Network Working Group
IETF Internet Draft

JP Vasseur (Editor)
Cisco System Inc.
JL Le Roux
France Telecom
Arthi Ayyangar
Juniper Networks
Eiji Oki
Yuichi Ikejiri
NTT
Alia Atlas
Google, Inc
Andrew Dolganow
Alcatel

Proposed Status: Standard
Expires: March 2006
September 2005

Path Computation Element (PCE) communication Protocol (PCEP)
- Version 1 -

draft-vasseur-pce-pcep-02.txt

Status of this Memo

Abstract

This document specifies the Path Computation Element communication Protocol (PCEP) for communications between a Path Computation Client (PCC) and a Path Computation Element (PCE), or between two PCEs. Such interactions include path computation requests and path computation replies as well as notifications of specific states related to the use of a PCE in the context of MPLS and GMPLS Traffic Engineering. The PCEP protocol is designed to be flexible and extensible so as to easily allow for the addition of further messages and objects, should further requirements be expressed in the future.

Conventions used in this document

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 RFC-2119.

Table of Contents

1. Terminology

Terminology used in this document

IGP Area: OSPF Area or IS-IS level

Inter-domain TE LSP: A TE LSP whose path transits across at least two different domains where a domain can either be an IGP area, an Autonomous System or a sub-AS (BGP confederations).

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

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

PCEP Peer: an element involved in a PCEP session (i.e. a PCC or the

PCE).

PLR: Point of Local Repair. The head-end LSR of a backup tunnel or a Detour LSP.

TED: Traffic Engineering Database which contains the topology and resource information of the domain. The TED may be fed by IGP extensions or potentially by other means.

Explicit path: full explicit path from start to destination made of a list of strict hops where a hop may be an abstract node such as an AS.

Strict/loose path: mix of strict and loose hops comprising of at least one loose hop representing the destination where a hop may be an abstract node such as an AS.

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

2. Introduction

[PCE-ARCH] describes the motivations and architecture for a PCE-based model to perform path computation for MPLS and GMPLS TE LSPs. The model allows the separation of PCE from PCC, and allows cooperation between PCEs. This necessitates a communication protocol between PCC and PCE, and between PCEs.

[PCE-COM-GEN-REQ] states the generic requirements for such a protocol including a requirement that the same protocol must be used between PCC and PCE, and between PCEs. Additional application-specific requirements (for scenarios such as inter-area, inter-AS, etc.) are not included in [PCE-COM-GEN-REQ], but there is a requirement that any solution protocol must be easily extensible to handle other requirements as they are introduced in application-specific requirements documents.

This document specifies the Path Computation Element communication Protocol (PCEP) for communications between Path Computation Client (PCC) and a Path Computation Element (PCE),or between two PCEs. Such interactions include path computation requests and path computation replies as well as notifications of specific states related to the use of a PCE in the context of MPLS and GMPLS Traffic Engineering. The PCEP protocol is designed to be flexible and extensible so as to easily allow for the addition of further messages and objects, should further requirements be expressed in the future.

The compliance of PCEP to the set of requirements stated in [PCE-COM-GEN-REQ] is covered in Appendix A.

3. Assumptions

[PCE-ARCH] describes various types of PCE: it is important to note that no assumption is made on the nature of the PCE in this document.

Moreover, it is assumed that the PCE gets the required information so as to perform TE LSP path computation which usually requires network topology and resource information that can be gathered by routing protocols or by some other means. The retrieval of such information is out of the scope of this document.

Similarly, no assumption is made on the discovery method used by a PCC to discover a set of PCEs (e.g. via static configuration or dynamic discovery) and select a PCE to send its path computation request(s) to. For the sake of reference [PCE-DISC-REQ] defines a list of requirements for dynamic PCE discovery.

4. Transport protocol

PCEP operates over TCP using the well-known TCP port (TBD by IANA). This allows the requirements of reliable messaging and flow control to be met without further protocol work.

An implementation may decide to keep the TCP session alive for an unlimited time (this may for instance be the case should an implementation have to send new requests frequently in which case the TCP session will already be in place). Another motivation for leaving the TCP connection open would be to avoid TCP connection establishment time. This mode is also referred to as the "Permanent mode". Conversely, in some other circumstances, it may be desirable to systematically open and close the TCP connection for each PCEP request (this may for instance be the case if sending of PCEP path computation request is a rare event). This mode is referred to as the "Per-request mode".

Since there are circumstances where the TCP connection state is used to detect the PCC/PCE liveness (e.g case of a stateful PCE detecting a PCC failure thanks to the TCP state), the desired mode MUST be known by both the PCC and the PCE and is determined during the initialization phase.

5. Architectural Protocol Overview (Model)

The aim of this section is to describe the PCEP protocol model in the spirit of [WP]. An architecture protocol overview (the big picture of the protocol) is provided in this section where details of the protocol can be found in further sections.

5.1. Problem

The PCE-based architecture used for the computation of MPLS and GMPLS TE LSP paths is described in [PCE-ARCH]. When the PCC and the PCE are not collocated, a communication protocol between the PCC and the PCE is required. PCEP is such a protocol designed specifically for communications between a PCC and a PCE or between two PCEs: a PCC may use PCEP to send a path computation request for one or more TE LSP(s) to a PCE and such a PCE may reply with a set of computed path(s) if one or more path(s) obeying the set of constraints can be found.

5.2. Architectural Protocol Overview

PCEP operates over TCP, which allows the requirements of reliable messaging and flow control to be met without further protocol work.

Several PCEP messages are defined:
- Open and Keepalive messages are used to initiate and maintain a PCEP session respectively.
- PCReq: a message sent by a PCC to a PCE to request a path computation.
- PCRep: a message sent by a PCE to a PCC in reply to a path computation request. A PCRep message can either contain a set of computed path(s) if the request could be satisfied or a negative reply otherwise.
- PCNtf: a notification message either sent by a PCC to a PCE or a PCE to a PCC to notify of specific event.
- PCErr: a message related to a protocol error condition.

The set of available PCE(s) may be either statically configured on a PCC or dynamically discovered (the mechanism for that discovery is out of the scope of this document). A PCC may have PCEP sessions with more than one PCE and similarly a PCE may have PCEP sessions with multiple PCCs. A PCEP session establishment can either be triggered by the PCC or the PCE.

5.2.1 Initialization phase

The initialization phase consists of two successive steps:

1) Establishment of a TCP connection (3-way handshake) between the PCC and the PCE.

2) Establishment of a PCEP session over the TCP connection
Once the TCP connection is established, the PCC and the PCE (also referred to as "PCEP peers") initiate a PCEP session establishment during which various session parameters are advertised. Those parameters are carried within Open messages and include the keepalive timer, the PCEP session mode (per-request or permanent), potential detailed capabilities and policy rules that specify the conditions under which path computation requests may be sent to the PCE. If the PCEP session establishment phase fails because the PCEP peers disagree on the exchanged parameters or one of the peers does not answer, the transport connection is immediately closed. Successive retries are permitted but an implementation SHOULD make use of

exponential back-off. Keepalive messages are used to acknowledge Open messages and once the PCEP session is established Keepalive messages are exchanged between PCEP peers to ensure the liveness of the PCEP session.

```
                 +-+-+                 +-+-+
                 |PCC|                 |PCE|
                 +-+-+                 +-+-+
                   |                     |
                   |---- Open message --->|
                    |                     |
                   |<--- Open message ----|
                   |                     |
                   |                     |
                   |                     |
                   |<--- Keepalive -------|
                   |                     |
                   |---- Keepalive ------>|
```

Figure 1: PCEP Initialization phase (triggered by a PCC)

5.2.2. Path computation request sent by a PCC to a PCE

Consider the diagram depicted in figure 2.

```
                 +-+-+                 +-+-+
                 |PCC|                 |PCE|
                 +-+-+                 +-+-+
1)Path computation |                     |
event              |                     |
2)PCE Selection    |                     |
3)Path computation |---- PCReq message--->|
request sent to    |                     |
the selected PCE   |                     |
```

Figure 2: Path computation request

Once a PCC (or a PCE) has successfully established a PCEP session with one or more PCEs, if an event is triggered that requires the computation of a path, the PCC first selects the PCE it desires to send a path computation request to (note that the PCE selection may be performed prior to the PCEP session establishment). Once a PCC has selected a PCE, it sends a path computation request to the PCE (PCReq message) that contains a variety of objects that specify the set of constraints and attributes for the path to be computed. For example "Compute a TE LSP path with source IP address=x.y.z.t, destination IP address=x.y.z.t, bandwidth=X Mbit/s, Priority=Y, ...". Additionally, the PCC may desire to specify the urgency of such request by assigning a request priority. It is worth pointing out that each request is uniquely identified by a request-id number and the PCC-PCE addresses pair. The process is shown in a schematic form in figure 2.

5.2.3. Path computation reply sent by the PCE to a PCC

```
                   +-+-+                 +-+-+
                   |PCC|                 |PCE|
                   +-+-+                 +-+-+

                     |                     |
                     |---- PCReq message--->|
                     |                     |1) Path computation
                     |                     |request received
                     |                     |
                     |                     |2)Path successfully
                     |                     |computed
                     |                     |
                     |                     |3) Computed path(s) sent
                     |                     |to the PCC
                     |<--- PCRep message ---|
                     |    (Positive reply)  |
```

      Figure 3a: Path computation request with successful computation

```
                   +-+-+                 +-+-+
                   |PCC|                 |PCE|
                   +-+-+                 +-+-+
                     |                     |
                     |                     |
                     |---- PCReq message--->|
                     |                     |1) Path computation
                     |                     |request received
                     |                     |
                     |                     |2) No Path found that
                     |                     |satisfies the request
                     |                     |
                     |                     |3) Negative reply sent to
                     |                     |the PCC (optionally with
                     |                     |various additional
                     |                     |information)
                     |<--- PCRep message ---|
                     |   (Negative reply)   |
```

     Figure 3b: Path computation request with unsuccessfull computation

Upon receiving a path computation request from a PCC, the PCE triggers a path computation, the result of which can either be:
- Positive: the PCE manages to compute a path satisfying the set of required constraints and returns the set of computed path(s) (note that the PCEP protocol supports the capability to send a single request which refers to the computation of multiple paths: for

example, compute two link diverse paths). This is illustrated in figure 3a.

- Negative: no path could be computed that satisfies the request. In this case, a PCE may provide the set of constraints that led to path computation failure. Upon receiving a negative reply, a PCC may decide to resend a modified request or take any other appropriate action. This is illustrated in figure 3b.

5.2.4 Notifications

There are several circumstances whereby a PCE may want to notify a PCC of a specific event. For example, suppose that the PCE suddenly experiences some congestion that would lead to unacceptable response times. The PCE may want to notify one or more PCCs that some of their requests (listed in the notification) will not be satisfied, potentially resulting in path computation redirections on the PCC towards another PCE, if an alternate PCE is available. Similarly, a PCC may desire to notify a PCE of particular event such as the cancellation of pending request(s).

```
                 +-+-+                  +-+-+
                 |PCC|                  |PCE|
                 +-+-+                  +-+-+
1)Path computation |                      |
event              |                      |
2)PCE Selection    |                      |
3)Path computation |---- PCReq message--->|
request X sent to  |                      |4) Path computation
the selected PCE   |                      |triggered
                   |                      |
                   |                      |
5) Path computation|                      |
request X cancelled|                      |
                   |---- PCNtf message -->|
                   |                      |6) Path computation
                   |                      |request X cancelled
```

Figure 4: Example of PCC notification (request cancellation) sent to a PCE

```
                 +-+-+                  +-+-+
                 |PCC|                  |PCE|
                 +-+-+                  +-+-+
1)Path computation |                      |
event              |                      |
2)PCE Selection    |                      |
3)Path computation |---- PCReq message--->|
request X sent to  |                      |4) Path computation
the selected PCE   |                      |triggered
                   |                      |
```

```
                    |                      |
                    |                      |5) PCE experiencing
                    |                      |congestion
```

```
                     |                      |
                     |                      |6) Path computation
                     |                      |request X cancelled
                     |                      |
                     |<--- PCNtf message----|
```

Figure 5: Example of PCEP notification (request(s) cancellation) send to a PCC

5.2.5. Termination of the PCEP Session

When one of the PCEP peers desires to terminate a PCEP session it MUST close the TCP connection. If the PCEP session is terminated by the PCE, the PCC MUST clear all the states related to pending requests sent to the PCE. Similarly, if the PCC terminates a PCEP session the PCE MUST clear all pending path computation requests sent by the PCC in question as well as the related states.

In case of TCP connection failure, the PCEP session SHOULD be maintained for a period of time equal to the Deadtimer.

6. PCEP messages

A PCEP message consists of a common header followed by a variable length body made of a set of objects that can either be mandatory or optional. In the context of this document, an object is said to be mandatory in a PCEP message when the object must be included in such message for the message to be valid. Conversely, an object is said to be optional the object may or may not be present. As specified in section 7.1, a specific flag is also defined in each object that can be set by a PCEP peer to enforce a PCE to take into account the related information during the path computation. For example, the DELAY object allows a PCC to specify in a path computation request a bounded acceptable delay for the computed path. The DELAY object is optional (does not have to be present in each path computation request message) but a PCC may set a flag to ensure that the delay constraint is being taken into account when present in a message.

For each PCEP message type a set of rules is defined which specifies the set of possible objects that the message can carry. We use the Backus-Naur Form (BNF) to specify such rules. Square brackets refer to optional sub-sequences. An implementation MUST form the PCEP messages using the order specified in this document.

If a mandatory object is missing in a received PCEP message the recipient of the PCEP message MUST trigger a protocol error condition.

6.1. Common header

```
     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      |                 Message-Length                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Message-Type  |                     Reserved                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 6 - PCEP message common header

Ver (Version): 3 bits

PCEP protocol version number. The current version is version 1

Flags: 8 bits

No Flags are currently defined

Message Length: 24 bits

Total length of the PCEP message expressed in bytes including the common header.

Message-Type: 8 bits

The following message types are currently defined.

Value	Meaning
1	Open
2	Keepalive
3	Path Computation Request
4	Path Computation Reply
5	Notification
6	Error

6.2. Open message

Once the TCP connection has been successfully established, the first message sent by the PCC to the PCE or by the PCE to the PCC MUST be an Open message. The aim of the Open message is to establish a PCEP session between the PCEP peers. During that phase the PCEP peers exchange several session characteristics. If both parties agree on such characteristics the PCEP session is successfully established.

The Message-Type field of the PCEP common header for the Open message is set to 1.

```
<Open Message>::= <Common Header>
                  <OPEN>
```

The Open message MUST only contain a single OPEN object defined in section 7. The various session characteristics specified within the OPEN object are the keepalive frequency, session mode (permanent or per-request) and potentially some optional parameters such as the detailed PCE capabilities and policy rules that specify the conditions under which path computation requests may be sent to the PCE. Details related to PCE capabilities discovery by means of PCEP are out of the scope of this document.

Keepalive: PCEP has its own keepalive mechanism used to ensure of the liveness of the PCEP session. This requires the determination of the frequency at which each PCEP peer sends the keepalive messages. Asymmetric values may be chosen; thus there is no constraints mandating the use of identical keepalive frequencies by both PCEP peers. The Deadtimer is defined as the period of time after the expiration of which a PCEP peer declares the session down if no PCEP message has been received (keepalive or any other PCEP message: thus, any PCEP message acts as a keepalive message). The minimum Keepalive value is 1 second and the Deadtimer value is equal to 4 times the Keepalive value.

Session mode: PCEP supports two session modes referred to as the "permanent" and "per-request" modes. In the permanent mode, the PCEP peers maintained a permanent PCEP session (and thus the TCP session is also maintained) regardless of the rate at which PCEP messages are exchanged. Such mode would typically be used to speed-up response times. In the permanent mode, a loss of TCP session MUST be interpreted as a communication failure. Conversely, in the "per-request" mode, a PCEP session is established on-demand, when one or more path computation requests are required and then closed by the PCC once those path computation requests are satisfied. Both PCEP peers MUST agree on the session mode; in case of disagreement, the PCEP session establishment fails.

Elements of procedure:
- Once an Open message has been sent to a PCEP peer, the sender MUST start an initialization timer called INIT-OPEN after the expiration of which a similar Open message MUST be resent if no reply has been received from the PCEP peer. The INIT-OPEN timer has a fixed value of one minute. The maximum number of Open messages that can be sent without any response from the PCEP peer is equal to 3.
- Upon the receipt of an Open message, the receiving PCEP peer MUST determine whether the suggested PCEP session characteristics are acceptable. If one or more characteristic(s) is not acceptable by the receiving peer, it MUST send a PCErr message with Error-type=8, Error-value=1. The PCErr message MUST also comprise an Open object: for each unacceptable session parameter, an acceptable parameter value MUST be proposed in the appropriate field of the Open object in place of the originally proposed value. The PCEP peer may decide to resend an Open message with different session characteristics. Consecutive retries SHOULD make use of exponential back-off so as to avoid undesirable burden of session initialization. If a second Open

message is received with the same set of parameters or with parameters differing from the proposed values, the receiving peer

MUST send a PCErr message with Error-Type=8, Error-value=2 and it MUST immediately close the TCP connection.

If the PCEP session characteristics are acceptable, the receiving PCEP peer MUST immediately send a Keepalive message as an acknowledgment.

The PCEP session is considered as operational once both PCEP peers have received a Keepalive message from their peer.

6.3. Keepalive message

Keepalive messages are used either to acknowledge an Open message if the receiving PCEP peer agrees on the session characteristics and to ensure the liveness of the PCEP session. Keepalive messages are sent at the frequency specified in the OPEN object carried within an Open message.

The Message-Type field of the PCEP common header for the Open message is set to 2.

```
<Keepalive Message>::= <Common Header>
```

6.4. Path Computation Request (PCReq) message

A Path Computation Request message (also referred to as a PCReq message) is sent by a PCC to a PCE so as to request a path computation. The Message-Type field of the PCEP common header is set to 3.

There are two mandatory objects that MUST be included within a PCReq message: the RP and the END-POINTS objects (see section 7). If one of these objects is missing, the receiving PCE MUST send an error message to the requester (PCErr message). Other objects are optional.

The format of a PCReq message is as follows:

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

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

   <request>::= <RP>
                <END-POINTS>
                [<LSPA>]
                [<BANDWIDTH>]
                [<DELAY>]
```

[<RRO>]
[<XRO>]

```
                  [<IRO>]
```

The SVEC, RP, END-POINTS, LSPA, BANDWIDTH, DELAY, ERO, XRO and IRO objects are defined in section 7.

6.5. Path Computation Reply (PCRep) message

The PCEP Path Computation reply message (also referred to as a PCRep message) is sent by a PCE to a requesting PCC in response to a previously received PCReq message. The Message-Type field of the PCEP common header is set to 4.

The PCRep message MUST comprise a RP object with a Request-ID-number identical to the one specified in the RP object carried in the corresponding PCReq message (see section 7 for the definition of the RP object).

A PCRep may comprise multiple computed path(s) corresponding to multiple path computation requests originated by a common requesting PCC. The bundling of multiple responses within a single PCRep message is supported by the PCEP protocol. If a PCE receives non-synchronized path computation requests by means of one or more PCReq messages from a requesting PCC it may decide to bundle the computed paths within a single PCRep message so as to reduce the control plane load. Note that the counter side of such an approach is the introduction of additional delays for some path computation requests of the set.

If the path computation request can be successfully satisfied (the PCE manages to compute a set of path(s) that obey the requested constraint(s)), the set of computed path(s) specified by means of ERO object(s) is inserted in the PCRep message. Such a situation where multiple computed paths are provided in a PCRep message is discussed in detail in section 8.

If the path computation request cannot be satisfied, the PCRep message MUST include a NO-PATH object. The NO-PATH object (further described in section 7) may also comprise other information (e.g reasons for the path computation failure).

The format of a PCRep message is as follows:

```
<PCRep Message> ::= <Common Header>
                    [<svec-list>]
                    <path-list>

where:
   <svec-list>::=<SVEC>[<svec-list>]
   <path-list>::=<path>[<path-list>]
```

```
<path>::=<RP>
         [<NO-PATH>]
```

```
            [<ero-list>]
            [<LSPA>]
            [<BANDWIDTH>]
            [<DELAY>]
            [<XRO>]
            [<IRO>]
```

where:

```
   <ero-list>:==<ERO>[<ero-list]>
```

6.6. Notification (PCNtf) message

The PCEP Notification message (also referred to as the PCNtf message) can either be sent by a PCE to a PCC or by a PCC to a PCE so as to notify of a specific event. The Message-Type field of the PCEP common header is set to 5.

The PCNtf message MUST carry at least one NOTIFICATION object and may comprise several NOTIFICATION objects should the PCE or the PCC intend to notify of multiple events. The NOTIFICATION object is defined in section 7. The PCNtf message may also comprise an RP object when the notification refers to a particular path computation request.

The PCNtf message may be sent by a PCC or a PCE in response to a request or in an unsolicited manner.

The format of a PCNtf message is as follows:

```
<PCNtf Message>::=<Common Header>
                  <notify-list>

<notify-list>::=<notify> [<notify-list>]

<notify>::= [<request-id-list>]
                  <notification-list>

<request-id-list>:==<RP><request-id-list>
<notification-list>:=<NOTIFICATION><notification-list>
```

The procedure upon the reception of a PCNtf message is defined in section 9.

6.7. Error (PCErr) message

The PCEP Error message (also referred to as a PCErr message) is sent when a protocol error condition is met. The Message-Type field of the PCEP common header is set to 6.

The PCErr message may be sent by a PCC or a PCE in response to a request or in an unsolicited manner. In the former case, the PCErr message MUST include the set of RP objects related to the pending path computation request(s) which triggered the protocol error condition. In the later case (unsolicited), no RP object is inserted within the PCErr message. No RP object is inserted in a PCErr when the error condition occurred during the initialization phase. A PCErr message MUST comprise a PCEP-ERROR object specifying the PCEP error condition. The PCEP-ERROR object is defined in section 7.

The format of a PCErr message is as follows:

```
<PCErr Message> ::= <Common Header>
                    <error-list>
                    [<Open>]

<error-list>:==<error>[<error-list>]
<error>::=[<request-id-list>]
          <error-obj-list>

<request-id-list>:==<RP>[<request-id-list>]
<error-obj-list>:==<PCEP-ERROR>[<error-obj-list>]
```

The procedure upon the reception of a PCErr message is defined in section 9.

7. Object Formats

7.1. Common object header

A PCEP object carried within a PCEP message consists of one or more 32-bit words with a common header which has the following format:

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Object-Class  |   OT  |Res|I|P|   Object Length (bytes)       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
//                        (Object body)                        //
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 8 - PCEP common object header

Object-Class (to be managed by IANA)

8-bit field that identifies the PCEP object class

OT (Object-Type) (to be managed by IANA)

4-bit field that identifies the PCEP object type

P flag (Processing-Rule)

1-bit flag which specifies whether the object must be taken into account by the receiving PCEP peer or is just optional. When the P flag is cleared, the object MUST be taken into account by the receiving entity. If the PCC or the PCE does not understand the object or understands the object but decides to ignore the object, this MUST trigger a protocol error condition as defined in section 7. Conversely, when the P flag is set the object is optional and can be silently ignored.

I flag

1-bit flag: the PCE set the I flag when the object is carried within a PCRep message so as to indicate when the constraint was optional and was ignored during path computation.

Res flags: 2-bit flag reserved (MUST be set to 0)

Object Length

16-bit field containing the total object length in bytes. The Object Length field MUST always be a multiple of 4, and at least 4.

The maximum object content length is 65528 bytes. The Object-Class and Object-Type fields uniquely identify each PCEP object.

The P bit is used to determine what action a node should take if it does not recognize the Object-Class or Object-Type of a PCEP object or decides not to take into account the object: there are two possible ways a PCEP implementation can react. This choice is determined by the P bit, as follows.

If P flag=0

The entire PCEP message MUST be rejected and the receiving PCEP peer MUST send a PCErr message with a PCEP-ERROR Object ("Unkown Object" or "Not supported Object").

If P flag=1

The node MAY ignore the object and process the PCEP message if possible. In that case (the message can be processed by ignoring the object in question), the PCE SHOULD include the object in the corresponding PCERep message. The I flag of the common header for this object MUST be set. If the path computation cannot be performed, a PCErr message MUST be sent to the requesting entity with a PCEP-ERROR object (Error-type=2, "Unknown Object").

7.2. OPEN Object

The OPEN object MUST be present in each Open message. There MUST be only one OPEN object per Open message.

The OPEN object contains a set of fields used to specify the PCEP protocol version, Keepalive frequency, PCEP session ID along with various flags. The OPEN object may also contain a set of TLVs used to convey various session characteristics such as the detailed PCE capability, policy rules and so on. No TLV is currently defined.

OPEN Object-Class is to be assigned by IANA (recommended value=1)
OPEN Object-Type is to be assigned by IANA (recommended value=1)

The format of the OPEN object body is as follows:

```
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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Ver |            Keepalive           |             SID          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|              Reserved          |                             |R|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
//                         Optional TLV(s)                     //
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 9 - OPEN Object format

Version (Ver): 3 bits - Current version is 1.

Keepalive frequency (Keepalive): 16 bits.

Specifies the frequency in seconds at which the sender of the Open message will send Keepalive messages. The minimum value for the Keepalive is 1 second. When set to 0, no keepalive is sent to the remote peer. A RECOMMENDED value for the keepalive frequency is 30 seconds.

PCEP session-ID (SID): 13 bits.
Specifies a 2 octet unsigned PCEP session number that identifies the current session. The SID MUST be incremented each time a new PCEP session is established.

Flags

One flag is currently defined.

R flag: when cleared, this indicates that the sending PCEP peer

requires the establishment of a PCEP session in permanent mode.
When set, a per-request mode is requested.

Optional TLVs may be included within the Open message body to specify PCC or PCE characteristics.

7.3. RP Object

The RP (Request Parameters) object MUST be carried within every PCReq and PCRep messages and MAY be carried within PCNtf and PCErr messages.

RP Object-Class is to be assigned by IANA (recommended value=2)
RP Object-Type is to be assigned by IANA (recommended value=1)

The format of the RP object body is as follows:

```
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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|       Reserved    |              Flags          |O|C|B|R| Pri |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Request-ID-number                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
//                                                              //
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 10 - RP object body format

The RP object has a variable length and may contain additional TLVs. No TLV is currently defined.

Flags: 18 bits - The following flags are currently defined:

Pri (Priority) field (3 bits)

This field may be used by the requesting PCC to specify to the PCE the request's priority. The decision of which priority should be used for a specific request is of a local matter and MUST be set to 0 when unused. Furthermore, the use of the path computation request priority by the PCE's requests scheduler is implementation specific and out of the scope of this document. Note that it is not required for a PCE to support the priority field: in that case, the priority field SHOULD be set to 0 by the PCC in the RP object. If the PCE does not take into account the request priority, it is RECOMMENDED to set the priority field to 0 in the RP object carried within the corresponding PCRep message, regardless of the priority value contained in the RP object carried within the corresponding PCReq message. A higher numerical value of the priority field reflects a higher priority. Note that it is the responsibility of the network administrator to make use of the priority values in a consistent manner across the various PCC(s). The ability of a PCE to support requests prioritization may be dynamically discovered by the PCC(s) by means of PCE capability discovery. If not advertised by the PCE, a PCC may decide to set the request priority and will learn the ability of the PCE the support request prioritization by observing the Priority field of the RP object received in the PCRep message. If the value of the Pri field is set to 0, this means that the PCE does not support the Pri field: in other words, the path computation request has been honoured but without taking the request priority into account.

R (Reoptimization) bit: when set, the requesting PCC specifies that the PCReq message relates to the reoptimization of an existing TE LSP in which case the path of the existing TE LSP to be reoptimized MUST be provided in the PCReq message by means of an RRO object defined in section 7.

B (Bi-directional) bit: when set, the PCC specifies that the path computation request relates to a bidirectional TE LSP (LSPs that have the same traffic engineering requirements including fate sharing, protection and restoration, LSRs, and resource requirements (e.g., latency and jitter) in each direction). When cleared, the TE LSP is unidirectional.

C (Cost) bit: when set, the PCE MUST provide the cost of the computed path in the PCRep message.

O (strict/lOose): In a PCReq message, when set, this means that a strict/loose path is acceptable. Otherwise, when cleared, this indicates to the PCE that an explicit path is required. In a PCRep message, when the O bit is set this indicates that the returned path is strict/loose, otherwise (the O bit is cleared), the returned path is explicit.

Request-ID-number: 32 bits

This value (combined with the source IP address of the PCC) uniquely identifies the path computation request context and MUST be incremented each time a new request is sent to the PCE. If no path computation reply is received from the PCE, and the PCC wishes to resend its request, the same Request-ID-number MUST be used. Conversely, different Request-ID-number MUST be used for different requests sent to a PCE. The same Request-ID-number may be used for path computation requests sent to different PCEs. The path computation reply is unambiguously identified by the IP source address of the replying PCE.

7.4. NO-PATH Object

When a PCE cannot find a path satisfying a set of constraints, it MUST include a NO-PATH object in the corresponding PCRep message. In its simplest form, the NO-PATH object is limited to a set of flags and just reports the impossibility to find a path that satisfies the set of constraints. Optionally, if the PCE supports such capability, the PCRep message MAY also comprise a list of objects that specify the set of constraints that could not be satisfied. When an object specifies a variety of constraints, the set of unsatisfied constraints can be unambiguously determined by the PCC after a simple comparison with the original requested constraints.

NO-PATH Object-Class is to be assigned by IANA (recommended value=3)
NO-PATH Object-Type is to be assigned by IANA (recommended value=1)

The format of the NO-PATH object body is as follows:

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|C|S|     Flags                 |          Reserved             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 11 - NO-PATH object format

The NO-PATH object has a fixed length of 4 octets.

Flags: 16 bits - The following flags are currently defined:

C bit: when set, this indicates that the set of unsatisfied constraints (reasons why a path could not be found) is specified in the PCRep message by means of the relevant PCEP objects. When cleared, no reason is specified.

For example, consider the case of a PCC that sends a path computation request to a PCE for a TE LSP of X MBits/s. Suppose that PCE cannot find a path for X MBits/s. In this case, the PCE includes in its path computation reply a NO-PATH object with the C flag set. In addition, the PCRep message carries the BANDWIDTH object and the bandwidth field value is equal to X.

When the NO-PATH object is absent from a PCRep message, the path computation request has been fully satisfied and the corresponding path(s) is/are provided in the PCRep message.

7.5. END-POINTS Object

The END-POINTS object is used in a PCReq message to specify the source IP address and the destination IP address of the TE LSP for which a path computation is requested. Two END-POINTS objects (for IPv4 and IPv6) are defined.

END-POINTS Object-Class is to be assigned by IANA (recommended value=4)
END-POINTS Object-Type is to be assigned by IANA (recommended value=1 for IPv4 and 2 for IPv6)

The format of the END-POINTS object body for IPv4 (Object-Type=1) is as follows:

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     Source IPv4 address                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                  Destination IPv4 address                     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 12 - END-POINTS object body format for IPv4

The format of the END-POINTS object for IPv6 (Object-Type=2) is as follows:

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                Source IPv6 address (16 bytes)                 |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|              Destination IPv6 address (16 bytes)              |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 13 - END-POINTS object body format for IPv6

7.6. BANDWIDTH object

The BANDWIDTH object is optional and can be used to specify the requested bandwidth and may be carried within PCReq and PCRep messages. The absence of the BANDWIDTH object MUST be interpreted by the PCE as a path computation request related to a 0 bandwidth TE LSP.

When carried within a PCReq message, the BANDWIDTH object specifies a bandwidth constraint that must be satisfied by the computed path(s) if P flag is cleared and MAY be ignored if the P flag is set. In a PCRep message, the BANDWIDTH object indicates that the bandwidth belong to the set of one or more constraint(s) that could be not satisfied. When absent from the PCRep message that means that the computed path satisfies the requested bandwidth constraint.

BANDWIDTH Object-Class is to be assigned by IANA (recommended value=5)

BANDWIDTH Object-Type is to be assigned by IANA (recommended value=1)

The format of the BANDWIDTH object body is as follows:

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         BANDWIDTH                             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 14 - BANDWIDTH object body format

Bandwidth: 32 bits. The requested bandwidth is encoded in 32 bits in IEEE floating point format, expressed in bytes per second.

7.7. DELAY Object

The DELAY object can be used to specify a strict delay constraint for the TE LSP. The delay constraint MUST be taken into account during path computation if P flag is cleared and MAY be ignored if the P flag is set. Note that the mechanism used by the PCE to retrieve the delays of each link is outside of the scope of this document (for the sake of illustration the link delay could be the IGP metric or a Service Provider may choose to use the TE metric to represent link delays). It must be understood that such path metric is only meaningful if used consistently: for instance, if the delay of a path computation segment is exchanged between two PCE residing in different domains, consistent ways of defining the delay must be used. The delay metric may be carried within PCReq and PCRep messages. The absence of the DELAY object MUST be interpreted by the PCE as a path computation request without delay constraint. When carried within a PCReq message, the DELAY object specifies a delay constraint that must be satisfied by the computed path(s). In a PCRep message and when the path computation was successful, the DELAY object indicates the delay(s) of the computed path(s). When the path computation was unsuccessful and the delay constraint was one of the mandatory constraints that could be satisfied the DELAY object MUST be present in the PCRep message.

DELAY Object-Class is to be assigned by IANA (recommended value=6)
DELAY Object-Type is to be assigned by IANA (recommended value=1)

The format of the DELAY object body is as follows:

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           Delay                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 15 - DELAY object body format

Delay: 32 bits. The requested delay constraint is encoded in 32 bits in IEEE floating point format, expressed in milliseconds.

7.8. ERO Object

The ERO object is used to encode a TE LSP path. If can either be carried within a PCReq message to specify the existing path of a TE LSP to be reoptimize or within a PCRep message to provide a computed TE LSP.

The contents of this object are identical in encoding to the contents of the Explicit Route Object defined in [RSPV-TE], [GRSVP] and [RSVP-UNNUM]. That is, the object is constructed from a series of sub-objects. Any RSVP ERO sub-object already defined or that could be defined in the future for use in the ERO is acceptable in this object.

PCEP ERO sub-object types correspond to RSVP ERO sub-object types.

Since the explicit path is available for immediate signaling by the MPLS or GMPLS control plane, the meanings of all of the sub-objects and fields in this object are identical to those defined for the ERO.

ERO Object-Class is to be assigned by IANA (recommended value=7)
ERO Object-Type is to be assigned by IANA (recommended value=1)

7.9. RRO Object

The RRO object is used to record the route followed by a TE LSP. The PCEP RRO object is exclusively carried within a PCReq message so as to specify the route followed by a TE LSP for which a reoptimization is desired.

The contents of this object are identical in encoding to the contents of the Route Record Object defined in [RSPV-TE], [G-RSVP] and [RSVP-UNNUM]. That is, the object is constructed from a series of sub-objects. Any RSVP RRO sub-object already defined or that could be defined in the future for use in the RRO is acceptable in this object.

The meanings of all of the sub-objects and fields in this object are identical to those defined for the RRO.

PCEP RRO sub-object types correspond to RSVP RRO sub-object types.

RRO Object-Class is to be assigned by IANA (recommended value=8)
RRO Object-Type is to be assigned by IANA (recommended value=1)

7.10. LSPA Object

The LSPA object specifies various TE LSP attributes to be taken into account by the PCE during path computation. The LSPA (LSP Attributes) object can either be carried within a PCReq message or a PCRep message in case of unsuccessful path computation (in this case, the PCReq message also comprises a NO-PATH object and the LSPA object is used to indicate the set of constraint(s) that could not be satisfied). Most of the fields of the LSPA object are identical to the fields of the SESSION-ATTRIBUTE object defined in [RSVP-TE] and [FRR].

LSPA Object-Class is to be assigned by IANA (recommended value=9)

Two Objects-Types are defined for the LSPA object: LSPA without resource affinity (Object-Type to be assigned by IANA with recommended value=1) and LSPA with resource affinity (Object-type=2).

The format of the LSPA object body with and without resource affinity are as follows:

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Setup Prio   |  Holding Prio |  Flags   |N|B|L|    Reserved   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
//                     Optional TLV(s)                         //
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 16 - LSPA object body format (without resource affinity)

```
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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       Exclude-any                             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       Include-any                             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       Include-all                             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Setup Prio   |  Holding Prio |  Flags   |N|B|L|    Reserved   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
//                     Optional TLV(s)                         //
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 17 - LSPA object body format (with resource affinity)

Setup Priority (8 bits)

The priority of the session with respect to taking resources,

in the range of 0 to 7. The value 0 is the highest priority.
The Setup Priority is used in deciding whether this session can

preempt another session.

Holding Priority

The priority of the session with respect to holding resources, in the range of 0 to 7. The value 0 is the highest priority. Holding Priority is used in deciding whether this session can be preempted by another session.

Flags

The flags L, B and N correspond to the "Local protection desired" bit ([RSVP-TE]), "Bandwidth protection desired" bit ([FRR]) and the "Node protection desired" bit ([FRR]) of the SESSION-ATTRIBUTE Object respectively.

L Flag (Local protection desired)

When set, this means that the computed path MUST included links protected with Fast Reroute as defined in [FRR].

B Flag (Bandwidth protection desired)

When set, this means that the computed path MUST included links protected with Fast Reroute as defined in [FRR] and that benefit from bandwidth protection. The B flag MUST only be set if the L flag is set.

N Flag (Node protection desired)

When set, this means that the computed path MUST included links protected with Fast Reroute as defined in [FRR] and that such links MUST be protected with NNOP (Next-next hop backup tunnel). The N flag MUST only be set of the L flag is set.

Note that the B flag and N flag are not exclusive.

7.11. IRO Object

The IRO (Include Route Object) object is optional and can be used to specify that the computed path must traverse a set of specified network elements. The IRO object may be carried within PCReq and PCRep messages.

IRO Object-Class is to be assigned by IANA (recommended value=10)
IRO Object-Type is to be assigned by IANA (recommended value=1)

```
0
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
```

```
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                                                               |
//                        (subobjects)                           |
```

```
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 18 - IRO objet body format

Subobjects

The IRO object is made of sub-object(s) identical to the ones defined in [RSVP-TE], [G-MPLS] and [RSVP-UNNUM] for use in EROs.

The following subobject types are supported:

Type	Subobject
1	IPv4 prefix
2	IPv6 prefix
4	Unnumbered interface ID
32	Autonomous system number

The L bit of such sub-object has no meaning within an IRO object.

The ERO object carried within a PCReq message is exclusively used in the context of a reoptimization path computation request, thus the need to define a new object (IRO) to specify the inclusion of specified network element(s) in a path.

7.12. SVEC Object

Section 8 details the circumstances under which it may be desirable and/or required to correlate several path computation requests. This leads to the specification of the SVEC object (Synchronization VECtor). The SVEC object is optional in a PCEP message.

The aim of the SVEC object carried within a PCReq message is to specify the correlation of M path computation requests. The SVEC object is a variable length object that lists the set of M requests the computation of which MUST be synchronized. Each path computation request is uniquely identified by the Request-ID-number carried within the respective RP object. The SVEC object also contains a set of flags that specify the synchronization type.

The SVEC object is carried within PCReq messages.

SVEC Object-Class is to be assigned by IANA (recommended value=11)
SVEC Object-Type is to be assigned by IANA (recommended value=1)

One Object-Type is defined for this object to be assigned by IANA with a recommended value of 1.

The format of the SVEC object body is as follows:

```
    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
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

```
|   Reserved    |                  Flags                  |S|N|L|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     RP Object #1                              |
|                                                               |
//                                                             //
|                     RP Object #M                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 19 - SVEC body object format

Flags

Defines the synchronization type between multiple path computation requests.

L (Link diverse) bit: when set, this indicates that the computed paths corresponding to the requests specified by the following RP objects MUST not have any link in common.

N (Node diverse) bit: when set, this indicates that the computed paths corresponding to the requests specified by the following RP objects MUST not have any node in common.

S (SRLG diverse) bit: when set, this indicates that the computed paths corresponding to the requests specified by the following RP objects MUST not share any SRLG (Shared Risk Link Group).

The flags defined above are not exclusive.

7.13. NOTIFICATION object

The NOTIFICATION object is exclusively carried within a PCNtf message and can either be used in a message sent by a PCC to a PCE or by a PCE to a PCC so as to notify of an event.

NOTIFICATION Object-Class is to be assigned by IANA (recommended value=12)
NOTIFICATION Object-Type is to be assigned by IANA (recommended value=1)

One Object-Type is defined for this object to be assigned by IANA with a recommended value of 1.

The format of the NOTIFICATION body object is as follows:

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Length    |     Flags     | Notification- | Notification- |
|               |               | type          | value         |
```

```
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
```

```
//                      Optional TLV(s)                        //
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 20 - NOTIFICATION body object format

Length

The Length contains the total length of the object in bytes and includes the Type and Length fields. This length must be a multiple of 4 and must be at least 12.

Flags

No flags are currently defined

A NOTIFICATION object is characterized by a Notification-type that specifies the class of notification and the Notification-value that provides additional information related to the nature of the notification. Both the Notification-type and Notification-value should be managed by IANA (see IANA section).

The following Notification-type and Notification-value values are currently defined:

Notification-type=1: Pending Request cancelled

Notification-value=1: PCC cancels a set of pending request(s)

A Notification-type=1, Notification-value=1 indicates that the PCC wants to inform a PCE of the cancellation of a set of pending request(s). Such event could be triggered because of external conditions such as the receipt of a positive reply from another PCE (should the PCC have sent multiple requests to a set of PCEs for the same path computation request), a network event such as a network failure rendering the request obsolete or any other event(s) local to the PCC. A NOTIFICATION object with Notification-type=1, Notification-value=1 is exclusively carried within a PCNtf message sent by the PCC to the PCE. The RP object MUST also be present in the PCNtf message. Multiple RP objects may be carried within the PCNtf message in which case the notification applies to all of them. If such notification is received by a PCC from a PCE, the PCC MUST silently ignore the notification and no errors should be generated.

Notification-value=2: PCE cancels a set of pending request(s)

A Notification-type=1, Notification-value=2 indicates that the PCE wants to inform a PCC of the cancellation of a set of pending request(s). Such event could be triggered because of

some PCE congested state or because of some path computation
requests that are part the set of synchronized path computation

requests are missing. A NOTIFICATION object with Notification-type=1, Notification-value=2 is exclusively carried within a PCNtf message sent by a PCE to a PCC. The RP object MUST also be present in the PCNtf message. Multiple RP objects may be comprised within the PCNtf message in which case the notification applies to all of them. If such notification is received by a PCE from a PCC, the PCE MUST silently ignore the notification and no errors should be generated.

Notification-type=2: PCE congestion

Notification-value=1

A Notification-type=2, Notification-value=1 indicates to the PCC(s) that the PCE is currently in a congested state. If no RP objects are comprised in the PCNtf message, this indicates that no other requests SHOULD be sent to that PCE until the congested state is cleared: the pending requests are not affected and will be served. If some pending requests cannot be served due to the congested state, the PCE MUST also include a set of RP object(s) that identifies the set of pending requests which will not be honored and which will be cancelled by the PCE. In this case, the PCE does not have to send an additional PCNtf message with Notification-type=1 and Notification-value=2 since the list of cancelled requests is specified by including the corresponding set of RP object(s). If such notification is received by a PCE from a PCC, the PCE MUST silently ignore the notification and no errors should be generated.

Optionally, a TLV named CONGESTION-DURATION may be included in the NOTIFICATION object that specifies the duration during which no further request should be sent to the PCE. Once this period has expired the PCE should no longer be considered in congested state.

The CONGESTION-DURATION TLV is composed of 1 octet for the type, 1 octet specifying the number of bytes in the value field followed by a fix length value field of 4 octets specifying the estimated PCE congestion duration in seconds. The CONGESTION-DURATION TLV is padded to eight-octet alignment

TYPE: To be assigned by IANA
LENGTH: 4
VALUE: estimated congestion duration in seconds

Notification-value=2

A Notification-type=2, Notification-value=2 indicates that the PCE is no longer in congested state and is available to process new path computation requests. An implementation MUST make sure that a PCE sends such notification to every PCC to which a Notification message (with Notification-type=2, Notification-value=1) has been sent unless a CONGESTION-DURATION TLV has been included in the corresponding message and the PCE wishes to wait for the expiration of that period of time before receiving new requests. An implementation may decide to cancel such notification if the PCC is in down state for a specific period. A RECOMMENDED value for such delay is 1 hour. If such notification is received by a PCE from a PCC, the PCE MUST silently ignore the notification and no errors should be generated.

It is RECOMMENDED to support some dampening notification procedure on the PCE so as to avoid too frequent congestion notifications and releases. For example, an implementation could make use of an hysteresis approach using a dual-thresholds mechanism triggering the sending of congestion notifications and releases. Furthermore, in case of high instabilities of the PCE resources, an additional dampening mechanism SHOULD be used (linear or exponential) to pace the notification frequency and avoid path computation requests oscillation.

7.14. PCEP-ERROR object

The PCEP-ERROR object is exclusively carried within a PCErr message and can either be used in a message sent by a PCC to a PCE or by a PCE to a PCC to notify of a PCEP protocol error.

PCEP-ERROR Object-Class is to be assigned by IANA (recommended value=13)
PCEP-ERROR Object-Type is to be assigned by IANA (recommended value=1)

One Object-Type is defined for this object to be assigned by IANA with a recommended value of 1.

The format of the PCEP-ERROR object body is as follows:

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Length    |      Flags    |   Error-Type  |  Error-Value  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
//                      Optional TLV(s)                        //
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 21 - PCEP-ERROR object body format

A PCEP-ERROR object is used to report a PCEP protocol error and is characterized by an Error-Type that specifies the type of error and an Error-value that provides additional information about the error type. Both the Error-Type and the Error-Value should be managed by IANA (see the IANA section).

Length (8 bits)

The Length contains the total length of the object in bytes including the Type and Length fields. This length must be a multiple of 4 and must be at least 8.

Flags (8 bits)

No flag is currently defined.

Error-type (8 bits)

The Error-type defines the class of error.

Error-value (8 bits)

Provides additional details about the error.

Optionally the PCErr message may contain additional TLV so as to provide further information about the encountered error. No TLV is currently defined.

A single PCErr message may contain multiple PCEP-ERROR objects.

For each PCEP protocol error, an Error type and value is defined.

```
Error-Type    Meaning
   1          Capability not supported
   2          Unknown Object
               Error-value=1: Unrecognized object class
               Error-value=2: Unrecognized object Type
   3          Not supported object
               Error-value=1: Not supported object class
               Error-value=2: Not supported object Type
   4          Policy violation
               Error-value=1: C bit set (request rejected)
               Error-value=2: O bit set (request rejected)
   5          Required Object missing
               Error-value=1: RP object missing
                Error-value=2: RRO object missing for a reoptimization
                               request (R bit of the RP object set)
               Error-value=3: END-POINTS object missing
   6          Synchronized path computation request missing
   7          Unknown request reference
   8          Unacceptable PCEP session characteristics
               Error-value=1: parameter negotiation
               Error-value=2: parameters negotiation failed
```

9	Deadtimer expired

In case of the Error-Type 1, the PCE indicates that the path computation request cannot be completed because it does not support one or more required capability. The corresponding path computation request MUST then be cancelled.

If a PCEP message is received that carries a mandatory PCEP object (P flag cleared) not recognized by the PCEP peer or recognized but not supported, then the PCEP peer MUST send a PCErr message with a PCEP-ERROR object (Error-Type=2 and 3 respectively). The corresponding path computation request MUST be cancelled by the PCE without further notification.

If a path computation request is received which is not compliant with an agreed policy between the PCC and the PCE, the PCE MUST send a PCErr message with a PCEP-ERROR object (Error-Type=4). The corresponding path computation MUST be cancelled.

If a path computation request is received that does not contain a required object, the PCE MUST send a PCErr message with a PCEP-ERROR object (Error-Type=5). If there are multiple mandatory objects missing, the PCErr message MUST contain one PCEP-ERROR object per missing object. The corresponding path computation MUST be cancelled.

If a PCC sends a synchronized path computation request to a PCE and the PCE does not receive all the synchronized path computation requests listed within the corresponding SVEC object during a configurable period of time, the PCE MUST send a PCErr message with a PCEP-ERROR object (Error-Type=6). The corresponding synchronized path computation MUST be cancelled.

If a PCC receives a PCRep message related to an unknown path computation request, the PCC MUST send a PCErr message with a PCEP-ERROR object (Error-Type=6). In addition, the PCC MUST include in the PCErr message the unknown RP object.

If one or more characteristic(s) is not acceptable by the receiving peer, it MUST send a PCErr message with Error-type=8, Error-value=1. The PCErr message MUST also comprise an Open object: for each unacceptable session parameter, an acceptable parameter value MUST be proposed in the appropriate field of the Open object in place of the originally proposed value. If a second Open message is received with the same set of parameters or with parameters differing from the proposed values, the receiving peer MUST send a PCErr message with Error-Type=8, Error-value=2 and it MUST immediately close the TCP connection.

If a PCEP peer does not receive any PCEP message (Keepalive, PCReq, PCRep, PCNtf) during the Deadtimer period (equal to four times the Keepalive value advertised in the OPEN object) the PCEP peer MUST send a PCErr message with a PCEP-ERROR object (Error-type=9, Error-value=1). Additionally, the PCEP session MUST be terminated and the TCP connection MUST be closed.

8. Independent versus synchronized path computation requests

The PCEP protocol permits the bundling of multiple independent path computation requests within a single PCRep message. A set of path computation requests is said to be non synchronized if their respective treatment (path computations) can be performed by a PCE in a serialized and independent fashion.

There are various circumstances where the synchronization of a set of path computations may be beneficial or required.

Consider the case of a set of N TE LSPs for which a PCC needs to send path computation requests to a PCE so as to obtain their respective paths. The first solution consists of sending N separate PCReq messages to the selected PCE. In this case, the path computation requests are independent. Note that the PCC may chose to distribute the set of N requests across K PCEs for load balancing reasons. Considering that M (with M<N) requests are sent to a particular PCEi, as described above, such M requests can be sent in the form of successive PCReq messages destined to PCEi or grouped within a single PCReq message. This is of course a viable solution if and only if such requests are independent. That said, it can be desirable to request from the PCE the computation of their paths in a synchronized fashion that is likely to lead to more optimal path computations and/or reduced blocking probability if the PCE is a stateless PCE. In other words, the PCE should not compute the corresponding paths in a serialized and independent manner but it should rather simultaneously compute their paths.

For example, trying to simultaneously compute the paths of M TE LSPs may allow the PCE to improve the likelihood to meet multiple constraints. Consider the case of two TE LSPs requesting N1 MBits/s and N2 MBits/s respectively and a maximum tolerable end to end delay for each TE LSP of X ms. There may be circumstances where the computation of the first TE LSP irrespectively of the second TE LSP may lead to the impossibility to meet the delay criteria for the second TE LSP. A second example is related to the bandwidth constraint. It is quite straightforward to provide examples where a serialized independent path computation approach would lead to the impossibility to satisfy both requests (due to bandwidth fragmentation) while a synchronized path computation would successfully satisfy both requests. A last example relates to the ability to avoid the allocation of the same resource to multiple requests thus helping to reduce the call set up failure probability compared to the serialized computation of independent requests.

Furthermore, if the PCC has to send a large number of path computation requests, it may also be desirable to pack multiple requests within a single PCReq object so as to minimize the control plane overhead. Note that the algorithm used by the PCC to "pack" a set of requests introduces some unavoidable trade-off between control plane load and delays and such algorithm is outside of the scope of this document.

There are other cases where the computation of M requests must be synchronized an obvious example of which being the computation of M diverse paths. If such paths are computed in a non-synchronized fashion this seriously increases the probability of not being able to satisfy all requests (sometimes also referred to as the well-know "trapping problem"). Furthermore, this would not allow a PCE to implement objective functions such as trying to minimize the sum of the TE LSP costs. In such a case, the path computation requests are synchronized: they cannot be computed independently of each other.

The synchronization of a set of path computation requests is achieved by using the SVEC object that specifies the list of synchronized requests along with the nature of the synchronization.

9. Elements of procedure

9.1. Non recognized or non support object received in a PCReq message

If a PCEP message is received that carries a mandatory PCEP object (P flag cleared) not recognized by the PCE or recognized but not supported, then the PCE MUST send a PCErr message with a PCEP-ERROR object (Error-Type=2 and 3 respectively). In addition, the PCRep message MUST comprise the set of non recognized or non supported object(s). The corresponding path computation request MUST be cancelled by the PCE without further notification.

9.2. RP object

The absence of a RP object in the PCReq message MUST trigger the sending of a PCErr message with Error-type=5 and Error-value=1.

If the C bit of the RP message carried within a PCReq message is set and some local policy has been configured on the PCE not to provide such cost, a PCErr message MUST be sent by the PCE to the requesting PCC and the pending path computation request MUST be discarded. The Error-type and Error-value of the PCEP-ERROR object MUST be set to 4 and 1 respectively.

If the O bit of the RP message carried within a PCReq message is set and some local policy has been configured on the PCE to not provide explicit path(s) (for instance, for confidentiality reasons), then a PCErr message MUST be sent by the PCE to the requesting PCC and the pending path computation request MUST be discarded. The Error-type and Error-value of the PCEP-ERROR object MUST be set to 4 and 2 respectively.

R bit: when the R bit of the RP object is set in a PCReq message, this indicates that the path computation request relates to the reoptimization of an existing TE LSP. In this case, the PCC MUST provide the explicit or strict/loose path by including an RRO object in the PCReq message so as to avoid double bandwidth counting (unless the TE LSP is a 0-bandwidth TE LSP). If the PCC has previously requested a non-explicit path (O bit set), a reoptimization can still be requested by the PCC but this implies for the PCE to be either stateful (keep track of the previously computed path with the associated list of strict hops) or to have the ability to retrieve the complete required path segment, or for PCC to inform PCE of the working path with associated list of strict hops in PCReq. The absence of an RRO in the PCReq message when the R bit of the RP object is set MUST trigger the sending of a PCErr message with Error-type=5 and Error-value=2.

If the PCC receives a PCRep message which contains a RP object referring to an unknown Request-ID-Number, it MUST trigger the sending of a PCErr message with Error-Type=7 and Error-value=1.

9.3. SVEC object

When a requesting PCC desires to send multiple synchronized path computation requests, it MUST send all the path computation requests within a single PCReq message that contains all the synchronized path computation requests: in that case, the PCReq message MUST also comprise a SVEC object listing all the synchronized path computation requests. Note that such PCReq message may also comprise non-synchronized path computation requests. For example, the PCReq message may comprise N synchronized path computation requests related to RP 1, ... , RP N listed in the SVEC object along with any other path computation requests.

If some RPs objects carried with the SVEC object are missing in the PCReq message, the PCE MUST send a PCErr message with Error-Type = 6 to the PCC.

10. Manageability Considerations

It is expected and required to specify a MIB for the PCEP communication protocol (in a separate document).

Furthermore, additional tools related to performance, fault and diagnostic detection are required which will also be specified in separate documents.

11. IANA Considerations

11.1. TCP port

The PCEP protocol will use a well-known TCP port to be assigned by IANA.

11.2. PCEP Objects

Several new PCEP objects are defined in this document that have an Object-Class and an Object-Type. The new Object-Class and Object-Type should be assigned by IANA.

- Open Object

The Object-Class of the Open object is to be assigned by IANA (recommended value=1).

One Object-Type is defined for this object and should be assigned by IANA with a recommended value of 1.

- RP Object

The Object-Class of the RP object is to be assigned by IANA (recommended value=2).

One Object-Type is defined for this object and should be assigned by IANA with a recommended value of 1.

- NO-PATH Object

The Object-Class of the NO-PATH object is to be assigned by IANA (recommended value=3).

One Object-Type is defined for this object and should be assigned by IANA with a recommended value of 1.

- END-POINTS Object

The Object-Class of the END-POINTS object is to be assigned by IANA (recommended value=4).

Two Object-Type are defined for this object and should be assigned by IANA with a recommended value of 1 and 2 for IPv4 and IPv6 respectively.

- BANDWIDTH Object

The Object-Class of the BANDWIDTH object is to be assigned by IANA (recommended value=5).

One Object-Type is defined for this object and should be assigned by IANA with a recommended value of 1.

- DELAY Object

The Object-Class of the DELAY object is to be assigned by IANA (recommended value=6).

One Object-Type is defined for this object and should be assigned by IANA with a recommended value of 1.

- ERO Object

The Object-Class of the ERO object is to be assigned by IANA (recommended value=7).

One Object-Type is defined for this object and should be assigned by IANA with a recommended value of 1.

- RRO Object

The Object-Class of the RRO object is to be assigned by IANA (recommended value=8).

One Object-Type is defined for this object and should be assigned by IANA with a recommended value of 1.

- LSPA Object

The Object-Class of the LSPA object is to be assigned by IANA (recommended value=9).

Two Object-Types are defined for this object and should be assigned by IANA with a recommended value of 1 (without resource affinity) and 2 (with resource affinity).

- IRO Object

The Object-Class of the IRO object is to be assigned by IANA (recommended value=10).

One Object-Type is defined for this object and should be assigned by IANA with a recommended value of 1.

- SVEC Object

The Object-Class of the SVEC object is to be assigned by IANA (recommended value=11).

One Object-Type is defined for this object and should be assigned by IANA with a recommended value of 1.

- NOTIFICATION Object

The Object-Class of the NOTIFICATION object is to be assigned by IANA (recommended value=12).

One Object-Type is defined for this object and should be assigned by IANA with a recommended value of 1.

- PCEP-ERROR Object

The Object-Class of the PCEP-ERROR object is to be assigned by IANA (recommended value=13).

One Object-Type is defined for this object and should be assigned by IANA with a recommended value of 1.

11.3. Notification

A NOTIFICATION object is characterized by a Notification-type that specifies the class of notification and a Notification-value that provides additional information related to the nature of the notification. Both the Notification-type and Notification-value should be managed by IANA (see IANA section).

The following Notification-type and Notification-value values are currently defined:

Notification-type=1: Pending Request cancelled

Notification-value=1: PCC cancels a set of pending request(s)

Notification-value=2: PCE cancels a set of pending request(s)

Notification-type=2: PCE congestion

Notification-value=1: PCE in congested state

Notification-value=2: PCE no longer in congested state

11.4. PCEP Error

A PCEP-ERROR object is used to report a PCEP protocol error and is characterized by an Error-Type which specifies the type of error and an Error-value that provides additional information about the error type. Both the Error-Type and the Error-Value should be managed by IANA.

Error-Type	Meaning
1	Capability not supported
2	Unknown Object
	Error-value=1: Unrecognized object class
	Error-value=2: Unrecognized object Type
3	Not supported object
	Error-value=1: Not supported object class
	Error-value=2: Not supported object Type
4	Policy violation
	Error-value=1: C bit set (request rejected)
	Error-value=2: O bit set (request rejected)
5	Required Object missing

Error-value=1: RP object missing
Error-value=2: RRO object missing for a reoptimization

request (R bit of the RP object set).
Error-value=3: END-POINTS object missing

6 Synchronized path computation request missing

7 Unknown request reference

8 Unacceptable PCEP session characteristics
Error-value=1: parameter negotiation
Error-value=2: parameters negotiation failed

9 Deadtimer expired

12. Security Considerations

PCEP communication could be the target of the following attacks:
- -Spoofing (PCC or PCE impersonation)
- -Snooping (message interception)
- -Falsification
- -Denial of Service

A PCEP attack may have significant impact, particularly in an inter-AS context as PCEP facilitates inter-AS path establishment. Several mechanisms are proposed below, so as to ensure authentication, integrity and privacy of PCEP Communications, and also to protect against DoS attacks.

12.1. PCEP Authentication and Integrity

It is RECOMMENDED to use TCP-MD5 [RFC1321] signature option to provide for the authenticity and integrity of PCEP messages. This will allow protecting against PCE or PCC impersonation and also against message content falsification.

This requires the maintenance, exchange and configuration of MD-5 keys on PCCs and PCEs. Note that such maintenance may be especially onerous to the operators as pointed out in [BGP-SEC-REQ]. Hence it is important to limit the number of keys while ensuring the required level of security.

MD-5 signature faces some limitations, as per explained in [RFC2385]. Note that when one digest technique stronger than MD5 is specified and implemented, PCEP could be easily upgraded to use it.

12.2. PCEP Privacy

Ensuring PCEP communication privacy is of key importance, especially in an inter-AS context, where PCEP communication end-points do not reside in the same AS, as an attacker that intercept a PCE message could obtain sensitive information related to computed paths and resources. Privacy can be ensured thanks to encryption. To ensure privacy of PCEP communication, IPSec [IPSEC] tunnels MAY be used between PCC and PCEs or between PCEs. Note that this could also be used to ensure Authentication and Integrity, in which case, TCP MD-5 option would not be required.

12.3. Protection against Denial of Service attacks

PCEP can be the target of TCP DoS attacks, such as for instance SYN attacks, as all protocols running on top of TCP. PCEP can use the same mechanisms as defined in [LDP] to mitigate the threat of such attacks:

- A PCE should avoid promiscuous TCP listens for PCEP TCP session establishment. It should use only listens that are specific to authorized PCCs.
- The use of the MD5 option helps somewhat since it prevents a SYN from being accepted unless the MD5 segment checksum is valid. However, the receiver must compute the checksum before it can decide to discard an otherwise acceptable SYN segment.
- The use of access-list on the PCE so as to restrict access to authorized PCCs.

13. Intellectual Property Statement

14. Acknowledgment

We would like to thank Dave Oran, Dean Cheng, Jerry Ash, Igor Bryskin for their very valuable input. Special thank to Adrian Farrel for his very valuable suggestions.

15. References

15.1. Normative references

[RFC] Bradner, S., "Key words for use in RFCs to indicate requirements levels", RFC 2119, March 1997.

[RFC3667] Bradner, S., "IETF Rights in Contributions", BCP 78, RFC 3667, February 2004.

[RFC3979] Bradner, S., Ed., "Intellectual Property Rights in IETF Technology", BCP 79, RFC 3979, March 2005.

[RSVP] R. Braden et al., "Resource ReSerVation Protocol (RSVP) - Version 1 Functional Specification", RFC 2205, September 1997.

[RSVP-TE] Awduche, D., et. al., "RSVP-TE: Extensions to RSVP for LSP tunnels", RFC 3209, December 2001.

[G-RSVP] Berger, L, et. al., "GMPLS Signaling RSVP-TE extensions", RFC 3473, January 2003.

[RSVP-UNNUM] Kompella, K., Rekhter Y., "Signalling Unnumbered Links in Resource ReSerVation Protocol - Traffic Engineering (RSVP-TE)", RFC 3477, January 2003.

[COPS] Durham, D., "The COPS (Common Open Policy Service) Protocol", RFC 2748, January 2000.

[SCTP] Stewart et al., "Stream Control Transmission Protocol", RFC2960, October 2000.

[TCP] J. Postel, "Transmission Control Protocol", RFC 793, September 1981.

[DS-TE-PROTO] Le Faucheur et al,"Protocol extensions for support of Differentiated-Service-aware MPLS Traffic Engineering", RFC 4124, June 2005.

[G-RECV-E2E-SIG] J. P. Lang et al, "RSVP-TE Extensions in support of End-to-End Generalized Multi-Protocol Label Switching (GMPLS)-based Recovery", draft-ietf-ccamp-gmpls-recovery-e2e-signaling-03.txt (working in progress).

[FRR] P. Pan, G. Swallow, A. Atlas, JP. Vasseur, M. Jork, D.H Gan and D. Cooper, "Fast Reroute Extensions to RSVP-TE for LSP Tunnels", RFC4090, May 2005.

15.2. Informative References

[WP] E. Rescorla, "Writting Protocol Models", RFC 4101, June 2005.

[PCE-ARCH] A. Farrel, JP. Vasseur and J. Ash, "Path Computation Element (PCE) Architecture", draft-ietf-pce-arch, work in progress.

[PCE-GEN-COM-REQ] J. Ash, J.L Le Roux et al., "PCE Communication Protocol Generic Requirements", draft-ietf-pce-comm-protocol-gen-reqs, work Progress.

[GMPLS-RTG] Kompella, K., Rekhter, Y., "Routing Extensions in Support of Generalized Multi-Protocol Label Switching", draft-ietf-ccamp-gmpls-routing, work in progress.

[INT-AREA-REQ] Le Roux, J.L., Vasseur, J.P., Boyle, J. et al, "Requirements for inter-area MPLS Traffic Engineering", RFC4105, June 2005.

[INT-AS-REQ] Zhang, R., Vasseur, J.P. et al, "MPLS Inter-AS Traffic Engineering Requirements", draft-ietf-tewg-interas-mpls-te-req, work in progress.

[INT-DOMAIN-FRWK] Farrel, A., Vasseur, J.P., Ayyangar, A., "A Framework for Inter-Domain MPLS Traffic Engineering", draft-ietf-ccamp-inter-domain-framework, work in progress.

[MGT] A. Farrel et al., "Requirements for Manageability Sections in Routing Area Drafts", draft-farrel-rtg-manageability-requirements, work in progress.

[XRO] Lee et al, "Exclude Routes - Extension to RSVP-TE", drfat-ietf-ccamp-rsvp-te-exclude-route, work in progress.

[PCE-DISC-REQ] JL Le Roux et al., "Requirements for Path Computation Element (PCE) Discovery", draft-ietf-pce-discovery-reqs, work in progress.

[BGP-SEC-REQ] B. Christian Ed., "BGP Security Requirements", draft-ietf-rpsec-bgpsecrec, work in progress

[LDP] L. Andersson, et al., "LDP Specification", RFC3036, January 2001

[DOBB] H. Dobbertin, "The Status of MD5 After a Recent Attack", RSALabs' CryptoBytes, Vol. 2 No. 2, Summer 1996.

[RFC1321] Rivest, R., "The MD5 Message-Digest Algorithm",RFC 1321, April 1992.

[IPSEC] S. Kent, A. Atkinson, " IP Encapsulating Security Payload (ESP)", RFC2406, November 1998

16. Authors' Address

Jean-Philippe Vasseur (Editor)
Cisco Systems, Inc.
300 Beaver Brook Road
Boxborough , MA - 01719
USA
Email: jpv@cisco.com

Jean-Louis Le Roux
France Telecom
2, avenue Pierre-Marzin
22307 Lannion Cedex
FRANCE
Email: jeanlouis.leroux@francetelecom.com

Arthi Ayyangar
Juniper Networks, Inc.
1194 N.Mathilda Ave
Sunnyvale, CA 94089
USA
E-mail: arthi@juniper.net

Eiji Oki
NTT
Midori 3-9-11
Musashino, Tokyo 180-8585
JAPAN
Email: oki.eiji@lab.ntt.co.jp

Alia K. Atlas
Google Inc.
1600 Amphitheatre Parkway
Mountain View, CA 94043
EMail: akatlas@alum.mit.edu

Andrew Dolganow
Alcatel
600 March Rd., K2K 2E6 Ottawa, ON, Canada
Phone: +1 (613)784 6285
Email: andrew.dolganow@alcatel.com

Yuichi Ikejiri
NTT Communications Corporation

1-1-6, Uchisaiwai-cho, Chiyoda-ku
Tokyo 100-8019

JAPAN
Email: y.ikejiri@ntt.com

Appendix A - Compliance of PCEP to the set of requirements specified in draft-ietf-pce-comm-protocol-gen-reqs

[PCE-GEN-COM-REQ] lists a set of requirement for the PCE communication protocol. The aim of the appendix A is to list the compliance of PCEP to such requirements. Note that requirements that are not satisfied in the context of the present version may be satisfied in further revisions.

The following legend is used in the table below:

YES: PCEP fully fulfills the requirement
ME (Minor Extension): PCEP could satisfy the requirement with minor extension(s).
SE (Substantial Extension): PCEP could satisfy the requirement with substantial extension(s).
NO: PCEP cannot meet the requirement without substantial redesign of the protocol.

Requirement	Necessity	Compliance
Commonality of PCC-PCE and PCE-PCE Communication	MUST	YES
Client-Server Communication	MUST	YES
Support PCC/PCE request message to request path computation	MUST	YES
Support PCE response message with computed path	MUST	YES
Support unsolicited communication PCE-PCC	MUST	YES
Maintain PCC-PCE session	NON-RQMT	
Use of Existing Transport Protocol	MAY	YES
Transport protocol satisfy reliability & security requirements	MAY	YES
Transport Protocol Limits Size of Message	MUST NOT	YES
Support Path Computation Requests	MUST	YES
Include source & destination		
Support path constraints (e.g., bandwidth, hops, affinities) to include/exclude	MUST	YES
Support path reoptimization & inclusion of a previously computed path	MUST	YES
Allow to select/prefer from advertised list of standard objective functions/options	MUST	ME
Allow to customize objective function/options	MUST	ME
Request a less-constrained path	MAY	ME
Support request for less-constrained path, including constraint-relaxation policy's	SHOULD	ME
Support Path Computation Responses	MUST	YES
Negative response support reasons for failure, constraints to relax to achieve positive result, less-constrained path reflecting constraint-relaxation policy's	SHOULD	ME

Cancellation of Pending Requests	MUST	YES
Multiple Requests and Responses	MUST	YES
Limit by configuration number of requests within		

a message	MUST	YES
Support multiple computed paths in response	MUST	YES
Support "continuation correlation" where related requests or computed paths cannot fit within one message	MUST	YES
Maximum message size & maximum number of requests per message exchanged through PCE messages to PCC, or indicated in request message	MAY	ME
Reliable Message Exchange (achieved by PCEP itself or transport protocol	MUST	YES
Allow detection & recovery of lost messages to occur quickly & not impede operation of PCEP	MUST	ME
Handle overload situations without significant decrease in performance, e.g., through throttling of requests	MUST	YES
Provide acknowledged message delivery with retransmission, in order message delivery or facility to restore order, message corruption detection, flow control & back-pressure to throttle requests, rapid partner failure detection, informed rapidly of failure of PCE-PCC connection	MUST	YES
Functionality added to PCEP if transport protocol provides it	SHOULD NOT	N/A
Secure Message Exchange (provided by PCEP or transport protocol	MUST	YES
Support mechanisms to prevent spoofing (e.g., authentication), snooping (e.g., encryption), DOS attacks	MUST	YES
Request Prioritization	MUST	YES
Unsolicited Notifications	SHOULD	YES
Allow Asynchronous Communication	MUST	YES
PCC Has to Wait for Response Before Making Another Request	MUST NOT	YES
Allow order of responses differ from order of Requests	MUST	YES
Communication Overhead Minimization	SHOULD	YES
Give particular attention to message size	SHOULD	
Extensibility without requiring modifications to the protocol	MUST	YES
Easily extensible to support intra-area, inter-area, inter-AS intra provider, inter-AS inter-provider, multi-layer path & virtual network topology path computation	MUST	YES
Easily extensible to support future applications not in scope (e.g., P2MP path computations)	SHOULD	YES
Scalability at least linearly with increase in number of PCCs, PCEs, PCCs communicating with a single PCE, PCEs communicated to by a single PCC,		

PCEs communicated to by another PCE, domains, path requests, handling bursts of requests	MUST	YES
Support Path Computation Constraints	MUST	ME

Support Different Service Provider Environments (e.g., MPLS-TE and GMPLS networks, centralized & distributed PCE path computation, single & multiple PCE path computation)	MUST	YES
Policy Support for policies to accept/reject requests, PCC to determine reason for rejection, notification of policy violation	MUST	ME
Aliveness Detection of PCCs/PCEs, partner failure Detection	MUST	YES
PCC/PCE Failure Response procedures defined for PCE/PCC failures, PCC able to clear pending Request	MUST	YES
PCC select another PCE upon detection of PCE failure	MUST	YES
PCE able to clear pending requests from a PCC (e.g. when it detects PCC failure or request buffer full)	MUST	YES
Protocol Recovery support resynchronization of information & requests between sender & receiver	MUST	ME
Minimize repeat data transfer, allow PCE to respond to computation requests issued before failure without requests being re-issued	SHOULD	ME
Stateful PCE able to resynchronize/recover states (e.g., LSP status, paths) after restart	SHOULD	SE