

PCE Working Group
Internet-Draft
Intended status: Best Current Practice
Expires: October 29, 2015

R. Casellas, Ed.
CTTC
O. Gonzalez de Dios, Ed.
Telefonica I+D
A. Farrel, Ed.
Old Dog Consulting
C. Margaria
Juniper Networks
D. Dhody
X. Zhang
Huawei Technologies
April 27, 2015

**PCEP Best Current Practices - Message formats and extensions
draft-many-pce-pcep-bcp-02**

Abstract

A core standards track RFC defines the main underlying mechanisms, basic object format and message structure of the Path Computation Element (PCE) Communications Protocol (PCEP). PCEP has been later extended in several RFCs, focusing on specific functionalities. The proliferation of such companion RFCs may cause ambiguity when implementing a PCE based solution. This document aims at documenting best current practices and at providing a reference RBNF grammar for PCEP messages, including object ordering and precedence rules.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

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

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

This Internet-Draft will expire on October 29, 2015.

Copyright Notice

Copyright (c) 2015 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1.	Introduction and Motivation	3
1.1.	Object Ordering Issues and Inconsistencies	4
1.2.	Inconsistent Naming	5
1.3.	Semantics and Exclusive Rules	6
2.	Initial Considerations	7
3.	Requirements Language	7
4.	RBNF Grammars	7
4.1.	Common Constructs	7
4.1.1.	Object Sequences	7
4.1.2.	Synchronized Vectors	8
4.1.3.	Monitoring Metrics	8
4.1.4.	Monitoring Requests and Responses	9
4.1.5.	Attributes	9
4.1.6.	Paths	11
4.2.	PCEP Messages	11
4.2.1.	PCEP Open Message	11
4.2.2.	PCEP Keep Alive (KeepAlive) Message	11
4.2.3.	PCEP Request (PCReq) Message	11
4.2.4.	PCEP Reply (PCRep) Message	12
4.2.5.	PCEP Monitoring Request (PCMonReq) Message	13
4.2.6.	PCEP Monitoring Reply (PCMonRep) Message	13
4.2.7.	PCEP Notify (PCNtf) Message	14
4.2.8.	PCEP Error (PCErr) Message	14
4.2.9.	PCEP Report (PCRpt) Message	15
4.2.10.	PCEP Update (PCUpd) Message	16
4.2.11.	PCEP Initiate (PCInitiate) Message	16
5.	Management Considerations	17
6.	Security Considerations	17
7.	Contributing Authors	17
8.	Acknowledgments	18
9.	Normative References	18

Authors' Addresses [20](#)

1. Introduction and Motivation

The RBNF notation, defined in [[RFC5511](#)], is used to specify the message format for the Path Computation Element communication Protocol (PCEP). The core of PCEP has been defined in [[RFC5440](#)] and later extended, notably, in [[RFC7150](#)] to support Vendor Extensions; in [[RFC5455](#)], adding a CLASSTYPE object to support Diffserv-aware Traffic Engineering (DS-TE); in [[RFC5520](#)], for topology confidentiality by means of Path-Keys; in [[RFC5521](#)], in support of exclusions; in [[RFC5541](#)] to convey specific Objective Functions; in [[RFC5557](#)], for Global Concurrent Optimization, in [[RFC5886](#)], for monitoring and in [[RFC6006](#)] for point-to-multipoint (P2MP) computation.

At the time of writing, several I.-D. are also addressing specific aspects, such as PCEP extensions for GMPLS networks [[I-D.ietf-pce-gmpls-pcep-extensions](#)], for hierarchical PCE [[I-D.ietf-pce-hierarchy-extensions](#)] or for multi-layer, multi-region networks [[I-D.ietf-pce-inter-layer-ext](#)]. Stateful PCE capabilities are also being defined in [[I-D.ietf-pce-stateful-pce](#)], including the case where a PCE is able to initiate the establishment and release of LSPs in [[I-D.ietf-pce-pce-initiated-lsp](#)].

Most PCEP RFCs describe specific protocol extensions and, as such, they focus on their constructs extending some base RFCs. Although it is not the intention of each individual draft or RFC to provide the latest and most complete/full definition of the protocol messages, in practice combining all the extensions as defined in the respective RFCs is complex, and open to interpretation.

Message rules are sometimes provided within the text, resulting in ambiguity. Moreover, the fact that extensions may be defined in parallel may be a problem. The canonical example is the case where RFC X defines construct `p ::= A` and subsequent RFC Y extends RFC X stating that object C MUST follow object A and RFC Z also extends RFC X stating that object D MUST follow object A.

This document describes current practice when implementing existing PCEP RFCs. This involves extending the existing RBNF notations using more verbose constructs where appropriate, while being semantically equivalent, in order to avoid ambiguity and to facilitate message validation.

1.1. Object Ordering Issues and Inconsistencies

The use of RBNF [[RFC5511](#)] states that the ordering of objects and constructs in an assignment is explicit, and protocol specifications MAY opt to state that ordering is only RECOMMENDED (the elements of a list of objects and constructs MAY be received in any order).

The core PCEP document [[RFC5440](#)] states in [Section 6](#) that an implementation MUST form the PCEP messages using the object ordering specified in [[RFC5440](#)].

[RFC5886] equally states that "An implementation MUST form the PCEP messages using the object ordering specified in this document."

[RFC5521] only states that "the XRO is OPTIONAL and MAY be carried within Path Computation Request (PCReq) and Path Computation Reply (PCRep) messages." and no ordering is provided. For example, it does not mention SVEC objects or rules.

[RFC5541] specifies that "the OF object MAY be carried within a PCReq message. If an objective function is to be applied to a set of synchronized path computation requests, the OF object MUST be carried just after the corresponding SVEC (Synchronization VECtor) object and MUST NOT be repeated for each elementary request. Similarly, if a metric is to be applied to a set of synchronized requests, the METRIC object MUST follow the SVEC object and MUST NOT be repeated for each elementary request. (...) An OF object specifying an objective function that applies to an individual path computation request (non-synchronized case) MUST follow the RP object for which it applies". It should be understood that this last sentence introduces ambiguity and if interpreted as the OF object MUST strictly follow (right after) the RP object, it contradicts [[RFC5440](#)] where the RP object is followed by the ENDPOINTS object.

RFCs that extend the core PCEP protocol are not consistent with the object ordering.

[RFC5541] in [section 3.2](#) is not consistent with the ordering of OF and metric-list:

```
<svec-list>      ::= <SVEC>
                   [<OF>]
                   [<metric-list>]

<request>        ::= <RP>
                   (snip)
                   [<metric-list>]
                   [<OF>]

<attribute-list> ::= [<OF>]
                   [<LSPA>]
                   [<BANDWIDTH>]
                   [<metric-list>]
```

In view of the above considerations, this document aims at providing an object ordering for PCEP messages so implementations can interoperate.

1.2. Inconsistent Naming

PCEP RFCs may use inconsistent or ambiguous naming. For example [\[RFC5440\]](#) defines the Open message as having a common header and an OPEN object, and later uses Open to refer to the object that may appear in a PCErr message.

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

<PCErr Message> ::= <Common Header>
                   (<error-obj-list> [<Open>]) | <error>
                   [<error-list>]
```

It is common that a sequence or repetition of an object OBJ is noted as obj-list. It may happen that in extensions to core documents, the naming is kept although it no longer applies to such a sequence. For example, [\[RFC5886\]](#) states:

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

and later

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


1.3. Semantics and Exclusive Rules

The current RBNF notation does not capture the semantics/intent of the messages; notably, when two options are mutually exclusive and at least one is mandatory. In most cases, this is noted as both options being optional. For example [\[RFC5440\]](#) states:

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

with this example, a message that contains a response of the form <RP><NO-PATH><ERO><..> (that is, a NO-PATH object followed by a path) is correct and successfully parsed. Likewise, a response with just an RP object is valid. Although the actual text within the RFC may state the intention and disambiguate the grammar, the RBNF notation can be improved to better capture the semantics, message structure and original intent. Such enhancements allow the automated validation of message elements.

Similarly, if the intent is to specify a rule such as metric-pce which includes a PCE-ID object followed by a PROC-TIME object and/or an OVERLOAD object, the syntax:

```
<metric-pce> ::= <PCE-ID> [<PROC-TIME>] [<OVERLOAD>]
```

allows, amongst other combinations, that neither PROC-TIME nor OVERLOAD appears, which is not the intended behavior (there should be at least one metric). The alternative

```
<metric-pce> ::= <PCE-ID> <metric-argument-list>
<metric-argument-list> ::= <metric-argument> [<metric-argument-list>]
<metric-argument> ::= <PROC-TIME> | <OVERLOAD>
```

or equivalently

```
<metric-pce> ::= <PCE-ID> (<metric-argument>...)
<metric-argument> ::= <PROC-TIME> | <OVERLOAD>
```

does not reflect that each metric-argument should appear at most once. This can be addressed verbosely:

```
<metric-pce> ::= <PCE-ID>
                ( <PROC-TIME> | <OVERLOAD> | <PROC-TIME><OVERLOAD> )

<metric-pce> ::= <PCE-ID>
                ( <PROC-TIME>[<OVERLOAD>] | [<PROC-TIME>]<OVERLOAD> )
```

Here the semantic is that we require any object of the set {PROC-TIME, OVERLOAD} to be present, and there should be at least one. Note that currently there are only a few cases where the "non-empty set" case arises.

2. Initial Considerations

This document does not modify the content of defined PCEP objects and TLVs.

This document is not normative, the normative definition is included in the existing specs. This does not preclude integration with a future revision of such documents.

3. Requirements Language

This draft does not provide any new extensions to PCEP, but it includes requirements specified by existing RFCs for illustrative purpose.

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

4. RBNF Grammars

This section provides the proposed RBNF notation for the PCEP messages. Specific constructs or grammar rules that appear in several messages or deserve special considerations are described first.

4.1. Common Constructs

4.1.1. Object Sequences

```
<of-list> ::= <OF> [<of-list>]

<metric-list> ::= <METRIC> [<metric-list>]

<vendor-info-list> ::= <VENDOR-INFORMATION> [<vendor-info-list>]

<pce-id-list> ::= <PCE-ID> [<pce-id-list>]
  -- (note: named pce-list in original)
```

4.1.2. Synchronized Vectors

SVEC tuple:

A svec-tuple is a construct that associates a SVEC object with one or more constraining objects. The selected order follows the relative order of having OF and metric-list after the SVEC object, and the name svec-list has been changed since it no longer means a list of SVEC objects.

```
<svec-tuple> ::= <SVEC>
                [<OF>]
                [<metric-list>]
                [<vendor-info-list>]
                [<GC>]
                [<XRO>]

<svec-tuple-list> ::= <svec-tuple> [<svec-tuple-list>]
```

Note that, again, as an example [[RFC7150](#)] defines:

```
<svec-list> ::= <SVEC>
                [<OF>]
                [<GC>]
                [<XRO>]
                [<metric-list>]
                [<vendor-info-list>]
                [<svec-list>]
```

There are two problems, ordering and naming. So, we use the afore defined svec-tuple-list. The construct is updated to reflect the new name and to have the same relative order in the attributes that constrain a individual request

4.1.3. Monitoring Metrics

A metric-pce-id is a rule that associates a PCE identified by its PCE-ID to a list of metric arguments.


```
<metric-pce-id> ::= <PCE-ID>
                    (<PROC-TIME> [<OVERLOAD>] |
                     [<PROC-TIME>] <OVERLOAD> )

<metric-pce-id-list> ::= <metric-pce-id> [<metric-pce-id-list>]
```

4.1.4. Monitoring Requests and Responses

See [\[RFC5886\]](#) for the definition of specific/general and in-band/out-of-band.

```
<monitoring> ::= <MONITORING> <PCC-ID-REQ>

<monitoring-request> ::= <monitoring> [<pce-id-list>]

<monitoring-response> ::= <monitoring>
                          (<specific-monitoring-metrics-list> |
                           <general-monitoring-metrics-list>)

<specific-monitoring-metrics-list> ::=
    <specific-monitoring-metrics>
    [<specific-monitoring-metrics-list>]

<general-monitoring-metrics-list> ::=
    <general-monitoring-metrics>
    [<general-monitoring-metrics-list>]

<specific-monitoring-metrics> ::=
    <RP> <monitoring-metrics>

<general-monitoring-metrics> ::=
    <monitoring-metrics>

<monitoring-metrics> ::=
    <metric-pce-id-list>
```

4.1.5. Attributes

Attributes are used to constrain a request, or to qualify a path (defined later in this document). However, it is not straightforward to define an attributes construct, since it may change for P2P or P2MP paths, and some objects (e.g. BANDWIDTH) may appear multiple times, with different semantics:

In [\[RFC5440\]](#) the BANDWIDTH object can optionally appear as a path attribute or as a request constraint.

In [\[RFC5440\]](#) the RRO object is only used in requests "The RRO is exclusively carried within a PCReq message" for reoptimization. In such contexts, the RRO and an optional BANDWIDTH objects are bound together, in the so called rro-bw-pair construct which is also an attribute.

In some contexts (stateful) paths are defined as having an optional RRO object, outside the PCEP attributes construct.

In P2MP paths, multiple RRO objects may appear.

```
-- Note: it is expected that each attribute may appear
-- just once, even if the RBNF grammar allows it. If an
-- object is allowed to repeat a list is used (e.g.
-- metric-list
```

```
-- Note: the ordering is implied by the notation below.
```

```
-- For P2P reoptimizations
<rro-bw-pair>      ::= <RRO> [<BANDWIDTH>]

-- For P2MP reoptimizations
<rro-list-bw>      ::= <rro-list> [<BANDWIDTH>]

-- Some attributes only apply to P2MP computations
<attribute>       ::=
    <CLASSTYPE> |
    <LSPA> |
    <OF> |
    <BANDWIDTH> |
    <metric-list> |
    <vendor-info-list> |
    <IRO> |
    <BNC> |           -- Only in P2MP
    <XRO> |
    <RRO> |           -- Used in Reports
    <rro-bw-pair> |    -- Only in P2P
    <rro-list-bw> |    -- Only in P2MP
    <LOAD-BALANCING> |
    <INTER-LAYER> |
    <SWITCH-LAYER> |
    <REQ-ADAP-CAP>

<attributes> ::= <attribute> [<attributes>]
```


4.1.6. Paths

A path is defined consistently as a qualified ERO (or ERO/SERO for P2MP). Similar path constructs appear, notably, in PCEP responses, in solicited/unsolicited state reports and in update requests. The following remarks apply:

The <path> construct is then defined as:

```
<ero-sero-list> ::= (<ERO> | <SERO>) [<ero-sero-list>]
```

```
<path>          ::= <ERO> [<attributes>]
```

```
<p2mp-path>     ::= <ero-sero-list> [<attributes>]
```

```
<path-list>     ::= <path>|<p2mp-path> [<path-list>]
```

4.2. PCEP Messages

4.2.1. PCEP Open Message

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

4.2.2. PCEP Keep Alive (KeepAlive) Message

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

4.2.3. PCEP Request (PCReq) Message

Note that the actual parsing depends on the content (flags) of the Request Parameters (RP) object, notably expansion and P2MP. In some cases, this may be considered redundant, e.g. the presence of a PATH_KEY object and the corresponding flag.

[Editor's note: from a notation perspective, we lack a way to express "if object a field x has value v then include object b, else include object c". RNBF extensions can be considered in future revisions of the PCEP protocol, e.g. defining new constructs :

```
(<a with x=v> <b>) | (<a with x!=v> <c>)
```

this issue is still open.]

The PCReq message contains a possibly monitored list of requests, some of which may be grouped by SVEC tuples.

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

where:

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

-- A request is either an expansion, a P2P request or a P2MP request

```
<request>         ::= <expansion> |
                    <p2p_computation> |
                    <p2mp_computation>
```

```
<expansion>       ::= <RP><PATH-KEY>
```

```
<p2p_computation> ::= <RP><ENDPOINTS>
                    [<LSP>]
                    [<attributes>]
```

```
<p2mp_computation> ::= <RP><tree-list>
                    [<attributes>]
```

-- For a P2P computation

-- in [RFC6006](#) there is a bw per tree,

-- it is intended to be an optimization for an RRO list

```
<tree>            ::= <ENDPOINTS>(<rro-bw-pair>|<rro-list-bw>)
```

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

```
<tree> ::= <ENDPOINTS> <rro-bw-pair>
```

[4.2.4.](#) PCEP Reply (PCRep) Message

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

-- Note: should clarify the use of SVEC tuple list
where

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

-- An individual response may include monitoring info

<response>  ::= <RP> [<monitoring>] [<LSP>]
                (<success> | <failure>) [<monitoring-metrics>]

-- Note: should clarify P2MP attributes. P2MP response
-- also includes endpoint-path-pair-list. TBD

<success>   ::= <path-list>

<failure>   ::= <NO-PATH> [<attributes>]
```

4.2.5. PCEP Monitoring Request (PCMonReq) Message

The PCMonReq message is defined in [[RFC5886](#)] for out-of-band monitoring requests.

[RFC5886] specifies that there is one mandatory object but the grammar also includes PCC-ID-REQ as mandatory.

[Ed note:does it make sense to include a pce-id-list and a svec-list/request-list at the same time?]

```
<PCMonReq Message> ::= <Common Header>
                        <monitoring-request>
                        [[<svec-tuple-list>] <request-list>]
```

4.2.6. PCEP Monitoring Reply (PCMonRep) Message

The PCMonRep message is defined in [[RFC5886](#)] for out-of-band monitoring responses.

[RFC5886] specifies that there is one mandatory object but the grammar also includes PCC-ID-REQ as mandatory.

[RFC5886] does not allow bundling several specific monitoring responses. A PCMonReq message causes N PCMonRep messages.

```
<PCMonRep Message> ::= <Common Header>
                        <monitoring-response>
```

4.2.7. PCEP Notify (PCNtf) Message

```
<PCNtf Message> ::= <Common Header>
                    ( <solicited-notify> | <unsolicited-notify> )
```

where

```
<solicited-notify>  ::= <request-id-list> <notification-list>
```

```
<unsolicited-notify> ::= <notification-list>
```

```
<request-id-list>   ::= <RP> [<request-id-list>]
```

```
<notification-list> ::= <NOTIFICATION> [<notification-list>]
```

4.2.8. PCEP Error (PCErr) Message

Errors can occur during PCEP handshake, or bound to one or more requests.

An error during handshake is never solicited, i.e., not associated to a list of requests.

A solicited error binds one or more Requests (RPs) to one or more PCEP-ERROR objects.

```
<PCErr Message> ::=
    <Common Header>
    ( <solicited-error> | <unsolicited-error> )

where

-- Solicited error is bound to a Request Paramters (RP) list or
-- to a Stateful Request Parameters (SRP) list

<solicited-error> ::= <request-id-list> | <stateful-request-id-list>

-- Unsolicited Error can be due to handshake or asynchronous

<unsolicited-error> ::= <handshake-error> | <pcep-error-list>

-- Handshake Error is bound to an OPEN object

<handshake-error>    ::= <pcep-error-list> <OPEN>

<request-id-list>    ::= <RP> [<request-id-list>]

<stateful-request-id-list> ::= <SRP> [<stateful-request-id-list>]

<pcep-error-list>    ::= <PCEP-ERROR> [<pcep-error-list>]
```

4.2.9. PCEP Report (PCRpt) Message

The PCRpt format is defined in [[I-D.ietf-pce-stateful-pce](#)]. Note, however, that the end-of-sync, solicited-report and unsolicited-report are introduced for convenience, and that the RRO object is already part of the attributes construct.

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

Where:

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

```
<state-report> ::=
    <end-of-sync> |
    <solicited-report> |
    <unsolicited-report>
```

-- LSP flags signal end of synchronization

```
<end-of-sync> ::= <LSP>
```

```
<solicited-report> ::= <SRP> <LSP> <path>
```

```
<unsolicited-report> ::= <LSP> <path>
```

4.2.10. PCEP Update (PCUpd) Message

As [[I-D.ietf-pce-stateful-pce](#)].

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

Where:

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

```
<update-request> ::= <SRP>
                    <LSP>
                    <path>
```

4.2.11. PCEP Initiate (PCInitiate) Message

As [[I-D.ietf-pce-pce-initiated-lsp](#)]. Note that the <path> construct is used here.

```
<PCInitiate Message> ::= <Common Header>
                           <PCE-initiated-lsp-request-list>
```

Where:

```
<PCE-initiated-lsp-request-list> ::= <PCE-initiated-lsp-request>
    [<PCE-initiated-lsp-request-list>]
```

```
-- A request can be an instantiation or a deletion. SRP / LSP
-- flags are used to select
```

```
<PCE-initiated-lsp-request> ::=
    <PCE-initiated-lsp-instantiation> |
    <PCE-initiated-lsp-deletion>)
```

```
<PCE-initiated-lsp-instantiation> ::= <SRP>
                                       <LSP>
                                       <ENDPOINTS>
                                       <path>
```

```
<PCE-initiated-lsp-deletion> ::= <SRP>
                                   <LSP>
```

5. Management Considerations

This document does not define additional management considerations.

6. Security Considerations

This document does not define additional security considerations.

7. Contributing Authors

Robert Varga
Pantheon
robert.varga@pantheon.sk

Jonathan Harwick
Metaswitch
Jonathan.Hardwick@metaswitch.com

Olivier Dugeon
Orange
olivier.dugeon@orange.com

Julien Meuric
Orange
julien.meuric@orange.com

Ina Minei
Google
inaminei@google.com

8. Acknowledgments

This work was supported in part by the PACE Support Action (<http://ict-pace.net/>) project under grant agreement number 619712.

9. Normative References

- [I-D.ietf-pce-gmpls-pcep-extensions]
Margaria, C., Dios, O., and F. Zhang, "PCEP extensions for GMPLS", [draft-ietf-pce-gmpls-pcep-extensions-09](#) (work in progress), February 2014.
- [I-D.ietf-pce-hierarchy-extensions]
Zhang, F., Zhao, Q., Dios, O., Casellas, R., and D. King, "Extensions to Path Computation Element Communication Protocol (PCEP) for Hierarchical Path Computation Elements (PCE)", [draft-ietf-pce-hierarchy-extensions-01](#) (work in progress), February 2014.
- [I-D.ietf-pce-inter-layer-ext]
Oki, E., Takeda, T., Farrel, A., and F. Zhang, "Extensions to the Path Computation Element communication Protocol (PCEP) for Inter-Layer MPLS and GMPLS Traffic Engineering", [draft-ietf-pce-inter-layer-ext-08](#) (work in progress), January 2014.
- [I-D.ietf-pce-pce-initiated-lsp]
Crabbe, E., Minei, I., Sivabalan, S., and R. Varga, "PCEP Extensions for PCE-initiated LSP Setup in a Stateful PCE Model", [draft-ietf-pce-pce-initiated-lsp-01](#) (work in progress), June 2014.
- [I-D.ietf-pce-stateful-pce]
Crabbe, E., Minei, I., Medved, J., and R. Varga, "PCEP Extensions for Stateful PCE", [draft-ietf-pce-stateful-pce-09](#) (work in progress), June 2014.

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.
- [RFC5440] Vasseur, JP. and JL. Le Roux, "Path Computation Element (PCE) Communication Protocol (PCEP)", [RFC 5440](#), March 2009.
- [RFC5455] Sivabalan, S., Parker, J., Boutros, S., and K. Kumaki, "Diffserv-Aware Class-Type Object for the Path Computation Element Communication Protocol", [RFC 5455](#), March 2009.
- [RFC5511] Farrel, A., "Routing Backus-Naur Form (RBNF): A Syntax Used to Form Encoding Rules in Various Routing Protocol Specifications", [RFC 5511](#), April 2009.
- [RFC5520] Bradford, R., Vasseur, JP., and A. Farrel, "Preserving Topology Confidentiality in Inter-Domain Path Computation Using a Path-Key-Based Mechanism", [RFC 5520](#), April 2009.
- [RFC5521] Oki, E., Takeda, T., and A. Farrel, "Extensions to the Path Computation Element Communication Protocol (PCEP) for Route Exclusions", [RFC 5521](#), April 2009.
- [RFC5541] Le Roux, JL., Vasseur, JP., and Y. Lee, "Encoding of Objective Functions in the Path Computation Element Communication Protocol (PCEP)", [RFC 5541](#), June 2009.
- [RFC5557] Lee, Y., Le Roux, JL., King, D., and E. Oki, "Path Computation Element Communication Protocol (PCEP) Requirements and Protocol Extensions in Support of Global Concurrent Optimization", [RFC 5557](#), July 2009.
- [RFC5886] Vasseur, JP., Le Roux, JL., and Y. Ikejiri, "A Set of Monitoring Tools for Path Computation Element (PCE)-Based Architecture", [RFC 5886](#), June 2010.
- [RFC6006] Zhao, Q., King, D., Verhaeghe, F., Takeda, T., Ali, Z., and J. Meuric, "Extensions to the Path Computation Element Communication Protocol (PCEP) for Point-to-Multipoint Traffic Engineering Label Switched Paths", [RFC 6006](#), September 2010.
- [RFC7150] Zhang, F. and A. Farrel, "Conveying Vendor-Specific Constraints in the Path Computation Element Communication Protocol", [RFC 7150](#), March 2014.

Authors' Addresses

Ramon Casellas (editor)
CTTC
Av. Carl Friedrich Gauss n.7
Castelldefels 08860 Barcelona
Spain

Phone: +34 93 645 29 00
Email: ramon.casellas@cttc.es

Oscar Gonzalez de Dios (editor)
Telefonica I+D
Don Ramon de la Cruz 82-84
Madrid 28045
Spain

Phone: +34913128832
Email: oscar.gonzalezdedios@telefonica.com

Adrian Farrel (editor)
Old Dog Consulting

Email: adrian@olddog.co.uk

Cyril Margaria
Juniper Networks
88 Centennial Ave, Piscataway Township
New Jersey
US

Email: cmargaria@juniper.net

Dhruv Dhody
Huawei Technologies
Leela Palace
Bangalore, Karnataka 560008
INDIA

Email: dhruv.dhody@huawei.com

Xian Zhang
Huawei Technologies

Email: zhang.xian@huawei.com