Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: January 1, 2014

E. Crabbe
Google, Inc.
J. Medved
Cisco Systems, Inc.
I. Minei
Juniper Networks, Inc.
R. Varga
Pantheon Technologies SRO

June 30, 2013

# PCEP Extensions for Stateful PCE
# draft-ietf-pce-stateful-pce-05

Abstract

The Path Computation Element Communication Protocol (PCEP) provides mechanisms for Path Computation Elements (PCEs) to perform path computations in response to Path Computation Clients (PCCs) requests.

Although PCEP explicitly makes no assumptions regarding the information available to the PCE, it also makes no provisions for synchronization or PCE control of timing and sequence of path computations within and across PCEP sessions. This document describes a set of extensions to PCEP to enable stateful control of MPLS-TE and GMPLS LSPs via PCEP.

## Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

This Internet-Draft will expire on January 1, 2014.

Table of Contents

## 1. Introduction

[RFC5440] describes the Path Computation Element Protocol (PCEP. PCEP defines the communication between a Path Computation Client (PCC) and a Path Control Element (PCE), or between PCE and PCE, enabling computation of Multiprotocol Label Switching (MPLS) for Traffic Engineering Label Switched Path (TE LSP) characteristics. Extensions for support of GMPLS in PCEP are defined in [I-D.ietf-pce-gmpls-pcep-extensions]

This document specifies a set of extensions to PCEP to enable stateful control of LSPs between and across PCEP sessions in compliance with [RFC4657]. It includes mechanisms to effect LSP state synchronization between PCCs and PCEs, delegation of control over LSPs to PCEs, and PCE control of timing and sequence of path computations within and across PCEP sessions.

## 2. Terminology

This document uses the following terms defined in [RFC5440]: PCC, PCE, PCEP Peer.

This document uses the following terms defined in [RFC4655]: TED.

This document uses the following terms defined in [RFC4090]: MPLS TE Fast Reroute (FRR), FRR One-to-One Backup, FRR Facility Backup.

The following terms are defined in this document:

Stateful PCE: has access to not only the network state, but also to the set of active paths and their reserved resources for its computations. A stateful PCE might also retain information regarding LSPs under construction in order to reduce churn and resource contention. The additional state allows the PCE to compute constrained paths while considering individual LSPs and their interactions. Note that this requires reliable state synchronization mechanisms between the PCE and the network, PCE and PCC, and between cooperating PCEs.

Passive Stateful PCE: uses LSP state information learned from PCCs to optimize path computations. It does not actively update LSP state. A PCC maintains synchronization with the PCE.

Active Stateful PCE: is an extension of Passive Stateful PCE, in which the PCE may issue recommendations to the network. For example, an active stateful PCE may utilize the Delegation mechanism to update LSP parameters in those PCCs that delegated

control over their LSPs to the PCE.

Delegation: An operation to grant a PCE temporary rights to modify a subset of LSP parameters on one or more PCC's LSPs. LSPs are delegated from a PCC to a PCE, and are referred to as delegated LSPs. The PCC who owns the PCE state for the LSP has the right to delegate it. An LSP is owned by a single PCC at any given point in time.

Revocation An operation performed by a PCC on a previously delegated LSP. Revocation revokes the rights granted to the PCE in the delegation operation.

Redelegation Timeout Interval: when a PCEP session is terminated, a PCC waits for this time period before revoking LSP delegation to a PCE and attempting to redelegate LSPs associated with the terminated PCEP session to an alternate PCE. The Redelegation Timeout Interval is a PCC-local value that can be either operator-configured or dynamically computed by the PCC based on local policy.

State Timeout Interval: when a PCEP session is terminated, a PCC waits for this time period before flushing LSP state associated with that PCEP session and reverting to operator-defined default parameters. The state will not be flushed in two cases: a) the LSP is redelegated to another PCE before the expiration of the State Timeout Interval or b)the PCC makes changes to the LSP state. The State Timeout Interval is a PCC-local value that can be either operator-configured or dynamically computed by the PCC based on local policy. The State Timeout Interval value SHOULD be greater than or equal to the Redelegation Timeout Interval value in order to allow for redelegation of LSPs without unnecessary churn in or loss of state.

LSP State Report: an operation to send LSP state (Operational / Admin Status, LSP attributes configured and set by a PCE, etc.) from a PCC to a PCE.

LSP Update Request: an operation where an Active Stateful PCE requests a PCC to update one or more attributes of an LSP and to re-signal the LSP with updated attributes.

LSP Priority: a specific pair of MPLS setup and hold priority values as defined in [RFC3209].

LSP State Database: information about and attributes of all LSPs that are being reported to one or more PCEs via LSP State Reports.

Minimum Cut Set: the minimum set of links for a specific source destination pair which, when removed from the network, result in a specific source being completely isolated from specific destination. The summed capacity of these links is equivalent to the maximum capacity from the source to the destination by the max-flow min-cut theorem.

Within this document, PCE-PCE communications are described by having the requesting PCE fill the role of a PCC. This provides a saving in documentation without loss of function.

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

## 3. Motivation and Objectives for Stateful PCE

Editor's note: the content in this section is duplicated in [I-D.zhang-pce-stateful-pce-app]. To avoid loss of information, the use cases will be removed from this document only after [I-D.zhang-pce-stateful-pce-app] becomes a working group document. In the meantime, changes and updates to the use cases are reflected in [I-D.zhang-pce-stateful-pce-app].

### 3.1. Motivation

In the following sections, several use cases are described, showcasing scenarios that benefit from the deployment of a stateful PCE. The scenarios apply equally to MPLS-TE and GMPLS deployments.

#### 3.1.1. Background

Traffic engineering has been a goal of the MPLS architecture since its inception ([RFC3031], [RFC2702], [RFC3346]). In the traffic engineering system provided by [RFC3630], [RFC5305], and [RFC3209] information about network resources utilization is only available as total reserved capacity by traffic class on a per interface basis; individual LSP state is available only locally on each LER for its own LSPs. In most cases, this makes good sense, as distribution and retention of total LSP state for all LERs within in the network would be prohibitively costly.

Unfortunately, this visibility in terms of global LSP state may result in a number of issues for some demand patterns, particularly within a common setup and hold priority. This issue affects online

traffic engineering systems.

A sufficiently over-provisioned system will by definition have no issues routing its demand on the shortest path. However, lowering the degree to which network over-provisioning is required in order to run a healthy, functioning network is a clear and explicit promise of MPLS architecture. In particular, it has been a goal of MPLS to provide mechanisms to alleviate congestion scenarios in which "traffic streams are inefficiently mapped onto available resources; causing subsets of network resources to become over-utilized while others remain underutilized" ([RFC2702]).

### 3.1.2. Why a Stateful PCE?

[RFC4655] defines a stateful PCE to be one in which the PCE maintains "strict synchronization between the PCE and not only the network states (in term of topology and resource information), but also the set of computed paths and reserved resources in use in the network." [RFC4655] also expressed a number of concerns with regard to a stateful PCE, specifically:

o Any reliable synchronization mechanism would result in significant control plane overhead

o Out-of-band TED synchronization would be complex and prone to race conditions

o Path calculations incorporating total network state would be highly complex

In general, stress on the control plane will be directly proportional to the size of the system being controlled and the tightness of the control loop, and indirectly proportional to the amount of over-provisioning in terms of both network capacity and reservation overhead.

Despite these concerns in terms of implementation complexity and scalability, several TE algorithms exist today that have been demonstrated to be extremely effective in large TE systems, providing both rapid convergence and significant benefits in terms of optimality of resource usage [MXMN-TE]. All of these systems share at least two common characteristics: the requirement for both global visibility of a flow (or in this case, a TE LSP) state and for ordered control of path reservations across devices within the system being controlled. While some approaches have been suggested in order to remove the requirements for ordered control (See [MPLS-PC]), these approaches are highly dependent on traffic distribution, and do not allow for multiple simultaneous LSP priorities representing diffserv

classes.

The following use cases demonstrate a need for visibility into global inter-PCC LSP state in PCE path computations, and for a PCE control of sequence and timing in altering LSP path characteristics within and across PCEP sessions.  Reference topologies for the use cases described later in this section are shown in Figures 1 and 2.

Unless otherwise cited, use cases assume that all LSPs listed exist at the same LSP priority.

```
          +-------+
          |   A   |
          |       |
          +-------+
                    \
                      +-------+                         +-------+
                      |   C   |-------------------------|   E   |
                      |       |                         |       |
                      +-------+        +-------+        +-------+
                    /          \       |   D   |       /
          +-------+              ------|       |------
          |   B   |                    +-------+
          |       |
          +-------+
```

Figure 1: Reference topology 1

```
               +-------+        +-------+        +-------+
               |   A   |        |   B   |        |   C   |
               |       |        |       |        |       |
               +---+---+        +---+---+        +---+---+
                   |                |                |
                   |                |                |
               +---+---+        +---+---+        +---+---+
               |   E   |        |   F   |        |   G   |
               |       +---------+      +---------+      |
               +-------+        +-------+        +-------+
```

Figure 2: Reference topology 2

### 3.1.2.1.  Throughput Maximization and Bin Packing

Because LSP attribute changes in [RFC5440] are driven by PCReq messages under control of a PCC's local timers, the sequence of RSVP reservation arrivals occurring in the network will be randomized. This, coupled with a lack of global LSP state visibility on the part

of a stateless PCE may result in suboptimal throughput in a given network topology.

Reference topology 2 in Figure 2 and Tables 1 and 2 show an example in which throughput is at 50% of optimal as a result of lack of visibility and synchronized control across PCC's. In this scenario, the decision must be made as to whether to route any portion of the E-G demand, as any demand routed for this source and destination will decrease system throughput.

| Link | Metric | Capacity |
|---|---|---|
| A-E | 1 | 10 |
| B-F | 1 | 10 |
| C-G | 1 | 10 |
| E-F | 1 | 10 |
| F-G | 1 | 10 |

Table 1: Link parameters for Throughput use case

| Time | LSP | Src | Dst | Demand | Routable | Path |
|---|---|---|---|---|---|---|
| 1 | 1 | E | G | 10 | Yes | E-F-G |
| 2 | 2 | A | B | 10 | No | --- |
| 3 | 1 | F | C | 10 | No | --- |

Table 2: Throughput use case demand time series

In many cases throughput maximization becomes a bin packing problem. While bin packing itself is an NP-hard problem, a number of common heuristics which run in polynomial time can provide significant improvements in throughput over random reservation event distribution, especially when traversing links which are members of the minimum cut set for a large subset of source destination pairs.

Tables 3 and 4 show a simple use case using Reference Topology 1 in Figure 1, where LSP state visibility and control of reservation order across PCCs would result in significant improvement in total throughput.

| Link | Metric | Capacity |
| --- | --- | --- |
| A-C | 1 | 10 |
| B-C | 1 | 10 |
| C-E | 10 | 5 |
| C-D | 1 | 10 |
| D-E | 1 | 10 |

Table 3: Link parameters for Bin Packing use case

| Time | LSP | Src | Dst | Demand | Routable | Path |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | 1 | A | E | 5 | Yes | A-C-D-E |
| 2 | 2 | B | E | 10 | No | --- |

Table 4: Bin Packing use case demand time series

#### 3.1.2.2. Deadlock

Most existing RSVP-TE implementations will not tear down established LSPs in the event of the failure of the bandwidth increase procedure detailed in [RFC3209]. This behavior is directly implied to be correct in [RFC3209] and is often desirable from an operator's perspective, because either a) the destination prefixes are not reachable via any means other than MPLS or b) this would result in significant packet loss as demand is shifted to other LSPs in the overlay mesh.

In addition, there are currently few implementations offering ingress admission control at the LSP level. Again, having ingress admission control on a per LSP basis is not necessarily desirable from an operational perspective, as a) one must over-provision LSPs significantly in order to avoid deleterious effects resulting from stacked transport and flow control systems and b) there is currently no efficient commonly available northbound interface for dynamic configuration of per LSP ingress admission control (such an interface could easily be defined using the extensions present in this spec, but it beyond the scope of the current document).

Lack of ingress admission control coupled with the behavior in [RFC3209] effectively results in mis-signaled LSPs during periods of contention for network capacity between LSPs in a given LSP priority. This in turn causes information loss in the TED with regard to actual network state, resulting in LSPs sharing common network interfaces

with mis-signaled LSPs operating in a degraded state for significant periods of time, even when unused network capacity may potentially be available.

Reference Topology 1 in Figure 1 and Tables 5 and 6 show a use case that demonstrates this behavior. Two LSPs, LSP 1 and LSP 2 are signaled with demand 2 and routed along paths A-C-D-E and B-C-D-E respectively. At a later time, the demand of LSP 1 increases to 20. Under such a demand, the LSP cannot be resignaled. However, the existing LSP will not be torn down. In the absence of ingress policing, traffic on LSP 1 will cause degradation for traffic of LSP 2 (due to oversubscription on the links C-D and D-E), as well as information loss in the TED with regard to the actual network state.

The problem could be easily ameliorated by global visibility of LSP state coupled with PCC- external demand measurements and placement of two LSPs on disjoint links. Note that while the demand of 20 for LSP 1 could never be satisfied in the given topology, what could be achieved would be isolation from the ill-effects of the (unsatisfiable) increased demand.

| Link | Metric | Capacity |
|---|---|---|
| A-C | 1 | 10 |
| B-C | 1 | 10 |
| C-E | 10 | 5 |
| C-D | 1 | 10 |
| D-E | 1 | 10 |

Table 5: Link parameters for the 'Deadlock' example

| Time | LSP | Src | Dst | Demand | Routable | Path |
|---|---|---|---|---|---|---|
| 1 | 1 | A | E | 2 | Yes | A-C-D-E |
| 2 | 2 | B | E | 2 | Yes | B-C-D-E |
| 3 | 1 | A | E | 20 | No | --- |

Table 6: Deadlock LSP and demand time series

#### 3.1.2.3. Minimum Perturbation

As a result of both the lack of visibility into global LSP state and the lack of control over event ordering across PCE sessions, unnecessary perturbations may be introduced into the network by a

stateless PCE. Tables 7 and 8 show an example of an unnecessary network perturbation using Reference Topology 1 in Figure 1. In this case an unimportant (high LSP priority value) LSP (LSP1) is first set up along the shortest path. At time 2, which is assumed to be relatively close to time 1, a second more important (lower LSP-priority value) LSP is established, preempting LSP 1 and shifting it to the longer A-C-E path.

| Link | Metric | Capacity |
|---|---|---|
| A-C | 1 | 10 |
| B-C | 1 | 10 |
| C-E | 10 | 10 |
| C-D | 1 | 10 |
| D-E | 1 | 10 |

Table 7: Link parameters for the 'Minimum-Perturbation' example

| Time | LSP | Src | Dst | Demand | LSP Prio | Routable | Path |
|---|---|---|---|---|---|---|---|
| 1 | 1 | A | E | 7 | 7 | Yes | A-C-D-E |
| 2 | 2 | B | E | 7 | 0 | Yes | B-C-D-E |
| 3 | 1 | A | E | 7 | 7 | Yes | A-C-E |

Table 8: Minimum-Perturbation LSP and demand time series

#### 3.1.2.4. Predictability

Randomization of reservation events caused by lack of control over event ordering across PCE sessions results in poor predictability in LSP routing. An offline system applying a consistent optimization method will produce predictable results to within either the boundary of forecast error when reservations are over-provisioned by reasonable margins or to the variability of the signal and the forecast error when applying some hysteresis in order to minimize churn.

Reference Topology 1 and Tables 9, 10 and 11 show the impact of event ordering and predictability of LSP routing.

| Link | Metric | Capacity |
|---|---|---|
| A-C | 1 | 10 |
| B-C | 1 | 10 |
| C-E | 1 | 10 |
| C-D | 1 | 10 |
| D-E | 1 | 10 |

Table 9: Link parameters for the 'Predictability' example

| Time | LSP | Src | Dst | Demand | Routable | Path |
|---|---|---|---|---|---|---|
| 1 | 1 | A | E | 7 | Yes | A-C-E |
| 2 | 2 | B | E | 7 | Yes | B-C-D-E |

Table 10: Predictability LSP and demand time series 1

| Time | LSP | Src | Dst | Demand | Routable | Path |
|---|---|---|---|---|---|---|
| 1 | 2 | B | E | 7 | Yes | B-C-E |
| 2 | 1 | A | E | 7 | Yes | A-C-D-E |

Table 11: Predictability LSP and demand time series 2

#### 3.1.2.5. Global Concurrent Optimization

Global Concurrent Optimization (GCO) defined in [RFC5557] is a network optimization mechanism that is able to simultaneously consider the entire topology of the network and the complete set of existing TE LSPs and their existing constraints, and look to optimize or reoptimize the entire network to satisfy all constraints for all TE LSPs.  It allows for bulk path computations in order to avoid blocking problems and to achieve more optimal network-wide solutions.

Global control of LSP operation sequence in [RFC5557] is predicated on the use of what is effectively a stateful (or semi-stateful) NMS. The NMS can be either not local to the switch, in which case another northbound interface is required for LSP attribute changes, or local/ collocated, in which case there are significant issues with efficiency in resource usage.  Stateful PCE adds a few features that:

- Roll the NMS visibility into the PCE and remove the requirement for an additional northbound interface

- Allow the PCE to determine when re-optimization is needed

- Allow the PCE to determine which LSPs should be re-optimized

- Allow a PCE to control the sequence of events across multiple PCCs, allowing for bulk (and truly global) optimization, LSP shuffling etc.

### 3.1.3. Protocol vs. Configuration

Note that existing configuration tools and protocols can be used to set LSP state.  However, this solution has several shortcomings:

- Scale & Performance: configuration operations often require processing of additional configuration portions beyond the state being directly acted upon, with corresponding cost in CPU cycles, negatively impacting both PCC stability LSP update rate capacity.

- Scale & Performance: configuration operations often have transactional semantics which are typically heavyweight and require additional CPU cycles, negatively impacting PCC update rate capacity.

- Security: opening up a configuration channel to a PCE would allow a malicious PCE to take over a PCC.  The PCEP extensions described in this document only allow a PCE control over a very limited set of LSP attributes.

- Interoperability: each vendor has a proprietary information model for configuring LSP state, which prevents interoperability of a PCE with PCCs from different vendors.  The PCEP extensions described in this document allow for a common information model for LSP state for all vendors.

- Efficient State Synchronization: configuration channels may be heavyweight and unidirectional, therefore efficient state synchronization between a PCE and a PCE may be a problem.

## 3.2. Objectives

The objectives for the protocol extensions to support stateful PCE described in this document are as follows:

- Allow a single PCC to interact with a mix of stateless and stateful PCEs simultaneously using the same PCEP.

- Support efficient LSP state synchronization between the PCC and one or more active or passive stateful PCEs.

- Allow a PCC to delegate control of its LSPs to an active stateful PCE such that a single LSP is under the control a single PCE at any given time. A PCC may revoke this delegation at any time during the lifetime of the LSP. If LSP delegation is revoked while the PCEP session is up, the PCC MUST notify the PCE about the revocation. A PCE may return an LSP delegation at any point during the lifetime of the PCEP session.

- Allow a PCE to control computation timing and update timing across all LSPs that have been delegated to it.

- Enable uninterrupted operation of PCC's LSPs in the event PCE failure or while control of LSPs is being transferred between PCEs.

## 4. New Functions to Support Stateful PCEs

Several new functions will be required in PCEP to support stateful PCEs. A function can be initiated either from a PCC towards a PCE (C-E) or from a PCE towards a PCC (E-C). The new functions are:

Capability negotiation (E-C,C-E): both the PCC and the PCE must announce during PCEP session establishment that they support PCEP Stateful PCE extensions defined in this document.

LSP state synchronization (C-E): after the session between the PCC and a stateful PCE is initialized, the PCE must learn the state of a PCC's LSPs before it can perform path computations or update LSP attributes in a PCC.

LSP Update Request (E-C): A PCE requests modification of attributes on a PCC's LSP.

LSP State Report (C-E): a PCC sends an LSP state report to a PCE whenever the state of an LSP changes.

LSP control delegation (C-E,E-C): a PCC grants to a PCE the right to update LSP attributes on one or more LSPs; the PCE becomes the authoritative source of the LSP's attributes as long as the delegation is in effect (See Section 5.5); the PCC may withdraw the delegation or the PCE may give up the delegation at any time.

[I-D.sivabalan-pce-disco-stateful] defines the extensions needed to support autodiscovery of stateful PCEs when using OSPF ([RFC5088]) or

IS-IS ([RFC5089]) for PCE discovery.

## 5. Architectural Overview of Protocol Extensions

### 5.1. LSP State Ownership

In the PCEP protocol (defined in [RFC5440]), LSP state and operation are under the control of a PCC (a PCC may be an LSR or a management station). Attributes received from a PCE are subject to PCC's local policy. The PCEP protocol extensions described in this document do not change this behavior.

An active stateful PCE may have control of a PCC's LSPs be delegated to it, but the LSP state ownership is retained by the PCC. In particular, in addition to specifying values for LSP's attributes, an active stateful PCE also decides when to make LSP modifications.

Retaining LSP state ownership on the PCC allows for:

- a PCC to interact with both stateless and stateful PCEs at the same time

- a stateful PCE to only modify a small subset of LSP parameters, i.e. to set only a small subset of the overall LSP state; other parameters may be set by the operator through CLI commands

- a PCC to revert delegated LSP to an operator-defined default or to delegate the LSPs to a different PCE, if the PCC get disconnected from a PCE with currently delegated LSPs

### 5.2. New Messages

In this document, we define the following new PCEP messages:

Path Computation State Report (PCRpt): a PCEP message sent by a PCC to a PCE to report the status of one or more LSPs. Each LSP Status Report in a PCRpt message can contain the actual LSP's path,bandwidth, operational and administrative status, etc. An LSP Status Report carried on a PCRpt message is also used in delegation or revocation of control of an LSP to/from a PCE. The PCRpt message is described in Section 6.1.

Path Computation Update Request (PCUpd): a PCEP message sent by a PCE to a PCC to update LSP parameters, on one or more LSPs. Each LSP Update Request on a PCUpd message MUST contain all LSP parameters that a PCE wishes to set for a given LSP. An LSP Update Request carried on a PCUpd message is also used to return

LSP delegations if at any point PCE no longer desires control of an LSP. The PCUpd message is described in Section 6.2.

The new functions defined in Section 4 are mapped onto the new messages as shown in the following table.

| Function | Message |
|---|---|
| Capability Negotiation (E-C,C-E) | Open |
| State Synchronization (C-E) | PCRpt |
| LSP State Report (C-E) | PCRpt |
| LSP Control Delegation (C-E,E-C) | PCRpt, PCUpd |
| LSP Update Request (E-C) | PCUpd |
| ISIS stateful capability advertisement | ISIS PCE-CAP-FLAGS sub-TLV |
| OSPF stateful capability advertisement | OSPF RI LSA, PCE TLV, PCE-CAP-FLAGS sub-TLV |

Table 12: New Function to Message Mapping

## 5.3. Capability Negotiation

During PCEP Initialization Phase, PCEP Speakers (PCE pr PCC) negotiate the use of stateful PCEP extensions. A PCEP Speaker includes the "Stateful PCE Capability" TLV, described in Section 7.1.1, in the OPEN Object to advertise its support for PCEP stateful extensions. The Stateful Capability TLV includes the 'LSP Update' Flag that indicates whether the PCEP Speaker supports LSP parameter updates.

The presence of the Stateful PCE Capability TLV in PCC's OPEN Object indicates that the PCC is willing to send LSP State Reports whenever LSP parameters or operational status changes.

The presence of the Stateful PCE Capability TLV in PCE's OPEN message indicates that the PCE is interested in receiving LSP State Reports whenever LSP parameters or operational status changes.

The PCEP protocol extensions for stateful PCEs MUST NOT be used if one or both PCEP Speakers have not included the Stateful PCE Capability TLV in their respective OPEN message. If the PCEP Speakers support the extensions of this draft, then a PCErr with code "Stateful PCE capability not negotiated" (see Section 8.4) will be generated and the PCEP session will be terminated.

LSP delegation and LSP update operations defined in this document MAY

only be used if both PCEP Speakers set the LSP-UPDATE Flag in the "Stateful Capability" TLV to 'Updates Allowed (U Flag = 1)'. If this is not the case and LSP delegation or LSP update operations are attempted, then a PCErr with code "Delegation not negotiated" (see Section 8.4) SHOULD be generated. Note that even if the update capability has not been negotiated, a PCE can still receive LSP Status Reports from a PCC and build and maintain an up to date view of the state of the PCC's LSPs.

### 5.4. State Synchronization

The purpose of State Synchronization is to provide a checkpoint-in-time state replica of a PCC's LSP state in a PCE. State Synchronization is performed immediately after the Initialization phase ([RFC5440]).

During State Synchronization, a PCC first takes a snapshot of the state of its LSPs state, then sends the snapshot to a PCE in a sequence of LSP State Reports. Each LSP State Report sent during State Synchronization has the SYNC Flag in the LSP Object set to 1. The set of LSPs for which state is synchronized with a PCE is determined by negotiated stateful PCEP capabilities and PCC's local configuration (see more details in Section 9.1).

The end of synchronization marker is a PCRpt message with the SYNC Flag set to 0 for an LSP Object with PLSP-ID equal to the reserved value 0. The LSP Object does not include the SYMBOLIC-PATH-NAME TLV in this case. If both PCEP speakers had set the INCLUDE-DB-VERSION Flag in the OPEN object's STATEFUL-PCE-CAPABILITY TLV, then the LSP-DB-VERSION TLV MUST be included and contain the PCC's latest LSP State Database version.

A PCE SHOULD NOT send PCUpd messages to a PCC before State Synchronization is complete. A PCC SHOULD NOT send PCReq messages to a PCE before State Synchronization is complete. This is to allow the PCE to get the best possible view of the network before it starts computing new paths.

If the PCC encounters a problem which prevents it from completing the state transfer, it MUST send a PCErr message to the PCE and terminate the session using the PCEP session termination procedure.

In the event of a PCC resetting the session during synchronization, the PCE MUST clean up state it received from this PCC. Session reestablishment MUST be re-attempted per the procedures defined in [RFC5440].

The PCE does not send positive acknowledgements for properly received

synchronization messages. It MUST respond with a PCErr message indicating "PCRpt error" (see Section 8.4) if it encounters a problem with the LSP State Report it received from the PCC. Either the PCE or the PCC MAY terminate the session if the PCE encounters a problem during the synchronization.

The successful State Synchronization sequence is shown in Figure 3.

```
                  +-+-+                    +-+-+
                  |PCC|                    |PCE|
                  +-+-+                    +-+-+
                    |                        |
                    |-----PCRpt, SYNC=1----->| (Sync start)
                    |                        |
                    |-----PCRpt, SYNC=1----->|
                    |            .           |
                    |            .           |
                    |            .           |
                    |-----PCRpt, SYNC=1----->| (Sync done)
                    |            .           |
                    |            .           |
                    |            .           |
                    |                        |
                    |-----PCRpt, SYNC=0----->| (End of sync marker
                    |                        |  LSP State Report
                    |                        |  for PLSP-ID=0)
```

Figure 3: Successful state synchronization

The sequence where the PCE fails during the State Synchronization phase is shown in Figure 4.

```
                     +-+-+                    +-+-+
                     |PCC|                    |PCE|
                     +-+-+                    +-+-+
                       |                        |
                       |-----PCRpt, SYNC=1----->|
                       |                        |
                       |-----PCRpt, SYNC=1----->|
                       |            .           |
                       |            .           |
                       |            .           |
                       |-----PCRpt, SYNC=1----->|
                       |                        |
                       |-PCRpt, SYNC=1          |
                       |         \    ,-PCErr=?-|
                       |          \  /          |
                       |           \/           |
                       |           /\           |
                       |          /   `-------->| (Ignored)
                       |<--------`              |
```

Figure 4: Failed state synchronization (PCE failure)

The sequence where the PCC fails during the State Synchronization phase is shown in Figure 5.

```
                     +-+-+                    +-+-+
                     |PCC|                    |PCE|
                     +-+-+                    +-+-+
                       |                        |
                       |-----PCRpt, SYNC=1----->|
                       |                        |
                       |-----PCRpt, SYNC=1----->|
                       |            .           |
                       |            .           |
                       |            .           |
                       |-------- PCErr=? ------>|
                       |                        |
```

Figure 5: Failed state synchronization (PCC failure)

### 5.4.1. State Synchronization Avoidance

State Synchronization MAY be skipped following a PCEP session restart if the state of both PCEP peers did not change during the period prior to session re-initialization. To be able to make this determination, state must be exchanged and maintained by both PCE and PCC during normal operation. This is accomplished by keeping track of the changes to the LSP State Database, using a database version

called the LSP State Database Version.

The LSP State Database version is an unsigned 64-bit value that MUST be incremented by 1 for each successive change in the LSP state database. The LSP State Database version MUST start at 1 and may wrap around. Values 0 and 0xFFFFFFFFFFFFFFFF are reserved. The PCC is the owner of the LSP State Database version, which is incremented each time a change is made to the PCC's local LSP State Database. Operations that trigger a change to the local LSP State database include a change in the LSP operational state, delegation of an LSP, removal or addition of an LSP or change in any of the LSP attributes that would trigger a report to the PCE. When State Synchronization avoidance is enabled on a PCEP session, a PCC includes the LSP-DB-VERSION TLV as an optional TLV in the LSP Object on each LSP State Report. The LSP-DB-VERSION TLV contains a PCC's LSP State Database version.

State Synchronization Avoidance is negotiated on a PCEP session during session startup. To make sure that a PCEP peer can recognize a previously connected peer even if its IP address changed, each PCEP peer includes the PREDUNDANCY-GROUP-ID TLV in the OPEN message.

If both PCEP speakers set the INCLUDE-DB-VERSION Flag in the OPEN object's STATEFUL-PCE-CAPABILITY TLV to 1, the PCC will include the LSP-DB-VERSION TLV in each LSP Object. The TLV will contain the PCC's latest LSP State Database version.

If a PCE's LSP State Database survived the restart of a PCEP session, the PCE will include the LSP-DB-VERSION TLV in its OPEN object, and the TLV will contain the last LSP State Database version received on an LSP State Report from the PCC in a previous PCEP session. If a PCC's LSP State Database survived the restart, the PCC will include the LSP-DB-VERSION TLV in its OPEN object and the TLV will contain the last LSP State Database version sent on an LSP State Report from the PCC in the previous PCEP session. If a PCEP Speaker's LSP State Database did not survive the restart of a PCEP session, the PCEP Speaker MUST NOT include the LSP-DB-VERSION TLV in the OPEN Object.

If both PCEP Speakers include the LSP-DB-VERSION TLV in the OPEN Object and the TLV values match, the PCC MAY skip State Synchronization. Otherwise, the PCC MUST perform State Synchronization. If the PCC attempts to skip State Synchronization (i.e. the SYNC Flag = 0 on the first LSP State Report from the PCC), the PCE MUST send back a PCError with Error-type 20 Error-value 2 'LSP Database version mismatch', and close the PCEP session.

If state synchronization is required, then prior to completing the Initialization phase, the PCE MUST mark any LSPs in the LSP database

that were previously reported by the PCC as stale. When the PCC reports an LSP during state synchronization, if the LSP already exists in the LSP database, the PCE MUST update the LSP database and clear the stale marker from the LSP. When it has finished state synchronization, the PCC MUST immediately send a PCRpt message containing a state report with an LSP object containing an PLSP-ID of 0 and with the SYNC flag set to 0 (see Section 5.4). This state report indicates to the PCE that state synchronization has completed. On receiving this state report, the PCE MUST purge any LSPs from the LSP database that are still marked as stale.

Note that a PCE MAY force State Synchronization by not including the LSP-DB-VERSION TLV in its OPEN object.

Figure 6 shows an example sequence where State Synchronization is skipped.

```
                  +-+-+                    +-+-+
                  |PCC|                    |PCE|
                  +-+-+                    +-+-+
                    |                        |
                    |--Open--,               |
                    |  DBv=42 \    ,---Open--|
                    |   IDB=1  \  /   DBv=42 |
                    |           \/     IDB=1 |
                    |           /\           |
                    |          /   `-------->| (OK to skip sync)
        (Skip sync) |<--------`              |
                    |            .           |
                    |            .           |
                    |            .           |
                    |                        |
                    |--PCRpt,DBv=43,SYNC=0-->| (Regular
                    |                        |  LSP State Report)
                    |--PCRpt,DBv=44,SYNC=0-->| (Regular
                    |                        |  LSP State Report)
                    |--PCRpt,DBv=45,SYNC=0-->|
                    |                        |
```

Figure 6: State Synchronization skipped

Figure 7 shows an example sequence where State Synchronization is performed due to LSP State Database version mismatch during the PCEP session setup. Note that the same State Synchronization sequence would happen if either the PCC or the PCE would not include the LSP-DB-VERSION TLV in their respective Open messages.

```
                     +-+-+                    +-+-+
                     |PCC|                    |PCE|
                     +-+-+                    +-+-+
                       |                        |
                       |--Open--,               |
                       |  DBv=46 \    ,---Open--|
                       |   IDB=1  \  /   DBv=42 |
                       |           \/     IDB=1 |
                       |           /\           |
                       |          /   `-------->| (Expect sync)
             (Do sync) |<--------`              | (Purge LSP State)
                       |                        |
                       |--PCRpt,DBv=46,SYNC=1-->| (Sync start)
                       |            .           |
                       |            .           |
                       |            .           |
                       |--PCRpt,DBv=46,SYNC=1-->| (Sync done)
                       |            .           |
                       |            .           |
                       |            .           |
                       |--PCRpt,DBv=47,SYNC=0-->| (Regular
                       |                        |  LSP State Report)
                       |--PCRpt,DBv=48,SYNC=0-->| (Regular
                       |                        |  LSP State Report)
                       |--PCRpt,DBv=49,SYNC=0-->|
                       |                        |
```

Figure 7: State Synchronization performed

Figure 8 shows an example sequence where State Synchronization is skipped, but because one or both PCEP Speakers set the INCLUDE-DB-VERSION Flag to 0, the PCC does not send LSP-DB-VERSION TLVs to the PCE.  If the current PCEP session restarts, the PCEP Speakers will have to perform State Synchronization, since the PCE will not know the PCC's latest LSP State Database version.

```
                     +-+-+                    +-+-+
                     |PCC|                    |PCE|
                     +-+-+                    +-+-+
                       |                        |
                       |--Open--,               |
                       |  DBv=42 \    ,---Open--|
                       |   IDB=0  \  /   DBv=42 |
                       |           \/     IDB=0 |
                       |           /\           |
                       |          /   `-------->| (OK to skip sync)
          (Skip sync)  |<--------`              |
                       |            .           |
                       |            .           |
                       |            .           |
                       |------PCRpt,SYNC=0----->| (Regular
                       |                        |  LSP State Report)
                       |------PCRpt,SYNC=0----->| (Regular
                       |                        |  LSP State Report)
                       |------PCRpt,SYNC=0----->|
                       |                        |
```

Figure 8: State Synchronization skipped, no LSP-DB-VERSION TLVs sent from PCC

### 5.4.2. PCE-triggered State Synchronization

Because the accuracy of the computations performed by the PCE is tied to the accuracy of the view the PCE has on the state of the LSPs, it can be beneficial to be able to resynchronize this state even after the session has established. The PCE may use this approach to continuously sanity check its state against the network, or to recover from error conditions without having to tear down sessions.

To trigger a resynchronization with a PCC, the PCE MUST first mark any LSPs in the LSP database that were previously reported by the PCC as stale and then send a PCUpd for an LSP object containing a PLSP-ID of 0 and with the SYNC flag set to 1. This PCUpd message is the trigger for the PCC to enter the synchronization phase as described in Section 5.4.1 and start sending PCRpt messages. After the receipt of the end-of-synchronization marker, the PCE will purge LSPs which were not refreshed.

## 5.5. LSP Delegation

If during Capability negotiation both the PCE and the PCC have indicated that they support LSP Update, then the PCC may choose to grant the PCE a temporary right to update (a subset of) LSP attributes on one or more LSPs. This is called "LSP Delegation", and

it MAY be performed at any time after the Initialization phase, including during the State Synchronization phase.

LSP Delegation is controlled by operator-defined policies on a PCC. LSPs are delegated individually - different LSPs may be delegated to different PCEs. An LSP is delegated to at most one PCE at any given point in time. The delegation policy, when all PCC's LSPs are delegated to a single PCE at any given time, SHOULD be supported by all delegation-capable PCCs. Conversely, the policy revoking the delegation for all PCC's LSPs SHOULD also be supported

A PCE may return LSP delegation at any time if it no longer wishes to update the LSP's state. A PCC may revoke LSP delegation at any time. Delegation, Revocation, and Return are done individually for each LSP.

In the event of an delegation being rejected or returned by a PCE, the PCC should react based on local policy. It can, for example, either retry delegating to the same PCE using an exponentially increasing timer or delegate to an alternate PCE.

### 5.5.1. Delegating an LSP

A PCC delegates an LSP to a PCE by setting the Delegate flag in LSP State Report to 1. If the PCE does not accept the LSP Delegation, it MUST immediately respond with an empty LSP Update Request which has the Delegate flag set to 0. If the PCE accepts the LSP Delegation, it confirms this when it sends the first LSP Update Request for the delegated LSP to the PCC by setting the Delegate flag to 1 (note that this may occur at a later time).

The delegation sequence is shown in Figure 9.

```
                  +-+-+                    +-+-+
                  |PCC|                    |PCE|
                  +-+-+                    +-+-+
                    |                        |
                    |---PCRpt, Delegate=1--->| LSP Delegated
                    |                        |
                    |---PCRpt, Delegate=1--->|
                    |            .           |
                    |            .           |
                    |            .           |
                    |<--(PCUpd,Delegate=1)---| Delegation confirmed
                    |                        |
                    |---PCRpt, Delegate=1--->|
                    |                        |
```

Figure 9: Delegating an LSP

Note that for an LSP to remain delegated to a PCE, the PCC MUST set the Delegate flag to 1 on each LSP Status Report sent to the PCE.

### 5.5.2. Revoking a Delegation

When a PCC decides that a PCE is no longer permitted to modify an LSP, it revokes that LSP's delegation to the PCE. A PCC may revoke an LSP delegation at any time during the LSP's life time. A PCC revoking an LSP delegation MAY immediately clear the LSP state provided by the PCE, but to avoid traffic loss, it SHOULD do so in a make-before-break fashion. If the PCC has received but not yet acted on PCUpd messages from the PCE for the LSP whose delegation is being revoked, then it SHOULD ignore these PCUpd messages when processing the message queue. All effects of all messages for which processing started before the revocation took place MUST be allowed to complete and the result MUST be given the same treatment as any LSP that had been previously delegated to the PCE (e.g. the state MAY be immediately cleared). Any further PCUpd messages from the PCE are handled according to the PCUpd procedures described in this document.

If a PCEP session with the PCE to which the LSP is delegated exists in the UP state during the revocation, the PCC MUST notify that PCE by sending an LSP State Report with the Delegate flag set to 0, as shown in Figure 10.

```
+-+-+                     +-+-+
|PCC|                     |PCE|
+-+-+                     +-+-+
  |                         |
  |---PCRpt, Delegate=1---->|
  |                         |
  |<--(PCUpd,Delegate=1)---| Delegation confirmed
  |            .            |
  |            .            |
  |            .            |
  |---PCRpt, Delegate=0---->| PCC revokes delegation
  |                         |
```

Figure 10: Revoking a Delegation

After an LSP delegation has been revoked, a PCE can no longer update LSP's parameters; an attempt to update parameters of a non-delegated LSP will result in the PCC sending a PCErr message indicating "LSP is not delegated" (see Section 8.4).

When a PCC's PCEP session with a PCE terminates unexpectedly, the PCC

MUST wait the time interval specified in Redelegation Timeout Interval before revoking LSP delegations to that PCE and attempting to redelegate LSPs to an alternate PCE. If a PCEP session with the original PCE can be reestablished before the Redelegation Timeout Interval timer expires, LSP delegations to the PCE remain intact.

Likewise, when a PCC's PCEP session with a PCE terminates unexpectedly, the PCC MUST wait for the State Timeout Interval before flushing any LSP state associated with that PCE. Note that the State Timeout Interval timer may expire before the PCC has redelegated the LSPs to another PCE, for example if a PCC is not connected to any active stateful PCE or if no connected active stateful PCE accepts the delegation. In this case, the PCC SHALL flush any LSP state set by the PCE upon expiration of the State Timeout Interval and revert to operator-defined default parameters. This operation SHOULD be done in a make-before-break fashion.

The State Timeout Interval SHOULD be greater than or equal to the Redelegation Timeout Interval and MAY be set to infinity (meaning that until the PCC specifically takes action to change the parameters set by the PCE, they will remain intact).

If State Synchronization Avoidance is enabled, a PCC MUST increment its LSP State Database version when the 'Redelegation Timeout Interval' timer expires.

### 5.5.3. Returning a Delegation

A PCE that no longer wishes to update an LSP's parameters SHOULD return the LSP delegation back to the PCC by sending an empty LSP Update Request which has the Delegate flag set to 0. Note that in order to keep a delegation, the PCE MUST set the Delegate flag to 1 on each LSP Update Request sent to the PCC.

```
                  +-+-+                    +-+-+
                  |PCC|                    |PCE|
                  +-+-+                    +-+-+
                    |                        |
                    |---PCRpt, Delegate=1--->| LSP delegated
                    |            .           |
                    |            .           |
                    |            .           |
                    |<--PCUpd, Delegate=0----| Delegation returned
                    |                        |
                    |---PCRpt, Delegate=0--->| No delegation for LSP
                    |                        |
```

Figure 11: Returning a Delegation

If a PCC cannot delegate an LSP to a PCE (for example, if a PCC is not connected to any active stateful PCE or if no connected active stateful PCE accepts the delegation), the LSP delegation on the PCC will time out within a configurable Redelegation Timeout Interval and the PCC MUST flush any LSP state set by a PCE at the expiration of the State Timeout Interval.

#### 5.5.4. Redundant Stateful PCEs

Note that a PCE may not have any delegated LSPs: in a redundant configuration where one PCE is backing up another PCE, the backup PCE may have only a subset of LSPs delegated to it. The backup PCE does not update any LSPs that are not delegated to it, but receives all LSP State Reports from a PCC. When the primary PCE for a given LSP set fails, after expiry of the Redelegation Timeout Interval, the PCC SHOULD delegate to the redundant PCE all LSPs that had been previously delegated to the failed PCE. Assuming that the State Timeout Interval had been configured to be larger than the Redelegation Timeout Interval (as recommended), this delegation change will not cause any changes to the LSP parameters.

#### 5.5.5. Redelegation on PCE failure

On failure, the goal is to: 1) avoid any traffic loss on the LSPs that were updated by the PCE that crashed 2) minimize the churn in the network in terms of ownership of the LSPs, 3) not leave any "orphan" (undelegated) LSPs and 4) be able to control when the state that was set by the PCE can be changed or purged. The values chosen for the Redelegation Timeout and State Timeout values affect the ability to accomplish these goals.

This section summarizes the behaviour with regards to LSP delegation and LSP state on a PCE failure.

If the PCE crashses but recovers within the Redelegation Timeout, both the delegation state and the LSP state are kept intact.

If the PCE crashes but does not recover within the Redelegation Timeout, the delegation state is returned to the PCC. If the PCC can redelegate the LSPs to another PCE, and that PCE accepts the delegations, there will be no change in LSP state. If the PCC cannot redelegate the LSPs to another PCE, then upon expiration of the State Timeout Interval, the state set by the PCE is flushed, which may cause change in the LSP state. Note that an operator may choose to use an infinite State Timeout Interval if he wishes to maintain the PCE state indefinetely. Note also that flushing the state should be implemented using make-before-break to avoid traffic loss.

If there is a hot-standby PCE, the Redelegation Timeout may be set to 0 through policy on the PCC, causing the LSPs to be redelegated immediately to the PCC, which can delegate them immediately to the hot-standby PCE.  Assuming the State Timeout Interval is larger than the Redelegation Timeout, the LSP state will be kept intact.

## 5.6.  LSP Operations

### 5.6.1.  Passive Stateful PCE Path Computation Request/Response

```
                    +-+-+                    +-+-+
                    |PCC|                    |PCE|
                    +-+-+                    +-+-+
                      |                        |
1) Path computation   |----- PCReq message --->|
   request sent to    |                        |2) Path computation
   PCE                |                        |   request received,
                      |                        |   path computed
                      |                        |
                      |<---- PCRep message ----|3) Computed paths
                      |     (Positive reply)   |   sent to the PCC
                      |     (Negative reply)   |
4) LSP Status change  |                        |
   event              |                        |
                      |                        |
5) LSP Status Report  |----- PCRpt message --->|
   sent to all        |            .           |
   stateful PCEs      |            .           |
                      |            .           |
6) Repeat for each    |----- PCRpt message --->|
   LSP status change  |                        |
                      |                        |
```

Figure 12: Passive Stateful PCE Path Computation Request/Response

Once a PCC has successfully established a PCEP session with a passive stateful PCE and the PCC's LSP state is synchronized with the PCE (i.e. the PCE knows about all PCC's existing LSPs), if an event is triggered that requires the computation of a set of paths, the PCC sends a path computation request to the PCE ([RFC5440], Section 4.2.3).  The PCReq message MAY contain the LSP Object to identify the LSP for which the path computation is requested.

Upon receiving a path computation request from a PCC, the PCE triggers a path computation and returns either a positive or a negative reply to the PCC ([RFC5440], Section 4.2.4).

Upon receiving a positive path computation reply, the PCC receives a

set of computed paths and starts to setup the LSPs.  For each LSP, it sends an LSP State Report carried on a PCRpt message to the PCE, indicating that the LSP's status is 'Pending'.

Once an LSP is up, the PCC sends an LSP State Report carried on a PCRpt message to the PCE, indicating that the LSP's status is 'Up'. If the LSP could not be set up, the PCC sends an LSP State Report indicating that the LSP is "Down' and stating the cause of the failure.  Note that due to timing constraints, the LSP status may change from 'Pending' to 'Up' (or 'Down') before the PCC has had a chance to send an LSP State Report indicating that the status is 'Pending'.  In such cases, the PCC may choose to only send the PCRpt indicating the latest status ('Up' or 'Down').

Upon receiving a negative reply from a PCE, a PCC may decide to resend a modified request or take any other appropriate action.  For each requested LSP, it also sends an LSP State Report carried on a PCRpt message to the PCE, indicating that the LSP's status is 'Down'.

There is no direct correlation between PCRep and PCRpt messages.  For a given LSP, multiple LSP State Reports will follow a single PC Reply, as a PCC notifies a PCE of the LSP's state changes.

A PCC sends each LSP State Report to each stateful PCE that is connected to the PCC.

Note that a single PCRpt message MAY contain multiple LSP State Reports.

The passive stateful PCE is the model for stateful PCEs is described in [RFC4655], Section 6.8.

### 5.6.2. Active Stateful PCE LSP Update

```
                     +-+-+                    +-+-+
                     |PCC|                    |PCE|
                     +-+-+                    +-+-+
                       |                        |
   1) LSP State        |-- PCRpt, Delegate=1 -->|
      Synchronization  |            .           |
      or add new LSP   |            .           |2) PCE decides to
                       |            .           |   update the LSP
                       |                        |
                       |<---- PCUpd message ----|3) PCUpd message sent
                       |                        |   to PCC
                       |                        |
                       |                        |
   4) LSP Status Report|---- PCRpt message ---->|
      sent(->Pending)  |            .           |
                       |            .           |
                       |            .           |
   5) LSP Status Report|---- PCRpt message ---->|
      sent (->Up|Down) |                        |
                       |                        |
```

Figure 13: Active Stateful PCE

Once a PCC has successfully established a PCEP session with an active stateful PCE, the PCC's LSP state is synchronized with the PCE (i.e. the PCE knows about all PCC's existing LSPs) and LSPs have been delegated to the PCE, the PCE can modify LSP parameters of delegated LSPs.

A PCE sends an LSP Update Request carried on a PCUpd message to the PCC. The LSP Update Request contains a variety of objects that specify the set of constraints and attributes for the LSP's path. Each LSP Update Request has a unique identifier, the SRP-ID-number, carried in the SRP (Stateful PCE Request Parameters) Object described in Section 7.2. The SRP-ID-number is used to correlate errors and state reports to LSP Update Requests. A single PCUpd message MAY contain multiple LSP Update Requests.

Upon receiving a PCUpd message the PCC starts to setup LSPs specified in LSP Update Requests carried in the message. For each LSP, it sends an LSP State Report carried on a PCRpt message to the PCE, indicating that the LSP's status is 'Pending'. If the PCC decides that the LSP parameters proposed in the PCUpd message are unacceptable, it MUST report this error by including the LSP-ERROR-CODE TLV (Section 7.3.3) with LSP error value="Unacceptable PCUpd parameters" in the LSP object in the PCRpt message to the PCE. Based

on local policy, it MAY react further to this error by revoking the delegation. If the PCC receives a PCUpd message for an LSP object identified with a PLSP-ID that does not exist on the PCC, it MUST generate a PCErr with error type 19, error value 3, "Unknown PLSP-ID" (see Section 8.4).

Once an LSP is up, the PCC sends an LSP State Report (PCRpt message) to the PCE, indicating that the LSP's status is 'Up'. If the LSP could not be set up, the PCC sends an LSP State Report indicating that the LSP is 'Down' and stating the cause of the failure. A PCC may choose to compress LSP State Reports to only reflect the most up to date state, as discussed in the previous section.

A PCC sends each LSP State Report to each stateful PCE that is connected to the PCC.

PCErr and PCRpt messages triggered as a result of a PCUpd message MUST include the SRP-ID-number from the PCUpd. This provides correlation of requests and errors and acknowledgement of state processing. The PCC may choose to compress state when processing PCUpd. In this case, receipt of a higher SRP-ID-number implicitly acknowledges processing all the earlier updates for the specific LSP.

A PCC MUST NOT send to any PCE a Path Computation Request for a delegated LSP. Should the PCC decide it wants to issue a Path Computation Request on a delegated LSP, it MUST perform Delegation Revocation procedure first.

### 5.7. LSP Protection

LSP protection and interaction with stateful PCE, as well as the extensions necessary to implement this functionality will be discussed in a separate draft.

### 5.8. Transport

A Permanent PCEP session MUST be established between a stateful PCE and the PCC. In the case of session failure, session reestablishment MUST be re-attempted per the procedures defined in [RFC5440].

## 6. PCEP Messages

As defined in [RFC5440], a PCEP message consists of a common header followed by a variable-length body made of a set of objects that can be either mandatory or optional. An object is said to be mandatory in a PCEP message when the object must be included for the message to be considered valid. For each PCEP message type, a set of rules is

defined that specify the set of objects that the message can carry. An implementation MUST form the PCEP messages using the object ordering specified in this document.

## 6.1. The PCRpt Message

A Path Computation LSP State Report message (also referred to as PCRpt message) is a PCEP message sent by a PCC to a PCE to report the current state of an LSP. A PCRpt message can carry more than one LSP State Reports. A PCC can send an LSP State Report either in response to an LSP Update Request from a PCE, or asynchronously when the state of an LSP changes. The Message-Type field of the PCEP common header for the PCRpt message is set to [TBD].

The format of the PCRpt message is as follows:

```
   <PCRpt Message> ::= <Common Header>
                       <state-report-list>
Where:

   <state-report-list> ::= <state-report>[<state-report-list>]

   <state-report> ::= <SRP>
                      <LSP>
                      <path>

Where:
     <path> is defined in [RFC5440] and extended by PCEP extensions.
```

The SRP object (see Section 7.2) is mandatory, and it MUST be included for each LSP State Report in the PCRpt message. The value of the SRP-ID-number in the SRP Object MUST be the same as that sent in the PCUpd message that triggered the state that is reported, or the reserved value 0x00000000 if the state is not as a result of processing a PCUpd message. If the PCC compressed several PCUpd messages for the same LSP by only processing the latest one, then it should use the SRP-ID-number of that request. If the SRP object is missing, the receiving PCE MUST send a PCErr message with Error-type=6 (Mandatory Object missing) and Error-value=[TBD] (SRP object missing). No state compression is allowed for state reporting. The PCC MUST explicitly report state changes (including removal) for paths it manages.

The LSP object (see Section 7.3) is mandatory, and it MUST be included in each LSP State Report on the PCRpt message. If the LSP object is missing, the receiving PCE MUST send a PCErr message with Error-type=6 (Mandatory Object missing) and Error-value=[TBD] (LSP object missing).

If the LSP transitioned to non-operational state, the PCC SHOULD include the LSP-ERROR-TLV (Section 7.3.3) with the relevant LSP Error Code to report the error to the PCE.

## 6.2. The PCUpd Message

A Path Computation LSP Update Request message (also referred to as PCUpd message) is a PCEP message sent by a PCE to a PCC to update attributes of an LSP. A PCUpd message can carry more than one LSP Update Request. The Message-Type field of the PCEP common header for the PCUpd message is set to [TBD].

The format of a PCUpd message is as follows:

```
   <PCUpd Message> ::= <Common Header>
                       <udpate-request-list>
Where:

   <update-request-list> ::= <update-request>[<update-request-list>]

   <update-request> ::= <SRP>
                        <LSP>
                        <path>
Where:
   <path>::= <ERO><attribute-list>

Where:
   <attribute-list> is defined in [RFC5440] and extended by PCEP extensions.
```

There are three mandatory objects that MUST be included within each LSP Update Request in the PCUpd message: the SRP Object (see Section 7.2), the LSP object (see Section 7.3) and the ERO object (as defined in [RFC5440]. If the SRP object is missing, the receiving PCC MUST send a PCErr message with Error-type=6 (Mandatory Object missing) and Error-value=[TBD] (SRP object missing). If the LSP object is missing, the receiving PCC MUST send a PCErr message with Error-type=6 (Mandatory Object missing) and Error-value=[TBD] (LSP object missing). If the ERO object is missing, the receiving PCC MUST send a PCErr message with Error-type=6 (Mandatory Object missing) and Error-value=[TBD] (ERO object missing).

A PCC only acts on an LSP Update Request if permitted by the local policy configured by the network manager. Each LSP Update Request that the PCC acts on results in an LSP setup operation. An LSP Update Request MUST contain all LSP parameters that a PCE wishes to set for the LSP. A PCC MAY set missing parameters from locally configured defaults. If the LSP specified in the Update Request is already up, it will be re-signaled.

The PCC SHOULD use the make-before-break procedures described in [RFC3209] in the re-signaling operation. During this process, two paths will briefly co-exist. The PCC MUST send separate PCRpt messages for each, identified by the LSP-IDENTIFIERS TLV. When the old path is torn down after the head end switches over the traffic, this event MUST be reported by sending a PCRpt message with the LSP-IDENTIFIERS-TLV or the old path, and the R bit set. Thus, a make-before-break operation will typically result in at least two PCRpt messages, one for the new path and one for the removal of the old path (more messages may be possible if intermediate states are reported).

A PCC MUST respond with an LSP State Report to each LSP Update Request it processed to indicate the resulting state of the LSP in the network (even if this processing did not result in changing the state of the LSP). The SRP-ID-number included in the PCRpt MUST match that in the PCUpd. A PCC MAY respond with multiple LSP State Reports to report LSP setup progress of a single LSP. In that case, the SRP-ID-number MUST be included while the state of the LSP is "pending", afterwards the reserved value 0x00000000 SHOULD be used..

Note that a PCC MUST process all LSP Update Requests - for example, an LSP Update Request is sent when a PCE returns delegation or puts an LSP into non-operational state. The protocol relies on TCP for message-level flow control.

If the rate of PCUpd messages sent to a PCC for the same target LSP exceeds the rate at which the PCC can signal LSPs into the network, the PCC MAY perform state compression and only signal the last modification in its queue, as the last PCUpd contains the most up-to-date desired state for the LSP. If two PCUpd arrive for the same LSP in quick succession and the PCC started the signaling of the changes relevant to the first PCUpd, then it MUST wait until the signaling finishes (and report the new state via a PCRpt) before attempting to apply the changes indicated in the second PCUpd.

Note also that it is up to the PCE to handle inter-LSP dependencies; for example, if ordering of LSP set-ups is required, the PCE has to wait for an LSP State Report for a previous LSP before starting the update of the next LSP. If the PCUpd cannot be satisfied (for example due to unsupported object or TLV), the PCC MUST respond with an PCErr message

### 6.3. The PCErr Message

If the stateful PCE capability has been negotiated on the PCEP session, the PCErr message MUST include the SRP object. If the error reported is the result of an LSP update request, then the SRP-ID-

number MUST be the one from the PCUpd that triggered the error. If it is unsolicited, the SRP-ID-number MUST be the reserved value 0x00000000.

### 6.4. The PCReq Message

A PCC MAY include the LSP object in the PCReq message (see Section 7.3) if the stateful PCE capability has been negotiated on a PCEP session between the PCC and a PCE.

The definition of the PCReq message from [RFC5440] and [I-D.ietf-pce-gmpls-pcep-extensions] is extended to optionally include the LSP object after the END-POINTS object. For illustration purposes, the encoding from [RFC5440] will become:

```
   <PCReq Message>::= <Common Header>
                      [<svec-list>]
                      <request-list>
```

Where:

```
      <svec-list>::=<SVEC>[<svec-list>]
      <request-list>::=<request>[<request-list>]

      <request>::= <RP>
                   <END-POINTS>
                   [<LSP>]              <--- New Object
                   [<LSPA>]
                   [<BANDWIDTH>]
                   [<metric-list>]
                   [<RRO>[<BANDWIDTH>]]
                   [<IRO>]
                   [<LOAD-BALANCING>]
```

### 6.5. The PCRep Message

A PCE MAY include the LSP object in the PCRep message (see (Section 7.3) if the stateful PCE capability has been negotiated on a PCEP session between the PCC and the PCE and the LSP object was included in the corresponding PCReq message from the PCC.

The definition of the PCRep message from [RFC5440] and [I-D.ietf-pce-gmpls-pcep-extensions] is extended to optionally include the LSP object after the RP object. For illustration purposes, the encoding from [RFC5440] will become:

```
      <PCRep Message> ::= <Common Header>
                          <response-list>
```

Where:

```
      <response-list>::=<response>[<response-list>]

      <response>::=<RP>
                  [<LSP>]              <--- New Object
                  [<NO-PATH>]
                  [<attribute-list>]
                  [<path-list>]
```

## 7. Object Formats

The PCEP objects defined in this document are compliant with the PCEP object format defined in [RFC5440]. The P flag and the I flag of the PCEP objects defined in this document MUST always be set to 0 on transmission and MUST be ignored on receipt since these flags are exclusively related to path computation requests.

### 7.1. OPEN Object

This document defines two new optional TLVs for use in the OPEN Object.

#### 7.1.1. Stateful PCE Capability TLV

The STATEFUL-PCE-CAPABILITY TLV is an optional TLV for use in the OPEN Object for stateful PCE capability negotiation. Its format is shown in the following figure:

```
    0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |               Type=[TBD]      |            Length=4           |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                             Flags                         |S|U|
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 14: STATEFUL-PCE-CAPABILITY TLV format

The type of the TLV is [TBD] and it has a fixed length of 4 octets.

The value comprises a single field - Flags (32 bits):

U (LSP-UPDATE-CAPABILITY - 1 bit): if set to 1 by a PCC, the U Flag indicates that the PCC allows modification of LSP parameters; if set to 1 by a PCE, the U Flag indicates that the PCE is capable of updating LSP parameters. The LSP-UPDATE-CAPABILITY Flag must be advertised by both a PCC and a PCE for PCUpd messages to be allowed on a PCEP session.

S (INCLUDE-DB-VERSION - 1 bit): if set to 1 by both PCEP Speakers, the PCC will include the LSP-DB-VERSION TLV in each LSP Object.

Unassigned bits are considered reserved. They MUST be set to 0 on transmission and MUST be ignored on receipt.

Negotiation of the stateful PCE capability implies support of LSPs that are signaled via RSVP, as well as the objects, TLVs and procedures defined in this document.

### 7.1.2. LSP State Database Version TLV

LSP-DB-VERSION is an optional TLV that MAY be included in the OPEN Object when a PCEP Speaker wishes to determine if State Synchronization can be skipped when a PCEP session is restarted. If sent from a PCE, the TLV contains the local LSP State Database version from the last valid LSP State Report received from a PCC. If sent from a PCC, the TLV contains the PCC's local LSP State Database version, which is incremented each time the LSP State Database is updated.

The format of the LSP-DB-VERSION TLV is shown in the following figure:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Type=[TBD]          |            Length=8           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      LSP State DB Version                     |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 15: LSP-DB-VERSION TLV format

The type of the TLV is [TBD] and it has a fixed length of 8 octets. The value contains a 64-bit unsigned integer.

### 7.1.3. PCE Redundancy Group Identifier TLV

PREDUNDANCY-GROUP-ID is an optional TLV that MAY be included in the OPEN Object when a PCEP Speaker wishes to determine if State Synchronization can be skipped when a PCEP session is restarted. It contains a unique identifier for the node that does not change during the life time of the PCEP Speaker. It identifies the PCEP Speaker to its peers if the Speaker's IP address changed.

The format of the PREDUNDANCY-GROUP-ID TLV is shown in the following figure:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Type=[TBD]          |       Length (variable)       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
//                 PCE redundancy group Identifier             //
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 16: PREDUNDANCY-GROUP-ID TLV format

The type of the TLV is [TBD] and it has a a variable length, which MUST be greater than 0. The value contains a node identifier that MUST be unique in the network. The node identifier MAY be configured by an operator. If the node identifier is not configured by the operator, it can be derived from a PCC's MAC address or serial number.

## 7.2. SRP Object

The SRP (Stateful PCE Request Parameters) object MUST be carried within each PCUpd and PCRpt message and MAY be carried within PCNtf and PCEerr messages. The SRP object is used to correlate between update requests sent by the PCE and the error reports and state reports sent by the PCC. The P flag in the common object header of the SRP object MUST be set to 0.

SRP Object-Class is [TBD].

SRP Object-Type is 1.

The format of the SRP object body is shown in Figure 17:

```
        0                   1                   2                   3
        0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |                          Flags                                |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |                        SRP-ID-number                          |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |                                                               |
       //                      Optional TLVs                          //
       |                                                               |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 17: The SRP Object format

The SRP object body has a variable length and may contain additional TLVs. The SYMBOLIC-PATH-NAME TLV MAY be included as one of the optional TLVs.

Flags (32 bits): None defined yet.

SRP-ID-number (32 bits): The SRP-ID-number value combined with the source IP address of the PCC and the PCE uniquely identify the operation that the PCE has requested the PCC to perform on a given LSP. The SRP-ID-number is incremented each time a new request is sent to the PCC, and may wrap around.

The values 0x00000000 and 0xFFFFFFFF are reserved.

Every request to update an LSP receives a new SRP-ID-number. This number is unique per PCEP session and is incremented each time an operation is requested from the PCE. Thus, for a given LSP there may be more than one SRP-id-number unacknowledged at a given time. The value of the SRP-ID-number is echoed back by the PCC in PCErr and PCRpt messages to allow for correlation between requests made by the PCE and errors or state reports generated by the PCC. If the error or report were not as a result of a PCE operation (for example in the case of a link down event), then the reserved value of 0x00000000 is used instead. An SRP-ID-number is considered unacknowledged and cannot be reused until a PCErr or PCRpt arrives with an SRP-ID-number equal or higher for the same LSP. A PCRpt with state "Pending" is not considered as an acknowledgement.

### 7.3. LSP Object

The LSP object MUST be present within PCRpt and PCUpd messages. The LSP object MAY be carried within PCReq and PCRep messages if the stateful PCE capability has been negotiated on the session. The LSP

object contains a set of fields used to specify the target LSP, the operation to be performed on the LSP, and LSP Delegation.  It also contains a flag indicating to a PCE that the LSP state synchronization is in progress.  This document focuses on LSPs that are signaled with RSVP, many of the TLVs used with the LSP object mirror RSVP state.

LSP Object-Class is [TBD].

LSP Object-Type is 1.

The format of the LSP object body is shown in Figure 18:

```
    0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                PLSP-ID                |     Flags |  O|A|R|S|D|
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |         Reserved                            |  LSP-sig-type   |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   //                        TLVs                                 //
   |                                                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 18: The LSP Object format

PLSP-ID (20 bits): An identifier for the LSP.  A PCC creates a unique PLSP-ID for each LSP that is constant for the life time of a PCEP session.  The mapping of the Symbolic Path Name to PLSP-ID is communicated to the PCE by sending a PCRpt message containing the SYMBOLIC-PATH-NAME TLV.  All subsequent PCEP messages then address the LSP by the PLSP-ID.  The values of 0 and 0xFFFFF are reserved. Note that the PLSP-ID is a value that is constant for the life time of the PCEP session, during which time for an RSVP-signaled LSP there might be a different RSVP identifiers (LSP-id, tunnel-id) allocated it.

Flags (12 bits):

D (Delegate - 1 bit):  on a PCRpt message, the D Flag set to 1 indicates that the PCC is delegating the LSP to the PCE.  On a PCUpd message, the D flag set to 1 indicates that the PCE is confirming the LSP Delegation.  To keep an LSP delegated to the PCE, the PCC must set the D flag to 1 on each PCRpt message for the duration of the delegation - the first PCRpt with the D flag set to 0 revokes the delegation.  To keep the delegation, the PCE must set the D flag to 1 on each PCUpd message for the duration of the delegation - the first PCUpd with the D flag set to 0 returns

the delegation.

S (SYNC - 1 bit): the S Flag MUST be set to 1 on each LSP State Report sent from a PCC during State Synchronization. The S Flag MUST be set to 0 otherwise.

R(Remove - 1 bit): On PCRpt messages the R Flag indicates that the LSP has been removed from the PCC and the PCE SHOULD remove all state from its database. Upon receiving an LSP State Report with the R Flag set to 1 for an RSVP-signaled LSP, the PCE SHOULD remove all state for the path identified by the LSP Identifiers TLV from its database. When the all-zeros LSP-IDENTIFIERS-TLV is used, the PCE SHOULD remove all state for the PLSP-ID from its database.

A(Administrative - 1 bit): On PCRpt messages, the A Flag indicates the PCC's target operational status for this LSP. On PCUpd messages, the A Flag indicates the LSP status that the PCE desires for this LSP. In both cases, a value of '1' means that the desired operational state is active, and a value of '0' means that the desired operational state is inactive. A PCC ignores the A flag on a PCUpd message unless the operator's policy allows the PCE to control the corresponding LSP's administrative state.

O(Operational - 3 bits): On PCRpt messages, the O Field represents the operational status of the LSP.

The following values are defined:

0 - DOWN: not active.

1 - UP: signalled.

2 - ACTIVE: up and carrying traffic.

3 - GOING-DOWN: LSP is being torn down, resources are being released.

4 - GOING-UP: LSP is being signalled.

5-7 - Reserved: these values MUST be set to 0 on transmission and MUST be ignored on receipt.

Unassigned bits are considered reserved. They MUST be set to 0 on transmission and MUST be ignored on receipt.

LSP-sig-type (8 bits) - identifies the method used for signaling the LSP. If a PCEP speaker receives an LSP object with LSP-sig-type that

had not been previously negotiated, a PCErr with error type 19, error value 5, "Unsupported LSP signaling type", (see Section 8.4) MUST be sent. If there is a mismatch in the LSP signaling type for a particular LSP between the PCEP speakers, a PCErr with error type 19, error value 4, "Mismatched LSP signaling type" (see Section 8.4) MUST be sent by the party identifying the mismatch.

Optional TLVs that may be included in the LSP Object are described in the following sections.

### 7.3.1. LSP Identifiers TLVs

The LSP Identifiers TLV MUST be included in the LSP object in PCRpt messages for RSVP-signaled LSPs. If the TLV is missing, the PCE will generate an error with error-type 6 (mandatory object missing) and Error Value 11 (LSP-IDENTIFIERS TLV missing) and close the session. The LSP Identifiers TLV MAY be included in the LSP object in PCUpd messages for RSVP-signaled LSPs. The special value of all zeros for this TLV is used to refer to all paths pertaining to a particular PLSP-ID. There are two LSP Identifiers TLVs, one for IPv4 and one for IPv6.

It is the responsibility of the PCC to send to the PCE the identifiers for each RSVP incarnation of the tunnel. For exmple, in a make-before-break scenario, the PCC MUST send a separate PCRpt for the old and for the reoptimized paths, and explicitly report removal of any of these paths using the R bit in the LSP object.

The format of the IPV4-LSP-IDENTIFIERS TLV is shown in the following figure:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Type=[TBD]          |           Length=12           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                   IPv4 Tunnel Sender Address                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|             LSP ID            |           Tunnel ID           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Extended Tunnel ID                     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 19: IPV4-LSP-IDENTIFIERS TLV format

The type of the TLV is [TBD] and it has a fixed length of 12 octets. The value contains the following fields:

IPv4 Tunnel Sender Address: contains the sender node's IPv4 address, as defined in [RFC3209], Section 4.6.2.1 for the LSP_TUNNEL_IPv4 Sender Template Object.

LSP ID: contains the 16-bit 'LSP ID' identifier defined in [RFC3209], Section 4.6.2.1 for the LSP_TUNNEL_IPv4 Sender Template Object.

Tunnel ID: contains the 16-bit 'Tunnel ID' identifier defined in [RFC3209], Section 4.6.1.1 for the LSP_TUNNEL_IPv4 Session Object. Tunnel ID remains constant over the life time of a tunnel.

Extended Tunnel ID: contains the 32-bit 'Extended Tunnel ID' identifier defined in [RFC3209], Section 4.6.1.1 for the LSP_TUNNEL_IPv4 Session Object.

The format of the IPV6-LSP-IDENTIFIERS TLV is shown in l following figure:

```
    0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |           Type=[TBD]          |           Length=36           |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                                                               |
   +                                                               +
   |                  IPv6 tunnel sender address                   |
   +                          (16 octets)                          +
   |                                                               |
   +                                                               +
   |                                                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |             LSP ID            |           Tunnel ID           |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                                                               |
   +                                                               +
   |                       Extended Tunnel ID                      |
   +                          (16 octets)                          +
   |                                                               |
   +                                                               +
   |                                                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 20: IPV6-LSP-IDENTIFIERS TLV format

The type of the TLV is [TBD] and it has a fixed length of 36 octets. The value contains the following fields:

IPv6 Tunnel Sender Address: contains the sender node's IPv6 address, as defined in [RFC3209], Section 4.6.2.2 for the LSP_TUNNEL_IPv6 Sender Template Object.

LSP ID: contains the 16-bit 'LSP ID' identifier defined in [RFC3209], Section 4.6.2.2 for the LSP_TUNNEL_IPv6 Sender Template Object.

Tunnel ID: contains the 16-bit 'Tunnel ID' identifier defined in [RFC3209], Section 4.6.1.2 for the LSP_TUNNEL_IPv6 Session Object. Tunnel ID remains constant over the life time of a tunnel. However, when Global Path Protection or Global Default Restoration is used, both the primary and secondary LSPs have their own Tunnel IDs. A PCC will report a change in Tunnel ID when traffic switches over from primary LSP to secondary LSP (or vice versa).

Extended Tunnel ID: contains the 128-bit 'Extended Tunnel ID' identifier defined in [RFC3209], Section 4.6.1.2 for the LSP_TUNNEL_IPv6 Session Object.

### 7.3.2. Symbolic Path Name TLV

Each LSP (path) MUST have a symbolic name that is unique in the PCC. This symbolic path name MUST remain constant throughout a path's lifetime, which may span across multiple consecutive PCEP sessions and/or PCC restarts. The symbolic path name MAY be specified by an operator in a PCC's configuration. If the operator does not specify a unique symbolic name for a path, the PCC MUST auto-generate one.

The SYMBOLIC-PATH-NAME TLV MUST be included in the LSP State Report when during a given PCEP session an LSP is first reported to a PCE. A PCC sends to a PCE the first LSP State Report either during State Synchronization, or when a new LSP is configured at the PCC. The symbolic path name MAY be included in subsequent LSP State Reports for the LSP.

The SYMBOLIC-PATH-NAME TLV MAY appear as a TLV in both the LSP Object and the LSPA Object.

The format of the SYMBOLIC-PATH-NAME TLV is shown in the following figure:

```
      0                   1                   2                   3
      0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |           Type=[TBD]          |       Length (variable)       |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |                                                               |
     //                      Symbolic Path Name                     //
     |                                                               |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 21: SYMBOLIC-PATH-NAME TLV format

The type of the TLV is [TBD] and it has a variable length, which MUST be greater than 0.

### 7.3.3. LSP Error Code TLV

The LSP Error code TLV is an optional TLV for use in the LSP object to convey error information. When an LSP Update Request fails, an LSP State Report MUST be sent to report the current state of the LSP, and SHOULD contain the LSP-ERROR-CODE TLV indicating the reason for the failure. Similarly, when a PCRpt is sent as a result of an LSP transitioning to non-operational state, the LSP-ERROR-CODE TLV SHOULD be included to indicate the reason for the transition.

The format of the LSP-ERROR-CODE TLV is shown in the following figure:

```
      0                   1                   2                   3
      0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |           Type=[TBD]          |            Length=4           |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |                          LSP Error Code                       |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 22: LSP-ERROR-CODE TLV format

The type of the TLV is [TBD] and it has a fixed length of 4 octets. The value contains an error code that indicates the cause of the failure.

The following LSP Error Codes are defined:

| Value | Meaning |
|---|---|
| 1 | Unknown reason |
| 2 | Limit reached for PCE-controlled LSPs |
| 3 | Too many pending LSP update requests |
| 4 | Unacceptable PCUpd parameters |
| 5 | Internal error |
| 6 | LSP administratively brought down |
| 7 | LSP preempted |
| 8 | RSVP signaling error |

### 7.3.4. RSVP Error Spec TLV

The RSVP-ERROR-SPEC TLV is an optional TLV for use in the LSP object to carry RSVP error information. It includes the RSVP ERROR_SPEC or USER_ERROR_SPEC Object ([RFC2205] and [RFC5284]) which were returned to the PCC from a downstream node. If the set up of an LSP fails at a downstream node which returned an ERROR_SPEC to the PCC, the PCC SHOULD include in the PCRpt for this LSP the LSP-ERROR-CODE TLV with LSP Error Code = "RSVP signaling error" and the RSVP-ERROR-SPEC TLV with the relevant RSVP ERROR-SPEC or USER_ERROR_SPEC Object.

The format of the RSVP-ERROR-SPEC TLV is shown in the following figure:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Type=[TBD]          |            Length (variable)  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
+                RSVP ERROR_SPEC or USER_ERROR_SPEC Object      +
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 23: RSVP-ERROR-SPEC TLV format

The type of the TLV is [TBD] and it has a variable length. The value contains the RSVP ERROR_SPEC or USER_ERROR_SPEC object, including the object header.

### 7.3.5. LSP State Database Version TLV

The LSP-DB-VERSION TLV can be included as an optional TLV in the LSP object. The LSP-DB-VERSION TLV is discussed in Section 5.4.1 which covers state synchronization avoidance. The format of the TLV is described in Section 7.1.2, where the details of its use in the OPEN message are listed.

If State Synchronization Avoidance has been enabled on a PCEP session (as described in Section 5.4.1) , a PCC MUST include the LSP-DB-VERSION TLV in each LSP Object sent out on the session. If the TLV is missing, the PCE will generate an error with error-type 6 (mandatory object missing) and Error Value 12 (LSP-DB-VERSION TLV missing) and close the session. If State Synchronization Avoidance has not been enabled on a PCEP session, the PCC SHOULD NOT include the LSP-DB-VERSION TLV in the LSP Object and the PCE SHOULD ignore it were it to receive one.

Since a PCE does not send LSP updates to a PCC, a PCC should never encounter this TLV. A PCC SHOULD ignore the LSP-DB-VERSION TLV, were it to receive one from a PCE.

#### 7.3.6. Delegation Parameters TLVs

Multiple delegation parameters, such as sub-delegation permissions, authentication parameters, etc. need to be communicated from a PCC to a PCE during the delegation operation. Delegation parameters will be carried in multiple delegation parameter TLVs, which will be defined in future revisions of this document.

### 7.4. Optional TLVs for the LSPA Object

TLVs that may be included in the LSPA Object are described in the following sections and in separate technology-specific documents.

#### 7.4.1. Symbolic Path Name TLV

See section Section 7.3.2.

## 8. IANA Considerations

This document requests IANA actions to allocate code points for the protocol elements defined in this document. Values shown here are suggested for use by IANA.

### 8.1. PCEP Messages

This document defines the following new PCEP messages:

| Value | Meaning | Reference |
|---|---|---|
| 10 | Report | This document |
| 11 | Update | This document |

## 8.2. PCEP Objects

This document defines the following new PCEP Object-classes and Object-values:

| Object-Class Value | Name | Reference |
|---|---|---|
| 32 | LSP<br>Object-Type<br>1 | This document |
| 33 | SRP<br>Object-Type<br>1 | This document |

## 8.3. LSP Object

This document requests that a registry is created to manage the Flags field of the LSP object. New values are to be assigned by Standards Action [RFC5226]. Each bit should be tracked with the following qualities:

- o Bit number (counting from bit 0 as the most significant bit)
- o Capability description
- o Defining RFC

The following values are defined in this document:

| Bit | Description | Reference |
|---|---|---|
| 25-27 | Operational (3 bits) | This document |
| 28 | Administrative | This document |
| 29 | Remove | This document |
| 30 | SYNC | This document |
| 31 | Delegate | This document |

## 8.4. PCEP-Error Object

This document defines new Error-Type and Error-Value for the following new error conditions:

| Error-Type | Meaning |
|---|---|
| 6 | Mandatory Object missing |
| | Error-value=8: LSP Object missing |
| | Error-value=9: ERO Object missing |
| | Error-value=10: SRP Object missing |
| | Error-value=11: LSP-IDENTIFIERS TLV missing |
| | Error-value=12: LSP-DB-VERSION TLV missing |
| 19 | Invalid Operation |
| | Error-value=1: Attempted LSP Update Request for a non-delegated LSP. The PCEP-ERROR Object is followed by the LSP Object that identifies the LSP. |
| | Error-value=2: Attempted LSP Update Request if active stateful PCE capability was not negotiated active PCE. |
| | Error-value=3: Attempted LSP Update Request for an LSP identified by an unknown PLSP-ID. |
| | Error-value=4: Mismatched LSP signaling type. |
| | Error-value=5: Unsupported LSP signaling type. |
| 20 | LSP State synchronization error. |
| | Error-value=1: A PCE indicates to a PCC that it can not process (an otherwise valid) LSP State Report. The PCEP-ERROR Object is followed by the LSP Object that identifies the LSP. |
| | Error-value=2: LSP Database version mismatch. |
| | Error-value=3: The LSP-DB-VERSION TLV Missing when State Synchronization Avoidance enabled. |

## 8.5. PCEP TLV Type Indicators

This document defines the following new PCEP TLVs:

| Value | Meaning | Reference |
|---|---|---|
| 16 | STATEFUL-PCE-CAPABILITY | This document |
| 17 | SYMBOLIC-PATH-NAME | This document |
| 18 | IPV4-LSP-IDENTIFIERS | This document |
| 19 | IPV6-LSP-IDENTIFIERS | This document |
| 20 | LSP-ERROR-CODE | This document |
| 21 | RSVP-ERROR-SPEC | This document |
| 23 | LSP-DB-VERSION | This document |
| 24 | PREDUNDANCY-GROUP-ID | This document |
| 25 | TUNNEL-ID | This document |

## 8.6. STATEFUL-PCE-CAPABILITY TLV

This document requests that a registry is created to manage the Flags field in the STATEFUL-PCE-CAPABILITY TLV in the OPEN object. New values are to be assigned by Standards Action [RFC5226]. Each bit should be tracked with the following qualities:

- Bit number (counting from bit 0 as the most significant bit)
- Capability description
- Defining RFC

The following values are defined in this document:

| Bit | Description | Reference |
|---|---|---|
| 30 | INCLUDE-DB-VERSION | This document |
| 31 | LSP-UPDATE-CAPABILITY | This document |

## 8.7. LSP-ERROR-CODE TLV

This document requests that a registry is created to manage the value of the LSP error code field in this TLV. This field specifies the reason for failure to update the LSP.

| Value | Meaning |
|---|---|
| 1 | Unknown reason |
| 2 | Limit reached for PCE-controlled LSPs |
| 3 | Too many pending LSP update requests |
| 4 | Unacceptable PCUpd parameters |
| 5 | Internal error |
| 6 | LSP administratively brought down |
| 7 | LSP preempted |
| 8 | RSVP signaling error |

## 8.8. LSP-SIG-TYPE field in the LSP object

This document requests that a registry is created to manage the value of the LSP type field in the LSP object, which defines the technology used for LSP setup.

| Value | Meaning |
|---|---|
| 0 | RSVP |

## 9. Manageability Considerations

All manageability requirements and considerations listed in [RFC5440] apply to PCEP protocol extensions defined in this document. In addition, requirements and considerations listed in this section apply.

### 9.1. Control Function and Policy

In addition to configuring specific PCEP session parameters, as specified in [RFC5440], Section 8.1, a PCE or PCC implementation MUST allow configuring the stateful PCEP capability and the LSP Update capability. A PCC implementation SHOULD allow the operator to specify multiple candidate PCEs for and a delegation preference for each candidate PCE. A PCC SHOULD allow the operator to specify an LSP delegation policy where LSPs are delegated to the most-preferred online PCE. A PCC MAY allow the operator to specify different LSP delegation policies.

A PCE or PCC implementation SHOULD allow the operator to configure a PREDUNDANCY-GROUP-ID (Section 7.1.3).

A PCC implementation which allows concurrent connections to multiple PCEs SHOULD allow the operator to group the PCEs by administrative domains and it MUST NOT advertise LSP existence and state to a PCE if the LSP is delegated to a PCE in a different group.

A PCC implementation SHOULD allow the operator to specify whether the PCC will advertise LSP existence and state for LSPs that are not controlled by any PCE (for example, LSPs that are statically configured at the PCC).

A PCC implementation SHOULD allow the operator to specify both the Redelegation Timeout Interval and the State Timeout Interval. The default value of the Redelegation Timeout Interval SHOULD be set to 30 seconds. An operator MAY also configure a policy that will dynamically adjust the Redelegation Timeout Interval, for example setting it to zero when the PCC has an established session to a backup PCE. The default value for the State Timeout Interval SHOULD be set to 60 seconds.

After the expiration of the State Timeout Interval, the LSP reverts to operator-defined default parameters. A PCC implementation MUST allow the operator to specify the default LSP parameters. To achieve a behavior where the LSP retains the parameters set by the PCE until such time that the PCC makes a change to them, a State Timeout Interval of infinity SHOULD be used. Any changes to LSP parameters SHOULD be done in make-before-break fashion.

A PCC implementation SHOULD allow the operator to specify delegation priority for PCEs. This effectively defines the primary PCE and one or more backup PCEs to which primary PCE's LSPs can be delegated when the primary PCE fails.

Policies defined for stateful PCEs and PCCs should eventually fit in the Policy-Enabled Path Computation Framework defined in [RFC5394], and the framework should be extended to support Stateful PCEs.

### 9.2. Information and Data Models

PCEP session configuration and information in the PCEP MIB module SHOULD be extended to include negotiated stateful capabilities, synchronization status, and delegation status (at the PCC list PCEs with delegated LSPs).

### 9.3. Liveness Detection and Monitoring

PCEP protocol extensions defined in this document do not require any new mechanisms beyond those already defined in [RFC5440], Section 8.3.

### 9.4. Verifying Correct Operation

Mechanisms defined in [RFC5440], Section 8.4 also apply to PCEP protocol extensions defined in this document. In addition to monitoring parameters defined in [RFC5440], a stateful PCC-side PCEP implementation SHOULD provide the following parameters:

- o Total number of LSP updates
- o Number of successful LSP updates
- o Number of dropped LSP updates
- o Number of LSP updates where LSP setup failed

A PCC implementation SHOULD provide a command to show for each LSP whether it is delegated, and if so, to which PCE.

A PCC implementation SHOULD allow the operator to manually revoke LSP delegation.

### 9.5. Requirements on Other Protocols and Functional Components

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

### 9.6. Impact on Network Operation

Mechanisms defined in [RFC5440], Section 8.6 also apply to PCEP protocol extensions defined in this document.

Additionally, a PCEP implementation SHOULD allow a limit to be placed on the rate PCUpd and PCRpt messages sent by a PCEP speaker and processed from a peer. It SHOULD also allow sending a notification when a rate threshold is reached.

A PCC implementation SHOULD allow a limit to be placed on the rate of LSP Updates to the same LSP to avoid signaling overload discussed in Section 10.3.

## 10. Security Considerations

### 10.1. Vulnerability

This document defines extensions to PCEP to enable stateful PCEs. The nature of these extensions and the delegation of path control to PCEs results in more information being available for a hypothetical adversary and a number of additional attack surfaces which must be protected.

The security provisions described in [RFC5440] remain applicable to these extensions. However, because the protocol modifications outlined in this document allow the PCE to control path computation timing and sequence, the PCE defense mechanisms described in [RFC5440] section 7.2 are also now applicable to PCC security.

As a general precaution, it is RECOMMENDED that these PCEP extensions only be activated on authenticated and encrypted sessions across PCEs and PCCs belonging to the same administrative authority.

The following sections identify specific security concerns that may result from the PCEP extensions outlined in this document along with recommended mechanisms to protect PCEP infrastructure against related attacks.

### 10.2. LSP State Snooping

The stateful nature of this extension explicitly requires LSP status updates to be sent from PCC to PCE. While this gives the PCE the ability to provide more optimal computations to the PCC, it also provides an adversary with the opportunity to eavesdrop on decisions made by network systems external to PCE. This is especially true if the PCC delegates LSPs to multiple PCEs simultaneously.

Adversaries may gain access to this information by eavesdropping on unsecured PCEP sessions, and might then use this information in various ways to target or optimize attacks on network infrastructure. For example by flexibly countering anti-DDoS measures being taken to protect the network, or by determining choke points in the network where the greatest harm might be caused.

PCC implementations which allow concurrent connections to multiple PCEs SHOULD allow the operator to group the PCEs by administrative domains and they MUST NOT advertise LSP existence and state to a PCE if the LSP is delegated to a PCE in a different group.

### 10.3. Malicious PCE

The LSP delegation mechanism described in this document allows a PCC to grant effective control of an LSP to the PCE for the duration of a PCEP session. While this enables PCE control of the timing and sequence of path computations within and across PCEP sessions, it also introduces a new attack vector: an attacker may flood the PCC with PCUpd messages at a rate which exceeds either the PCC's ability to process them or the network's ability to signal the changes, either by spoofing messages or by compromising the PCE itself.

A PCC is free to revoke an LSP delegation at any time without needing any justification. A defending PCC can do this by enqueueing the appropriate PCRpt message. As soon as that message is enqueued in the session, the PCC is free to drop any incoming PCUpd messages without additional processing.

### 10.4. Malicious PCC

A stateful session also result in increased attack surface by placing a requirement for the PCE to keep an LSP state replica for each PCC. It is RECOMMENDED that PCE implementations provide a limit on resources a single PCC can occupy.

Delegation of LSPs can create further strain on PCE resources and a PCE implementation MAY preemptively give back delegations if it finds itself lacking the resources needed to effectively manage the delegation. Since the delegation state is ultimately controlled by the PCC, PCE implementations SHOULD provide throttling mechanisms to prevent strain created by flaps of either a PCEP session or an LSP delegation.

## 11. Acknowledgements

We would like to thank Adrian Farrel, Cyril Margaria and Ramon

Casellas for their contributions to this document.

We would like to thank Shane Amante, Julien Meuric, Kohei Shiomoto, Paul Schultz and Raveendra Torvi for their comments and suggestions. Thanks also to Cyril Margaria, Jon Hardwick, Dhruv Dhoddy, Oscar Gonzales de Dios, Tomas Janciga, Stefan Kobza, Kexin Tang, Matej Spanik, Jon Parker, Marek Zavodsky, Ambrose Kwong, Ashwin Sampath, Calvin Ying and Xian Zhang for helpful comments and discussions.

## 12. References

### 12.1. Normative References

[I-D.ietf-pce-gmpls-pcep-extensions]
Margaria, C., Dios, O., and F. Zhang, "PCEP extensions for GMPLS", draft-ietf-pce-gmpls-pcep-extensions-07 (work in progress), October 2012.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC2205] Braden, B., Zhang, L., Berson, S., Herzog, S., and S. Jamin, "Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification", RFC 2205, September 1997.

[RFC3209] Awduche, D., Berger, L., Gan, D., Li, T., Srinivasan, V., and G. Swallow, "RSVP-TE: Extensions to RSVP for LSP Tunnels", RFC 3209, December 2001.

[RFC3473] Berger, L., "Generalized Multi-Protocol Label Switching (GMPLS) Signaling Resource ReserVation Protocol-Traffic Engineering (RSVP-TE) Extensions", RFC 3473, January 2003.

[RFC4090] Pan, P., Swallow, G., and A. Atlas, "Fast Reroute Extensions to RSVP-TE for LSP Tunnels", RFC 4090, May 2005.

[RFC5088] Le Roux, JL., Vasseur, JP., Ikejiri, Y., and R. Zhang, "OSPF Protocol Extensions for Path Computation Element (PCE) Discovery", RFC 5088, January 2008.

[RFC5089] Le Roux, JL., Vasseur, JP., Ikejiri, Y., and R. Zhang, "IS-IS Protocol Extensions for Path Computation Element (PCE) Discovery", RFC 5089, January 2008.

[RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226,

May 2008.

[RFC5284] Swallow, G. and A. Farrel, "User-Defined Errors for RSVP", RFC 5284, August 2008.

[RFC5440] Vasseur, JP. and JL. Le Roux, "Path Computation Element (PCE) Communication Protocol (PCEP)", RFC 5440, March 2009.

[RFC5511] Farrel, A., "Routing Backus-Naur Form (RBNF): A Syntax Used to Form Encoding Rules in Various Routing Protocol Specifications", RFC 5511, April 2009.

## 12.2. Informative References

[I-D.sivabalan-pce-disco-stateful]
Sivabalan, S., Medved, J., and X. Zhang, "IGP Extensions for Stateful PCE Discovery", draft-sivabalan-pce-disco-stateful-01 (work in progress), April 2013.

[I-D.zhang-pce-stateful-pce-app]
Zhang, X. and I. Minei, "Applicability of Stateful Path Computation Element (PCE)", draft-zhang-pce-stateful-pce-app-04 (work in progress), May 2013.

[MPLS-PC] Chaieb, I., Le Roux, JL., and B. Cousin, "Improved MPLS-TE LSP Path Computation using Preemption", Global Information Infrastructure Symposium, July 2007.

[MXMN-TE] Danna, E., Mandal, S., and A. Singh, "Practical linear programming algorithm for balancing the max-min fairness and throughput objectives in traffic engineering", pre-print, 2011.

[NET-REC] Vasseur, JP., Pickavet, M., and P. Demeester, "Network Recovery: Protection and Restoration of Optical, SONET-SDH, IP, and MPLS", The Morgan Kaufmann Series in Networking, June 2004.

[RFC2702] Awduche, D., Malcolm, J., Agogbua, J., O'Dell, M., and J. McManus, "Requirements for Traffic Engineering Over MPLS", RFC 2702, September 1999.

[RFC3031] Rosen, E., Viswanathan, A., and R. Callon, "Multiprotocol Label Switching Architecture", RFC 3031, January 2001.

[RFC3346] Boyle, J., Gill, V., Hannan, A., Cooper, D., Awduche, D., Christian, B., and W. Lai, "Applicability Statement for Traffic Engineering with MPLS", RFC 3346, August 2002.

[RFC3630] Katz, D., Kompella, K., and D. Yeung, "Traffic Engineering (TE) Extensions to OSPF Version 2", RFC 3630, September 2003.

[RFC4655] Farrel, A., Vasseur, J., and J. Ash, "A Path Computation Element (PCE)-Based Architecture", RFC 4655, August 2006.

[RFC4657] Ash, J. and J. Le Roux, "Path Computation Element (PCE) Communication Protocol Generic Requirements", RFC 4657, September 2006.

[RFC5305] Li, T. and H. Smit, "IS-IS Extensions for Traffic Engineering", RFC 5305, October 2008.

[RFC5394] Bryskin, I., Papadimitriou, D., Berger, L., and J. Ash, "Policy-Enabled Path Computation Framework", RFC 5394, December 2008.

[RFC5557] Lee, Y., Le Roux, JL., King, D., and E. Oki, "Path Computation Element Communication Protocol (PCEP) Requirements and Protocol Extensions in Support of Global Concurrent Optimization", RFC 5557, July 2009.

Authors' Addresses

Edward Crabbe
Google, Inc.
1600 Amphitheatre Parkway
Mountain View, CA 94043
US

Email: edc@google.com

Jan Medved
Cisco Systems, Inc.
170 West Tasman Dr.
San Jose, CA 95134
US

Email: jmedved@cisco.com

Ina Minei
Juniper Networks, Inc.
1194 N. Mathilda Ave.
Sunnyvale, CA  94089
US

Email: ina@juniper.net

Robert Varga
Pantheon Technologies SRO
Mlynske Nivy 56
Bratislava  821 05
Slovakia

Email: robert.varga@pantheon.sk