PCE Working Group
Internet-Draft
Updates: 8231, 8664, 8281 (if approved)
Intended status: Standards Track
Expires: 24 July 2026

A. S. (Editor)
Nokia
M. Koldychev
S. Sivabalan
Ciena
D. Achavel
Nokia
S. Peng
Huawei Technologies
H. Kotni
Juniper Networks, Inc
S. Sidor
Cisco

20 January 2026

Amendments to Stateful PCE Communication Protocol (PCEP)
draft-many-pce-stateful-amendment-03

Abstract

This document updates RFC8231, RFC8664 and RFC8281 to reflect operationalized implementations and define optimizations in the PCEP protocol.

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 24 July 2026.

Copyright Notice

Table of Contents

1. Introduction

The PCEP protocol has evolved from a stateless protocol [RFC5440] to a stateful protocol [RFC8231], incorporating numerous extensions.

During interoperability testing it was observed that various implementations have implemented optimizations in the protocol. This document serves to optimize the original procedure in [RFC8231] to optionally drop the PCReq and PCReply exchange, which greatly simplifies implementation and optimizes the protocol.

In addition, [RFC8664] introduced extensions for Segment Routing and the encoding of segments in the ERO and RRO objects in PCEP. This document serves as an update to [RFC8664] to permit the exclusion of the RRO object for Segment Routed paths.

Lastly, [RFC8281] describes two mechanisms for handling orphaned LSPs, one of which requires a PCE to request delegation of the orphaned LSP. However, this mechanism is incompletely specified, which has led most implementations to follow PCC originated redelegation when an LSP becomes orphaned. This document updates [RFC8281] to clarify the ambiguity and promote interoperability by mandating that the PCC attempt to redelegate orphaned LSPs.

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 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 may internally maintain 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 but 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.

3. Stateful Bringup

[RFC8231] Section 5.8.2 allows delegation of an LSP in an operationally down state, but at the same time mandates the use of PCReq before sending PCRpt. This document clarifies that sending PCReq is optional.

The process of sending PCReq before PCRpt is referred to in this document as "stateless bringup". In practice, stateless bringup introduces overhead and the PcRpt sent from PCC cannot be enforced by the PCE, because a stateless PCE is not required to maintain any per-LSP state about previous PCReq messages. It has been observed that many implementations choose to ignore this requirement and send the PCRpt directly, without first sending a PCReq. Although this behavior is not compliant with [RFC8231], it offers message processing advantages and simplifications. As a result, this document updates [RFC8231].

The adoption of stateful PCE does not eliminate the utility of stateless PCEP. A characteristic of stateless PCEP is that PCReq messages does not require altering the LSP path state information in the PCE. As a result, PCReq messages can be used in scenarios such

as OAM functions (e.g., ping and traceroute), where it is necessary to probe the network topology without impacting existing LSPs and LSP state management in the PCE.

This document uses the concept of a PCRPT-LSP-DB to represent the database of actual LSP state in the network, as reported by PCCs. It is used to illustrate the internal state maintained by PCEP speakers in response to PCRpt messages. This datastore is modified only by PCRpt messages. In contrast, additional information that a PCE implementation may maintain such as desired state, policy metadata, or telemetry is considered part of the EXTENDED-LSP-DB. The EXTENDED-LSP-DB is an implementation-specific logical store which is outside the scope of this document.

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".

3.1. Updates to RFC 8231 - Stateful bringup

[RFC8231] Section 5.8.2, says "The only explicit way for a PCC to request a path from the PCE is to send a PCReq message. The PCRpt message MUST NOT be used by the PCC to attempt to request a path from the PCE." This document updates [RFC8231] to remove the quoted text.

As part of the new bringup procedure, the PCC MAY delegate an empty LSP (no ERO or empty ERO) to the PCE and then wait for the PCE to send a PCUpd, without first sending a PCReq. This process is referred to as "stateful bringup". The PCE MUST support the original stateless bringup for backward compatibility.

An example of stateful bringup follows. In this example, the PCC starts by using an LSP-ID of 0. The value 0 does not hold any special meaning; any other 16-bit value could have been used.

PCC has no LSP yet, but wants to establish a path. PCC sends PCRpt(R-FLAG=0, D-flag=1, OPER-FLAG=DOWN, PLSP-ID=100, LSP-ID=0, ERO={}).

TUNNEL	LSP
PLSP-ID=100	OLSP-ID=0, D-flag=1, OPER=DOWN, ERO={}

Table 1: Content of LSP DB after first PcRpt

PCC received a PCUpd from the PCE and has decided to install the ERO={A} from that PCUpd. PCC sends PCRpt(R-FLAG=0, D-flag=1, OPER-FLAG=UP, PLSP-ID=100, LSP-ID=0, ERO={A}).

TUNNEL	LSP
PLSP-ID=100	LSP-ID=0, D-flag=1, OPER=UP, ERO={A}

Table 2: Content of LSP DB after PcUpd

3.2. Backwards Compatibility

The stateful bringup mechanism is compatible with legacy PCEP implementations. The PCE continues to support stateless bringup (via PCReq) for legacy PCCs. Supporting stateful bringup does not require introducing new behavior on the PCE, since, as previously noted, a PCE implementation only modifies the conceptual PCRPT-LSP-DB state based on PcRpt messages. Therefore, regardless of whether a PCReq has been received, the PCE processes the PCRpt in the same manner.

4. Updates to RFC 8664 - Use of SR-RRO and SRv6-RRO objects

[RFC8231] defines a PCRpt message which contains <intended-path> known as the ERO object and <actual-path> known as the RRO object. [RFC8664] defines SR-ERO and SR-RRO sub-objects for SR-TE LSPs. [RFC9603] further defines SRv6-ERO and SRv6-RRO sub-objects for SRv6-TE paths.

In practice RRO data is the result of signalling via a protocol such as RSVP-TE, which allows collection of per-hop information along the path. The ERO and RRO values may be different as the path encoded in the ERO may differ than the RRO such as during protection conditions or if the ERO contains loose hops which are expanded upon. As Segment Routing LSP does not perform any signalling, the values of an SR-ERO/SRv6-ERO and SR-RRO/SRv6-RRO (respectively) are in practice the same, therefore some implementations have omitted the RRO when reporting a SR-TE LSP while others continue to send both ERO and RRO values.

This document updates [RFC8664] by clarifying and relaxing requirement for both an ERO and RRO object to exist for SR-TE paths. If both ERO and RRO are present for the same LSP, it SHOULD be interpreted as the RRO being the actual path the LSP is taking but MAY interpret only the ERO as the actual path. In the absence of RRO a PCE MUST interpret the ERO as the actual path for the LSP. Until SR-TE introduces some form of signaling similar to RSVP-TE, the use of RRO is discouraged for SR-TE LSPs.

4.1. Backwards Compatibility

The update to [RFC8664] permitting PCC to omit carrying SR-RRO/SRv6-RRO may create interoperability problems between different implementations of newer PCC and a legacy PCE. It is possible that an implementation of PCE which requires reading the SR-RRO/SRv6-RRO, may result with incomplete data processing on the PCE for the LSP. However, as this document is attempting to reflect operationalized implementations, PCE implementations are likely already capable of falling back to processing the SR-ERO/SRv6-ERO objects.

5. Updates to RFC 8281 - Orphaned LSP without delegation

[RFC8281] Section 6, describes two mechanisms for handling the case when an LSP becomes orphaned. The relevant text is as follows:

"The PCC MAY attempt to redelegate an orphaned LSP by following the procedures of [RFC8231]. Alternatively, if the orphaned LSP was PCE-initiated, then a PCE MAY obtain control over it, as follows"

The second mechanism goes on to describe the messaging procedures by which a PCE may obtain control of an orphaned PCE-initiated LSP. However, the document does not define how a PCE determines whether a PCC expects it to take action to obtain control, nor does it specify when such action should be taken. This ambiguity is problematic in deployments with backup or redundant PCEs, as a PCE may be completely unaware of the current delegation status of an LSP with respect to another PCE.

To address this issue, this document updates the previously quoted text in [RFC8281] as follows:

"PCC MUST attempt to redelegate an orphaned LSP to a connected PCE by following the procedures of [RFC8231] and in accordance with local policy."

5.1. Backwards Compatibility

The update to [RFC8281] mandating that a PCC MUST attempt to redelegate orphaned LSPs introduces considerations for interoperability between updated and legacy implementations.

PCC Perspective: A PCC implementing this document MUST attempt to redelegate orphaned LSPs to an active PCE. From the perspective of a legacy PCE, these redelegations will appear as standard [RFC8231] procedures. Since legacy PCEs are already capable of processing redelegation of LSPs driven from PCC, this update is backwards compatible.

PCE Perspective: For a PCE implementing this document, the primary change is the shift in expectation regarding PCC behavior. A PCE operates with the expectation that the PCC will initiate redelegation. However, if the PCC is a legacy implementation that does not perform the redelegation, the PCE MAY apply local policy to decide when to revert to [RFC8281] procedures and explicitly request delegation of orphaned LSPs.

Capability Advertisement: A capability mechanism to indicate support for this document may be defined in a future revision. This capability is currently informational; it serves to notify the PCE that the PCC explicitly supports the mandated redelegation behavior. This allows the PCE to distinguish between a PCC that is expected to redelegate (per this document) and a legacy PCC, requiring the PCE to follow local policy and therefore MAY explicitly request delegation of orphaned LSPs.

6. Security Considerations

The security considerations described in [RFC8231] and [RFC8281] apply to this document.

7. Managability Considerations

TODO

8. IANA Considerations

This document has no IANA actions.

9. References

9.1. 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/rfc/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/rfc/rfc5440>.

[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>.

[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/rfc/rfc8281>.

[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/rfc/rfc8664>.

9.2. Informative References

[I-D.draft-koldychev-pce-operational] Koldychev, M., Sivabalan, S., Peng, S., Achaval, D., Kotni, H., and A. Stone, "PCEP Operational Clarification", Work in Progress, Internet-Draft, draft-koldychev-pce-operational-09, 28 March 2025, <https://datatracker.ietf.org/doc/html/draft-koldychev-pce-operational-09>.

[RFC9603] Li, C., Ed., Kaladharan, P., Sivabalan, S., Koldychev, M., and Y. Zhu, "Path Computation Element Communication Protocol (PCEP) Extensions for IPv6 Segment Routing", RFC 9603, DOI 10.17487/RFC9603, July 2024, <https://www.rfc-editor.org/rfc/rfc9603>.

Acknowledgments

The authors would like to thank Adrian Farrel for useful review comments on [I-D.draft-koldychev-pce-operational] which have been carried over and have been applied into this document. The authors would also like to thank Dhruv Dhody for content discussion and review.

Authors' Addresses

Andrew Stone (Editor)
Nokia
Email: andrew.stone@nokia.com

Mike Koldychev
Ciena
Email: mkoldych@proton.me

Siva Sivabalan
Ciena
Email: ssivabal@ciena.com

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

Shuping Peng
Huawei Technologies
Email: pengshuping@huawei.com

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

Samuel Sidor
Cisco
Email: ssidor@cisco.com