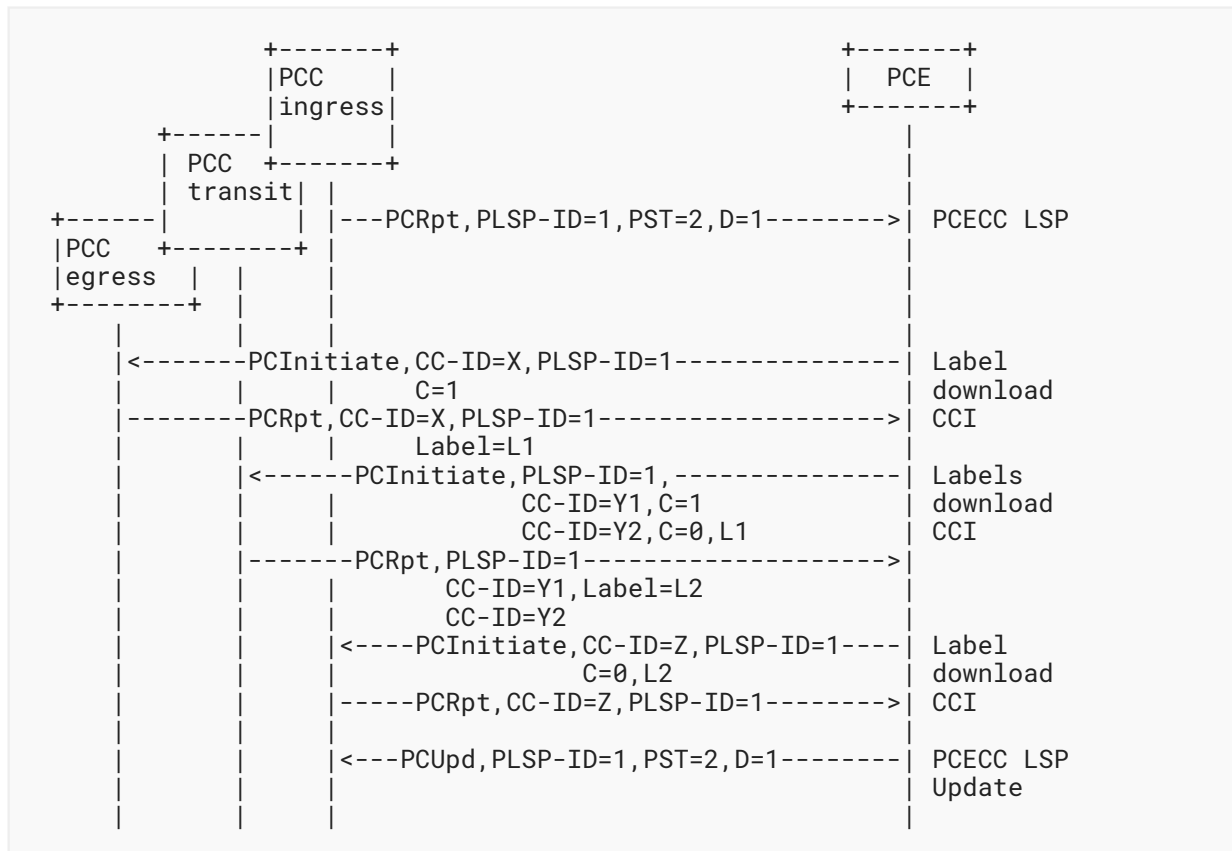


Figure 4: PCC-Initiated PCECC LSP (PCC Allocation)

Note:

The O bit is set as before (and thus not included).

In the case where the label allocations are made by the PCC itself (see Section 5.5.8), the procedure remains the same, with just an additional constraint on the configuration sequence.

The rest of the PCC-initiated PCECC LSP setup operations are the same as those described in Section 5.5.1.

5.5.3. Central Controller Instructions

The new CCI for the label operations in PCEP are done via the PCInitiate message (Section 6.1) by defining a new PCEP object for CCI operations. The local label range of each PCC is assumed to be known by both the PCC and the PCE.

5.5.3.1. Label Download CCI

In order to set up an LSP based on the PCECC, the PCE sends a PCInitiate message to each node along the path to download the label instructions, as described in Sections 5.5.1 and 5.5.2.

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

If a node (PCC) receives a PCInitiate message that 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=31 (PCECC failure) and Error-value=1 (Label out of range) and **MUST** include the SRP object to specify the error is for the corresponding label update via a PCInitiate message. If a PCC receives a PCInitiate message but fails to download the label entry, it **MUST** send a PCErr message with Error-Type=31 (PCECC failure) and Error-value=2 (Instruction failed) and **MUST** include the SRP object to specify the error is for the corresponding label update via a PCInitiate message.

A new PCEP object for CCI is defined in [Section 7.3](#).

#### 5.5.3.2. Label Cleanup CCI

In order to delete an LSP based on the PCECC, the PCE sends Central Controller Instructions via a PCInitiate message to each node along the path of the LSP to clean up 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=18 (Unknown Label) and **MUST** include the SRP object to specify the error is for the corresponding label cleanup (via a PCInitiate message).

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

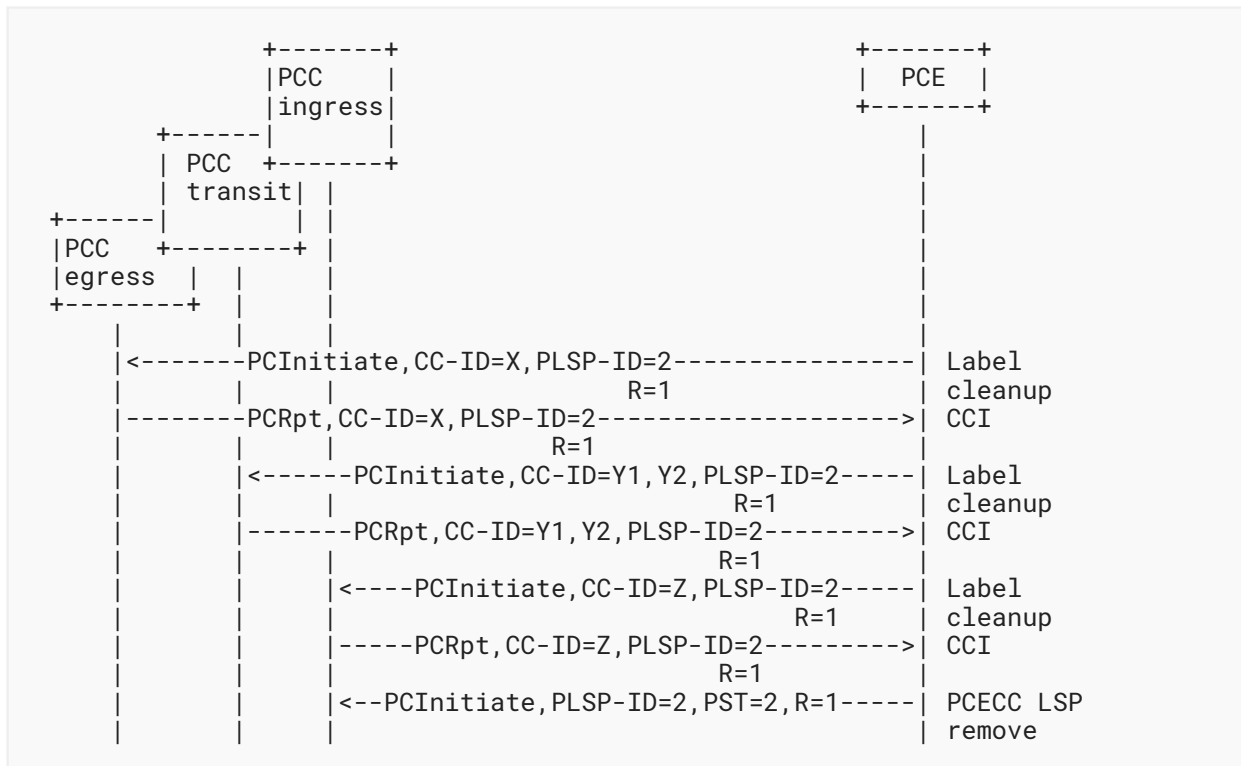


Figure 5: Label Cleanup

As per [RFC8281], following the removal of the label-forwarding instruction, the PCC **MUST** send a PCRpt message. The SRP object in the PCRpt message **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 the case where the label allocation is made by the PCC itself (see Section 5.5.8), the removal procedure remains the same, adding the sequence constraint.

#### 5.5.4. PCECC LSP Update

The update is done as per the make-before-break procedures, i.e., the PCECC first updates new label instructions based on the updated path and then informs the ingress to switch traffic before cleaning up the former instructions. New CC-IDs are used to identify the updated instructions; the identifiers in the LSP object uniquely 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 the PCRpt message, the PCE does the cleanup operation for the former LSP, as described in Section 5.5.3.2.



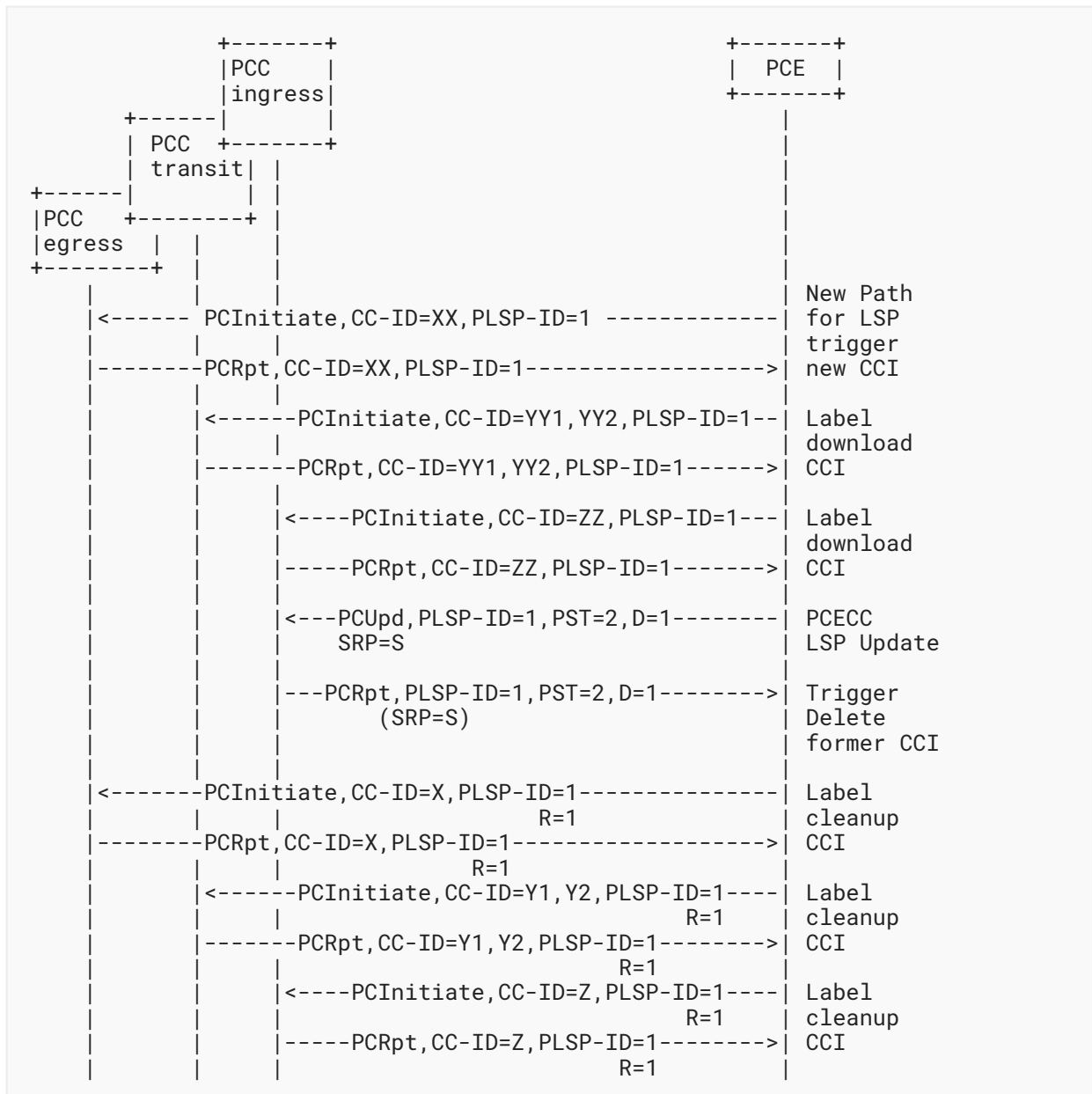


Figure 6: PCECC LSP Update

The modified PCECC LSPs are considered to be 'up' by default. The ingress could further choose to deploy a data-plane check mechanism and report the status back to the PCE via a PCRpt message. The exact mechanism is out of scope of this document.

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

### 5.5.5. Re-delegation and Cleanup

As described in [RFC8281], a new PCE can gain control over an orphaned LSP. In the case of a PCECC LSP, the new PCE **MUST** also gain control over the CCI 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 instructions that 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 the removal of CCI as one of the behaviors applied on expiration of the State Timeout Interval timer.

In the case of the PCC-initiated PCECC LSP, the control over the orphaned LSP at the ingress PCC is taken over by the mechanism specified in [RFC8741] to request delegation. The control over the CCI is described above using [RFC8281].

### 5.5.6. Synchronization of Central Controller Instructions

The purpose of CCI 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 [RFC8232].

As per LSP State Synchronization [RFC8231], a PCC reports the state of its LSPs to the PCE using PCRpt messages and, as per [RFC8281], the PCE would initiate any missing LSPs and/or remove any LSPs that are not wanted. The same PCEP messages and procedures are also used for the CCI 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 Operations, Administration, and Maintenance (OAM) mechanism to check the status of the LSP in the data plane and **MAY** further send its status in a 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 made by the PCC. The PCC **MUST** try to allocate the label and **MUST** report to the PCE via a PCRpt or PCErr message.

If the value of the label is 0 and the C flag is set to '1', it indicates that the PCE is requesting the allocation to be made by the PCC. If the label is 'n' and the C flag is set to '1' in the CCI object, it indicates that the PCE requests a specific value 'n' for the label. If the allocation is successful, the PCC **MUST** report via the PCRpt message with the CCI object. If the value of the label in the CCI object is invalid, it **MUST** send a PCErr message with Error-Type=31 (PCECC failure) and Error-

value=3 (Invalid CCI). If it is valid but the PCC is unable to allocate it, it **MUST** send a PCErr message with Error-Type=31 (PCECC failure) and Error-value=4 (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 label cleanup of the older label, as per [Section 5.5.3.2](#).

## 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, which specifies the set of objects that the message can carry. An implementation **MUST** form the PCEP messages using the object ordering specified in this document.

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

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

### 6.1. The PCInitiate Message

The PCInitiate message [\[RFC8281\]](#) can be used to download or remove the labels; this document extends the message, 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 a PCInitiate message is used for the CCI (labels), the SRP, LSP, and CCI objects **MUST** be present. The SRP object is defined in [RFC8231]; 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=17 (CCI object missing). More than one CCI object **MAY** be included in the PCInitiate message for a transit LSR.

To clean up entries, the R (remove) bit **MUST** be set in the SRP object to be encoded along with the LSP and CCI objects.

The CCI object received at the ingress node **MUST** have the O bit (out-label) set. The CCI object received at the egress **MUST** have the O bit unset. If this is not the case, the PCC **MUST** send a PCErr message with Error-Type=31 (PCECC failure) and Error-value=3 (Invalid CCI). Other instances of the CCI object, if present, **MUST** be ignored.

For the point-to-point (P2P) LSP setup via the PCECC technique, at the transit LSR, two CCI objects are expected for incoming and outgoing labels associated with the LSP object. If any other CCI object is included in the PCInitiate message, it **MUST** be ignored. If the transit LSR did not receive two CCI objects, with one of them having the O bit set and another with the O bit unset, it **MUST** send a PCErr message with Error-Type=31 (PCECC failure) and Error-value=3 (Invalid CCI).

Note that, on receipt of the PCInitiate message with CCI object, the ingress, egress, or transit role of the PCC is identified via the ingress and egress IP address encoded in the LSP-IDENTIFIERS TLV.

## 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 or as an acknowledgment to a PCInitiate message.

```
<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 objects are also defined in [RFC8231].

When a PCRpt message is used to report the CCI (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 PCE **MUST** send a PCErr message with Error-Type=6 (Mandatory Object missing) and Error-value=17 (CCI object missing). Two CCI objects can be included in the PCRpt message for a 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 PST (2) to be included in the PATH-SETUP-TYPE-CAPABILITY TLV in the OPEN object. Further, a new sub-TLV for the PCECC capability exchange is also defined.

#### 7.1.1. PCECC Capability Sub-TLV

The PCECC-CAPABILITY sub-TLV is an optional TLV for use in the OPEN object in the PATH-SETUP-TYPE-CAPABILITY TLV when the Path Setup Type list includes the PCECC Path Setup Type 2. A PCECC-CAPABILITY sub-TLV **MUST** be ignored if the PST list does not contain PST=2.

Its format is shown in Figure 7.

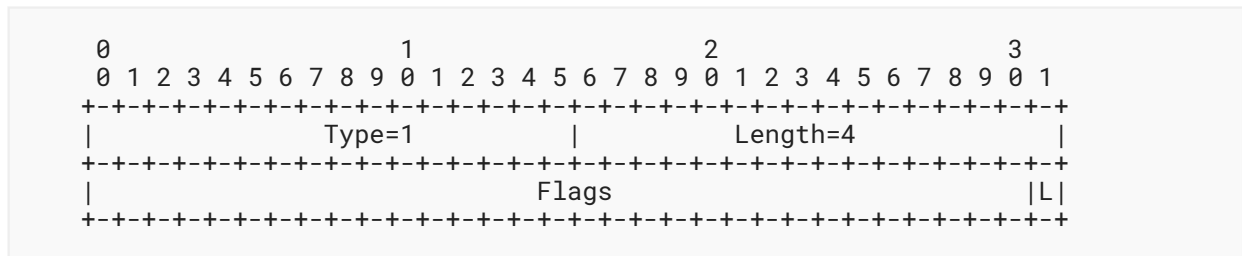


Figure 7: PCECC Capability Sub-TLV

The type of the TLV is 1, and it has a fixed length of 4 octets.

The value comprises a single field: Flags (32 bits). Currently, the following flag bit is defined:

**L bit (Label):** If set to '1' by a PCEP speaker, the L flag indicates that the PCEP speaker will support and is willing to handle the PCEC-based Central Controller Instructions for label download. The bit **MUST** be set to '1' by both a PCC and a PCE for the PCECC label download/report on a PCEP session.

Unassigned bits **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:

PST=2: Path is set up via the PCECC mode.

On a PCRpt/PCUpd/PCInitiate message, the PST=2 in the PATH-SETUP-TYPE TLV in the SRP object **MUST** be included for an LSP set up via the PCECC-based mechanism.

## 7.3. CCI Object

The 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 a PCInitiate or PCRpt message for label download/report.

CCI Object-Class is 44.

CCI Object-Type is 1 for the MPLS label.

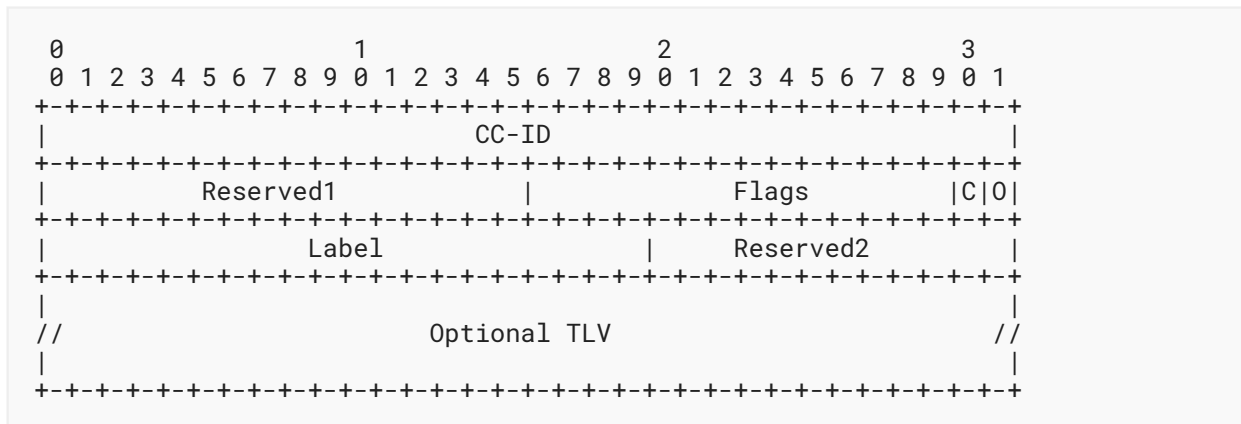


Figure 8: CCI Object

The fields in the CCI object are as follows:

**CC-ID:** A PCEP-specific identifier for the CCI information. A PCE creates a 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. Note that [SECURITY-ID] gives advice on assigning transient numeric identifiers, such as the CC-ID, so as to minimize security risks.

**Reserved1 (16 bit):** Set to 'zero' while sending; ignored on receipt.

**Flags (16 bit):** A field used to carry any additional information pertaining to the CCI. Currently, the following flag bits are defined:

- **O bit (out-label) :** If the bit is set to '1', it specifies the label is the out-label, and it is mandatory to encode the next-hop information (via Address TLVs (Section 7.3.1) 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 Address TLVs in the CCI object).
- **C Bit (PCC allocation):** If the bit is set to '1', it indicates that the label allocation needs to be done by the PCC for the Central Controller Instruction. A PCE sets 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 label and report it to the PCE.
- All unassigned bits **MUST** be set to 'zero' at transmission and ignored at receipt.

**Label (20-bit):** The label information.

**Reserved2 (12 bit):** Set to 'zero' while sending; ignored on receive.

### 7.3.1. Address TLVs

[RFC8779] defines the IPV4-ADDRESS, IPV6-ADDRESS, and UNNUMBERED-ENDPOINT TLVs for the use of Generalized Endpoint. The same TLVs can also be used in the CCI object to associate the next-hop information in the case of an outgoing label and local interface information in the case of an incoming label. The next-hop information encoded in these TLVs needs to be a directly

connected IP address/interface information. If the PCC is not able to resolve the next-hop information, it **MUST** reject the CCI and respond with a PCErr message with Error-Type=31 (PCECC failure) and Error-value=5 (Invalid next-hop information).

## 8. Security Considerations

As per [RFC8283], the security considerations for a PCE-based controller are a little different from those for any other PCE system. That is, the operation relies heavily on the use and security of PCEP, so consideration should be given to the security features discussed in [RFC5440] and the additional mechanisms described in [RFC8253]. It further lists the vulnerability of a central controller architecture, such as a central point of failure, denial of service, and a focus for interception and modification of messages sent to individual Network Elements (NEs).

In the PCECC operations, the PCEP sessions are also required to the internal routers, thus increasing the resources required for the session management at the PCE.

The PCECC extension builds on the existing PCEP messages; thus, the security considerations described in [RFC5440], [RFC8231], and [RFC8281] continue to apply. [RFC8253] specifies the support of Transport Layer Security (TLS) in PCEP, as it provides support for peer authentication, message encryption, and integrity. It further provides mechanisms for associating peer identities with different levels of access and/or authoritative via an attribute in X.509 certificates or a local policy with a specific accept-list of X.509 certificates. This can be used to check the authority for the PCECC operations. Additional considerations are discussed in following sections.

### 8.1. Malicious PCE

In this extension, the PCE has complete control over the PCC to download/remove the labels and can cause the LSPs to behave inappropriately and cause a major impact to the network. As a general precaution, it is **RECOMMENDED** that this PCEP extension be activated on mutually authenticated and encrypted sessions across PCEs and PCCs belonging to the same administrative authority, using TLS [RFC8253], as per the recommendations and best current practices in BCP 195 [RFC7525].

Further, an attacker may flood the PCC with the PCECC-related messages at a rate that exceeds either the PCC's ability to process them or the network's ability to send them, by either spoofing messages or compromising the PCE itself. [RFC8281] provides a mechanism to protect the PCC by imposing a limit. The same can be used for the PCECC operations as well.

As specified in Section 5.5.3.1, a PCC needs to check if the label in the CCI object is in the range set aside for the PCE; otherwise, it **MUST** send a PCErr message with Error-Type=31 (PCECC failure) and Error-value=1 (Label out of range).

### 8.2. Malicious PCC

The PCECC mechanism described in this document requires the PCE to keep labels (CCI) that it downloads and relies on the PCC responding (with either an acknowledgment or an error message) to request for LSP instantiation. This is an additional attack surface by placing a



requirement for the PCE to keep a CCI/label replica for each PCC. It is **RECOMMENDED** that PCE implementations provide a limit on resources (in this case the CCI) a single PCC can occupy. [RFC8231] provides a notification mechanism when such threshold is reached.

## 9. Manageability Considerations

### 9.1. Control of Function and Policy

A PCE or PCC implementation **SHOULD** allow the PCECC capability to be enabled/disabled as part of the global configuration. Section 6.1 of [RFC8664] list various controlling factors regarding the Path Setup Type. They are also applicable to the PCECC Path Setup Types. Further, Section 6.2 of [RFC8664] describes the migration steps when the Path Setup Type of an existing LSP is changed.

### 9.2. Information and Data Models

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

The PCEP YANG module [PCEP-YANG] could be extended to enable/disable the PCECC capability.

### 9.3. Liveness Detection and Monitoring

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

### 9.4. Verify Correct Operations

The operator needs the following information to verify that PCEP is operating correctly with respect to the PCECC Path Setup Type.

- An implementation **SHOULD** allow the operator to view whether the PCEP speaker sent the PCECC PST capability to its peer.
- An implementation **SHOULD** allow the operator to view whether the peer sent the PCECC PST capability.
- An implementation **SHOULD** allow the operator to view whether the PCECC PST is enabled on a PCEP session.
- If one PCEP speaker advertises the PCECC PST capability, but the other does not, then the implementation **SHOULD** create a log to inform the operator of the capability mismatch.
- If a PCEP speaker rejects a CCI, then it **SHOULD** create a log to inform the operator, giving the reason for the decision (local policy, label issues, etc.).

### 9.5. Requirements on Other Protocols

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

### 9.6. Impact on Network Operations

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

## 10. IANA Considerations

### 10.1. PATH-SETUP-TYPE-CAPABILITY Sub-TLV Type Indicators

[RFC8408] detailed the creation of the "PATH-SETUP-TYPE-CAPABILITY Sub-TLV Type Indicators" subregistry. Further, IANA has allocated the following codepoint:

Value	Meaning	Reference
1	PCECC-CAPABILITY	RFC 9050

*Table 2: PATH-SETUP-TYPE-CAPABILITY Sub-TLV Type Indicators Subregistry Addition*

### 10.2. PCECC-CAPABILITY Sub-TLV's Flag Field

This document defines the PCECC-CAPABILITY sub-TLV; IANA has created a new subregistry to manage the value of the PCECC-CAPABILITY sub-TLV's 32-bit Flag field. New values are to be assigned by Standards Action [RFC8126]. Each bit should be tracked with the following qualities:

- bit number (counting from bit 0 as the most significant bit)
- capability description
- defining RFC

Currently, there is one allocation in this registry.

Bit	Name	Reference
0-30	Unassigned	RFC 9050
31	Label	RFC 9050

*Table 3: Initial Contents of the PCECC-CAPABILITY Sub-TLV Subregistry*

### 10.3. PCEP Path Setup Type Registry

[RFC8408] created a subregistry within the "Path Computation Element Protocol (PCEP) Numbers" registry called "PCEP Path Setup Types". IANA has allocated a new codepoint within this registry, as follows:

Value	Description	Reference
2	Traffic engineering path is set up using PCECC mode	RFC 9050

*Table 4: Path Setup Type Registry Codepoint Addition*

## 10.4. PCEP Object

IANA has allocated new codepoints in the "PCEP Objects" subregistry for the CCI object as follows:

Object-Class Value	Name	Object-Type	Reference
44	CCI Object-Type	0: Reserved 1: MPLS Label 2-15: Unassigned	RFC 9050

*Table 5: PCEP Objects Subregistry Additions*

## 10.5. CCI Object Flag Field

IANA has created a new subregistry to manage the Flag field of the CCI object called "CCI Object Flag Field for MPLS Label". New values are to be assigned by Standards Action [RFC8126]. Each bit should be tracked with the following qualities:

- bit number (counting from bit 0 as the most significant bit)
- capability description
- defining RFC

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

Bit	Description	Reference
0-13	Unassigned	
14	C Bit - PCC allocation	RFC 9050
15	O Bit - Specifies label is out-label	RFC 9050

*Table 6: CCI Object Flag Field for MPLS Label Initial Contents*

## 10.6. PCEP-Error Object

IANA has allocated new error types and error values within the "PCEP-ERROR Object Error Types and Values" subregistry of the "Path Computation Element Protocol (PCEP) Numbers" registry for the following errors:

Error-Type	Meaning	Error-value	Reference
6	Mandatory Object missing	17: CCI object missing	RFC 9050
10	Reception of an invalid object	33: Missing PCECC Capability sub-TLV	RFC 9050
19	Invalid Operation	16: Attempted PCECC operations when PCECC capability was not advertised 17: Stateful PCE capability was not advertised 18: Unknown Label	RFC 9050
31	PCECC failure	1: Label out of range 2: Instruction failed 3: Invalid CCI 4: Unable to allocate the specified CCI 5: Invalid next-hop information	RFC 9050

Table 7: PCEP-ERROR Object Error Types and Values Additions

## 11. References

### 11.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC5440] Vasseur, JP., Ed. and JL. Le Roux, Ed., "Path Computation Element (PCE) Communication Protocol (PCEP)", RFC 5440, DOI 10.17487/RFC5440, March 2009, <<https://www.rfc-editor.org/info/rfc5440>>.
- [RFC5511] Farrel, A., "Routing Backus-Naur Form (RBNF): A Syntax Used to Form Encoding Rules in Various Routing Protocol Specifications", RFC 5511, DOI 10.17487/RFC5511, April 2009, <<https://www.rfc-editor.org/info/rfc5511>>.
- [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>>.
- [RFC8253] Lopez, D., Gonzalez de Dios, O., Wu, Q., and D. Dhody, "PCEPS: Usage of TLS to Provide a Secure Transport for the Path Computation Element Communication Protocol (PCEP)", RFC 8253, DOI 10.17487/RFC8253, October 2017, <<https://www.rfc-editor.org/info/rfc8253>>.
- [RFC8281] Crabbe, E., Minei, I., Sivabalan, S., and R. Varga, "Path Computation Element Communication Protocol (PCEP) Extensions for PCE-Initiated LSP Setup in a Stateful PCE Model", RFC 8281, DOI 10.17487/RFC8281, December 2017, <<https://www.rfc-editor.org/info/rfc8281>>.
- [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>>.
- [RFC8664] Sivabalan, S., Filsfils, C., Tantsura, J., Henderickx, W., and J. Hardwick, "Path Computation Element Communication Protocol (PCEP) Extensions for Segment Routing", RFC 8664, DOI 10.17487/RFC8664, December 2019, <<https://www.rfc-editor.org/info/rfc8664>>.
- [RFC8779] Margaria, C., Ed., Gonzalez de Dios, O., Ed., and F. Zhang, Ed., "Path Computation Element Communication Protocol (PCEP) Extensions for GMPLS", RFC 8779, DOI 10.17487/RFC8779, July 2020, <<https://www.rfc-editor.org/info/rfc8779>>.

## 11.2. Informative References

- [RFC4655] Farrel, A., Vasseur, JP., 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>>.
- [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>>.

- 
- [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>>.
- [RFC8232]** Crabbe, E., Minei, I., Medved, J., Varga, R., Zhang, X., and D. Dhody, "Optimizations of Label Switched Path State Synchronization Procedures for a Stateful PCE", RFC 8232, DOI 10.17487/RFC8232, September 2017, <<https://www.rfc-editor.org/info/rfc8232>>.
- [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>>.
- [RFC8741]** Raghuram, A., Goddard, A., Karthik, J., Sivabalan, S., and M. Negi, "Ability for a Stateful Path Computation Element (PCE) to Request and Obtain Control of a Label Switched Path (LSP)", RFC 8741, DOI 10.17487/RFC8741, March 2020, <<https://www.rfc-editor.org/info/rfc8741>>.
- [PCECC]** Li, Z. (., Dhody, D., Zhao, Q., Ke, K., Khasanov, B., Fang, L., Zhou, C., Zhang, B., Rachitskiy, A., and A. Gulida, "The Use Cases for Path Computation Element (PCE) as a Central Controller (PCECC).", Work in Progress, Internet-Draft, draft-ietf-teas-pcecc-use-cases-07, 8 March 2021, <<https://datatracker.ietf.org/doc/html/draft-ietf-teas-pcecc-use-cases-07>>.
- [PCEP-YANG]** Dhody, D., Ed., Hardwick, J., Beeram, V., and J. Tantsura, "A YANG Data Model for Path Computation Element Communications Protocol (PCEP)", Work in Progress, Internet-Draft, draft-ietf-pce-pcep-yang-16, 22 February 2021, <<https://datatracker.ietf.org/doc/html/draft-ietf-pce-pcep-yang-16>>.
- [PCECC-SR]** Li, Z., Peng, S., Negi, M. S., Zhao, Q., and C. Zhou, "PCEP Procedures and Protocol Extensions for Using PCE as a Central Controller (PCECC) for Segment Routing (SR) MPLS Segment Identifier (SID) Allocation and Distribution.", Work in Progress, Internet-Draft, draft-ietf-pce-pcep-extension-pce-controller-sr-02, 25 March 2021, <<https://datatracker.ietf.org/doc/html/draft-ietf-pce-pcep-extension-pce-controller-sr-02>>.
- [PCECC-SRv6]** Li, Z., Peng, S., Geng, X., and M. S. Negi, "PCEP Procedures and Protocol Extensions for Using PCE as a Central Controller (PCECC) for SRv6", Work in Progress, Internet-Draft, draft-dhody-pce-pcep-extension-pce-controller-srv6-06, 21 February 2021, <<https://datatracker.ietf.org/doc/html/draft-dhody-pce-pcep-extension-pce-controller-srv6-06>>.
- [PCE-ID]** Li, C., Chen, M., Wang, A., Cheng, W., and C. Zhou, "PCE Controlled ID Space", Work in Progress, Internet-Draft, draft-li-pce-controlled-id-space-08, 22 February 2021, <<https://datatracker.ietf.org/doc/html/draft-li-pce-controlled-id-space-08>>.

**[SECURITY-ID]** Gont, F. and I. Arce, "Security Considerations for Transient Numeric Identifiers Employed in Network Protocols", Work in Progress, Internet-Draft, draft-gont-numeric-ids-sec-considerations-06, 5 December 2020, <<https://datatracker.ietf.org/doc/html/draft-gont-numeric-ids-sec-considerations-06>>.

## Acknowledgments

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

Thanks to Julien Meuric for shepherding this document and providing valuable comments.  
Thanks to Deborah Brungard for being the responsible AD.

Thanks to Victoria Pritchard for a very detailed RTGDIR review. Thanks to Yaron Sheffer for the SECDIR review. Thanks to Gyan Mishra for the Gen-ART review.

Thanks to Alvaro Retana, Murray Kucherawy, Benjamin Kaduk, Roman Danyliw, Robert Wilton, Éric Vyncke, and Erik Kline for the IESG review.

## Contributors

### **Dhruv Dhody**

Huawei Technologies  
Divyashree Techno Park, Whitefield  
Bangalore 560066  
Karnataka  
India  
Email: [dhruv.ietf@gmail.com](mailto:dhruv.ietf@gmail.com)

### **Satish Karunanithi**

Huawei Technologies  
Divyashree Techno Park, Whitefield  
Bangalore 560066  
Karnataka  
India  
Email: [satishk@huawei.com](mailto:satishk@huawei.com)

### **Adrian Farrel**

Old Dog Consulting  
United Kingdom  
Email: [adrian@olddog.co.uk](mailto:adrian@olddog.co.uk)

### **Xuesong Geng**

Huawei Technologies  
China  
Email: [gengxuesong@huawei.com](mailto:gengxuesong@huawei.com)

**Udayasree Palle**

Email: [udayasreereddy@gmail.com](mailto:udayasreereddy@gmail.com)

**Katherine Zhao**

Futurewei Technologies

Email: [katherine.zhao@futurewei.com](mailto:katherine.zhao@futurewei.com)

**Boris Zhang**

Telus Ltd.

Toronto

Canada

Email: [boris.zhang@telus.com](mailto:boris.zhang@telus.com)

**Alex Tokar**

Cisco Systems

Slovakia

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

## Authors' Addresses

**Zhenbin Li**

Huawei Technologies

Huawei Bld., No.156 Beiqing Rd.

Beijing

100095

China

Email: [lizhenbin@huawei.com](mailto:lizhenbin@huawei.com)

**Shuping Peng**

Huawei Technologies

Huawei Bld., No.156 Beiqing Rd.

Beijing

100095

China

Email: [pengshuping@huawei.com](mailto:pengshuping@huawei.com)

**Mahendra Singh Negi**

RtBrick Inc

N-17L, 18th Cross Rd, HSR Layout

Bangalore 560102

Karnataka

India

Email: [mahend.ietf@gmail.com](mailto:mahend.ietf@gmail.com)



**Quintin Zhao**

Ethereic Networks

1009 S Claremont St.

San Mateo, CA 94402

United States of America

Email: [qzhao@ethericnetworks.com](mailto:qzhao@ethericnetworks.com)**Chao Zhou**

HPE

Email: [chaozhou\\_us@yahoo.com](mailto:chaozhou_us@yahoo.com)