PCE Working Group
Internet-Draft
Intended status: Standards Track
Expires: 30 November 2025

H. Zheng, Ed.
Huawei Technologies
S. Litkowski
Cisco
S. Sivabalan
Ciena Corporation
C. Li
Huawei Technologies

29 May 2025

Procedures for Communication between Stateful Path Computation Elements
draft-ietf-pce-state-sync-12

Abstract

The Path Computation Element (PCE) Communication Protocol (PCEP) provides mechanisms for PCEs to perform path computation in response to a Path Computation Client (PCC) request. The Stateful PCE extensions allow stateful control of Multi-Protocol Label Switching (MPLS) Traffic Engineering (TE) and Generalized Multi-Protocol Label Switching (GMPLS) Label Switched Paths (LSPs) using PCEP.

A Path Computation Client (PCC) can synchronize LSP state information to a Stateful Path Computation Element (PCE). A PCC can have multiple PCEP sessions towards multiple PCEs. In some use cases, inter-PCE stateful communication can bring additional resiliency in the design, for instance, when some PCC-PCE session fails.

This document describes the procedures to allow stateful communication between PCEs for various use cases and also the procedures to prevent computational loops.

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 30 November 2025.

Table of Contents

1. Introduction and Problem Statement

The Path Computation Element communication Protocol (PCEP) [RFC5440] provides mechanisms for Path Computation Elements (PCEs) to perform path computations in response to Path Computation Clients' (PCCs) requests.

A stateful PCE [RFC8231] is capable of considering, for the purposes of path computation, not only the network state in terms of links and nodes (referred to as the Traffic Engineering Database or TED) but also the status of active services (previously computed paths), and currently reserved resources, stored in the Label Switched Paths Database (LSP-DB).

[RFC8051] describes general considerations for a stateful PCE deployment and examines its applicability and benefits, as well as its challenges and limitations through a number of use cases.

A PCC can synchronize LSP state information to a Stateful PCE. The stateful PCE extension allows a redundancy scenario where a PCC can have redundant PCEP sessions towards multiple PCEs. In such a case, a PCC gives control of an LSP to a single PCE, and only one PCE is responsible for path computation for this delegated LSP. [RFC8231] does not state the procedures related to inter-PCE stateful communication.

There are some use cases, where inter-PCE stateful communication can bring additional resiliency to the design, for instance, when some PCC-PCE session fails. The inter-PCE stateful communication may also provide a faster update of the LSP states when such an event occurs. Finally, when, in a redundant PCE scenario, there is a need to compute a set of paths that are part of a group (so there is a dependency between the paths), there may be some cases where the computation of all paths in the group is not handled by the same PCE: this situation is called a split-brain. This split-brain scenario may lead to computation loops between PCEs or suboptimal path computation.

In the scope of this document, the term ’computation loop’ is used to describe the behaviour of PCEP message exchange looping between PCC and PCE or between PCEs, resulting in frequent path calculations, path reporting and path updates to the network resulting in constant load on the PCE and oscillation of data plane traffic after each subsequent path update.

This document describes the procedures to allow stateful communication between PCEs for various use cases and also the procedures to prevent computational loops.

This section contains illustrative examples to showcase the need for inter-PCE stateful PCEP sessions.

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.

1.2. Reporting LSP Changes

When using a stateful PCE ([RFC8231]), a PCC can synchronize LSP state information to the stateful PCE. If the PCC grants the control of the LSP to the PCE (called delegation [RFC8231]), the PCE can update the LSP parameters at any time.

In a multi-PCE deployment (redundancy, load-balancing...), with the specification defined in [RFC8231], when a PCE makes an update, the PCC is responsible for reporting the LSP parameter updates all PCEs.

This delay may affect the reaction time of the other PCEs if they need to take action after being notified of the LSP parameter change.

Apart from the synchronization from the PCC, it is also useful if there is a synchronization mechanism between the stateful PCEs. As a stateful PCE makes changes to its delegated LSPs, these changes (pending LSPs and the sticky resources [RFC7399]) can be synchronized to the other PCEs.

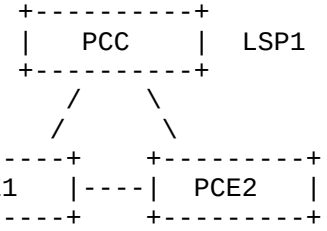


Figure 1: Active and Standby PCEs

In Figure 1, we consider PCE1 is responsible for computing paths for PCC, and PCE2 is standing by. When there is a change in LSP1, the PCC should report to PCE1. From PCE2's perspective, PCC1 reporting the update of LSP1 to PCE2 can be slower than sync from PCE1 to PCE2 (such as when the PCE instances are hosted at the same server).

1.3. Split-Brain

In a resiliency case, a PCC has redundant PCEP sessions towards multiple PCEs. In such a case, a PCC gives control of an LSP to a single PCE only, and only this PCE is responsible for the path computation for the delegated LSP: the PCC achieves this by setting the D flag only towards the active PCE [RFC8231] selected for delegation. The election of the active PCE to delegate an LSP is controlled by each PCC. The PCC usually elects the active PCE by a locally configured policy (by setting a priority). Upon PCEP session failure, or active PCE failure, the PCC may decide to elect a new active PCE by sending a new Path Compute report (PCRpt) message with the D flag set to this new active PCE. When the failed PCE or PCEP session comes back online, it will be up to the implementation whether to revert back to the original primary PCE. Reverting may lead to some disruption on the existing path if computation results

from both PCEs are not exactly the same. By considering a network with multiple PCCs and implementing multiple stateful PCEs for redundancy purposes, it is not likely that all PCCs delegate their LSPs to the same PCE.

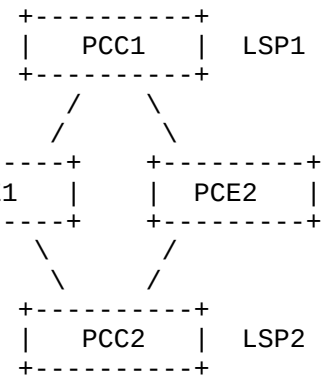


Figure 2: Two PCEs with Shared Responsibility

In the example in Figure 2, we consider that by configuration, both PCCs will first delegate their LSPs to PCE1. So, PCE1 is responsible for computing a path for both LSP1 and LSP2. If the PCEP session between PCC2 and PCE1 fails, PCC2 will delegate LSP2 to PCE2. So PCE1 becomes responsible only for LSP1 path computation while PCE2 is responsible for the path computation of LSP2. When the PCC2-PCE1 session is back online, PCC2 will keep using PCE2 as active PCE (consider no preemption in this example). So the result is a permanent situation where each PCE is responsible for a subset of path computation.

This situation is called a split-brain scenario, as there are multiple computation brains running at the same time while a central computation unit is required in some deployments/use cases.

Further, there are use cases where a particular LSP path computation is linked to another LSP path computation: the most common use case is path disjointness (see [RFC8800]) and Bidirectional LSPs (see [RFC9059]). The set of LSPs that are dependent to each other may start from different head-ends.

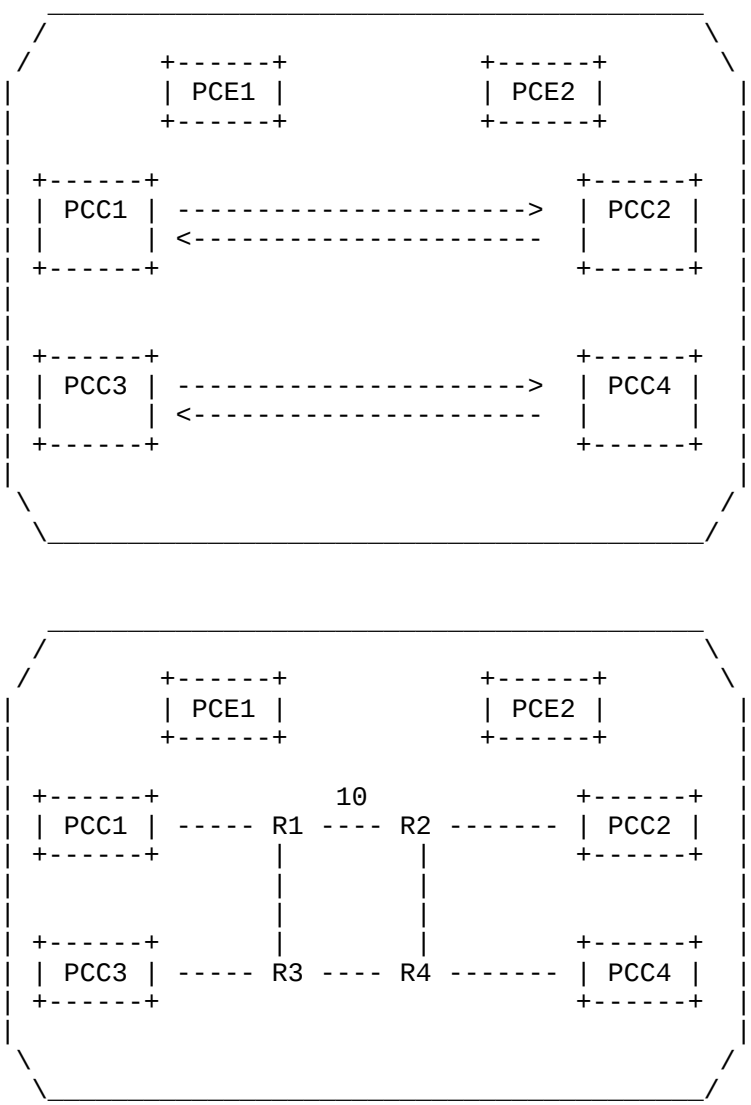


Figure 3: Managing Link-Disjoint LSPs

In Figure 3, the requirement is to create two link-disjoint LSPs: PCC1->PCC2 and PCC3->PCC4. In the topology, all link cost metrics are set to 1 except for the link 'R1-R2' which has a metric of 10. The PCEs are responsible for the path computation and PCE1 is the active primary PCE for all PCCs in the nominal case.

Appendix B provides several scenarios for illustrative purposes. There are many other cases where the solutions defined in this document are also applicable.

1.4. Applicability to H-PCE

[RFC8751] describes general considerations and use cases for the deployment of Stateful PCE(s) using the Hierarchical PCE [RFC6805] architecture. In this architecture, there is a clear need to communicate between a child stateful PCE and a parent stateful PCE. As per [RFC8751], the procedures and extensions as described in Section 3 are also applicable to the H-PCE scenario.

2. Solution

The solution specified in this document is based on:

* The creation of the inter-PCE stateful PCEP session with specific procedures.

* A Primary/Secondary relationship between stateful PCEs.

The solution builds upon the protocol extensions for stateful PCE in [RFC8231], synchronization optimizations in [RFC8232], and PCE-initiation in [RFC8281].

2.1. State-sync Session

This document specifies a mechanism to set up a PCEP session between the stateful PCEs. Creating a PCEP session between PCEs is already enabled for multiple scenarios like the one described in [RFC4655] (multiple PCEs that are handling part of a path computation) and [RFC6805] (hierarchical PCE). However, that earlier work focused only on the sessions between stateless PCEs.

Stateful PCE brings additional features (LSP state synchronization, path update, delegation, ...). Thus some new behaviors need to be defined on the inter-PCE PCEP session.

This inter-PCE PCEP session will allow the exchange of LSP states between PCEs and can help in some scenarios where PCEP sessions are lost between PCCs and PCEs. This inter-PCE PCEP session is called a "state-sync session" in this document.

For example, in the scenario in Figure 4, there is no possibility to compute disjointness as there is no PCE that is aware of both LSPs.

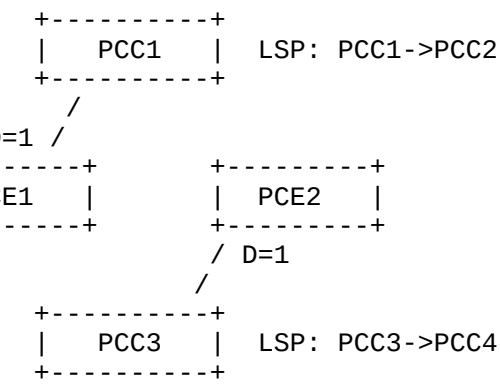


Figure 4: Partitioned Visibility Amongst PCEs

If we add a state-sync session as shown in Figure 5, PCE1 will be able to do state synchronization via PCRpt messages for its LSPs to PCE2 and PCE2 will do the same. All the PCEs will be aware of all LSPs even if a PCC->PCE session is down. PCEs will then be able to compute disjoint paths.

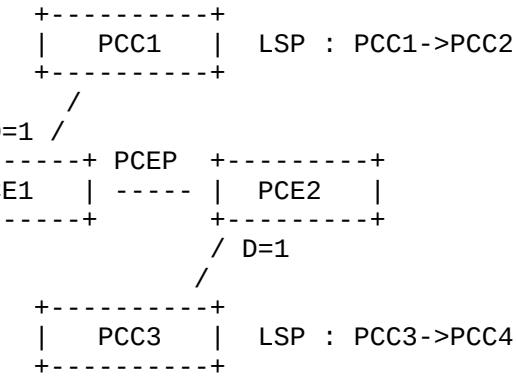


Figure 5: Partitioned Visibility With State Synchronization

The procedures associated with this state-sync session are defined in Section 3.

By just adding this state-sync session, it does not ensure that a path with LSP association-based constraints can always be computed and does not prevent the computation loop, but it increases resiliency and ensures that PCEs will have the state information for

all LSPs. Also, this session will allow a PCE to update the other PCEs providing a faster synchronization mechanism than relying on PCCs only.

2.2. Primary/Secondary Relationship between PCE

As seen in Section 1, performing a path computation in a split-brain scenario (multiple PCEs responsible for computation) may provide a non-optimal LSP placement, no path, or computation loops. To achieve better efficiency, an LSP association constraint-based computation may require that a single PCE performs the path computation for all LSPs in the association group. Note that, it could be all LSPs belonging to a particular association group, or all LSPs from a particular PCC, or all LSPs in the network that need to be delegated to a single PCE based on the deployment scenarios.

This document specifies a mechanism to add a priority mechanism between PCEs to elect a single computing 'primary' PCE. Using this priority mechanism, PCEs can agree on the PCE that will be responsible for the computation for a particular association group, or set of LSPs. The priority could be set per association, per PCC, or for all PCEs. The rest of the text considers the association group as an example.

When a single PCE is performing the computation for a particular association group, no computation loop can happen and an optimal placement will be provided. The other PCEs will only act as state collectors and forwarders.

In the scenario described in Section 2.1, PCE1 and PCE2 will decide that PCE1 will be responsible for the path computation of both LSPs. If we first configure PCC1->PCC2, PCE1 computes the shortest path as it is the only LSP in the disjoint-group that it is aware of: R1->R3->R4->R2->PCC2 (shortest path). When PCC3->PCC4 is configured, PCE2 will not perform computation even if it has delegation but forwards the delegation via PCRpt message to PCE1 through the state-sync session. PCE1 will then perform disjointness computation and will move PCC1->PCC2 onto R1->R2->PCC2 and provides an ERO to PCE2 for PCC3->PCC4: R3->R4->PCC4. The PCE2 will further update the PCC3 with the new path.

3. Procedures and Protocol Extensions

3.1. Opening a state-sync session

3.1.1. Capability Advertisement

A PCE indicates its support of state-sync procedures during the PCEP Initialization phase [RFC5440]. The OPEN object in the Open message MUST contain the "Stateful PCE Capability" TLV defined in [RFC8231]. A new P (INTER-PCE-CAPABILITY) flag is introduced to indicate the support of state-sync.

This document adds a new bit in the Flags field with:

* P (INTER-PCE-CAPABILITY - 1 bit - TBD4): If set to 1 by a PCE, the PCE indicates that the session MUST follow the state-sync procedures as described in this document. If the P bit is set by both PCEP speakers on the session, the procedures MUST be used. If a PCEP speaker receives a STATEFUL-PCE-CAPABILITY TLV with P=0 while it advertised P=1 or if both PCEP speakers set the P flag to 0, the session SHOULD be set up but the state-sync procedures MUST NOT be applied on this session. A PCE MAY decide to close a session if the received setting of the P flag is not acceptable.

The U flag [RFC8231] MUST be set when sending the STATEFUL-PCE-CAPABILITY TLV with the P flag set. In case the U flag is not set along with the P flag, the state sync capability is not enabled and it is considered as if the P flag is not set. The S flag MAY be set if optimized synchronization is required as per [RFC8232].

3.2. State Synchronization

When the state sync capability has been negotiated between stateful PCEs, each PCEP speaker will behave as a PCE and as a PCC at the same time regarding the state synchronization as defined in [RFC8231]. This means that each PCEP Speaker:

* MUST send a PCRpt message towards its neighbour with the S flag set for each LSP in its LSP database learned from a PCC. (PCC role)

* MUST send the End Of Synchronization Marker towards its neighbour when all LSPs have been reported. (PCC role)

* MUST wait for the LSP synchronization from its neighbour to end (receiving an End Of Synchronization Marker). (PCE role)

The process of synchronization runs in parallel on each PCE (with no defined order).

The optimized state synchronization procedures MAY be used, as defined in [RFC8232].

When a PCEP Speaker sends a PCRpt on a state-sync session, it MUST add the SPEAKER-ENTITY-ID TLV (defined in [RFC8232]) in the LSP Object, the value used will refer to the 'owner' PCC of the LSP. If a PCEP Speaker receives a PCRpt on a state-sync session without this TLV, it MUST discard the PCRpt message and it MUST reply with a PCErr message using error-type=6 (Mandatory Object missing) and error-value=TBD1 (SPEAKER-ENTITY-ID TLV missing).

3.3. Incremental Updates and Report Forwarding Rules

During the life of an LSP, its state may change (path, constraints, operational state...) and a PCC will advertise a new PCRpt to the PCE for each such change.

When propagating LSP state changes from a PCE to other PCEs, PCE uses the freshest state coming from the PCC (based on the LSP-DB version).

When a PCE receives a new PCRpt from a PCC with the LSP-DB-VERSION (defined in [RFC8232]) , the PCE MUST forward the PCRpt to all its state-sync sessions and MUST add the appropriate SPEAKER-ENTITY-ID TLV in the PCRpt. In addition, it MUST add a new ORIGINAL-LSP-DB-VERSION TLV (described below). The ORIGINAL-LSP-DB-VERSION contains the LSP-DB-VERSION coming from the PCC.

When a PCE receives a new PCRpt from a PCC without the LSP-DB-VERSION, it SHOULD NOT forward the PCRpt on any state-sync sessions and SHOULD log such an event on the first occurrence.

When a PCE receives a new PCRpt from a PCC with the R flag (Remove) set and an LSP-DB-VERSION TLV, the PCE MUST forward the PCRpt to all its state-sync sessions keeping the R flag set (Remove) and MUST add the appropriate SPEAKER-ENTITY-ID TLV and ORIGINAL-LSP-DB-VERSION TLV in the PCRpt message.

When a PCE receives a PCRpt from a state-sync session, it MUST NOT forward the PCRpt to other state-sync sessions. This helps to prevent message loops between PCEs. As a consequence, a full mesh of PCEP sessions between PCEs is REQUIRED.

When a PCRpt is forwarded, all the original objects and values are kept. As an example, the PLSP-ID used in the forwarded PCRpt will be the same as the original one used by the PCC. Thus an implementation supporting this document MUST consider SPEAKER-ENTITY-ID TLV and PLSP-ID together to uniquely identify an LSP on the state-sync session.

The ORIGINAL-LSP-DB-VERSION TLV is encoded as shown in Figure 6 and MUST always contain the LSP-DB-VERSION received from the owner PCC of the LSP.

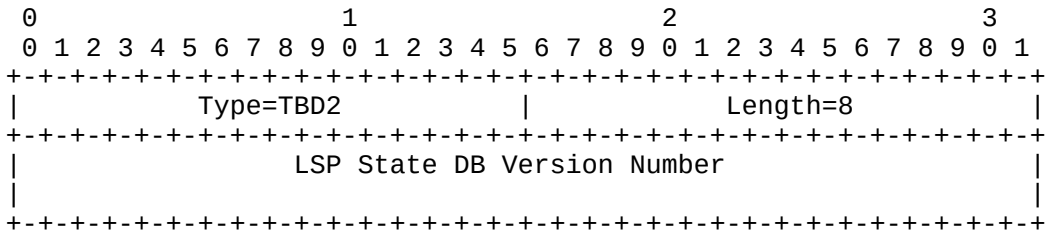


Figure 6: The ORIGINAL-LSP-DB-VERSION TLV

Using the ORIGINAL-LSP-DB-VERSION TLV allows a PCE to keep using optimized synchronization ([RFC8232]) with another PCE. In such a case, the PCE will send a PCRpt to another PCE with both ORIGINAL-LSP-DB-VERSION TLV and LSP-DB-VERSION TLV. The ORIGINAL-LSP-DB-VERSION TLV will contain the version number as allocated by the PCC while the LSP-DB-VERSION will contain the version number allocated by the local PCE.

3.4. Maintaining LSP States from Different Sources

When a PCE receives a PCRpt on a state-sync session, it stores the LSP information into the original PCC address context (as the LSP belongs to the PCC). A PCE SHOULD maintain a single state for a particular LSP and SHOULD maintain the list of sources it learned a particular state from.

A PCEP speaker may receive state information for a particular LSP from different sources: the PCC that owns the LSP (through a regular PCEP session) and some PCEs (through PCEP state-sync sessions). A PCEP speaker MUST always keep the freshest state in its LSP database, overriding the previously received information.

A PCE, receiving a PCRpt from a PCC, updates the state of the LSP in its LSP-DB with the newly received information. When receiving a PCRpt from another PCE, a PCE SHOULD update the LSP state only if the ORIGINAL-LSP-DB-VERSION present in the PCRpt indicates it is newer than the current ORIGINAL-LSP-DB-VERSION of the stored LSP state taking wrap-around into account. This ensures that a PCE never tries to update its stored LSP state with old information. Each time a PCE updates an LSP state in its LSP-DB, it SHOULD reset the source list associated with the LSP state and SHOULD add the source speaker

address in the source list. When a PCE receives a PCRpt that has an ORIGINAL-LSP-DB-VERSION (if coming from a PCE) or an LSP-DB-VERSION (if coming from the PCC) equal to the current ORIGINAL-LSP-DB-VERSION of the stored LSP state, it SHOULD add the source speaker address in the source list.

When a PCE receives a PCRpt requesting an LSP deletion from a particular source, it SHOULD remove this particular source from the list of sources associated with this LSP.

When the list of sources becomes empty for a particular LSP, the LSP state MUST be removed. This means that all the sources must send a PCRpt with R=1 for an LSP to make the PCE remove the LSP state.

3.5. Computation Priority between PCEs and Sub-delegation

A computation priority is necessary to ensure that a single PCE will perform the computation for all the LSPs in an association group: this will allow for a more optimized LSP placement and will prevent computation loops.

All PCEs in the network that are handling LSPs in a common LSP association group SHOULD be aware of each other including the computation priority of each PCE. Note that there is no need for PCC to be aware of this. The computation priority is a number and the PCE having the highest priority MUST be responsible for the computation. If several PCEs have the same priority value, their IP address MUST be used as a tie-breaker to provide a rank: the highest IP address has more priority.

The computation priorities could be set through local configurations. The priority for local and remote PCEs could be set at the global level so the highest priority PCE will handle all path computations or more granular, so a PCE may have the highest priority for only a subset of LSPs or association groups. See Section 8.1 for more details. In future, PCEs could also advertise and discover these parameters via PCEP, those details are out of the scope of this document and left for future specification.

A PCEP Speaker receiving a PCRpt from a PCC with the D flag set that does not have the highest computation priority, SHOULD forward the PCRpt on all state-sync sessions (as per Section 3.3) and SHOULD set the D flag on the state-sync session towards the highest priority PCE, the D flag will be unset to all other state-sync sessions. This behavior is similar to the delegation behaviour handled at the PCC side and is called a sub-delegation (the PCE sub-delegates the control of the LSP to another PCE). When a PCEP Speaker sub-delegates an LSP to another PCE, it loses control of the LSP and

cannot update it anymore by its own decision. When a PCE receives a PCRpt with the D flag set on a state-sync session, as a regular PCE, it is granted control over the LSP.

If the highest priority PCE is failing or if the state-sync session between the local PCE and the highest priority PCE failed, the operator MAY decide to instruct a switch-over to delegate the LSP to the next highest priority PCE or to take back control of the LSP. It is a local policy decision.

When a PCE has the delegation for an LSP and needs to update this LSP, it MUST send a Path Compute Update (PCUpd) message to all state-sync sessions and to the PCC session on which it received the delegation. The D-Flag would be unset in the PCUpd for state-sync sessions whereas the D-Flag would be set for the PCC. In the case of sub-delegation, the computing PCE will send the PCUpd only to all state-sync sessions (as it has no direct delegation from a PCC). The D-Flag would be set for the state-sync session to the PCE that sub-delegated this LSP and the D-Flag would be unset for other state-sync sessions.

The PCUpd sent over a state-sync session MUST contain the SPEAKER-ENTITY-ID TLV in the LSP Object (the value used must identify the target PCC). The PLSP-ID used is the original PLSP-ID generated by the PCC and learned from the forwarded PCRpt. If a PCE receives a PCUpd on a state-sync session without the SPEAKER-ENTITY-ID TLV, it MUST discard the PCUpd and MUST reply with a PCErr message using error-type=6 (Mandatory Object missing) and error-value=TBD1 (SPEAKER-ENTITY-ID TLV missing).

When a PCE receives a valid PCUpd on a state-sync session, it SHOULD forward the PCUpd to the appropriate PCC (identified based on the SPEAKER-ENTITY-ID TLV value) that delegated the LSP originally and SHOULD remove the SPEAKER-ENTITY-ID TLV from the LSP Object. The acknowledgement of the PCUpd is done through a cascaded mechanism, and the PCC is only responsible for triggering the acknowledgement: when the PCC receives the PCUpd from the local PCE, it acknowledges it with a PCRpt as per [RFC8231]. When receiving the new PCRpt from the PCC, the local PCE uses the defined forwarding rules on the state-sync session so the acknowledgement is relayed to the computing PCE.

3.5.1. Information Received via Open Message from PCC

To ensure uniform information across all PCEs, each PCE needs to relay the information it receives from the PCCs in the Open message to other PCEs via the state-sync session. This includes various PCC capabilities and parameters such as Maximum Segment Identifier (SID) Depth (MSD).

As per [RFC5440], the PCEP Notification message (PCNtf) can be sent by a PCEP speaker to notify its peer of a specific event. A PCE should notify the other state-sync PCEs of the information it receives from the PCC's open message. Section 7.14 of [RFC5440] specifies the NOTIFICATION object. This document adds a new Notification-type=TBD6 (Inter-PCE State-sync) and two Notification-values (Notification-value=1 (Add PCC's Open Information) and Notification-value=2 (Remove PCC's Open Information)).

For Notification-type=TBD6, the NOTIFICATION object encodes the SPEAKER-ENTITY-ID TLV (with values that identify the PCC) and any other TLV that can be carried inside the OPEN object as a way to signal the PCC's information it received via the open message to other state-sync PCEs.

* Notification-value=1: Add PCC's Open Information. On session establishment with a PCC, a PCE with state-sync capability MUST send this notification to other state-sync PCEs with the SPEAKER-ENTITY-ID TLV with values that identify the PCC and any other TLVs encoded in the OPEN object received from the PCC. On session establishment with a state-sync PCE, the PCE MUST also exchange notifications for each of the PCCs it already has a session established. The PCE MUST exchange this notification prior to the State Synchronization (described in Section 3.2). Note that the PCNtf can be used to carry multiple NOTIFICATION objects, one for each PCC. On receiving this notification, PCE adds the information to its database.

* Notification-value=2: Remove PCC's Open Information. On session down with a PCC, a PCE with state-sync capability MUST send this notification to other state-sync PCEs with the SPEAKER-ENTITY-ID TLV with values that identify the PCC to remove the information from the database.

A PCE may receive this Notification from multiple PCEs that a given PCC has a session and can use a similar mechanism as described in Section 3.4 to keep the freshest state. In case of the termination of a state-sync session, this information is also cleaned up alongside LSP-DB.

3.5.2. Association Group

All LSPs belonging to the same association group SHOULD have the same computation priorities for the PCEs. A PCE SHOULD only compute a path using an association-group constraint if it has delegation for all of the LSPs in the association group. In this case, an implementation MAY use a local policy on PCE to decide if PCE does not compute a path at all for this set of LSP or if it can compute a path by relaxing the association-group constraint.

3.6. Passive Stateful Procedures

In the passive stateful PCE architecture, the PCC is responsible for triggering a path computation request using a PCReq message to its PCE. Similarly to PCRpt Message, which remains unchanged for passive mode, if a PCE receives a PCReq for an LSP and if this PCE finds that it does not have the highest computation priority of this LSP, or groups, it MUST forward the PCReq message to the highest priority PCE over the state-sync session. When the highest priority PCE receives the PCReq, it computes the path and generates a PCRep message towards the PCE that made the request. This PCE will then forward the PCRep to the requesting PCC. The handling of the LSP object and the SPEAKER-ENTITY-ID TLV in PCReq and PCRep is similar to PCRpt/PCUpd messages.

3.7. PCE Initiation Procedures

It is possible that a PCE does not have a PCEP session with the headend to initiate an LSP as per [RFC8281]. A PCE could send the Path Compute Initiate (PCInitiate) message on the state-sync sessions to another PCE to request it to create a PCE-Initiated LSP on its behalf. If the PCE is able to initiate the LSP it would report it on the state-sync session via PCRpt message. If the PCE does not have a session to the headend, it MUST send a PCErr message with Error-type=24 (PCE instantiation error) and Error-value=TBD5 (No PCEP session with the headend). PCE could try to initiate via another state-sync PCE if available.

3.8. Loop Prevention

This specification allows PCEP messages to be propagated among PCEP speakers. Thus, it is useful to track information about the propagation of PCEP messages. One of the use cases is a PCEP message loop detection mechanism, but other use cases like hop-by-hop information recording may also be possible, but are out of scope of this document.

3.8.1. PCEP-PATH-VECTOR TLV

This document introduces the PCEP-PATH-VECTOR TLV (type TBD3) to be encoded in the LSP Object with the format shown in Figure 7.

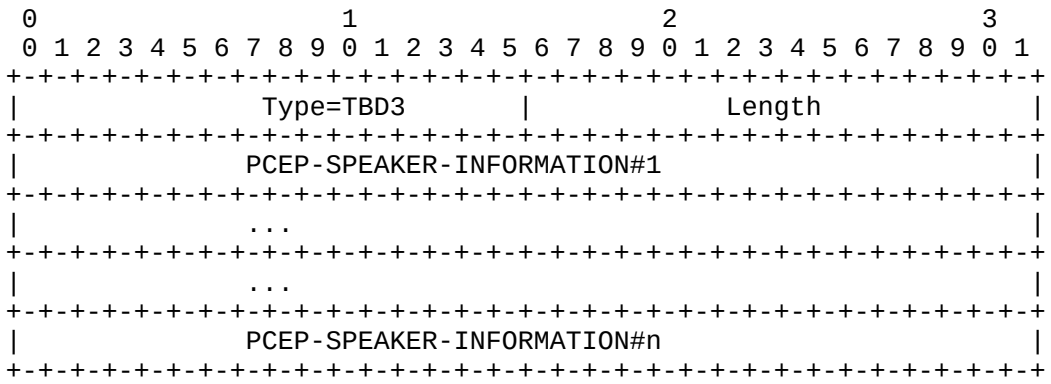


Figure 7: The PCEP-PATH-VECTOR TLV

The TLV format and padding rules are as per [RFC5440].

The PCEP-SPEAKER-INFORMATION field has the format shown in Figure 8.

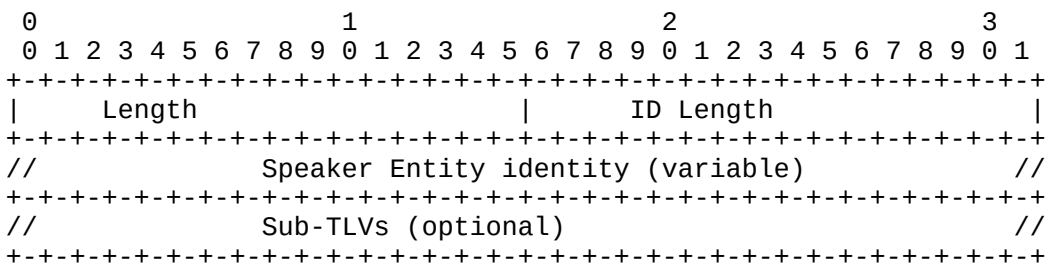


Figure 8: The PCEP-SPEAKER-INFORMATION

* Length: defines the total length of the PCEP-SPEAKER-INFORMATION field.

* ID Length: defines the length of the Speaker Entity identity field, not counting any padding.

* Speaker Entity identity: same as the value portion of the SPEAKER-ENTITY-ID TLV. Padded with trailing zeros to a 4-byte boundary.

* The PCEP-SPEAKER-INFORMATION may also carry some optional sub-TLVs, so each PCEP speaker can add local information that could be recorded. This document does not define any sub-TLVS.

The PCEP-PATH-VECTOR TLV MAY be carried in the LSP Object. Its usage is purely optional.

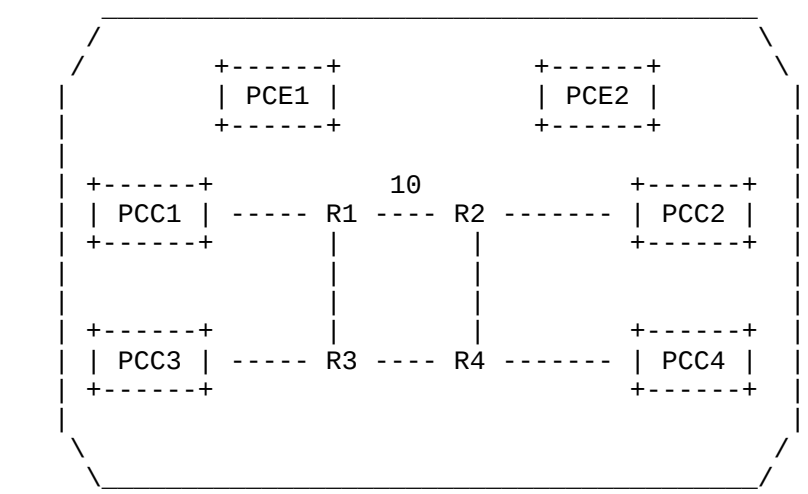
If a PCEP speaker receives a message with PCEP-PATH-VECTOR TLV and finds its speaker information already present in the PCEP-PATH-VECTOR TLV, it MUST ignore the PCEP message and SHOULD log it as an error because this represents a message loop.

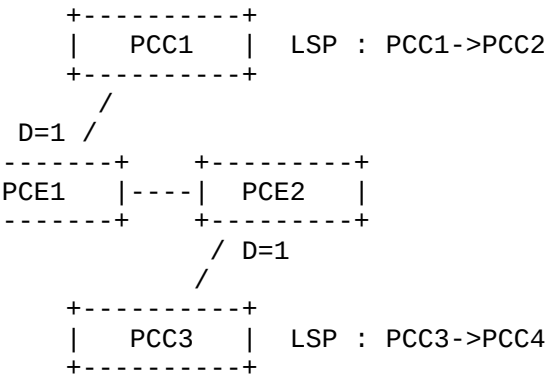
The list of speakers within the PCEP-PATH-VECTOR TLV MUST be ordered. When sending a PCEP message (PCRpt, PCUpd, or PCInitiate), a PCEP Speaker MAY add the PCEP-PATH-VECTOR TLV with a PCEP-SPEAKER-INFORMATION containing its own information. If the PCEP message sent is the result of a previously received PCEP message, and if the PCEP-PATH-VECTOR TLV was already present in the initial message, the PCEP speaker MAY append a new PCEP-SPEAKER-INFORMATION containing its own information at the end of the TLV.

4. Examples

The examples in this section are for illustrative purpose only, to show how the behavior of the state sync inter-PCE session works.

4.1. Example 1 - Successful disjoint paths (requiring reroute)





PCE1 computation priority 100
PCE2 computation priority 200

Figure 9: Disjoint Paths Requiring Reroute

Consider the PCEP sessions in Figure 9, where computation priority is global for all the LSPs and a link disjoint path between LSPs PCC1->PCC2 and PCC3->PCC4 is required.

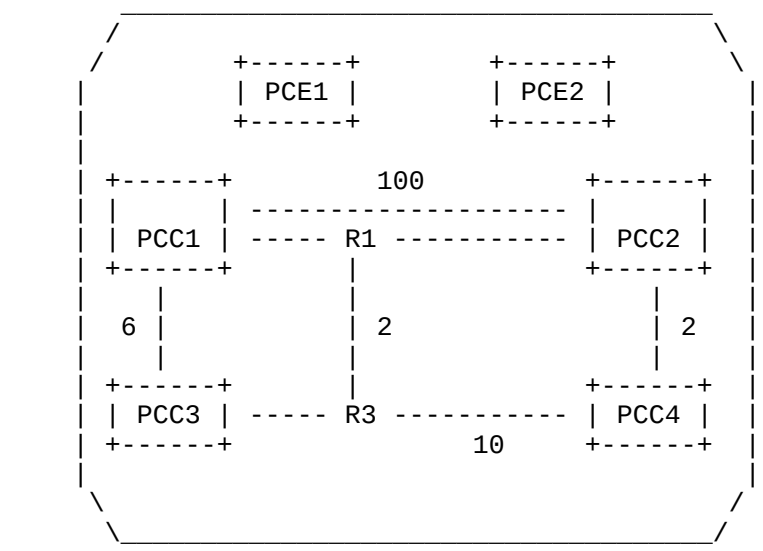
Consider the PCC1->PCC2 is configured first and PCC1 delegates the LSP to PCE1, but as PCE1 does not have the highest computation priority, it sub-delegates the LSP to PCE2 by sending a PCRpt with D=1 and including the SPEAKER-ENTITY-ID TLV over the state-sync session. PCE2 receives the PCRpt and as it has delegation for this

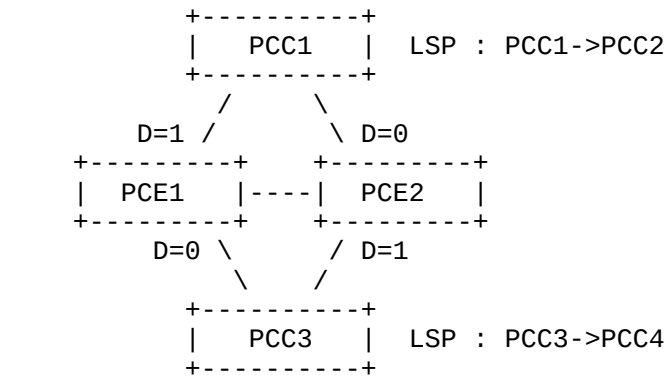
LSP, it computes the shortest path: R1->R3->R4->R2->PCC2. It then sends a PCUpd to PCE1 (including the SPEAKER-ENTITY-ID TLV) with the computed ERO. PCE1 forwards the PCUpd to PCC1 (removing the SPEAKER-ENTITY-ID TLV). PCC1 acknowledges the PCUpd by a PCRpt to PCE1. PCE1 forwards the PCRpt to PCE2.

When PCC3->PCC4 is configured, PCC3 delegates the LSP to PCE2, PCE2 can compute a disjoint path as it has knowledge of both LSPs and has delegation also for both. The only solution found is to move PCC1->PCC2 LSP on another path, PCE2 can move PCC1->PCC2 as it has sub-delegation for it. It creates a new PCUpd with a new ERO: R1->R2-PCC2 towards PCE1 which forwards to PCC1. PCE2 sends a PCUpd to PCC3 with the path: R3->R4->PCC4.

In this set-up, PCEs are able to find a disjoint path while without state-sync and computation priority, they could not.

4.2. Example 2 - Successful disjoint paths (simultaneous turnup)



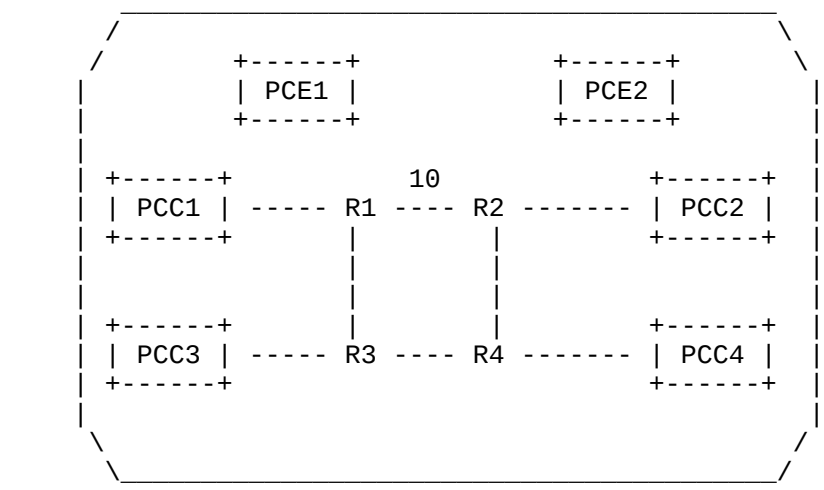


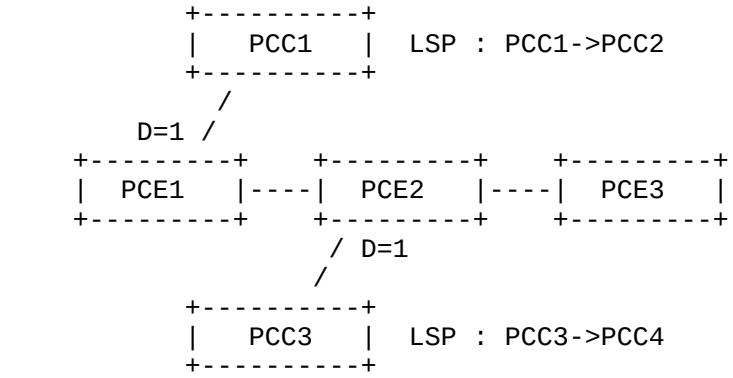
PCE1 computation priority 200
PCE2 computation priority 100

Figure 10: Disjoint Paths with Simultaneous Turnup

In this example (see Figure 10), suppose both LSPs are configured almost at the same time. PCE1 sub-delegates PCC1->PCC2 to PCE2 while PCE2 keeps delegation for PCC3->PCC4, PCE2 computes a path for PCC1->PCC2 and PCC3->PCC4 and can achieve disjointness computation easily. No computation loop happens in this case.

4.3. Example 3 - Unfeasible disjoint paths (insufficient state-sync sessions)





PCE1 computation priority 100
PCE2 computation priority 200
PCE3 computation priority 300

Figure 11: Unfeasible Disjoint Paths

With the PCEP sessions as in Figure 11, consider the need to have link disjoint LSPs PCC1->PCC2 and PCC3->PCC4.

Suppose PCC1->PCC2 is configured first, PCC1 delegates the LSP to PCE1, but as PCE1 does not have the highest computation priority, it will sub-delegate the LSP to PCE2 (as it is not aware of PCE3 and has no way to reach it). PCE2 cannot compute a path for PCC1->PCC2 as it does not have the highest priority and is not allowed to sub-delegate the LSP again towards PCE3 as per Section 3.

When PCC3->PCC4 is configured, PCC3 delegates the LSP to PCE2 that performs sub-delegation to PCE3. As PCE3 will have knowledge of only one LSP in the group, it cannot compute disjointness and can decide to fall back to a less constrained computation to provide a path for PCC3->PCC4. In this case, it will send a PCUpd to PCE2 that will be forwarded to PCC3.

Disjointness cannot be achieved in this scenario because of lack of state-sync session between PCE1 and PCE3, but no computation loop happens. Thus it is required for all PCEs that support state-sync to have full mesh sessions between each other.

5. Using Primary/Secondary Computation and State-sync Sessions to Increase Scaling

The Primary/Secondary computation and state-sync sessions architecture can be used to increase the scaling of the PCE architecture. If the number of PCCs is really high, it may be too resource-consuming for a single PCE instance to maintain all the PCEP sessions while at the same time performing all path computations. Using primary/secondary computation and state-sync sessions may allow to create groups of PCEs that manage a subset of the PCCs and perform some or no path computations. Decoupling PCEP session maintenance and computation will allow increasing scaling of the PCE architecture.

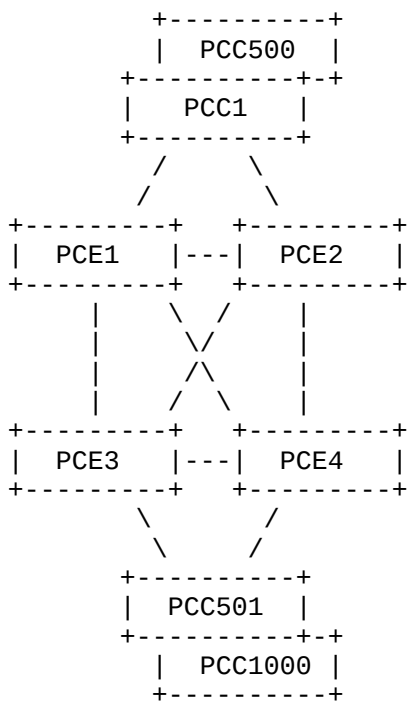


Figure 12: Improved Scalability

In Figure 12, two groups of PCEs are created: PCE1/2 maintain PCEP sessions with PCC1 up to PCC500, while PCE3/4 maintain PCEP sessions with PCC501 up to PCC1000. A granular primary/secondary policy is set-up as follows to load-share computation between PCEs:

* PCE1 has priority 200 for association ID 1 up to 300, association source 0.0.0.0. All other PCEs have a decreasing priority for those associations.

* PCE3 has priority 200 for association ID 301 up to 500, association source 0.0.0.0. All other PCEs have a decreasing priority for those associations.

If some PCCs delegate LSPs with association ID 1 up to 300 and association source 0.0.0.0, the receiving PCE (if not PCE1) will sub-delegate the LSPs to PCE1. PCE1 becomes responsible for the computation of these LSP associations while PCE3 is responsible for the computation of another set of associations.

The procedures described in this document could help greatly in load-sharing between a group of stateful PCEs.

6. Security Considerations

The security considerations described in [RFC8231] and [RFC5440] apply to the extensions described in this document as well. Additional considerations related to state synchronization and sub-delegation between stateful PCEs are introduced, as they could be spoofed and could be used as an attack vector. An attacker could attempt to create too much state in an attempt to load the PCEP peer. The PCEP peer could respond with a PCErr message as described in [RFC8231]. An attacker could impact LSP operations by creating a bogus state. Further, state synchronization between stateful PCEs could provide an adversary with the opportunity to eavesdrop on the network. Thus, securing the PCEP session using Transport Layer Security (TLS) [RFC8253], as per the recommendations and best current practices in [RFC9325], is RECOMMENDED.

7. 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".

At the time of posting this document, there are no known implementations of this mechanism. It is believed that some vendors are considering implementations, but these plans are too vague to make any further assertions.

8. Manageability Considerations

8.1. Control of Function and Policy

An operator MUST be allowed to configure the capability to support state-sync procedures for an inter-PCE session. They MUST be allowed to configure a computation priority of the local and remote PCEs at the global level. They MAY also be allowed to configure the computation priority of the local and remote PCEs per association (or a range of them). Further, they MAY also be allowed to configure computation priority per PCC (or range of them). An implementation MAY support other such configuration levels for computation priority of the local and remote PCEs.

8.2. Information and Data Models

An implementation SHOULD allow the operator to view the capability defined in this document. To serve this purpose, the PCEP YANG module [I-D.ietf-pce-pcep-yang] could be extended in the future.

8.3. 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].

8.4. Verify Correct Operations

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

8.5. Requirements On Other Protocols

Mechanisms defined in this document do not imply any new requirements on other protocols.

8.6. Impact On Network Operations

Mechanisms defined in this document improve the network operations by alleviating the problems described in Section 1.

9. Acknowledgements

Thanks to [I-D.knodel-terminology] urging for better use of terms.

10. IANA Considerations

This document requests IANA actions to allocate code points for the protocol elements defined in this document.

10.1. PCEP-Error Object

This document defines one new Error-Value within the "Mandatory Object Missing" Error-Type and "LSP instantiation error" Error-Type. IANA is requested to allocate new error values within the "PCEP-ERROR Object Error Types and Values" registry of the "Path Computation Element Protocol (PCEP) Numbers" registry group, as follows:

Error-Type	Meaning	Reference
6	Mandatory Object Missing	[RFC5440]
	Error-value=TBD1: SPEAKER-ENTITY-ID TLV missing	This document
24	LSP instantiation error	[RFC8281]
	Error-value=TBD5: No PCEP session with the headend	This document

Table 1

10.2. PCEP TLV Type Indicators

IANA is requested to allocate new TLV Type Indicator values within the "PCEP TLV Type Indicators" registry of the "Path Computation Element Protocol (PCEP) Numbers" registry group, as follows:

Value	Meaning	Reference
TBD2	ORIGINAL-LSP-DB-VERSION TLV	This document
TBD3	PCEP-PATH-VECTOR TLV	This document

Table 2

10.3. STATEFUL-PCE-CAPABILITY TLV

IANA is requested to allocate a new bit value in the "STATEFUL-PCE-CAPABILITY TLV Flag Field" registry of the "Path Computation Element Protocol (PCEP) Numbers" registry group, as follows:

Bit	Description	Reference
TBD4	INTER-PCE-CAPABILITY	This document

Table 3

10.4. Notification Object

IANA is requested to allocate a new Notification Type and Notification Values within the "Notification Object" registry of the "Path Computation Element Protocol (PCEP) Numbers" registry group, as follows:

Notification-type	Meaning	Reference
TBD6	Inter-PCE State-sync	This document
	Notification-value=1: Add PCC's Open Information	This document
	Notification-value=2: Remove PCC's Open Information	This document

Table 4

11. References

11.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC5440] Vasseur, JP., Ed. and JL. Le Roux, Ed., "Path Computation Element (PCE) Communication Protocol (PCEP)", RFC 5440, DOI 10.17487/RFC5440, March 2009, <https://www.rfc-editor.org/info/rfc5440>.

[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>.

[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>.

[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>.

11.2. Informative References

[I-D.ietf-pce-pcep-yang] Dhody, D., Beeram, V. P., Hardwick, J., and J. Tantsura, "A YANG Data Model for Path Computation Element Communications Protocol (PCEP)", Work in Progress, Internet-Draft, draft-ietf-pce-pcep-yang-30, 26 January 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-pce-pcep-yang-30>.

[I-D.knodel-terminology] Knodel, M. and N. ten Oever, "Terminology, Power, and Inclusive Language in Internet-Drafts and RFCs", Work in Progress, Internet-Draft, draft-knodel-terminology-14, 24 August 2023, <https://datatracker.ietf.org/doc/html/draft-knodel-terminology-14>.

[RFC4655] Farrel, A., Vasseur, J.-P., 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>.

[RFC6805] King, D., Ed. and A. Farrel, Ed., "The Application of the Path Computation Element Architecture to the Determination of a Sequence of Domains in MPLS and GMPLS", RFC 6805, DOI 10.17487/RFC6805, November 2012, <https://www.rfc-editor.org/info/rfc6805>.

[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>.

[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>.

[RFC8051] Zhang, X., Ed. and I. Minei, Ed., "Applicability of a Stateful Path Computation Element (PCE)", RFC 8051, DOI 10.17487/RFC8051, January 2017, <https://www.rfc-editor.org/info/rfc8051>.

[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>.

[RFC8751] Dhody, D., Lee, Y., Ceccarelli, D., Shin, J., and D. King, "Hierarchical Stateful Path Computation Element (PCE)", RFC 8751, DOI 10.17487/RFC8751, March 2020, <https://www.rfc-editor.org/info/rfc8751>.

[RFC8800] Litkowski, S., Sivabalan, S., Barth, C., and M. Negi, "Path Computation Element Communication Protocol (PCEP) Extension for Label Switched Path (LSP) Diversity Constraint Signaling", RFC 8800, DOI 10.17487/RFC8800, July 2020, <https://www.rfc-editor.org/info/rfc8800>.

[RFC9059] Gandhi, R., Ed., Barth, C., and B. Wen, "Path Computation Element Communication Protocol (PCEP) Extensions for Associated Bidirectional Label Switched Paths (LSPs)", RFC 9059, DOI 10.17487/RFC9059, June 2021, <https://www.rfc-editor.org/info/rfc9059>.

[RFC9325] Sheffer, Y., Saint-Andre, P., and T. Fossati, "Recommendations for Secure Use of Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS)", BCP 195, RFC 9325, DOI 10.17487/RFC9325, November 2022, <https://www.rfc-editor.org/info/rfc9325>.

[RFC9552] Talaulikar, K., Ed., "Distribution of Link-State and Traffic Engineering Information Using BGP", RFC 9552, DOI 10.17487/RFC9552, December 2023, <https://www.rfc-editor.org/info/rfc9552>.

Appendix A. Contributors

Dhruv Dhody
Huawei
India

Email: dhruv.ietf@gmail.com

Appendix B. Scenarios

This appendix provides several scenarios for illustrative purposes. There are many other cases where the solution defined in this document are also applicable.

B.1. Scenario 1

In the normal case (PCE1 as active primary PCE), consider that PCC1->PCC2 LSP is configured first with the link disjointness constraint, PCE1 sends a PCUpd message to PCC1 with the Explicit Routing Object (ERO): R1->R3->R4->R2->PCC2 (shortest path). PCC1 signals and installs the path. When PCC3->PCC4 is configured, the PCEs already knows the path of PCC1->PCC2 and can compute a link-disjoint path: the solution requires to move PCC1->PCC2 onto a new path to let room for the new LSP. PCE1 sends a PCUpd message to PCC1 with the new ERO: R1->R2->PCC2 and a PCUpd to PCC3 with the following ERO: R3->R4->PCC4. In the normal case, there is no issue for PCE1 to compute a link-disjoint path.

B.2. Scenario 2

Consider that PCC1 lost its PCEP session with PCE1 (all other PCEP sessions are UP) as shown in Figure 13. PCC1 delegates its LSP to PCE2.

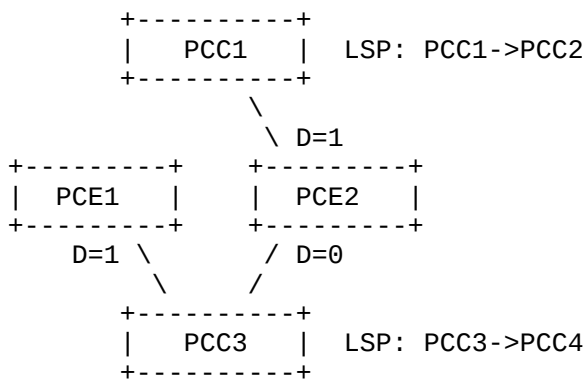


Figure 13: Scenario 2

Consider that the PCC1->PCC2 LSP is configured first with the link disjointness constraint, PCE2 (which is the new active primary PCE for PCC1) sends a PCUpd message to PCC1 with the ERO: R1->R3->R4->R2->PCC2 (shortest path). When PCC3->PCC4 is configured, PCE1 is not aware of LSPs from PCC1 any more, so it cannot compute a disjoint path for PCC3->PCC4 and will send a PCUpd message to PCC3 with the shortest path ERO: R3->R4->PCC4. When PCC3->PCC4 LSP will be reported to PCE2 by PCC3, PCE2 will ensure disjointness computation and will correctly move PCC1->PCC2 (as it owns delegation for this LSP) on the following path: R1->R2->PCC2. With this sequence of events and these PCEP sessions, disjointness is ensured.

B.3. Scenario 3

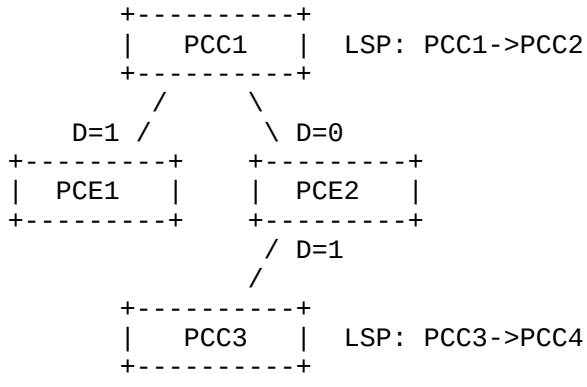


Figure 14: Scenario 3

Consider the PCEP sessions in Figure 14, and the PCC1->PCC2 LSP is configured first with the link disjointness constraint, PCE1 computes the shortest path as it is the only LSP in the disjoint association group that it is aware of: R1->R3->R4->R2->PCC2 (shortest path). When PCC3->PCC4 is configured, PCE2 must compute a disjoint path for this LSP. The only solution found is to move PCC1->PCC2 LSP on another path, but PCE2 cannot do it as it does not have delegation for this LSP. In this set-up, PCEs are not able to find a disjoint path.

B.4. Scenario 4

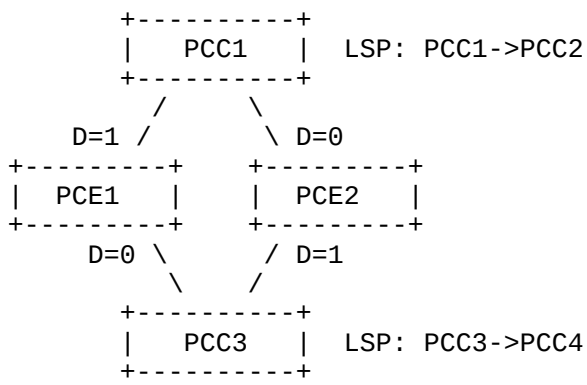


Figure 15: Scenario 4

Consider the PCEP sessions in Figure 15. and that PCEs are configured to fall-back to the shortest path if disjointness cannot be found as described in [RFC8800]. The PCC1->PCC2 LSP is configured first, PCE1 computes the shortest path as it is the only LSP in the disjoint association group that it is aware of: R1->R3->R4->R2->PCC2 (shortest path). When PCC3->PCC4 is configured, PCE2 must compute a disjoint path for this LSP. The only solution found is to move PCC1->PCC2 LSP on another path, but PCE2 cannot do it as it does not have delegation for this LSP. PCE2 then provides the shortest path for PCC3->PCC4: R3->R4->PCC4. When PCC3 receives the ERO, it reports it back to both PCEs. When PCE1 becomes aware of the PCC3->PCC4 path, it recomputes the constrained shortest path first (CSPF) algorithm and provides a new path for PCC1->PCC2: R1->R2->PCC2. The new path is reported back to all PCEs by PCC1. PCE2 recomputes also CSPF to take into account the new reported path. The new computation does not lead to any path update.

B.5. Scenario 5

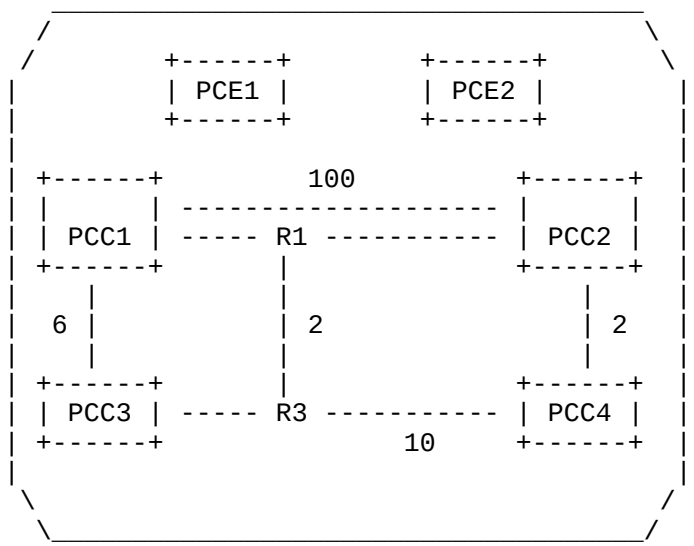


Figure 16: Scenario 5

Now, consider a new network topology in Figure 16 with the same PCEP sessions as the previous example. Suppose that both LSPs are configured almost at the same time. PCE1 will compute a path for PCC1->PCC2 while PCE2 will compute a path for PCC3->PCC4. As each PCE is not aware of the path of the second LSP in the association group (not reported yet), each PCE is computing the shortest path for the LSP. PCE1 computes ERO: R1->PCC2 for PCC1->PCC2 and PCE2 computes ERO: R3->R1->PCC2->PCC4 for PCC3->PCC4. When these shortest paths will be reported to each PCE. Each PCE will recompute disjointness. PCE1 will provide a new path for PCC1->PCC2 with ERO: PCC1->PCC2. PCE2 will provide also a new path for PCC3->PCC4 with ERO: R3->PCC4. When those new paths will be reported to both PCEs, this will trigger CSPF again. PCE1 will provide a new more optimal path for PCC1->PCC2 with ERO: R1->PCC2 and PCE2 will also provide a more optimal path for PCC3->PCC4 with ERO: R3->R1->PCC2->PCC4. So we come back to the initial state. When those paths will be reported to both PCEs, this will trigger CSPF again. An infinite loop of CSPF computation is then happening with a permanent flap of paths because of the split-brain situation.

Another common example to note would be two LSPs with link-diverse paths that share a common node in its path but delegated to different PCEs. In case of the common node failure, both PCEs would detect the same and each could independently compute a new path that might both choose the same new link.

This permanent computation loop comes from the inconsistency between the state of the LSPs as seen by each PCE due to the split-brain: each PCE is trying to modify at the same time its delegated path based on the last received path information which de facto invalidates this received path information.

B.6. Scenario 6

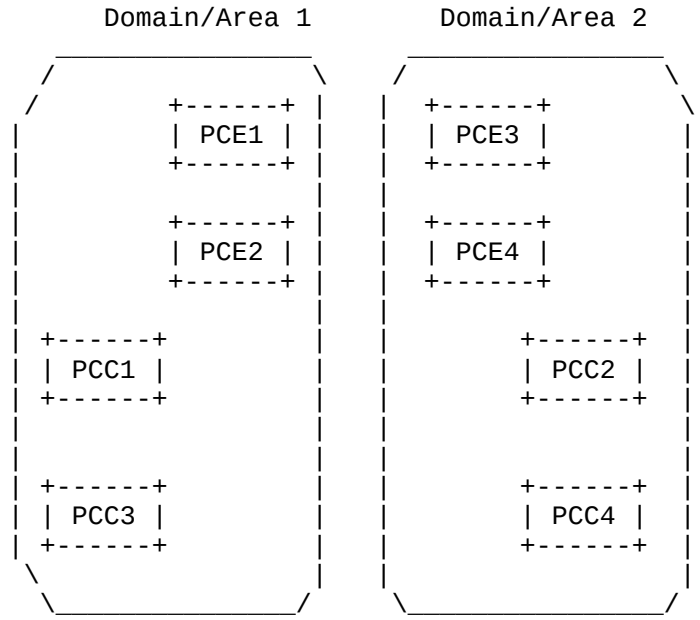


Figure 17: Scenario 6

In the example in Figure 17, suppose that the disjoint LSPs from PCC1 to PCC2 and from PCC4 to PCC3 are created. All the PCEs have the knowledge of both domain topologies (e.g. using BGP-LS [RFC9552]). For operation/management reasons, each domain uses its own group of redundant PCEs. PCE1/PCE2 in domain 1 have PCEP sessions with PCC1 and PCC3 while PCE3/PCE4 in domain 2 have PCEP sessions with PCC2 and PCC4. As PCE1/2 does not know about LSPs from PCC2/4 and PCE3/4 do not know about LSPs from PCC1/3, there is no possibility to compute

the disjointness constraint. This scenario can also be seen as a split-brain scenario. This multi-domain architecture (with multiple groups of PCEs) can also be used in a single domain, where an operator wants to limit the failure domain by creating multiple groups of PCEs maintaining a subset of PCCs. As for the multi-domain example, there will be no possibility to compute the disjoint path starting from head-ends managed by different PCE groups.

In this document, we specify a solution that addresses the possibility to compute LSP association based constraints (like disjointness) in split-brain scenarios while preventing computation loops.

Authors' Addresses

Haomian Zheng (editor)
Huawei Technologies
H1, Huawei Xiliu Beipo Village, Songshan Lake
Dongguan
Guangdong, 523808
China
Email: zhenghaomian@huawei.com

Stephane Litkowski
Cisco
Email: slitkows.ietf@gmail.com

Siva Sivabalan
Ciena Corporation
Email: msiva282@gmail.com

Cheng Li
Huawei Technologies
Huawei Campus, No. 156 Beiqing Rd.
Beijing
100095
China
Email: c.l@huawei.com