

Path Computation Element  
Internet-Draft  
Intended status: Informational  
Expires: 25 June 2026

M. Koldychev  
S. Sivabalan  
Ciena Corporation  
S. Peng  
Huawei Technologies  
D. Achaval  
Nokia  
H. Kotni  
Juniper Networks, Inc  
A. Stone  
Nokia  
22 December 2025

PCEP Operational Clarification  
draft-ietf-pce-operational-02

## Abstract

This document clarifies certain operational behavior aspects of the PCEP protocol. The content of this document has been compiled based on several interop exercises.

## Discussion Venues

This note is to be removed before publishing as an RFC.

Discussion of this document takes place on the Path Computation Element mailing list ([pce@ietf.org](mailto:pce@ietf.org)), which is archived at <https://mailarchive.ietf.org/arch/browse/pce/>.

Source for this draft and an issue tracker can be found at <https://github.com/ietf-wg-pce/draft-ietf-pce-operational>.

## Status of This Memo

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

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

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

This Internet-Draft will expire on 25 June 2026.

## Copyright Notice

Copyright (c) 2025 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 Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

## Table of Contents

1. Introduction . . . . .	2
2. Requirements Language . . . . .	3
3. Terminology . . . . .	3
4. PCEP LSP Database . . . . .	4
4.1. Structure . . . . .	4
4.2. Synchronization . . . . .	5
4.3. Make before Break (MBB) . . . . .	6
4.3.1. Successful MBB . . . . .	6
4.3.2. Aborted MBB . . . . .	7
5. PCEP Association Database . . . . .	8
5.1. LSPs in same Association . . . . .	8
5.2. Switch Association during MBB . . . . .	9
6. Computation Constraints . . . . .	11
7. Overloaded PCE . . . . .	11
8. Security Considerations . . . . .	12
9. Managability Considerations . . . . .	12
10. IANA Considerations . . . . .	12
11. Normative References . . . . .	12
Acknowledgments . . . . .	12
Contributors . . . . .	12
Authors' Addresses . . . . .	13

## 1. Introduction

Due to different interpretations of PCEP standards, it was found that implementations often had to adjust their behavior in order to interoperate. The current document serves to clarify certain aspects of PCEP to make it easier to produce interoperable implementations of PCEP.

## 2. 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. Terminology

The following terminologies are used in this document:

PCC: Path Computation Client. Any client application requesting a path computation to be performed by a Path Computation Element.

PCE: Path Computation Element. An entity (component, application, or network node) that is capable of computing a network path or route based on a network graph and applying computational constraints.

PCEP: Path Computation Element Protocol.

MBB: Make-Before-Break. A procedure during which the head-end of a traffic-engineered path wishes to move traffic to a new path without losing any traffic, by first "making" a new path and then "breaking" the old path.

Association parameters: As described in [RFC8697], the combination of the mandatory fields Association type, Association ID and Association Source in the ASSOCIATION object uniquely identify the association group. If the optional TLVs - Global Association Source or Extended Association ID are included, then they are included in combination with mandatory fields to uniquely identify the association group.

Association information: As described in [RFC8697], the ASSOCIATION object could also include other optional TLVs based on the association types, that provides 'information' related to the association type.

ERO: Explicit Route Object is the path of the LSP encoded into a PCEP object. In this document, an empty ERO object, i.e., without any subobjects, is represented with notation "ERO={}". An ERO object containing a given sequence of subobjects is represented as "ERO={A}".

PCRPT-LSP-DB: PCEP Reported Label Switched Path Database. A logical datastore that captures the reported state information of Label Switched Paths (LSPs) within a PCEP speaker. This term is not defined in the PCE architecture; however, it is used in this document to describe how a PCEP speaker internally maintains LSP-related state information reported via PCRpt messages.

EXTENDED-LSP-DB: Extended Label Switched Path Database. An implementation-specific logical datastore used to capture information related to a Label Switched Path. It may be keyed using the same identifiers as the PCRPT-LSP-DB. This term is not defined in the PCE architecture; however is used in this document to refer to a conceptual datastore that can include additional attributes—such as desired state, telemetry data, and other information not defined within IETF PCE working group documents.

PLSP-ID (Path LSP Identifier): Introduced in [RFC8231]. A unique identifier used in PCEP to distinguish a specific LSP between a PCC and a PCE which is constant for the lifetime of a PCEP session.

#### 4. PCEP LSP Database

This document uses the concept of the PCRPT-LSP-DB, a database of actual LSP state in the network, to illustrate the internal state of PCEP speakers in response to Pcrpt messages.

Note that the term "LSP", which stands for "Label Switched Path", if taken too literally, would restrict the discussion to the MPLS dataplane only. In this document, the term "LSP" is applied to non-MPLS paths as well, to avoid renaming the term. Alternatively, the term "LSP" could be replaced with "Instance".

##### 4.1. Structure

The distinction between Tunnels and LSPs in the PCRPT-LSP-DB is essential for accurately modeling state information in PCEP, including procedures such as make-before-break, and for maintaining consistent state management across PCEP speakers.

The PCRPT-LSP-DB contains two types of information: LSPs and Tunnels. An LSP is identified by the LSP-IDENTIFIERS TLV, while a Tunnel is identified by the PLSP-ID in the LSP object and/or the SYMBOLIC-NAME (see [RFC8231]).

A Tunnel may or may not correspond to an actual tunnel on the router. For example, working and protect paths can be implemented as a single tunnel interface, but in PCEP, these are represented as two different Tunnels with different PLSP-IDs.

An LSP can be viewed as an instance of a Tunnel. In steady state, a Tunnel typically has a single LSP; however, during make-before-break procedures (see [RFC3209]), multiple LSPs may exist simultaneously to represent both the new and old instances during the transition.

#### 4.2. Synchronization

Since the PCRPT-LSP-DB contains PCEP-specific information such as PLSP-IDs, which remain constant for the duration of a PCEP session, both the PCE and PCC maintain their own local copies of the PCRPT-LSP-DB. The PCE PCRPT-LSP-DB is only modified by PCRpt messages, no other PCEP message may modify the PCE PCRPT-LSP-DB. The PCC PCRPT-LSP-DB is built from actual forwarding state that PCC has installed. PCC uses PCRpt messages to synchronize its local PCRPT-LSP-DB to the PCE.

The PCE is intended to act based on the most recent state available in the PCRPT-LSP-DB, which represents the actual state of Label Switched Paths (LSPs) in the network. This does not preclude the PCE from leveraging information obtained outside the PCRPT-LSP-DB. For example, implementation-specific mechanisms may be used to collect traffic statistics that can be considered during path computation. Such information is not part of the PCRPT-LSP-DB itself, but may reference it. Additional data such as desired state, telemetry, or operator-intended configurations—may be maintained in a separate, implementation-specific logical datastore referred to as the EXTENDED-LSP-DB.

Both the PCE and PCC maintain a PCRPT-LSP-DB that reflects the reported (actual) state of LSPs. Implementations may choose to store desired state information or other attributes in the EXTENDED-LSP-DB. It is important to note that the PCRPT-LSP-DB reflects only the live view of the network and does not imply alignment with the desired state.

For example, in the case of a PCE-initiated LSP, a change in the LSP configuration made by an operator represents a modification to the desired state. However, the actual state does not change until the PCE sends a PCInit or PCUpd message to the PCC. Upon receipt of such a message, the PCC may act on the request, update its local PCRPT-LSP-DB, and generate a PCRpt message to inform the PCE of the change. The PCE then updates its own PCRPT-LSP-DB accordingly. Once this exchange is complete, the PCRPT-LSP-DBs on both the PCC and the PCE are synchronized.

### 4.3. Make before Break (MBB)

Due to variations in how different implementations interpret or handle MBB procedures—sometimes resulting in incorrect message processing or misinterpretation—the following section provides illustrative examples to clarify expected MBB behavior.

#### 4.3.1. Successful MBB

Below is an example of performing MBB to switch a Tunnel from one path to another. The path encoded into the ERO object is represented as  $ERO=\{A\}$  and  $ERO=\{B\}$ .

PCC has an existing LSP in UP state, with LSP-ID=2. PCC sends  $PCRpt(R-FLAG=0, PLSP-ID=100, LSP-ID=2, ERO=\{A\}, OPER-FLAG=UP)$ .

TUNNEL	LSP
PLSP-ID=100	LSP-ID=2, $ERO=\{A\}$ , OPER=UP

Figure 1: Content of LSP DB

PCC initiates the MBB procedure by creating a new LSP with LSP-ID=3. It does not matter what triggered the creation of the new LSP, it could have been due to a new path received via PCUpd (if the given Tunnel is delegated), or it could have been local computation on the PCC (if the Tunnel is locally computed on the PCC), or it could have been a change in configuration on the PCC (if the Tunnel's path is explicitly configured on the PCC). It is important to emphasize that the procedure for updating the PCRPT-LSP-DB is common, regardless of the trigger that caused the change.

PCC sends  $PCRpt(R-FLAG=0, PLSP-ID=100, LSP-ID=3, ERO=\{B\}, OPER-FLAG=UP)$ .

TUNNEL	LSP
PLSP-ID=100	LSP-ID=2, $ERO=\{A\}$ , OPER=UP
	LSP-ID=3, $ERO=\{B\}$ , OPER=UP

Figure 2: Content of LSP DB

After traffic has successfully switched to the new LSP, the PCC cleans up the old LSP. PCC sends PCRpt(R-FLAG=1, PLSP-ID=100, LSP-ID=2).

TUNNEL	LSP
PLSP-ID=100	LSP-ID=3, ERO={B}, OPER=UP

Figure 3: Content of LSP DB

#### 4.3.2. Aborted MBB

The MBB process can abort when the newly created LSP is destroyed before it is installed as traffic carrying. This scenario is described below.

PCC has an existing LSP in UP state, with LSP-ID=2. PCC sends PCRpt(R-FLAG=0, OPER-FLAG=UP, PLSP-ID=100, LSP-ID=2).

TUNNEL	LSP
PLSP-ID=100	LSP-ID=2, OPER=UP

Figure 4: Content of LSP DB

MBB procedure is initiated, a new LSP is created with LSP-ID=3. LSP is currently being established, so its oper state is DOWN. PCC sends PCRpt(R-FLAG=0, OPER-FLAG=DOWN, PLSP-ID=100, LSP-ID=3).

TUNNEL	LSP
PLSP-ID=100	LSP-ID=2, OPER=UP
	LSP-ID=3, OPER=DOWN

Figure 5: Content of LSP DB

MBB procedure is aborted. PCC sends PCRpt(R-FLAG=1, PLSP-ID=100, LSP-ID=3).

TUNNEL	LSP
PLSP-ID=100	LSP-ID=2, OPER=UP

Figure 6: Content of LSP DB

## 5. PCEP Association Database

PCEP Association is the instantiate of a group containing at least one LSP.

The PCE ASSO DB is populated by PCRpt messages and/or via configuration on the PCE itself. An Association is identified by the Association Parameters. As the Association Parameters contain many fields, all fields are grouped into a single value for convenience. The notation ASSO\_PARAM=A and ASSO\_PARAM=B is used to refer to different PCEP Associations: A and B, respectively.

### 5.1. LSPs in same Association

The following example illustrates how LSPs join the same Association.

PCC creates the first LSP. PCC sends PCRpt(R-FLAG=0, PLSP-ID=100, LSP-ID=1, ASSO\_PARAM=A, ASSO\_R\_FLAG=0).

ASSO	LSP
ASSO_PARAM=A	PLSP-ID=100, LSP-ID=1

Figure 7: Content of PCE ASSO DB

PCC creates the second LSP. PCC sends PCRpt(R-FLAG=0, PLSP-ID=200, LSP-ID=1, ASSO\_PARAM=A, ASSO\_R\_FLAG=0).

ASSO	LSP
ASSO_PARAM=A	PLSP-ID=100, LSP-ID=1
	PLSP-ID=200, LSP-ID=1

Figure 8: Content of PCE ASSO DB

PCC updates the first LSP. As [RFC8697] indicates, subsequent Pcrpt should include only the associations that are being modified or removed. Therefore it is optional as to send the ASSOCIATION object in this PCRpt, since the LSP is already in the Association. PCC sends PCRpt(R-FLAG=0, PLSP-ID=100, LSP-ID=1). The content of the PCE ASSO DB is unchanged. Note that the PCC sends the ASSOCIATION OBJECT in the first PCRpt during SYNC state, even if it has already issued a PCRpt with the association object sometime in the past with this PCE. The synchronization steps outlined in [RFC8697] are to be followed.

ASSO	LSP
ASSO_PARAM=A	PLSP-ID=100, LSP-ID=1
	PLSP-ID=200, LSP-ID=1

Figure 9: Content of PCE ASSO DB

PCC decides to delete the second LSP. PCC sends PCRpt(R-FLAG=1, PLSP-ID=200, LSP-ID=1).

ASSO	LSP
ASSO_PARAM=A	PLSP-ID=100, LSP-ID=1

Figure 10: Content of PCE ASSO DB

PCC decides to remove the first LSP from the Association, but not delete the LSP itself. PCC sends PCRpt(R-FLAG=0, PLSP-ID=100, LSP-ID=1, ASSO\_PARAM=A, ASSO\_R\_FLAG=1). The PCE ASSO DB is now empty.

ASSO	LSP
-	

Figure 11: Content of PCE ASSO DB

## 5.2. Switch Association during MBB

The following example illustrates how a Tunnel goes through MBB and switches from Association A to Association B.

Each new LSP (identified by the LSP-ID) does not inherit the Association membership of any previous LSPs within the same Tunnel. This is so that a Tunnel can have two LSPs that are in different Associations, this may be done when switching from one Association to another.

PCC creates the first LSP. PCC sends PCRpt(R-FLAG=0, PLSP-ID=100, LSP-ID=1, ASSO\_PARAM=A, ASSO\_R\_FLAG=0).

ASSO	LSP
ASSO_PARAM=A	PLSP-ID=100, LSP-ID=1

Figure 12: Content of PCE ASSO DB

PCC creates the MBB LSP in a different Association. PCC sends PCRpt(R-FLAG=0, PLSP-ID=100, LSP-ID=2, ASSO\_PARAM=B, ASSO\_R\_FLAG=0).

ASSO	LSP
ASSO_PARAM=A	PLSP-ID=100, LSP-ID=1
ASSO_PARAM=B	PLSP-ID=100, LSP-ID=2

Figure 13: Content of PCE ASSO DB

PCC deletes the old LSP. PCC sends PCRpt(R-FLAG=1, PLSP-ID=100, LSP-ID=1).

ASSO	LSP
ASSO_PARAM=B	PLSP-ID=100, LSP-ID=2

Figure 14: Content of PCE ASSO DB

## 6. Computation Constraints

As well as reporting any state change in the network on a PCRpt message, a PCC may also change the parameters of a delegated LSP. For example, it may remove or modify the computation constraints that it wishes the PCE to apply as it computes any updated paths in the future. For any PCEP object that specifies a path computation constraint and that does not have a defined explicit removal flag, the absence of that entire object on a repeat or follow-up PCRpt message indicates removal of the constraint previously specified by that object. For example, suppose the first PCRpt contains an LSPA object with some affinity constraints. Then if a subsequent PCRpt does not contain an LSPA object, then this means that the previously specified affinity constraints do not apply anymore. Same applies to all PCEP objects, like METRIC, BANDWIDTH, etc., which do not have an explicit flag for removal. This simply ensures that it is possible to remove a constraint without using an explicit removal flag.

## 7. Overloaded PCE

[RFC5440] defines the concept of an Overloaded PCE and explains how a PCE may signal to a PCC that it is congested, instructing that "no requests should be sent to that PCE until the overload state is cleared."

In the case of an overloaded PCE, a PCC implementation could choose to wait for the PCE to no longer be overloaded or instead send a PcReq to a backup, non-overloaded PCE instead.

[RFC8231] builds upon [RFC5440] by introducing the concept of a Stateful PCE, which allows delegation of LSP control to a single PCE. However, it does not address the case of an overloaded PCE. According to [RFC8231], a PCE maintains delegation until it is revoked by the PCC or returned back to PCC by the PCE. The PCC may revoke delegation and re-assign it to another PCE.

As a result, a PCE in an overload state still retains LSP delegation. For PCC-initiated LSPs, the PCC may revoke delegation from the overloaded PCE and maintain delegation for itself or delegate it to another PCE. For PCE-initiated LSPs, since the PCC cannot revoke delegation as per [RFC8281], the overloaded PCE may return the delegation to the PCC.

The PCE will continue to send PCRpt messages to PCC even though it may indicate it is overloaded, otherwise the the PCRPT-LSP-DB on PCC may go out of sync.

## 8. Security Considerations

None at this time.

## 9. Managability Considerations

None at this time.

## 10. IANA Considerations

None at this time.

## 11. 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/rfc/rfc2119>>.
- [RFC3209] Awduche, D., Berger, L., Gan, D., Li, T., Srinivasan, V., and G. Swallow, "RSVP-TE: Extensions to RSVP for LSP Tunnels", RFC 3209, DOI 10.17487/RFC3209, December 2001, <<https://www.rfc-editor.org/rfc/rfc3209>>.
- [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/rfc/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/rfc/rfc8231>>.
- [RFC8697] Minei, I., Crabbe, E., Sivabalan, S., Ananthakrishnan, H., Dhody, D., and Y. Tanaka, "Path Computation Element Communication Protocol (PCEP) Extensions for Establishing Relationships between Sets of Label Switched Paths (LSPs)", RFC 8697, DOI 10.17487/RFC8697, January 2020, <<https://www.rfc-editor.org/rfc/rfc8697>>.

## Acknowledgments

The authors would like to thank Adrian Farrel and Tom Petch for useful review and feedback. comments.

## Contributors

Dhruv Dhody  
Huawei Technologies  
Email: dhruv.ietf@gmail.com

Samuel Sidor  
Cisco Systems  
Email: ssidor@cisco.com

Mahendra Singh Negi  
RtBrick Inc  
Email: mahend.ietf@gmail.com

#### Authors' Addresses

Mike Koldychev  
Ciena Corporation  
Email: mkoldych@proton.me

Siva Sivabalan  
Ciena Corporation  
Email: ssivabal@ciena.com

Shuping Peng  
Huawei Technologies  
Email: pengshuping@huawei.com

Diego Achaval  
Nokia  
Email: diego.achaval@nokia.com

Hari Kotni  
Juniper Networks, Inc  
Email: hkotni@juniper.net

Andrew Stone  
Nokia  
Email: andrew.stone@nokia.com