PCE Working Group   Z. Li
Internet-Draft   S. Peng
Intended status: Standards Track   Huawei Technologies
Expires: September 2, 2021   M. Negi
                                                       RtBrick Inc
                                                           Q. Zhao
                                             Etheric Networks
                                                           C. Zhou
                                                              HPE
                                                    March 1, 2021

PCEP Procedures and Protocol Extensions for Using PCE as a Central Controller (PCECC) of LSPs
draft-ietf-pce-pcep-extension-for-pce-controller-13

Abstract

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

   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/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 provisioning labels along the
   path of the static LSP.

Status of This Memo

   This Internet-Draft will expire on September 2, 2021.

Copyright Notice

Table of Contents

1. Introduction

   The Path Computation Element (PCE) [RFC4655] was developed to offload
   the path computation function from routers in an MPLS traffic-
   engineered 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 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].

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 MPLS and GMPLS 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.

[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 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 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
PCECC for Segment Routing (SR) are specified in
[I-D.ietf-pce-pcep-extension-pce-controller-sr] and
[I-D.dhody-pce-pcep-extension-pce-controller-srv6].

1.1. Requirements Language

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

2. Terminology

The terminology used in this document is the same as that described
in the [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.

[RFC8283] examines the motivations and applicability for PCECC and
use of PCEP as an SBI. Section 3.1.2. of [RFC8283] highlights the
use of 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
[I-D.ietf-teas-pcecc-use-cases] describes some of the scenarios where
the PCECC technique could be useful. Section 4 of [RFC8283] also
describe the implications on the protocol when used as an SDN SBI.
The operator needs to evaluate the advantages offered by 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
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 [I-D.li-pce-controlled-id-space]). 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 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 need means to identify PCECC-based LSP 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]. 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 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 set up PCE-Initiated LSP based on 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]. PCRpt message is used
      to send PCECC LSP Reports. It is also extended to report the set
      of Central Controller Instructions (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 PCRpt message during synchronization.

   PCUpd: a PCEP message described in [RFC8231]. PCUpd message is used
      to send 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
   central controller instructions 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
   PCECC using these elements in the OPEN message:

   o A new Path Setup Type (PST) ([Section 7.2](#)) in the PATH-SETUP-TYPE-
      CAPABILITY TLV to indicate support for PCEP extensions for PCECC -
      TBD1 (Path is set up via PCECC mode)

   o 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 PCECC based central
      controller instructions for label download

   o The STATEFUL-PCE-CAPABILITY TLV ([[RFC8231](#)]) (with the I flag set
      [[RFC8281](#)])

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

   The new PCECC-CAPABILITY sub-TLV is included in PATH-SETUP-TYPE-
   CAPABILITY TLV in the OPEN object to indicate a willingness to use
   the PCEP extensions for 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:

   o Send a PCErr message with Error-Type=19 (Invalid Operation) and
      Error-value=TBD4 (stateful PCE capability was not advertised)

   o Terminate the session

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

   o Send a PCErr message with Error-Type 10 (Reception of an invalid
      object) and Error-Value TBD2 (Missing PCECC-CAPABILITY sub-TLV)

   o Terminate the PCEP session

   The PCECC-CAPABILITY sub-TLV MUST NOT be used without the
   corresponding Path Setup Type being listed in the PATH-SETUP-TYPE-
   CAPABILITY TLV. If it is present without the corresponding Path
   Setup Type 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 PCECC, the PCEP extensions
for 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:

o Send a PCErr message with Error-Type=19 (Invalid Operation) and
   Error-Value=TBD3 (Attempted PCECC operations when PCECC capability
   was not advertised)

o 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 RP (Request Parameter) /SRP (Stateful PCE
Request Parameters) Object will:

o Send a PCErr message with Error-Type = 21 (Invalid traffic
   engineering path setup type) and Error-value = 1 (Unsupported path
   setup type)

o Terminate the PCEP session

5.5. LSP Operations

The PCEP messages pertaining to a PCECC MUST include PATH-SETUP-TYPE
TLV [RFC8408] in the SRP object [RFC8231] with PST set to TBD1 to
clearly identify that 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
PCInitiate message with PST set to TBD1 for PCECC (see Section 7.2)
to the ingress PCC.

The label forwarding instructions (see Section 5.5.3) from 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 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 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 central controller instructions (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 LSR
could have multiple central controller instructions that have the
same PLSP-ID. The PLSP-ID in combination with the source (in 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 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 instruction within a PCEP session. Each node
along the path (PCC) responds with a PCRpt message to acknowledge the
central controller instruction with the PCRpt messages including the
central controller instruction (CCI) and the LSP objects.

The ingress node would receive one CCI object with O bit (out-label)
set. The transit node(s) would receive two CCI objects with the in-
label CCI without an O bit set and the out-label CCI with O bit set.
The egress node would receive one CCI object without 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 PCE-Initiated PCECC LSP is the same as
defined in [RFC8281]. The PCE should further perform Label entry
clean up operation as described in Section 5.5.3.2 for the
corresponding LSP.

The PCE-Initiated PCECC LSP setup sequence is shown in Figure 1.

```
                    +-------+                        +-------+
                    |PCC    |                        | PCE   |
                    |ingress|                        +-------+
            +------|        |                        |
            | PCC  +-------+                          |
            | transit| |                             |
    +------|        | |<--PCInitiate,PLSP-ID=0,PST=TBD1------| PCECC LSP
    |PCC   +--------+ |                               | Initiate
    |egress |  |      |----PCRpt,PLSP-ID=2,D=1,C=1---------->| PCECC LSP
    +--------+ |      |         (GOING-UP)            |
        |      |      |                               |
        |<-------PCInitiate,CC-ID=X,PLSP-ID=2---------------| Label
        |      |      |                               | download
        |--------PCRpt,CC-ID=X,PLSP-ID=2-------------------->| CCI
        |      |      |                               |
        |         |<------PCInitiate,CC-ID=Y1,Y2,PLSP-ID=2-----| Label
        |      |      |                               | download
        |         |-------PCRpt,CC-ID=Y1,Y2,PLSP-ID=2-------->| CCI
        |      |      |                               |
        |      |         |<----PCInitiate,CC-ID=Z,PLSP-ID=2-----| Label
        |      |      |                               | download
        |      |         |-----PCRpt,CC-ID=Z,PLSP-ID=2-------->| CCI
        |      |      |                               |
        |      |         |<---PCUpd,PLSP-ID=2,PST=TBD1,D=1------| PCECC LSP
        |      |      |          (UP)                 | Update
        |      |         |----PCRpt,PLSP-ID=2,D=1,C=1---------->|
        |      |      |          (UP)                 |
```

                  Figure 1: PCE-Initiated PCECC LSP

   Once the label operations are completed, the PCE MUST send a PCUpd
   message to the ingress PCC. The PCUpd message is as per [RFC8231]
   with D flag set.

   The PCECC LSPs are considered to be 'up' by default (on receipt of
   PCUpd message from 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, and then the PCC would send a PCRpt 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.

```
                  +-------+                        +-------+
                  |PCC    |                        |  PCE  |
                  |ingress|                        +-------+
           +------|       |                            |
           | PCC  +-------+                            |
           | transit| |                               |
   +------|        | |<--PCInitiate,PLSP-ID=0,PST=TBD1,-----| PCECC LSP
   |PCC    +--------+ |                               | Initiate
   |egress  | |     |----PCRpt,PLSP-ID=2,D=1,C=1---------->| PCECC LSP
   +--------+ |     |        (GOING-UP)              |
       |      |     |                                |
       |<-------PCInitiate,CC-ID=X,PLSP-ID=2---------------| Label
       |      |     |       C=1,O=0                   | download
       |--------PCRpt,CC-ID=X,PLSP-ID=2-------------------->| CCI
       |      |     |       Label=L1                  |
       |      |<------PCInitiate,PLSP-ID=2,---------------| Labels
       |      |     |            CC-ID=Y1,C=1,O=0      | download
       |      |     |            CC-ID=Y2,C=0,O=1,L1   | CCI
       |      |-------PCRpt,PLSP-ID=2--------------------->|
       |      |     |        CC-ID=Y1,O=0,Label=L2     |
       |      |     |        CC-ID=Y2,O=1              |
       |      |     |<----PCInitiate,CC-ID=Z,PLSP-ID=2-----| Label
       |      |     |               C=0,O=1,L2         | download
       |      |     |-----PCRpt,CC-ID=Z,PLSP-ID=2--------->| CCI
       |      |     |                                |
       |      |     |<---PCUpd,PLSP-ID=2,PST=TBD1,D=1------| PCECC LSP
       |      |     |          (UP)                   | Update
```

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

   It should be noted that in this example, 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 inform
   the transit PCC without setting the C bit to 1 in the CCI object for
   out-label but the C bit is set to 1 for in-label so the transit node
   make the label allocation (for the in-label) and report to the PCE.
   Similarly, the C bit is unset towards the ingress to complete all the
   label allocation 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 PST set for 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 D flag and PST
Type set to TBD1, it SHOULD calculate the path and 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
(see Figure 3). The CC-ID uniquely identifies the central controller
instruction within a PCEP session. Each PCC further responds with
the PCRpt messages including the central controller instruction (CCI)
and the LSP objects.

Once the central controller instructions (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 PCECC LSPs is the same as defined in
[RFC8231]. If the PCE receives a PCRpt message for LSP deletion then
it does label clean up operation as described in Section 5.5.3.2 for
the corresponding LSP.

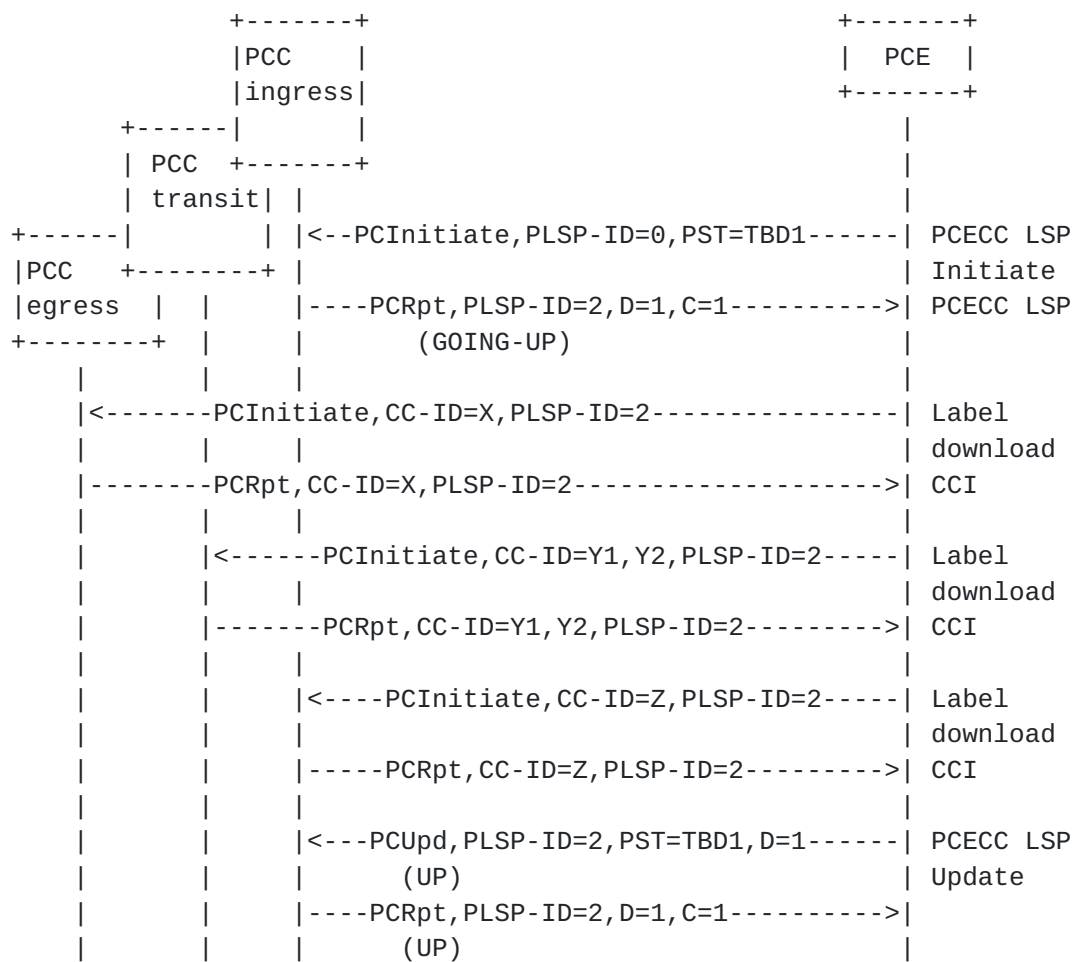The Basic PCECC LSP setup sequence is as shown in Figure 3.

```
                    +-------+                          +-------+
                    |PCC    |                          |  PCE  |
                    |ingress|                          +-------+
              +------|       |                          |
              | PCC  +-------+                          |
              | transit| |                             |
        +------|        | |---PCRpt,PLSP-ID=1,PST=TBD1,D=1----->| PCECC LSP
        |PCC    +--------+ |                            |
        |egress |  |     |                             |
        +--------+  |     |                             |
            |      |     |                             |
            |<-------PCInitiate,CC-ID=X,PLSP-ID=1---------------| Label
            |      |     |     L1,O=0                  | download
            |--------PCRpt,CC-ID=X,PLSP-ID=1------------------>| CCI
            |      |     |                             |
            |          |<------PCInitiate,PLSP-ID=1,---------------| Labels
            |      |     |            CC-ID=Y1,O=0,L2   | download
            |      |     |            CC-ID=Y2,O=1,L1   | CCI
            |          |-------PCRpt,CC-ID=Y1,Y2,PLSP-ID=1-------->|
            |      |     |                             |
            |      |     |<----PCInitiate,CC-ID=Z,PLSP-ID=1----| Label
            |      |     |            L2,O=1           | download
            |      |     |-----PCRpt,CC-ID=Z,PLSP-ID=1-------->| CCI
            |      |     |                             |
            |      |     |<---PCUpd,PLSP-ID=1,PST=TBD1,D=1-----| PCECC LSP
            |      |     |                             | Update
            |      |     |                             |
```

                    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, and then the PCC would send a PCRpt with the allocated
   label encoded in the CC-ID object as shown in Figure 4.

```
                  +-------+                          +-------+
                  |PCC    |                          |  PCE  |
                  |ingress|                          +-------+
          +------|       |                               |
          | PCC  +-------+                               |
          | transit| |                                   |
     +------|       | |---PCRpt,PLSP-ID=1,PST=TBD1,D=1----->| PCECC LSP
     |PCC    +--------+ |                                 |
     |egress | |     |                                    |
     +--------+ |     |                                   |
         |      |     |                                   |
         |<-------PCInitiate,CC-ID=X,PLSP-ID=1--------------| Label
         |      |     |      C=1                         | download
         |--------PCRpt,CC-ID=X,PLSP-ID=1------------------>| CCI
         |      |     |     Label=L1                      |
         |      |<------PCInitiate,PLSP-ID=1,---------------| Labels
         |      |     |           CC-ID=Y1,C=1           | download
         |      |     |           CC-ID=Y2,C=0,L1        | CCI
         |      |-------PCRpt,PLSP-ID=1-------------------->|
         |      |     |        CC-ID=Y1,Label=L2          |
         |      |     |        CC-ID=Y2                    |
         |      |     |<----PCInitiate,CC-ID=Z,PLSP-ID=1----| Label
         |      |     |              C=0,L2              | download
         |      |     |-----PCRpt,CC-ID=Z,PLSP-ID=1-------->| CCI
         |      |     |                                   |
         |      |     |<---PCUpd,PLSP-ID=1,PST=TBD1,D=1-----| PCECC LSP
         |      |     |                                   | Update
         |      |     |                                   |
```
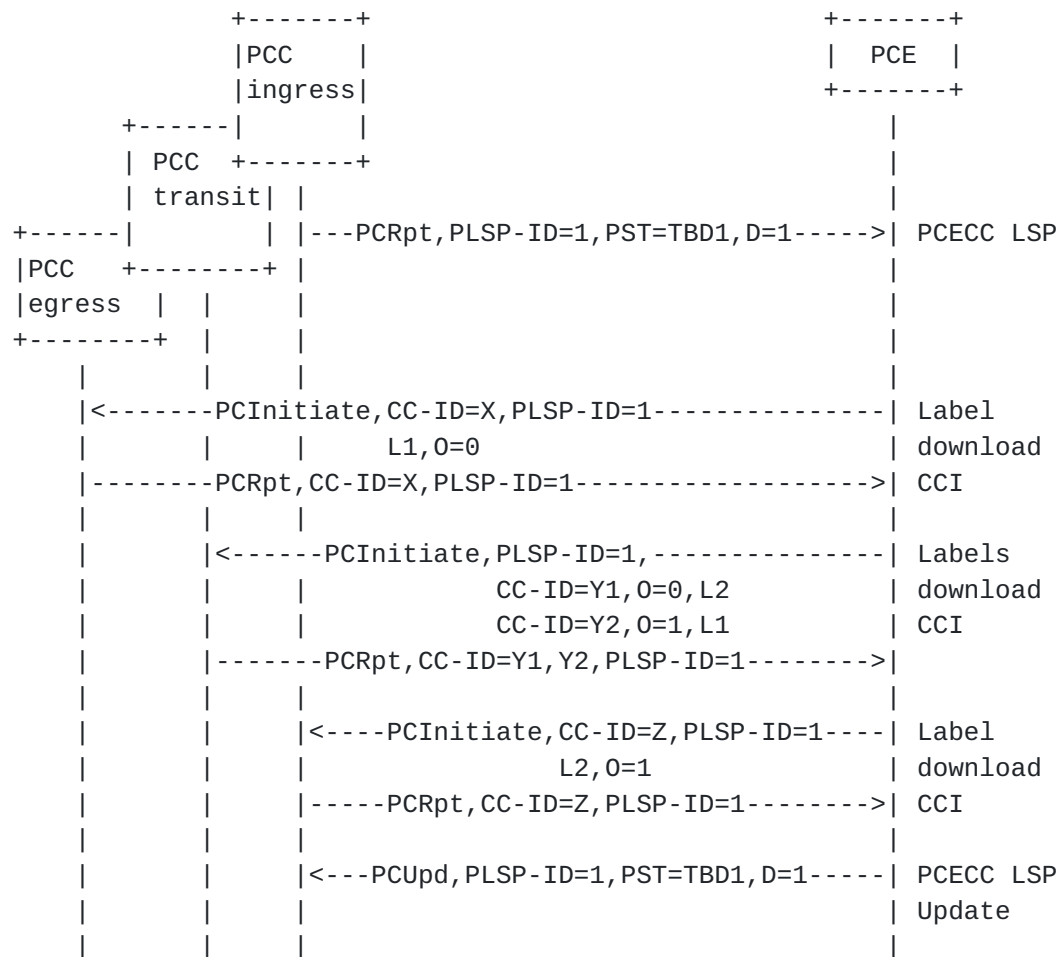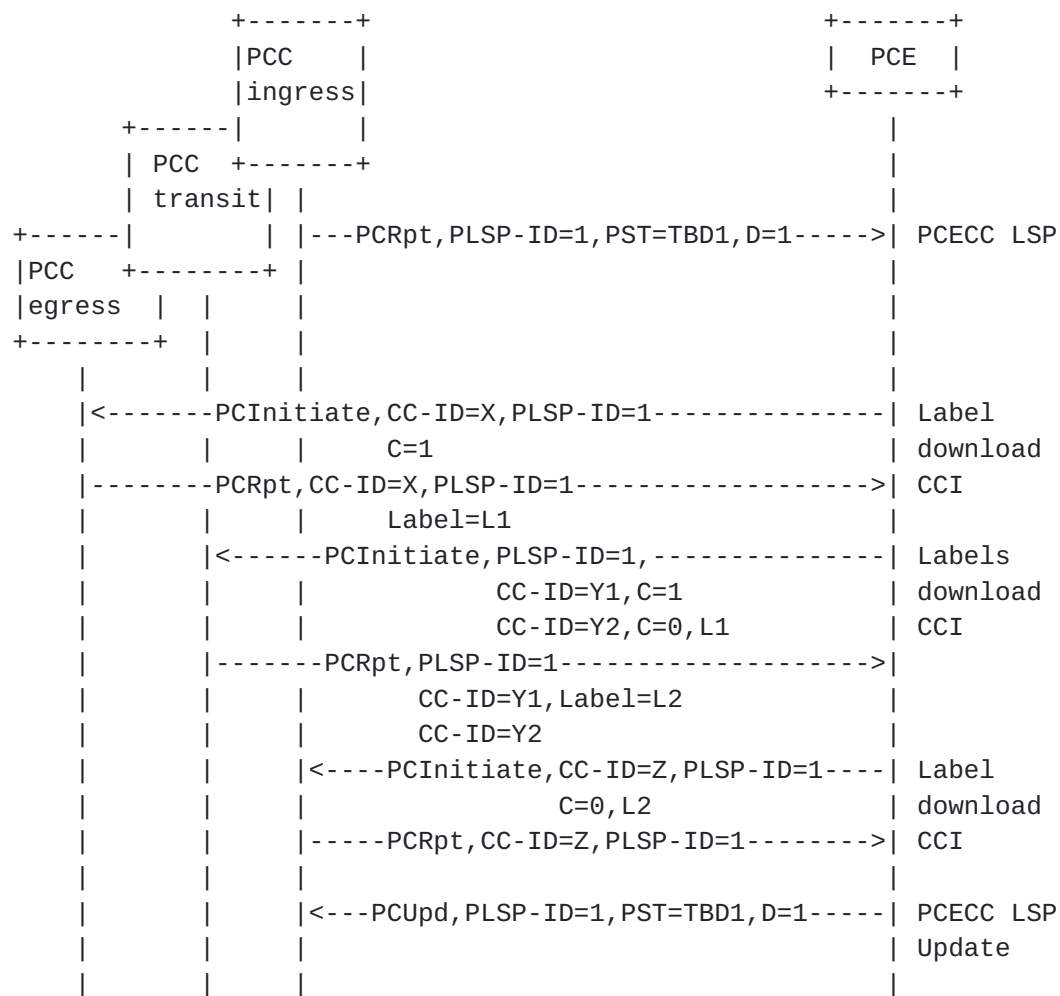
     - The O bit is set as before (and thus not included)

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

   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 central controller instructions (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 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 and Section 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 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 fails 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.

   A new PCEP object for central controller instructions (CCI) is
   defined in Section 7.3.

5.5.3.2. Label Clean up 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 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=TBD8, "Unknown Label" and
   MUST include the SRP object to specify the error is for the
   corresponding label clean up (via PCInitiate message).

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

```
                 +-------+                           +-------+
                 |PCC    |                           |  PCE  |
                 |ingress|                           +-------+
           +------|       |                               |
           | PCC  +-------+                               |
           | transit| |                                   |
      +------|       | |                                   |
      |PCC   +--------+ |                                   |
      |egress |   |     |                                   |
      +--------+  |     |                                   |
          |       |     |                                   |
          |<-------PCInitiate,CC-ID=X,PLSP-ID=2----------------| Label
          |       |     |                 R=1               | clean up
          |--------PCRpt,CC-ID=X,PLSP-ID=2------------------->| CCI
          |       |     |             R=1                   |
          |       |<------PCInitiate,CC-ID=Y1,Y2,PLSP-ID=2-----| Label
          |       |     |                     R=1           | clean up
          |       |-------PCRpt,CC-ID=Y1,Y2,PLSP-ID=2--------->| CCI
          |       |     |                       R=1         |
          |       |<----PCInitiate,CC-ID=Z,PLSP-ID=2-----| Label
          |       |     |                       R=1       | clean up
          |       |-----PCRpt,CC-ID=Z,PLSP-ID=2--------->| CCI
          |       |     |                       R=1       |
          |       |<--PCInitiate,PLSP-ID=2,PST=TBD1,R=1--| PCECC LSP
          |       |     |                                 | remove
```
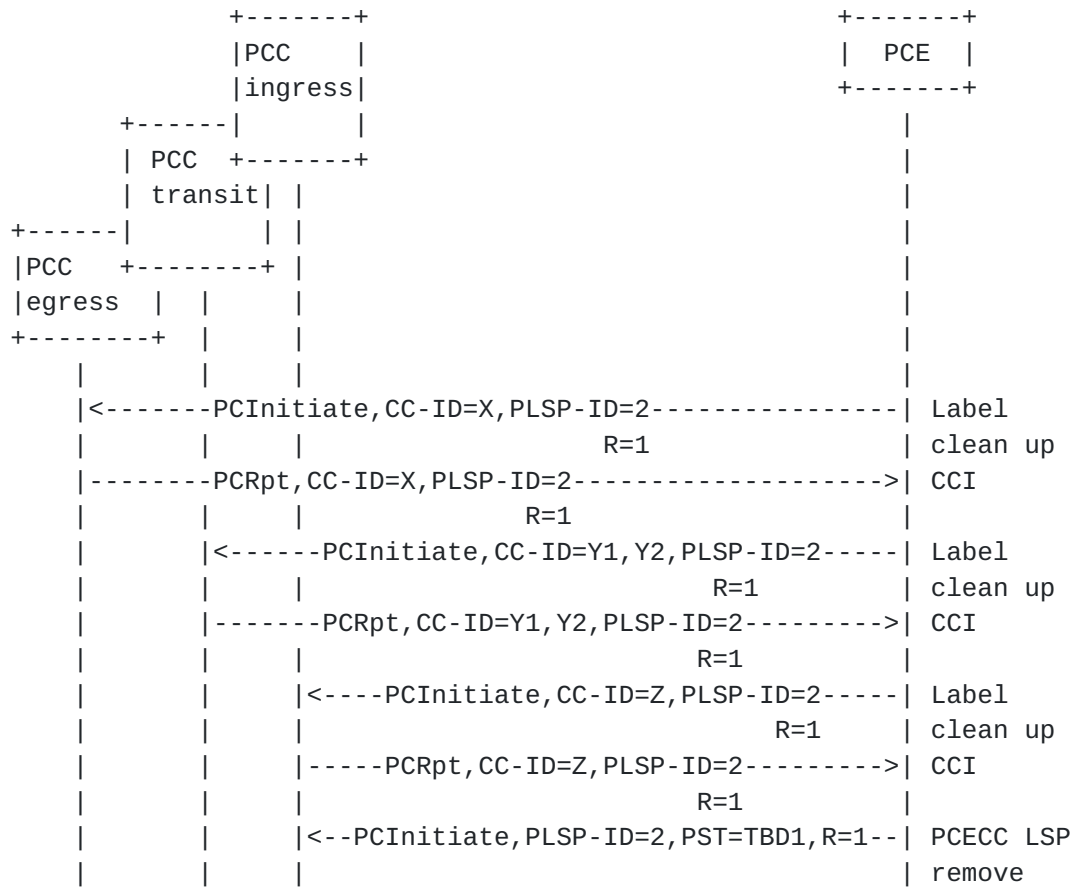
                     Figure 5: Label Cleanup

   As per [RFC8281], following the removal of the Label forwarding
   instruction, the PCC MUST send a PCRpt message. The SRP object in
   the PCRpt 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 clean up
operation for the former LSP as described in Section 5.5.3.2.

The PCECC LSP Update sequence is shown in Figure 6.

```
                    +-------+                        +-------+
                    |PCC    |                        | PCE   |
                    |ingress|                        +-------+
            +------|       |                              |
            | PCC  +-------+                              |
            | transit| |                                 |
     +------|       | |                                   |
     |PCC   +--------+ |                                   |
     |egress |  |     |                                   |
     +--------+  |     |                                   |
          |      |     |                                   | New Path
          |<------ PCInitiate,CC-ID=XX,PLSP-ID=1 -------------| for LSP
          |      |     |                                   | trigger
          |--------PCRpt,CC-ID=XX,PLSP-ID=1----------------->| new CCI
          |      |     |                                   |
          |      |<------PCInitiate,CC-ID=YY1,YY2,PLSP-ID=1--| Label
          |      |     |                                   | download
          |      |-------PCRpt,CC-ID=YY1,YY2,PLSP-ID=1----->| CCI
          |      |     |                                   |
          |      |     |<----PCInitiate,CC-ID=ZZ,PLSP-ID=1---| Label
          |      |     |                                   | download
          |      |     |-----PCRpt,CC-ID=ZZ,PLSP-ID=1------->| CCI
          |      |     |                                   |
          |      |     |<---PCUpd,PLSP-ID=1,PST=TBD1,D=1-----| PCECC
          |      |     |     SRP=S                          | LSP Update
          |      |     |                                   |
          |      |     |---PCRpt,PLSP-ID=1,PST=TBD1,D=1----->| Trigger
          |      |     |          (SRP=S)                   | Delete
          |      |     |                                   | former CCI
          |      |     |                                   |
          |<-------PCInitiate,CC-ID=X,PLSP-ID=1---------------| Label
          |      |     |                     R=1           | clean up
          |--------PCRpt,CC-ID=X,PLSP-ID=1------------------->| CCI
          |      |     |              R=1                  |
          |      |<------PCInitiate,CC-ID=Y1,Y2,PLSP-ID=1----| Label
          |      |     |                         R=1       | clean up
          |      |-------PCRpt,CC-ID=Y1,Y2,PLSP-ID=1-------->| CCI
          |      |     |                     R=1           |
          |      |     |<----PCInitiate,CC-ID=Z,PLSP-ID=1----| Label
          |      |     |                         R=1       | clean up
          |      |     |-----PCRpt,CC-ID=Z,PLSP-ID=1-------->| CCI
          |      |     |                     R=1           |
```
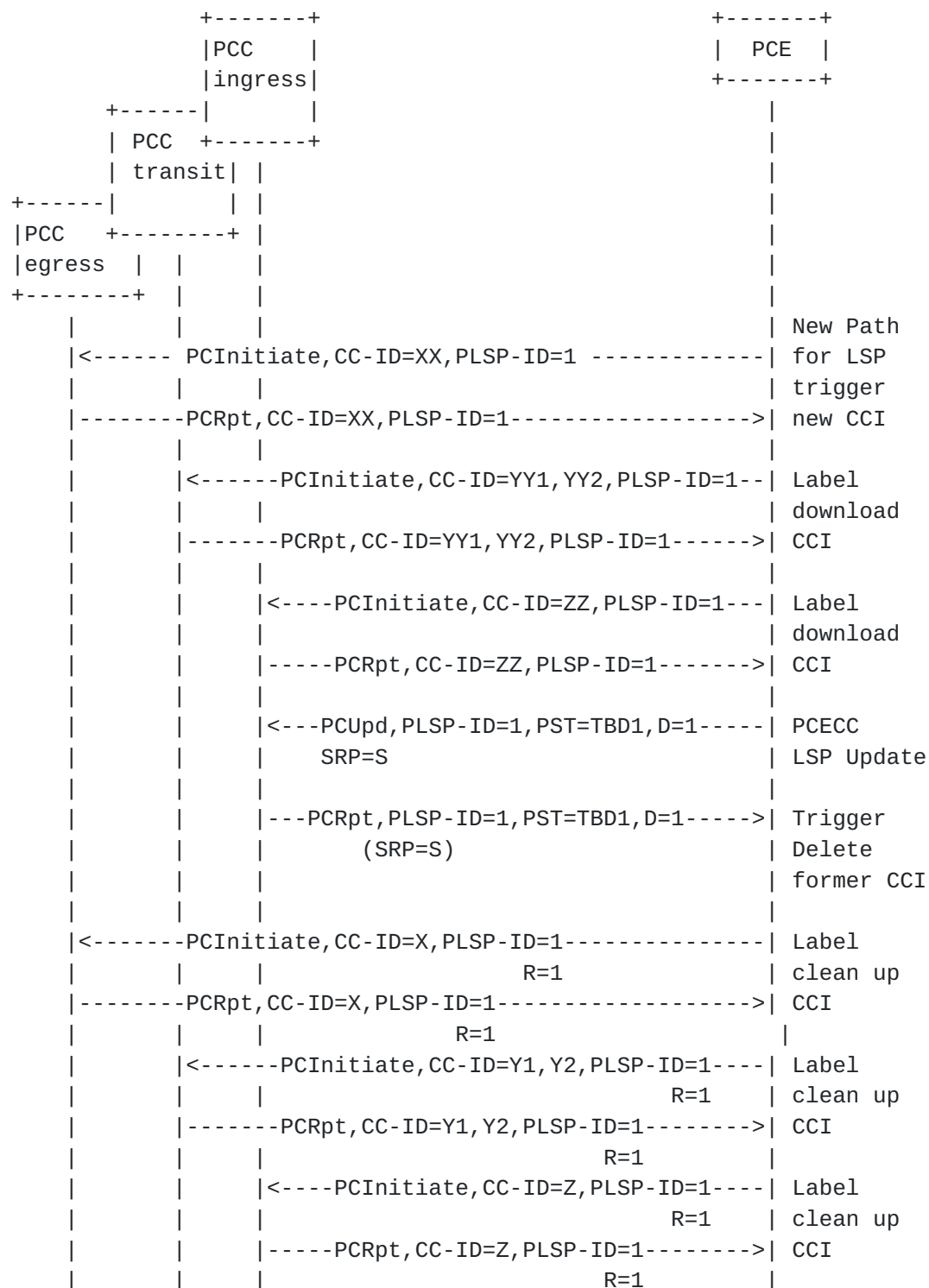
                        Figure 6: PCECC LSP Update

   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 Clean up

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 central controller 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
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 clean up 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 case of 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 central
controller instructions is described above using [RFC8281].

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 [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], 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
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 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 done by the PCC. The PCC
MUST try to allocate the Label and MUST report to the PCE via 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 done 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 = TBD5 ("PCECC failure") and Error Value = TBD9 ("Invalid CCI").
If it 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 Label cleanup of 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 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.

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 PCInitiate message is used for the central controller
   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 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 the CCI object.

   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, PCC MUST send a PCErr message
   with Error-Type = TBD5 ("PCECC failure") and Error Value = TBD9
   ("Invalid CCI"). Other instances of the CCI object if present, MUST
   be ignored.

   For the P2P LSP setup via PCECC technique, at the transit LSR two CCI
   objects are expected for in-coming and outgoing label 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 object with one of them having the O bit set and
   another with O bit unset, it MUST send a PCErr message with Error-
   Type = TBD5 ("PCECC failure") and Error Value = TBD9 ("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 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 object are
        also defined in [RFC8231].
```

   When PCRpt message is used to report the central controller
   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 PCE MUST send a PCErr message with Error-
   type=6 (Mandatory Object missing) and Error-value=TBD11 (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 (TBD1) to be included in the PATH-
   SETUP-TYPE-CAPABILITY TLV in the OPEN Object. Further, a new sub-TLV
   for 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 TBD1. A PCECC-
   CAPABILITY sub-TLV MUST be ignored if the PST list does not contain
   PST=TBD1.

   Its format is shown in Figure 7.

```
    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=TBD12          |           Length=4            |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                           Flags                             |L|
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
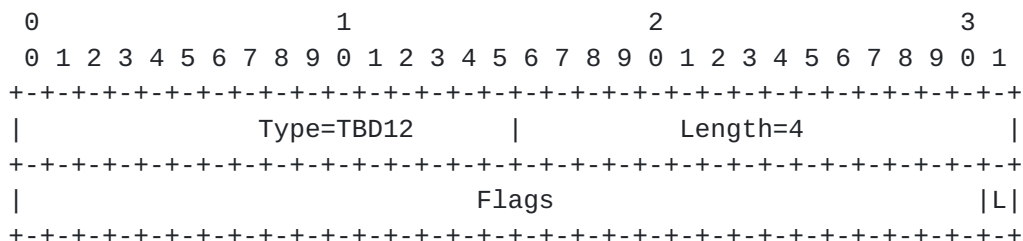
                   Figure 7: PCECC Capability sub-TLV

   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). Currently, the
   following flag bit is defined:

   o L bit (Label): if set to 1 by a PCEP speaker, the L flag indicates
      that the PCEP speaker support and is willing to handle the PCECC
      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.

   o 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:

   o PST = TBD1: Path is set up via PCECC mode.

   On a PCRpt/PCUpd/PCInitiate message, the PST=TBD1 in the PATH-SETUP-
   TYPE TLV in the SRP object MUST be included for a LSP set up via the
   PCECC-based mechanism.

7.3. CCI Object

   The Central Controller 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/report.

   CCI Object-Class is TBD13.

   CCI Object-Type is 1 for the MPLS Label.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                            CC-ID                             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Reserved1           |         Flags          |C|O|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|             Label            |         Reserved2            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                              |
//                      Optional TLV                          //
|                                                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

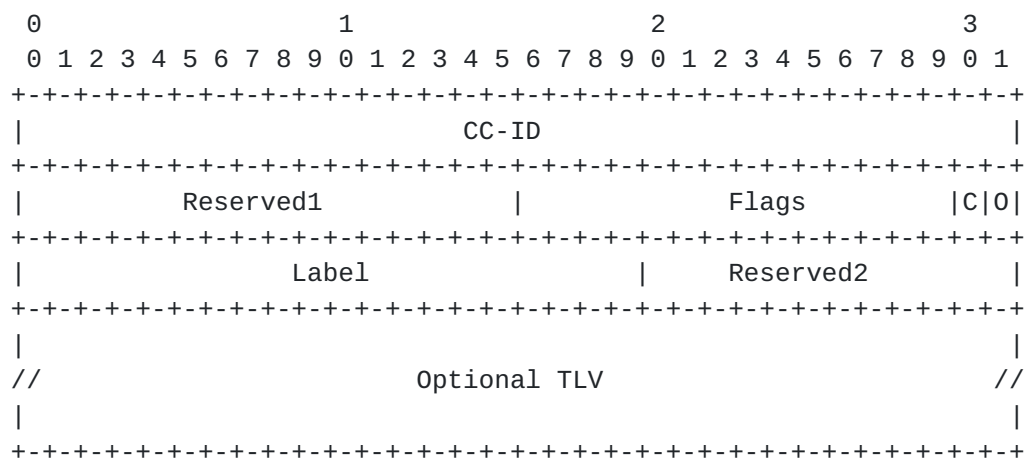   Figure 8: CCI Object

   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 [I-D.gont-numeric-ids-sec-considerations] 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 receive.

   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 this
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 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 =
TBD5 ("PCECC failure") and Error Value = TBD15 ("Invalid next-hop
information").

Further, this document specifies a new TLV called LINKLOCAL-
IPV6-ADDRESS TLV to describe an IPv6 adjacency for an interface that
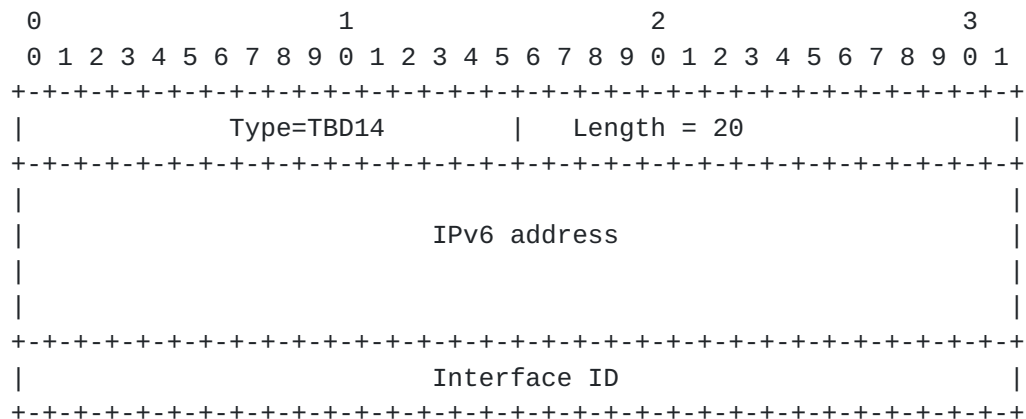does not have a global IPv6 address assigned.

```
 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 = 20               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                         IPv6 address                          |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Interface ID                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 9: LINKLOCAL-IPV6-ADDRESS TLV

The type of the LINKLOCAL-IPV6-ADDRESS TLV is TBD14 and it has a
fixed length of 20 octets. The value portion of the TLV includes:

o IPv6 address: A 128-bit IPv6 address of the node.

o Interface ID: A 32-bit identifier assigned to the link.

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

   As per [RFC8283], the security considerations for a PCE-based
   controller is 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 NEs.

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

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

9.1. Malicious PCE

In this extension, the PCE has complete control over the PCC to
download/remove the labels and can cause the LSP's 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 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=TBD5 (PCECC failure) and
Error-value=TBD6 (Label out of range).

9.2. 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 requests 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.

10. Manageability Considerations

10.1. 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 path setup type.
They are also applicable to the PCECC path setup types. Further,
Section 6.2 of [RFC8664] describe the migration steps when path setup
type of an existing LSP is changed.

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

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

   o An implementation SHOULD allow the operator to view whether the
      PCEP speaker sent the PCECC PST capability to its peer.

   o An implementation SHOULD allow the operator to view whether the
      peer sent the PCECC PST capability.

   o An implementation SHOULD allow the operator to view whether the
      PCECC PST is enabled on a PCEP session.

   o 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.

   o 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.).

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 value
   within the "PCEP TLV Type Indicators" sub-registry of the PCEP
   Numbers registry:

```
        Value           Meaning                       Reference
        TBD14           LINKLOCAL-IPV6-ADDRESS TLV This document
```

[11.2](#). **PATH-SETUP-TYPE-CAPABILITY Sub-TLV Type Indicators**

   [RFC8408] requested the 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](#). **PCECC-CAPABILITY sub-TLV's Flag field**

   This document defines the PCECC-CAPABILITY sub-TLV and requests that
   IANA to create a new sub-registry to manage the value of the PCECC-
   CAPABILITY sub-TLV's 32-bits 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

   Currently, there is one allocation in this registry.

```
        Bit             Name                  Reference
        31              Label                 This document
        0-30            Unassigned            This document
```

[11.4](#). **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  This document
                         set up using PCECC mode
```

11.5. 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
```

11.6. 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 for MPLS
   Label". 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  This document
                         is out-label
```

11.7. 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
   ----------   -------
   6            Mandatory Object missing.

                Error-value = TBD11 :                CCI object missing
```

```
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


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.
              Error-value = TBD15 :             Invalid next-hop
                                                information.
```

[12]. 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 I-D 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 GENART review.

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

[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>.

[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>.

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>.

   [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>.

   [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>.

   [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>.

   [I-D.ietf-teas-pcecc-use-cases]
              Li, Z., Khasanov, B., Dhody, D., Zhao, Q., 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-06 (work in progress), September 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-15 (work in progress), October 2020.

   [I-D.ietf-pce-pcep-extension-pce-controller-sr]
              Li, Z., Peng, S., Negi, M., 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.",
              draft-ietf-pce-pcep-extension-pce-controller-sr-00 (work
              in progress), December 2020.

   [I-D.dhody-pce-pcep-extension-pce-controller-srv6]
              Li, Z., Peng, S., Geng, X., and M. Negi, "PCEP Procedures
              and Protocol Extensions for Using PCE as a Central
              Controller (PCECC) for SRv6", draft-dhody-pce-pcep-
              extension-pce-controller-srv6-05 (work in progress),
              November 2020.

   [I-D.li-pce-controlled-id-space]
              Li, C., Chen, M., Wang, A., Cheng, W., and C. Zhou, "PCE
              Controlled ID Space", draft-li-pce-controlled-id-space-07
              (work in progress), October 2020.

   [I-D.gont-numeric-ids-sec-considerations]
              Gont, F. and I. Arce, "Security Considerations for
              Transient Numeric Identifiers Employed in Network
              Protocols", draft-gont-numeric-ids-sec-considerations-06
              (work in progress), December 2020.

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
   Old Dog Consulting
   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

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

      EMail: lizhenbin@huawei.com

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

      EMail: pengshuping@huawei.com

      Mahendra Singh Negi
      RtBrick Inc
      N-17L, 18th Cross Rd, HSR Layout
      Bangalore, Karnataka 560102
      India

      EMail: mahend.ietf@gmail.com

      Quintin Zhao
      Etheric Networks
      1009 S CLAREMONT ST
      SAN MATEO, CA 94402
      USA

      EMail: qzhao@ethericnetworks.com

      Chao Zhou
      HPE

      EMail: chaozhou_us@yahoo.com