Internet Engineering Task Force
Internet-Draft
Intended status: Standards Track
Expires: January 5, 2015

S. Alvarez
S. Sivabalan
Z. Ali
Cisco Systems, Inc.
L. Tomotaki
Verizon
V. Lopez
Telefonica I+D
R. Shakir
BT

July 4, 2014

PCE Path Profiles
draft-alvarez-pce-path-profiles-03

Abstract

This document describes extensions to the Path Computation Element (PCE) Communication Protocol (PCEP) to signal path profile identifiers. A profile represents a list of path parameters or policies that a PCEP peer may invoke on a remote peer using an opaque identifier. When a path computation client (PCC) initiates a path computation request, the PCC can signal profile identifiers to invoke path parameters or policies defined on the PCE which would influence the path computation. Similarly, when a PCE initiates or updates a path, the PCE can signal profile identifiers to invoke path parameters or policies defined on the PCC which would influence the path setup.

Table of Contents

1. Introduction 2
1.1. Requirements Language 2
2. Path Profiles 3
3. Procedures 3
3.1. Capability Advertisement 3
3.2. PCC-Initiated Paths 3
3.2.1. Point-to-Point Paths 4
3.2.2. Point-to-Multipoint Paths 5
3.3. PCE-Initiated Paths 6
4. Object Extensions 7
4.1. OPEN Object 7
4.2. PATH-PROFILE Object 8
5. Error Codes for PATH-PROFILE Object 9
6. Acknowledgements 9
7. IANA Considerations 9
8. Security Considerations 10
9. References 10
9.1. Normative References 10
9.2. Informative References 11
Authors' Addresses 11

1. Introduction

1.1. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

2. Path Profiles

A path profile represents a list of path parameters or policies that a PCEP peer may invoke on a remote peer using a profile identifier. The receiving peer interprets the identifier according to a local path profile definition. The PATH-PROFILE object defined in Section 4.2 can signal one or more profile identifiers. PCEP carries profile identifiers as opaque values. PCEP peers do not exchange the details of a path profile. The PCE may be stateful or stateless.

3. Procedures

3.1. Capability Advertisement

PCEP peers advertise their capability to support path profile identifiers during the session initialization phase. They include the PATH-PROFILE-CAPABILITY TLV defined in Section 4.1 as part of the OPEN object. A PCEP peer can only signal path profile identifiers if both peers advertised this capability. A peer MUST send a PCErr message with Error-Type=4 (Not supported object), Error-value=1 (Not supported object class) and close the session if it receives a message with a path profile identifier, it supports the extensions in this document and both peers did not advertise this capability.

3.2. PCC-Initiated Paths

A PCC MAY include a PATH-PROFILE object when sending a PCReq message. The PCE uses the path profile identifiers to select path parameters or path policies to fulfill the request. The PCE MUST process the identifiers in the PATH-PROFILE object in the order received. The means by which the PCC learns about a particular path profile identifier and decides to include it in a PCReq message are outside the scope of this document. Similarly, the means by which the PCE selects a set of parameters or policies based on the profile identifier for a specific request are outside the scope of this document. The P flag of the PATH-PROFILE object MUST be set.

A PCE may receive a path computation request with one or more unexpected path profile identifiers. The PCE sends a PCErr message with Error-Type=[TBA] (PATH-PROFILE Error), Error-value=1 (Unknown path profile) if the path profile identifier is not known to the PCE. The PCE sends a PCErr message with Error-Type=[TBA] (PATH-PROFILE Error), Error-value=2 (Invalid path profile) if the PCE knows about the path profile identifier, but considers the request invalid. As an example, the profile may be invalid because of the path type, the PCEP session type or the originating PCC. The PCE sends a PCErr message with Error-Type=[TBA] (PATH-PROFILE Error), Error-value=3 (Incompatible path profiles) if two or more path profile identifiers

are incompatible. That is, they are known and valid, but can not occur simultanously. The PCEP-ERROR object SHOULD include the path profile identifiers that generated the error condition.

The PCE will determine whether to consider any additional optional objects included in a PCReq message based on policy. As illustrated in Section 3.2.1 and Section 3.2.2, the PCC MAY include other optional objects along with a PATH-PROFILE object as part of a path computation request. The PCC will use the processing-rule (P) flag in the common object header to signal whether it considers those objects mandatory or optional when the PCE performs path computation. Those objects may overlap with the path parameters that the PCE associates with the path profile identifier.

PCE policy may place different kinds of restrictions on PCReq messages that include a PATH-PROFILE object and additional parameters. A PCE MUST send an error message if it receives a request with optional objects signaled as mandatory (P flag = 1) for path computation and PCE policy does not allow such behavior from the originating PCC. In that case, the PCE sends a PCErr message with Error-Type=[TBA] (PATH-PROFILE Error), Error-value=3 (Unexpected mandatory object). If the objects are signaled as optional (P flag = 0) for path computation, the PCE will decide based on policy whether to consider them or not. When sending the PCRep message for the request, the PCE will use the ignore (I) flag in the common object header to indicate to the PCC whether an object was ignored.

3.2.1. Point-to-Point Paths

[RFC5440] defines the basic structure of a PCReq message for point-to-point paths. This documents extends the message format 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>
             [<PATH-PROFILE>]
             [<path-computation>]
```

where:

<path-computation> is the list of optional objects used for path computation as defined initially in [RFC5440] and modified in subsequent PCEP extensions.

If present in a PCReq message, the PATH-PROFILE object MUST be the first optional object in the request portion of the message.

3.2.2. Point-to-Multipoint Paths

[RFC6006] defines the basic structure of a PCReq message for point-to-multipoint paths. This documents extends the message format as follows:

```
<PCReq Message>::= <Common Header>
                   <request>
```

where:

```
   <request>::= <RP>
                <end-point-rro-pair-list>
                [<PATH-PROFILE>]
                [<OF>]
                [<LSPA>]
                [<BANDWIDTH>]
                [<metric-list>]
                [<IRO>]
                [<LOAD-BALANCING>]
```

where:

```
   <end-point-rro-pair-list>::=
                     <END-POINTS>[<RRO-List>][<BANDWIDTH>]
                     [<end-point-rro-pair-list>]

   <RRO-List>::=<RRO>[<BANDWIDTH>][<RRO-List>]
   <metric-list>::=<METRIC>[<metric-list>]
```

If present in a PCReq message, the PATH-PROFILE object MUST be the first optional object in the request portion of the message.

3.3. PCE-Initiated Paths

A PCE MAY include a PATH-PROFILE object when sending a PCInitiate message as defined in [I-D.ietf-pce-pce-initiated-lsp]. The PCC uses the path profile identifiers to select path parameters or path policies to be applied during the instantiation of the path. The PCC MUST process the identifiers in the PATH-PROFILE object in the order received. The means by which the PCE learns about a particular path profile identifier and decides to include it in a PCInitiate message are outside the scope of this document. Similarly, the means by which the PCC selects a set of parameters or policies based on the profile identifier for a specific path are outside the scope of this document.

A PCC may receive a path instantiation request with one or more unexpected path profile identifiers. The PCC sends a PCErr message with Error-Type=[TBA] (PATH-PROFILE Error), Error-value=1 (Unknown path profiles) if the path profile identifier is not known to the PCC. The PCC sends a PCErr message with Error-Type=[TBA] (PATH-PROFILE Error), Error-value=2 (Invalid path profiles) if the PCC knows about the path profile identifier, but considers the request invalid. As an example, the profile may be invalid because of the path type, the PCEP session type or the originating PCE. The PCC sends a PCErr message with Error-Type=[TBA] (PATH-PROFILE Error), Error-value=3 (Incompatible path profiles) if two or more path profile identifiers are incompatible. That is, they are known and valid, but can not occur simultanously. The PCEP-ERROR object SHOULD include the path profile identifiers that generated the error condition.

[I-D.ietf-pce-pce-initiated-lsp] defines the basic structure of a PCInitiate message. This documents extends the message format as follows:

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

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

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

  <PCE-initiated-lsp-instantiation> ::= <SRP>
                                        <LSP>
                                        <END-POINTS>
                                        <ERO>
                                        [PATH-PROFILE>
                                        [<attribute-list>]

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

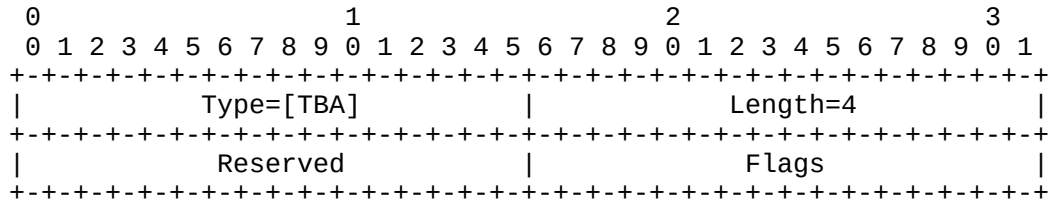
where:

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

4. Object Extensions

4.1. OPEN Object

This documents defines a new optional PATH-PROFILE-CAPABILITY TLV in the OPEN object.



PATH-PROFILE-CAPABILITY TLV

Figure 1

Reserved (16 bits):

MUST be set to zero on transmission and ignored on receipt.

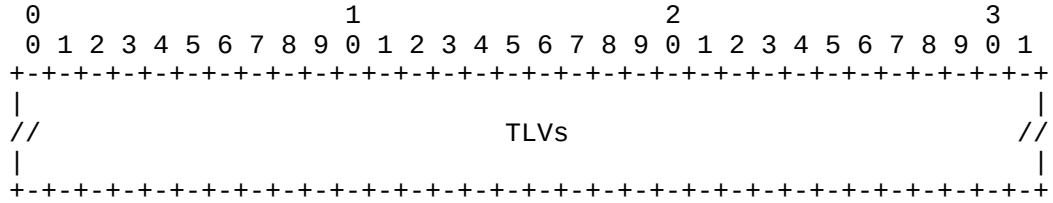
Flags (16 bits):
Unassigned bits are considered reserved. They MUST be set to zero on transmission and ignored on receipt. No flags are currently defined.

4.2. PATH-PROFILE Object

The PATH-PROFILE object may be carried in PCReq, PCInitiate and PCUpd messages.

PATH-PROFILE Object-Class is [TBA].

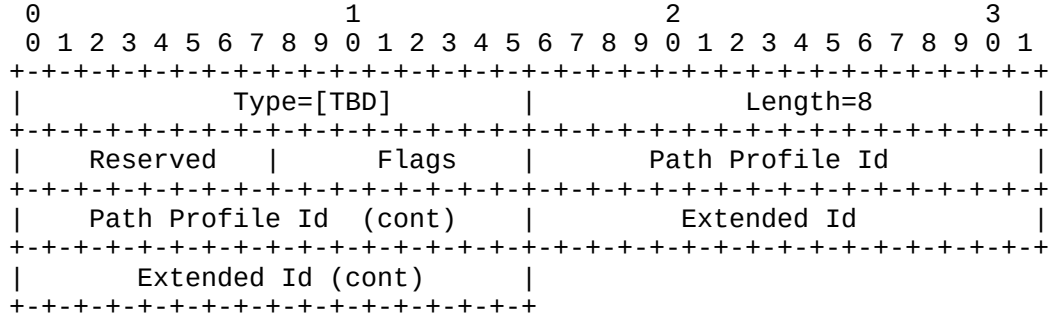
PATH-PROFILE Object-Type is 1.



PATH-PROFILE Object

Figure 2

The PATH-PROFILE object has a variable length and contains one or more PATH-PROFILE-ID TLVs.



PATH-PROFILE-ID TLV

Figure 3

Reserved (8 bits):
MUST be set to zero on transmission and ignored on receipt.

Flags (8 bits):

0x01 (X) - Extended Id Flag

It indicates to the receiver that an extended identifier associated with Path Profile Id is present.

Path Profile Id (32 bits):
(non-zero) unsigned path profile identifier.

Extended Id (32 bits):
Extended identifier associated with Path Profile Id. MUST be set to zero on transmission and ignored on receipt unless the Extended Id flag is set.

If more than one PATH-PROFILE object is present, the first one MUST be processed and subsequent objects ignored.

5. Error Codes for PATH-PROFILE Object

Error-Type	Meaning	Error-Value
<TBA>	PATH-PROFILE Error	1: Unknown path profiles
		2: Invalid path profiles
		3: Incompatible path profiles
		4: Unexpected mandatory object

6. Acknowledgements

The authors would like to thank Clarence Filsfils for his valuable comments.

7. IANA Considerations

IANA is requested to assign the following code points.

PATH-PROFILE-CAPABILITY TLV

PATH-PROFILE Object-Class

PATH-PROFILE Object-Type

PATH-PROFILE Error-Type

8. Security Considerations

TBD

9. References

9.1. Normative References

[I-D.ali-pce-remote-initiated-gmpls-lsp]
Ali, Z., Sivabalan, S., Filsfils, C., Varga, R., Lopez, V., Dios, O., and X. Zhang, "Path Computation Element Communication Protocol (PCEP) Extensions for remote-initiated GMPLS LSP Setup", draft-ali-pce-remote-initiated-gmpls-lsp-03 (work in progress), February 2014.

[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-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.

[I-D.sivabalan-pce-segment-routing]
Sivabalan, S., Medved, J., Filsfils, C., Crabbe, E., and R. Raszuk, "PCEP Extensions for Segment Routing", draft-sivabalan-pce-segment-routing-02 (work in progress), October 2013.

[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.

[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.

9.2. Informative References

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

Authors' Addresses

Santiago Alvarez
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134
USA

Email: saalvare@cisco.com

Siva Sivabalan
Cisco Systems, Inc.
2000 Innovation Drive
Kanata, ON K2K-3E8
Canada

Email: msiva@cisco.com

Zafar Ali
Cisco Systems, Inc.
2000 Innovation Drive
Kanata, ON K2K-3E8
Canada

Email: zali@cisco.com

Luis Tomotaki
Verizon
400 International
Richardson, TX 75081
US

Email: luis.tomotaki@verizon.com

Victor Lopez
Telefonica I+D
c/ Don Ramon de la Cruz 84
Madrid 28006
Spain

Email: vlopez@tid.es

Rob Shakir
BT
London
UK

Email: rob.shakir@bt.com

Internet Engineering Task Force
Internet-Draft
Intended status: Standards Track
Expires: May 11, 2015

S. Alvarez
S. Sivabalan
Z. Ali
Cisco Systems, Inc.
L. Tomotaki
Verizon
V. Lopez
Telefonica I+D
R. Shakir
BT
J. Tantsura
Ericsson

November 7, 2014

PCE Path Profiles
draft-alvarez-pce-path-profiles-04

Abstract

This document describes extensions to the Path Computation Element (PCE) Communication Protocol (PCEP) to signal path profile identifiers. A profile represents a list of path parameters or policies that a PCEP peer may invoke on a remote peer using an opaque identifier. When a path computation client (PCC) initiates a path computation request, the PCC can signal profile identifiers to invoke path parameters or policies defined on the PCE which would influence the path computation. Similarly, when a PCE initiates or updates a path, the PCE can signal profile identifiers to invoke path parameters or policies defined on the PCC which would influence the path setup.

Table of Contents

1. Introduction 2
1.1. Requirements Language 3
2. Motivation 3
3. Path Profiles 4
4. Procedures 4
4.1. Capability Advertisement 5
4.2. PCC-Initiated Paths 5
4.2.1. Point-to-Point Paths 6
4.2.2. Point-to-Multipoint Paths 7
4.3. PCE-Initiated Paths 7
5. Object Extensions 9
5.1. OPEN Object 9
5.2. PATH-PROFILE Object 9
6. Error Codes for PATH-PROFILE Object 11
7. Acknowledgements 11
8. IANA Considerations 11
9. Security Considerations 11
10. References 11
10.1. Normative References 11
10.2. Informative References 12
Authors' Addresses 12

1. Introduction

[RFC4655] specifies an architecture to address path computation requirements in large, multi-domain, multi-region and multi-layer networks. The architecture defines two main functional nodes: a path computation client (PCC) and a path computation element (PCE). It includes considerations for centralized versus distributed computation, synchronization, PCE discovery, PCE load balancing, PCE liveness detection, PCC-PCE and PCE-PCE communication, Traffic

Engineering Database (TED) synchronization, stateful versus stateless PCEs, monitoring, policy, confidentiality, and evaluation metrics.

[RFC5440] specifies the PCE Protocol (PCEP) for communications between a PCC and a PCE, or between two PCEs. [I-D.ietf-pce-stateful-pce] specifies PCEP extensions for stateful control of LSPs including LSP state synchronization between PCCs and PCEs, delegation of LSP control to PCEs, and PCE control of timing and sequence of path computations within and across PCEP sessions. [I-D.ietf-pce-pce-initiated-lsp] introduces PCEP extensions to allow a stateful PCE to set up, maintain and tear down LSPs without the need for local configuration on the PCC.

This document describes PCEP extensions to signal path profile identifiers. A profile represents a list of path parameters or policies that a PCEP peer may invoke on a remote peer using an opaque identifier. The PCE may be stateful or stateless. When a path computation client (PCC) initiates a path computation request, the PCC can signal profile identifiers to invoke path parameters or policies defined on the PCE which would influence the path computation. Similarly, when a PCE initiates or updates a path, the PCE can signal profile identifiers to invoke path parameters or policies defined on the PCC which would influence the path setup.

1.1. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

2. Motivation

PCEP peers may need to specify request-specific parameters and policies without signaling them explicitly. The signaling of one or more path profile identifiers allows peers to make use of opaque identifiers to implicitly communicate such information. An important characteristic of this approach is that the transmitting peer does not need to know the specifics of the profiles and can invoke new functional enhancements on the receiving peer without requiring changes to its implementation.

There are multiple reasons why the explicit communication of some parameters and policies may not be possible or desirable. The transmitting peer may not implement the protocol extensions required or such extensions do not exist. The defintion of some parameters and policies may be located on the receiving peer as a matter of operational preference. The parameters and policies may not be directly related to computation or instantiation of the path, but may

be related to other functionality associated with the path (e.g. traffic steering, accounting, monitoring, etc).

A PCC may use path profiles in numerous scenarios when requesting a path computation. For example, a PCE may be provisioned with a policy profile that enforces path diversity, elaborate dependencies between paths or time-based behaviors. Alternative, a PCE may be provisioned with a set of configuration profiles that define path computation parameters. These policies and configuration parameters can be centrally managed on the PCE and made effective across multiple PCCs. A PCC does not need to know the specifics of the profiles and is able to invoke new PCE functionality without changes to its implementation.

Similarly, a PCE may use path profiles in numerous scenarios when initiating or updating a path on a PCC. A PCC may be provisioned with a set of configuration and policy profiles that may be applied to paths. For example, those profiles could specify a policy to steer traffic into the path or configuration parameters related to traffic accounting, event logging, path monitoring, etc. A PCE can invoke these policies and configuration, so the PCC can establish a more completly configured path. A PCE does not need to know the specifics of the profiles and is able to invoke new PCC functionality without changes to its implementation.

3. Path Profiles

A path profile represents a list of path parameters or policies that a PCEP peer may invoke on a remote peer using a profile identifier. The receiving peer interprets the identifier according to a local path profile definition. The PATH-PROFILE object defined in Section 5.2 can signal one or more profile identifiers. PCEP carries profile identifiers as opaque values. PCEP peers do not exchange the details of a path profile.

Regarding policies in particular, the PCE path profile specifications in this document enable a new type of policy realization in the PCE architecture. They define an approach where request-specific policies may be communicated implicitly to achieve some level of coordination of policy between PCEP peers. [RFC4655] defines the current policy realization options and policy types in the PCE architecture.

4. Procedures

4.1. Capability Advertisement

PCEP peers advertise their capability to support path profile identifiers during the session initialization phase. They include the PATH-PROFILE-CAPABILITY TLV defined in Section 5.1 as part of the OPEN object. A PCEP peer can only signal path profile identifiers if both peers advertised this capability. A peer MUST send a PCErr message with Error-Type=4 (Not supported object), Error-value=1 (Not supported object class) and close the session if it receives a message with a path profile identifier, it supports the extensions in this document and both peers did not advertise this capability.

4.2. PCC-Initiated Paths

A PCC MAY include a PATH-PROFILE object when sending a PCReq message. The PCE uses the path profile identifiers to select path parameters or path policies to fulfill the request. The PCE MUST process the identifiers in the PATH-PROFILE object in the order received. The means by which the PCC learns about a particular path profile identifier and decides to include it in a PCReq message are outside the scope of this document. Similarly, the means by which the PCE selects a set of parameters or policies based on the profile identifier for a specific request are outside the scope of this document. The P flag of the PATH-PROFILE object MUST be set.

A PCE may receive a path computation request with one or more unexpected path profile identifiers. The PCE sends a PCErr message with Error-Type=[TBA] (PATH-PROFILE Error), Error-value=1 (Unknown path profile) if the path profile identifier is not known to the PCE. The PCE sends a PCErr message with Error-Type=[TBA] (PATH-PROFILE Error), Error-value=2 (Invalid path profile) if the PCE knows about the path profile identifier, but considers the request invalid. As an example, the profile may be invalid because of the path type, the PCEP session type or the originating PCC. The PCE sends a PCErr message with Error-Type=[TBA] (PATH-PROFILE Error), Error-value=3 (Incompatible path profiles) if two or more path profile identifiers are incompatible. That is, they are known and valid, but can not occur simultanously. The PCEP-ERROR object SHOULD include the path profile identifiers that generated the error condition.

The PCE will determine whether to consider any additional optional objects included in a PCReq message based on policy. As illustrated in Section 4.2.1 and Section 4.2.2, the PCC MAY include other optional objects along with a PATH-PROFILE object as part of a path computation request. The PCC will use the processing-rule (P) flag in the common object header to signal whether it considers those objects mandatory or optional when the PCE performs path computation.

Those objects may overlap with the path parameters that the PCE associates with the path profile identifier.

PCE policy may place different kinds of restrictions on PCReq messages that include a PATH-PROFILE object and additional parameters. A PCE MUST send an error message if it receives a request with optional objects signaled as mandatory (P flag = 1) for path computation and PCE policy does not allow such behavior from the originating PCC. In that case, the PCE sends a PCErr message with Error-Type=[TBA] (PATH-PROFILE Error), Error-value=3 (Unexpected mandatory object). If the objects are signaled as optional (P flag = 0) for path computation, the PCE will decide based on policy whether to consider them or not. When sending the PCRep message for the request, the PCE will use the ignore (I) flag in the common object header to indicate to the PCC whether an object was ignored.

4.2.1. Point-to-Point Paths

[RFC5440] defines the basic structure of a PCReq message for point-to-point paths. This documents extends the message format 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>
             [<PATH-PROFILE>]
             [<path-computation>]
```

where:

<path-computation> is the list of optional objects used for path computation as defined initially in [RFC5440] and modified in subsequent PCEP extensions.

If present in a PCReq message, the PATH-PROFILE object MUST be the first optional object in the request portion of the message.

4.2.2. Point-to-Multipoint Paths

[RFC6006] defines the basic structure of a PCReq message for point-to-multipoint paths. This documents extends the message format as follows:

```
<PCReq Message>::= <Common Header>
                   <request>
```

where:

```
   <request>::= <RP>
                <end-point-rro-pair-list>
                [<PATH-PROFILE>]
                [<OF>]
                [<LSPA>]
                [<BANDWIDTH>]
                [<metric-list>]
                [<IRO>]
                [<LOAD-BALANCING>]
```

where:

```
   <end-point-rro-pair-list>::=
                     <END-POINTS>[<RRO-List>][<BANDWIDTH>]
                     [<end-point-rro-pair-list>]

   <RRO-List>::=<RRO>[<BANDWIDTH>][<RRO-List>]
   <metric-list>::=<METRIC>[<metric-list>]
```

If present in a PCReq message, the PATH-PROFILE object MUST be the first optional object in the request portion of the message.

4.3. PCE-Initiated Paths

A PCE MAY include a PATH-PROFILE object when sending a PCInitiate message as defined in [I-D.ietf-pce-pce-initiated-lsp]. The PCC uses the path profile identifiers to select path parameters or path policies to be applied during the instantiation of the path. The PCC MUST process the identifiers in the PATH-PROFILE object in the order received. The means by which the PCE learns about a particular path profile identifier and decides to include it in a PCInitiate message are outside the scope of this document. Similarly, the means by which the PCC selects a set of parameters or policies based on the

profile identifier for a specific path are outside the scope of this document.

A PCC may receive a path instantiation request with one or more unexpected path profile identifiers. The PCC sends a PCErr message with Error-Type=[TBA] (PATH-PROFILE Error), Error-value=1 (Unknown path profiles) if the path profile identifier is not known to the PCC. The PCC sends a PCErr message with Error-Type=[TBA] (PATH-PROFILE Error), Error-value=2 (Invalid path profiles) if the PCC knows about the path profile identifier, but considers the request invalid. As an example, the profile may be invalid because of the path type, the PCEP session type or the originating PCE. The PCC sends a PCErr message with Error-Type=[TBA] (PATH-PROFILE Error), Error-value=3 (Incompatible path profiles) if two or more path profile identifiers are incompatible. That is, they are known and valid, but can not occur simultanously. The PCEP-ERROR object SHOULD include the path profile identifiers that generated the error condition.

[I-D.ietf-pce-pce-initiated-lsp] defines the basic structure of a PCInitiate message. This documents extends the message format as follows:

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

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

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

  <PCE-initiated-lsp-instantiation> ::= <SRP>
                                        <LSP>
                                        <END-POINTS>
                                        <ERO>
                                        [PATH-PROFILE>
                                        [<attribute-list>]

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

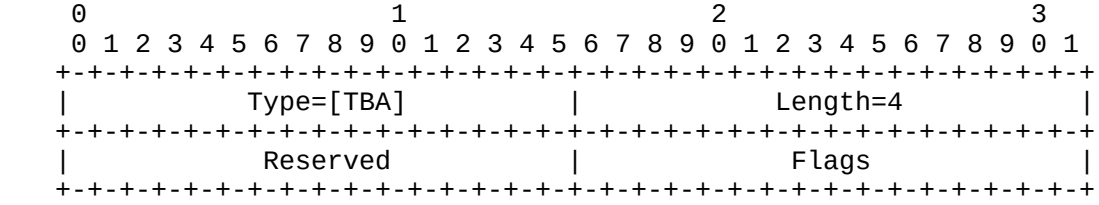
where:

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

5. Object Extensions

5.1. OPEN Object

This documents defines a new optional PATH-PROFILE-CAPABILITY TLV in the OPEN object.



PATH-PROFILE-CAPABILITY TLV

Figure 1

Reserved (16 bits):
MUST be set to zero on transmission and ignored on receipt.

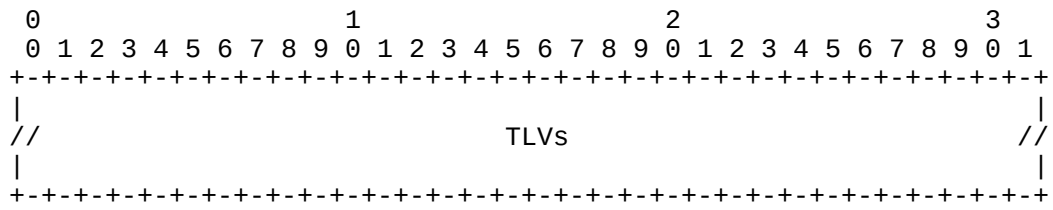
Flags (16 bits):
Unassigned bits are considered reserved. They MUST be set to zero on transmission and ignored on receipt. No flags are currently defined.

5.2. PATH-PROFILE Object

The PATH-PROFILE object may be carried in PCReq, PCInitiate and PCUpd messages.

PATH-PROFILE Object-Class is [TBA].

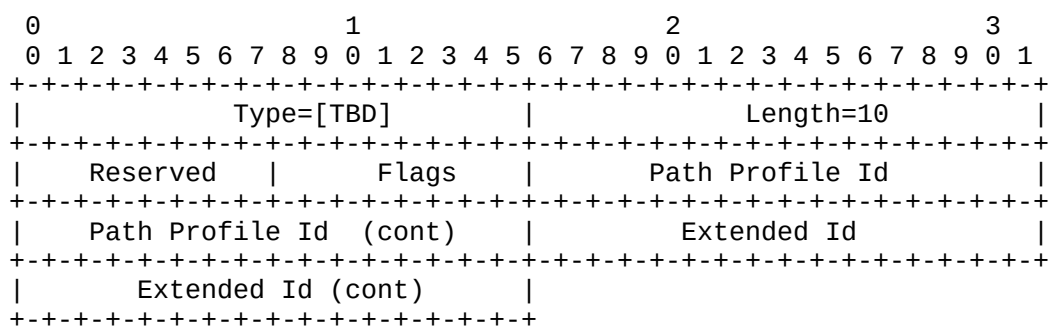
PATH-PROFILE Object-Type is 1.



PATH-PROFILE Object

Figure 2

The PATH-PROFILE object has a variable length and contains one or more PATH-PROFILE-ID TLVs.



PATH-PROFILE-ID TLV

Figure 3

Reserved (8 bits):
MUST be set to zero on transmission and ignored on receipt.

Flags (8 bits):

0x01 (X) - Extended Id Flag

It indicates to the receiver that an extended identifier associated with Path Profile Id is present.

Path Profile Id (32 bits):
(non-zero) unsigned path profile identifier.

Extended Id (32 bits):

Extended identifier associated with Path Profile Id. MUST be set to zero on transmission and ignored on receipt unless the Extended Id flag is set.

If more than one PATH-PROFILE object is present, the first one MUST be processed and subsequent objects ignored.

6. Error Codes for PATH-PROFILE Object

Error-Type	Meaning	Error-Value
<TBA>	PATH-PROFILE Error	1: Unknown path profiles
		2: Invalid path profiles
		3: Incompatible path profiles
		4: Unexpected mandatory object

7. Acknowledgements

The authors would like to thank Clarence Filsfils for his valuable comments.

8. IANA Considerations

IANA is requested to assign the following code points.

PATH-PROFILE-CAPABILITY TLV

PATH-PROFILE Object-Class

PATH-PROFILE Object-Type

PATH-PROFILE Error-Type

9. Security Considerations

This document does not introduce new security concerns. The security considerations in [RFC4655], [I-D.ietf-pce-stateful-pce] and [I-D.ietf-pce-pce-initiated-lsp] remain relevant.

10. References

10.1. Normative References

[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-00 (work in progress), December 2013.

[I-D.ietf-pce-stateful-pce]
Crabbe, E., Medved, J., Minei, I., and R. Varga, "PCEP Extensions for Stateful PCE", draft-ietf-pce-stateful-pce-08 (work in progress), February 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.

[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.

10.2. Informative References

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

Authors' Addresses

Santiago Alvarez
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134
USA

Email: saalvare@cisco.com

Siva Sivabalan
Cisco Systems, Inc.
2000 Innovation Drive
Kanata, ON K2K-3E8
Canada

Email: msiva@cisco.com

Zafar Ali
Cisco Systems, Inc.
2000 Innovation Drive
Kanata, ON K2K-3E8
Canada

Email: zali@cisco.com

Luis Tomotaki
Verizon
400 International
Richardson, TX 75081
US

Email: luis.tomotaki@verizon.com

Victor Lopez
Telefonica I+D
c/ Don Ramon de la Cruz 84
Madrid 28006
Spain

Email: vlopez@tid.es

Rob Shakir
BT
London
UK

Email: rob.shakir@bt.com

Jeff Tantsura
Ericsson
300 Holger Way
San Jose, CA 95134
US

Email: Jeff.Tantsura@ericsson.com

PCE Working Group
Internet-Draft
Intended status: Informational
Expires: April 26, 2015

D. Dhody
Huawei Technologies

October 23, 2014

Informal Survey into Include Route Object (IRO) Implementations in Path Computation Element communication Protocol (PCEP) draft-dhody-pce-iro-survey-01

Abstract

During discussions of a document to provide a standard representation and encoding of Domain-Sequence within the Path Computation Element (PCE) communication Protocol (PCEP) for communications between a Path Computation Client (PCC) and a PCE, or between two PCEs. It was determined that there was a need for clarification with respect to the ordered nature of the Include Route Object (IRO).

Since there was a proposal to have a new IRO type with ordering, as well as handling of Loose bit, it felt necessary to conduct a survey of the existing and planned implementations.

This document summarizes the survey questions and captures the results. Some conclusions are also presented.

This survey was informal and conducted via email. Responses were collected and anonymized by the PCE working group chairs.

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 April 26, 2015.

Table of Contents

1. Introduction 2
2. Survey Details 3
2.1. Survey Preamble 3
2.2. Survey Questions 3
3. Respondents 5
4. Results 5
5. Conclusions 7
5.1. Proposed Action 7
6. Security Considerations 8
7. IANA Considerations 8
8. Acknowledgments 8
9. References 8
9.1. Normative References 8
9.2. Informative References 8
Appendix A. Contributor Addresses 10

1. Introduction

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

[RFC5440] defines the Include Route Object (IRO) to specify that the computed path must traverse a set of specified network elements. The specification did not mention if IRO is an ordered or un-ordered list of sub-objects. It mentioned that the L bit (loose) has no meaning within an IRO.

[RFC5441] suggested the use of IRO to indicate the sequence of domains to be traversed during inter-domain path computation.

During discussion of [I-D.ietf-pce-pcep-domain-sequence] it was proposed to have a new IRO type with ordered nature, as well as handling of L bit.

In order to discover the current state of affairs amongst implementations a survey of the existing and planned implementations was conducted. This survey was informal and conducted via email. Responses were collected and anonymized by the PCE working group chair.

This document summarizes the survey questions and captures the results. Some conclusions are also presented.

2. Survey Details

2.1. Survey Preamble

The survey was introduced with the following text.

Hi PCE WG.

To address the issues associated with draft-ietf-pce-pcep-domain-sequence and "Include Route Object" in PCEP, Dhruv has proposed to start a small survey. If implementers agree that we need to clarify this, they would be much welcome to answer the attached questions.

Dhruv will process the results, but to improve confidentiality, answers may be sent privately to the chairs.

Thanks,

JP & Julien, on behalf of Dhruv

2.2. Survey Questions

The following survey questions were asked, the survey questionnaire is listed verbatim below.

> During discussion of draft-ietf-pce-pcep-domain-sequence-05, it has been noted that RFC 5440 does not define whether the sub-objects in the IRO are ordered or unordered.
>
> We would like to do an informal and *confidential* survey of current implementations, to help clarify this situation.
>
> 1. IRO Encoding

a. Does your implementation construct IRO?

b. If your answer to part (a) is Yes, does your implementation construct the IRO as an ordered list always, sometimes or never?

c. If your answer to part (b) is Sometimes, what criteria do you use to decide if the IRO is an ordered or unordered list?

d. If your answer to part (b) is Always or Sometimes, does your implementation construct the IRO as a sequence of strict hops or as a sequence of loose hops?

2. IRO Decoding

a. Does your implementation decode IRO?

b. If your answer to part (a) is Yes, does your implementation interpret the decoded IRO as an ordered list always, sometimes or never?

c. If your answer to part (b) is Sometimes, what criteria do you use to decide if the IRO is an ordered or unordered list?

d. If your answer to part (b) is Always or Sometimes, does your implementation interpret the IRO as a sequence of strict hops or as a sequence of loose hops?

3. Impact

a. Will there be an impact to your implementation if RFC 5440 is updated to state that the IRO is an ordered list?

b. Will there be an impact to your implementation if RFC 5440 is updated to state that the IRO is an unordered list?

c. If RFC 5440 is updated to state that the IRO is an ordered list, will there be an impact to your implementation if RFC 5440 is also updated to allow IRO sub-objects to use the loose bit (L-bit)?

4. Respondents

a. Are you a Vendor/Research Lab/Software House/Other (please specify)?

b. If your answer to part (a) is Vendor, is the implementation for a shipping product, product under development or a prototype?

3. Respondents

Total 9 responses were received from vendors, software houses, and research labs. Vendors made responses for their current shipping products as well as products that they currently have under development.

- Total Number of Respondents: 9
  - Vendors: 4
    - Shipping Product: 1
    - Product Under Development: 1
    - Prototype: 1
    - Unknown: 1
  - Software House: 1
  - Research Labs: 2
    - Operator’s Research Facility: 1
  - Open Source: 1
    - Shipped Release: 1
  - Others (or Unknown): 1

4. Results

	Questions	Response
1a	Does your implementation construct IRO?	yes (9)
1b	Does your implementation construct the IRO as an ordered list always, sometimes or never?	always (8), never (1)
1c	What criteria do you use to decide if the IRO is an ordered or unordered list?	none (9)
1d	Does your implementation construct the IRO as a sequence of strict hops or as a sequence of loose hops?	strict (5), loose (2), both (2)

Table 1: IRO Encoding

Regarding IRO encodings, most implementations construct IRO in an ordered fashion and consider it to be an ordered list. More than half of implementation under survey consider the IRO sub-objects as strict hops, others consider loose or support both.

	Questions	Response
2a	Does your implementation decode IRO?	yes (9)
2b	Does your implementation interpret the decoded IRO as an ordered list always, sometimes or never?	always (7), sometimes (1), never (1)
2c	What criteria do you use to decide if the IRO is an ordered or unordered list?	none (9)
2d	Does your implementation interpret the IRO as a sequence of strict hops or as a sequence of loose hops?	strict (5), loose (2), both (2)

Table 2: IRO Decoding

Regarding IRO decoding, most implementations interpret IRO as an ordered list. More than half of implementation under survey consider the IRO sub-objects as strict hops, others consider loose or support both.

	Questions	Response
3a	Will there be an impact to your implementation if [RFC5440] is updated to state that the IRO is an ordered list?	none (9)
3b	Will there be an impact to your implementation if [RFC5440] is updated to state that the IRO is an unordered list?	yes (5), no (4)
3c	will there be an impact to your implementation if [RFC5440] is also updated to allow IRO sub-objects to use the loose bit (L-bit)?	none (5), yes(1), yes-but-small (3)

Table 3: Impact

It is interesting to note that most implementation that responded to the survey finds that there is no impact to their existing or under-development implementation if [RFC5440] is updated to state that the IRO as an ordered list. Further most implementations find that support for loose bit (L-bit) for IRO has minimal or no impact on their implementation.

5. Conclusions

The results shown in this survey seems to suggest that most implementations would be fine with updating [RFC5440] to specify IRO as an ordered list with no impact on the shipping or under-development products. It is also the conclusion of this survey to suggest that it would be helpful to update [RFC5440] to enable support for loose bit (L-bit) such that both strict and loose hops could be supported in the IRO.

5.1. Proposed Action

The proposed action is as follows:

- Update [RFC5440] to specify IRO as an ordered list.
- Update [RFC5440] to specify support for loose bit (L-bit) for IRO.
- Remove the new IRO option from draft-ietf-pce-pcep-domain-sequence-05.

An update to IRO specification are proposed in [I-D.dhody-pce-iro-update].

6. Security Considerations

This survey defines no protocols or procedures and so includes no security-related protocol changes. Clarification in the supported IRO ordering or loose bit handling will not have any negative security impact. The survey responses in this document were collected by email and that email was not authenticated, although responses were sent to the respondents that might have triggered alarms if the responses were spoofed. Spoofed or malicious responses could represent an attack on the IETF process and so this survey should be treated with some caution where there is reason to suspect such an attack. Further, this survey was compiled and anonymized by the working group chairs.

7. IANA Considerations

This informational document makes no requests to IANA for action.

8. Acknowledgments

A special thanks to author of [I-D.farrel-ccamp-ero-survey], this document borrow some of the structure and text from it.

9. References

9.1. Normative References

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

9.2. Informative References

[RFC5441] Vasseur, JP., Zhang, R., Bitar, N., and JL. Le Roux, "A Backward-Recursive PCE-Based Computation (BRPC) Procedure to Compute Shortest Constrained Inter-Domain Traffic Engineering Label Switched Paths", RFC 5441, April 2009.

[I-D.ietf-pce-pcep-domain-sequence] Dhody, D., Palle, U., and R. Casellas, "Standard Representation Of Domain-Sequence", draft-ietf-pce-pcep-domain-sequence-06 (work in progress), October 2014.

[I-D.farrel-ccamp-ero-survey]
Farrel, A., "Informal Survey into Explicit Route Object Implementations in Generalized Multiprotocol Labels Switching Signaling Implementations", draft-farrel-ccamp-ero-survey-00 (work in progress), May 2006.

[I-D.dhody-pce-iro-update]
Dhody, D., "Update to Include Route Object (IRO) specification in Path Computation Element communication Protocol (PCEP)", draft-dhody-pce-iro-update-00 (work in progress), October 2014.

Appendix A. Contributor Addresses

Julien Meuric
Orange

EMail: julien.meuric@orange.com

Jean Philippe Vasseur
Cisco Systems, Inc.

EMail: jpv@cisco.com

Jonathan Hardwick
Metaswitch
100 Church Street
Enfield EN2 6BQ
UK

EMail: jonathan.hardwick@metaswitch.com

Author’s Address

Dhruv Dhody
Huawei Technologies
Leela Palace
Bangalore, Karnataka 560008
INDIA

EMail: dhruv.ietf@gmail.com

PCE Working Group
Internet-Draft
Updates: 5440 (if approved)
Intended status: Standards Track
Expires: April 27, 2015

D. Dhody
Huawei Technologies

October 24, 2014

Update to Include Route Object (IRO) specification in Path Computation Element communication Protocol (PCEP) draft-dhody-pce-iro-update-01

Abstract

During discussions of a document to provide a standard representation and encoding of Domain-Sequence within the Path Computation Element (PCE) communication Protocol (PCEP) for communications between a Path Computation Client (PCC) and a PCE, or between two PCEs. It was determined that there was a need for clarification with respect to the ordered nature of the Include Route Object (IRO).

An informal survey was conducted to determine the state of current and planned implementation with respect to IRO ordering and handling of Loose bit.

This document updates the IRO specification based on the survey conclusion and recommendation.

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 April 27, 2015.

Table of Contents

1. Introduction 2
1.1. Requirements Language 3
2. Update in IRO specification 3
3. Other Considerations 4
4. Security Considerations 4
5. IANA Considerations 4
6. Acknowledgments 4
7. References 4
7.1. Normative References 4
7.2. Informative References 5
Appendix A. Details of IRO survey 6

1. Introduction

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

[RFC5440] defines the Include Route Object (IRO) to specify that the computed path must traverse a set of specified network elements. The specification did not mention if IRO is an ordered or un-ordered list of sub-objects. It mentioned that the L bit (loose) has no meaning within an IRO.

[RFC5441] suggested the use of IRO to indicate the sequence of domains to be traversed during inter-domain path computation.

During discussion of [I-D.ietf-pce-pcep-domain-sequence] it was proposed to have a new IRO type with ordered nature, as well as handling of L bit.

In order to discover the current state of affairs amongst implementations a survey of the existing and planned implementations was conducted. This survey [I-D.dhody-pce-iro-survey] was informal and conducted via email. Responses were collected and anonymized by the PCE working group chair.

This document updates the IRO specifications in [RFC5440] as per the conclusion and action points presented in [I-D.dhody-pce-iro-survey].

1.1. Requirements Language

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

2. Update in IRO specification

[RFC5440] describes IRO as an optional object used to specify that the computed path MUST traverse a set of specified network elements. It further state that the L bit of such sub-object has no meaning within an IRO. It did not mention if IRO is an ordered or un-ordered list of sub-objects.

A survey of the existing and planned implementations was conducted in order to discover the current state of affairs amongst implementations. [I-D.dhody-pce-iro-survey] describe the questionnaire, results and presents some conclusions and proposed action items. More details in Appendix A.

The survey suggest that most implementations construct or interpret IRO in an ordered fashion and consider it to be an ordered list. More than half of implementation under survey consider the IRO sub-objects as strict hops, others consider loose or support both. The results shown in this survey seems to suggest that most implementations would be fine with updating [RFC5440] to specify IRO as an ordered list as well as to enable support for loose bit (L-bit) such that both strict and loose hops could be supported in the IRO.

This document thus updates [RFC5440] regarding the IRO specification making IRO as an ordered list as well as support for loose bit (L-bit).

The content of an IRO object is an ordered list of subobjects representing a series of abstract nodes. An abstract node may just be a simple abstract node comprising one node or a group of nodes for example an AS (comprising of multiple hops within the AS) (refer [RFC3209] for details). Further each subobject has an attribute called ’L-bit’, which is set if the subobject represents a loose hop.

If the bit is not set, the subobject represents a strict hop. The interpretation of L-bit is as per section 4.3.3.1 of [RFC3209].

3. Other Considerations

Based on the survey, it should be noted that most implementation already support the update in the IRO specification as per this document. The other implementation are expected to make an update to the IRO procedures.

4. Security Considerations

This update in IRO specification does not introduce any new security considerations, apart from those mentioned in [RFC5440]. Clarification in the supported IRO ordering or Loose bit handling will not have any negative security impact.

It is worth noting that PCEP operates over TCP. An analysis of the security issues for routing protocols that use TCP (including PCEP) is provided in [RFC6952] while [I-D.ietf-pce-pceps] discusses an experimental approach to provide secure transport for PCEP.

5. IANA Considerations

This informational document makes no requests to IANA for action.

6. Acknowledgments

A special thanks to PCE chairs for guidance regarding this work.

Thanks to Francesco Fondelli for his suggestions in clarifying the L bit.

7. References

7.1. Normative References

[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.

7.2. Informative References

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

[RFC5441] Vasseur, JP., Zhang, R., Bitar, N., and JL. Le Roux, "A Backward-Recursive PCE-Based Computation (BRPC) Procedure to Compute Shortest Constrained Inter-Domain Traffic Engineering Label Switched Paths", RFC 5441, April 2009.

[RFC6952] Jethanandani, M., Patel, K., and L. Zheng, "Analysis of BGP, LDP, PCEP, and MSDP Issues According to the Keying and Authentication for Routing Protocols (KARP) Design Guide", RFC 6952, May 2013.

[I-D.ietf-pce-pcep-domain-sequence] Dhody, D., Palle, U., and R. Casellas, "Standard Representation Of Domain-Sequence", draft-ietf-pce-pcep-domain-sequence-06 (work in progress), October 2014.

[I-D.ietf-pce-pceps] Lopez, D., Dios, O., Wu, W., and D. Dhody, "Secure Transport for PCEP", draft-ietf-pce-pceps-02 (work in progress), October 2014.

[I-D.dhody-pce-iro-survey] Dhody, D., "Informal Survey into Include Route Object (IRO) Implementations in Path Computation Element communication Protocol (PCEP)", draft-dhody-pce-iro-survey-01 (work in progress), October 2014.

Appendix A. Details of IRO survey

During discussions of this document to provide a standard representation and encoding of Domain-Sequence within PCEP. It was determined that there was a need for clarification with respect to the ordered nature of the IRO.

Since there was a proposal to have a new IRO type with ordering, as well as handling of Loose bit, in an earlier version of this document (refer - draft-ietf-pce-pcep-domain-sequence-05), it was deemed necessary to conduct a survey of the existing and planned implementations. An informal survey was conducted via email. Responses were collected and anonymized by the PCE working group chairs.

[I-D.dhody-pce-iro-survey] summarizes the survey questions and captures the results. It further list some conclusions and action points.

This document was considered as one possible venue to handle the proposed action points.

Author's Address

Dhruv Dhody
Huawei Technologies
Leela Palace
Bangalore, Karnataka 560008
INDIA

EMail: dhruv.ietf@gmail.com

PCE Working Group
Internet-Draft
Updates: 5440 (if approved)
Intended status: Standards Track
Expires: July 3, 2015

D. Dhody
Huawei Technologies

December 30, 2014

Update to Include Route Object (IRO) specification in Path Computation Element communication Protocol (PCEP) draft-dhody-pce-iro-update-02

Abstract

During discussions of a document to provide a standard representation and encoding of Domain-Sequence within the Path Computation Element (PCE) communication Protocol (PCEP) for communications between a Path Computation Client (PCC) and a PCE, or between two PCEs. It was determined that there was a need for clarification with respect to the ordered nature of the Include Route Object (IRO).

An informal survey was conducted to determine the state of current and planned implementation with respect to IRO ordering and handling of Loose bit (L bit).

This document updates the IRO specification based on the survey conclusion and recommendation.

Status of This Memo

Table of Contents

1. Introduction 2
1.1. Requirements Language 3
2. Update in IRO specification 3
3. Other Considerations 4
4. Security Considerations 4
5. IANA Considerations 4
6. Acknowledgments 4
7. References 4
7.1. Normative References 4
7.2. Informative References 5
Appendix A. Details of IRO survey 6

1. Introduction

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

[RFC5440] defines the Include Route Object (IRO) to specify that the computed path must traverse a set of specified network elements. The specification did not mention if IRO is an ordered or un-ordered list of sub-objects. It mentioned that the Loose bit (L bit) has no meaning within an IRO.

[RFC5441] suggested the use of IRO to indicate the sequence of domains to be traversed during inter-domain path computation.

During discussion of [I-D.ietf-pce-pcep-domain-sequence] it was proposed to have a new IRO type with ordered nature, as well as handling of Loose bit (L bit).

In order to discover the current state of affairs amongst implementations a survey of the existing and planned implementations was conducted. This survey [I-D.dhody-pce-iro-survey] was informal and conducted via email. Responses were collected and anonymized by the PCE working group chair.

This document updates the IRO specifications in [RFC5440] as per the conclusion and action points presented in [I-D.dhody-pce-iro-survey].

1.1. Requirements Language

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

2. Update in IRO specification

[RFC5440] describes IRO as an optional object used to specify that the computed path MUST traverse a set of specified network elements. It further state that the Loose bit (L bit) of such sub-object has no meaning within an IRO. It did not mention if IRO is an ordered or un-ordered list of sub-objects.

A survey of the existing and planned implementations was conducted in order to discover the current state of affairs amongst implementations. [I-D.dhody-pce-iro-survey] describe the questionnaire, results and presents some conclusions and proposed action items. More details in Appendix A.

The survey suggest that most implementations construct or interpret IRO in an ordered fashion and consider it to be an ordered list. More than half of implementation under survey consider the IRO sub-objects as strict hops, others consider loose or support both. The results shown in this survey seems to suggest that most implementations would be fine with updating [RFC5440] to specify IRO as an ordered list as well as to enable support for Loose bit (L bit) such that both strict and loose hops could be supported in the IRO.

This document thus updates [RFC5440] regarding the IRO specification making IRO as an ordered list as well as support for Loose bit (L bit).

The content of an IRO object is an ordered list of subobjects representing a series of abstract nodes. An abstract node may just be a simple abstract node comprising one node or a group of nodes for example an AS (comprising of multiple hops within the AS) (refer [RFC3209] for details). Further each subobject has an attribute called ’L bit’, which is set if the subobject represents a loose hop.

If the bit is not set, the subobject represents a strict hop. The interpretation of Loose bit (L bit) is as per section 4.3.3.1 of [RFC3209].

3. Other Considerations

Based on the survey, it should be noted that most implementation already support the update in the IRO specification as per this document. The other implementation are expected to make an update to the IRO procedures.

4. Security Considerations

This update in IRO specification does not introduce any new security considerations, apart from those mentioned in [RFC5440]. Clarification in the supported IRO ordering or Loose bit handling will not have any negative security impact.

It is worth noting that PCEP operates over TCP. An analysis of the security issues for routing protocols that use TCP (including PCEP) is provided in [RFC6952], while [I-D.ietf-pce-pceps] discusses an experimental approach to provide secure transport for PCEP.

5. IANA Considerations

This informational document makes no requests to IANA for action.

6. Acknowledgments

A special thanks to PCE chairs for guidance regarding this work.

Thanks to Francesco Fondelli for his suggestions in clarifying the L bit usage.

7. References

7.1. Normative References

[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.

7.2. Informative References

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

[RFC5441] Vasseur, JP., Zhang, R., Bitar, N., and JL. Le Roux, "A Backward-Recursive PCE-Based Computation (BRPC) Procedure to Compute Shortest Constrained Inter-Domain Traffic Engineering Label Switched Paths", RFC 5441, April 2009.

[RFC6952] Jethanandani, M., Patel, K., and L. Zheng, "Analysis of BGP, LDP, PCEP, and MSDP Issues According to the Keying and Authentication for Routing Protocols (KARP) Design Guide", RFC 6952, May 2013.

[I-D.ietf-pce-pcep-domain-sequence] Dhody, D., Palle, U., and R. Casellas, "Standard Representation Of Domain-Sequence", draft-ietf-pce-pcep-domain-sequence-06 (work in progress), October 2014.

[I-D.ietf-pce-pceps] Lopez, D., Dios, O., Wu, W., and D. Dhody, "Secure Transport for PCEP", draft-ietf-pce-pceps-02 (work in progress), October 2014.

[I-D.dhody-pce-iro-survey] Dhody, D., "Informal Survey into Include Route Object (IRO) Implementations in Path Computation Element communication Protocol (PCEP)", draft-dhody-pce-iro-survey-02 (work in progress), December 2014.

Appendix A. Details of IRO survey

During discussions of this document to provide a standard representation and encoding of Domain-Sequence within PCEP. It was determined that there was a need for clarification with respect to the ordered nature of the IRO.

Since there was a proposal to have a new IRO type with ordering, as well as handling of Loose bit, in an earlier version of this document (refer - draft-ietf-pce-pcep-domain-sequence-05), it was deemed necessary to conduct a survey of the existing and planned implementations. An informal survey was conducted via email. Responses were collected and anonymized by the PCE working group chairs.

[I-D.dhody-pce-iro-survey] summarizes the survey questions and captures the results. It further list some conclusions and action points.

This document was considered as one possible venue to handle the proposed action points.

Author's Address

Dhruv Dhody
Huawei Technologies
Leela Palace
Bangalore, Karnataka 560008
India

EMail: dhruv.ietf@gmail.com

PCE Working Group
Internet-Draft
Intended status: Standards Track
Expires: April 30, 2015

D. Dhody
Y. Lee
Huawei Technologies
D. Ceccarelli
Ericsson
October 27, 2014

PCEP Extension for Transporting TE Data
draft-dhodylee-pce-pcep-te-data-extn-01

Abstract

In order to compute and provide optimal paths, Path Computation Elements (PCEs) require an accurate and timely Traffic Engineering Database (TED). Traditionally this TED has been obtained from a link state routing protocol supporting traffic engineering extensions.

This document extends the Path Computation Element Communication Protocol (PCEP) with TED population capability.

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 April 30, 2015.

Copyright Notice

Table of Contents

1. Introduction 3
1.1. Requirements Language 4
2. Terminology 4
3. Applicability 4
4. Requirements for PCEP extension 4
5. New Functions to Support TED via PCEP 5
6. Overview of Extension to PCEP 5
6.1. New Messages 5
6.2. Capability Advertisement 6
6.3. Initial TED Synchronization 6
6.3.1. Optimizations for TED Synchronization 9
6.4. TE Report 9
7. Transport 9
8. PCEP Messages 9
8.1. TE Report Message 10
8.2. The PCErr Message 10
9. Objects and TLV 11
9.1. Open Object 11
9.1.1. TED Capability TLV 11
9.2. TE Object 12
9.2.1. Routing Universe TLV 13
9.2.2. Local TE Node Descriptors TLV 14
9.2.3. Remote TE Node Descriptors TLV 15
9.2.4. TE Node Descriptors Sub-TLVs 15
9.2.5. TE Link Descriptors TLV 16
9.2.6. TE Node Attributes TLV 17
9.2.7. TE Link Attributes TLV 18
10. Other Considerations 20
10.1. Inter-AS Links 20
11. Security Considerations 20
12. Manageability Considerations 20
12.1. Control of Function and Policy 20
12.2. Information and Data Models 20
12.3. Liveness Detection and Monitoring 20
12.4. Verify Correct Operations 20
12.5. Requirements On Other Protocols 20
12.6. Impact On Network Operations 20
13. IANA Considerations 20
14. Acknowledgments 21
15. References 21
15.1. Normative References 21

15.2. Informative References 21
Appendix A. Contributor Addresses 23

1. Introduction

In Multiprotocol Label Switching (MPLS) and Generalized MPLS (GMPLS), a Traffic Engineering Database (TED) is used in computing paths for connection oriented packet services and for circuits. The TED contains all relevant information that a Path Computation Element (PCE) needs to perform its computations. It is important that the TED be complete and accurate each time, the PCE performs a path computation.

In MPLS and GMPLS, interior gateway routing protocols (IGPs) have been used to create and maintain a copy of the TED at each node running the IGP. One of the benefits of the PCE architecture [RFC4655] is the use of computationally more sophisticated path computation algorithms and the realization that these may need enhanced processing power not necessarily available at each node participating in an IGP.

Section 4.3 of [RFC4655] describes the potential load of the TED on a network node and proposes an architecture where the TED is maintained by the PCE rather than the network nodes. However, it does not describe how a PCE would obtain the information needed to populate its TED. PCE may construct its TED by participating in the IGP ([RFC3630] and [RFC5305] for MPLS-TE; [RFC4203] and [RFC5307] for GMPLS). An alternative is offered by BGP-LS [I-D.ietf-idr-ls-distribution] .

[I-D.lee-pce-transporting-te-data] proposes some other approaches for creating and maintaining the TED directly on a PCE as an alternative to IGPs and BGP flooding and investigate the impact from the PCE, routing protocol, and node perspectives.

[RFC5440] describes the specifications for the Path Computation Element Communication Protocol (PCEP). PCEP specifies the communication between a Path Computation Client (PCC) and a Path Computation Element (PCE), or between two PCEs based on the PCE architecture [RFC4655].

This document specifies a PCEP extension for TED population capability to support functionalities described in [I-D.lee-pce-transporting-te-data].

1.1. Requirements Language

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

2. Terminology

The terminology is as per [RFC4655] and [RFC5440].

3. Applicability

As per [I-D.lee-pce-transporting-te-data], the mechanism specified in this draft is applicable to:

- Where there is no IGP-TE or BGP-LS running at the PCE to learn TED.
- Where there is IGP-TE or BGP-LS running but with a need for a faster TED population and convergence at the PCE.
  - A PCE may receive partial information (say basic TE) from IGP-TE and other information (optical and impairment) from PCEP.
  - A PCE may receive full information from both IGP-TE and PCEP.

A PCC may further choose to send only local TE information or both local and remote learned TED information.

How a PCE manages the TED information is implementation specific and thus out of scope of this document.

4. Requirements for PCEP extension

Following key requirements associated with TED population are identified for PCEP:

1. The PCEP speaker supporting this draft MUST be a mechanism to advertise the TED capability.
2. PCC supporting this draft MUST have the capability to report the TED to the PCE. This includes self originated TE information and remote TE information learned via routing protocols. PCC MUST be capable to do the initial bulk sync at the time of session initialization as well as changes to TED after.
3. A PCE MAY learn TED from PCEP as well as from existing mechanism like IGP-TE/BGP-LS. PCEP extension MUST have a mechanism to link

the TED information learned via other means. There MUST NOT be any changes to the existing TED population mechanism via IGP-TE/ BGP-LS. PCEP extension SHOULD keep the TE properties in a routing protocol (IGP-TE or BGP-LS) neutral way, such that an implementation which do want to learn about a Link-state topology do not need to know about any OSPF or IS-IS or BGP protocol specifics.

4. It SHOULD be possible to encode only the changes in TED properties (after the initial sync) in PCEP messages.

5. The same mechanism should be used for both MPLS TE as well as GMPLS, optical and impairment aware properties.

6. The extension in this draft SHOULD be extensible to support various architecture options listed in [I-D.lee-pce-transporting-te-data].

5. New Functions to Support TED via PCEP

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

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

- TE synchronization (C-E): after the session between the PCC and a PCE is initialized, the PCE must learn PCC’s TED before it can perform path computations. In case of stateful PCE it is RECOMENDED that this operation be done before LSP state synchronization.

- TE Report (C-E): a PCC sends a TE report to a PCE whenever the TED changes.

6. Overview of Extension to PCEP

6.1. New Messages

In this document, we define a new PCEP messages called TE Report (TERpt), a PCEP message sent by a PCC to a PCE to report TED. Each TE Report in a TERpt message can contain the TE node or TE Link properties. An unique PCEP specific TE identifier (TE-ID) is also carried in the message to identify the TE node or link and that remains constant for the lifetime of a PCEP session. This identifier on its own is sufficient when no IGP-TE or BGP-LS running in the

network for PCE to learn TED. Incase PCE learns some information from PCEP and some from the existing mechanism, the PCC SHOULD include the mapping of IGP-TE or BGP-LS identifier to map the TED information populated via PCEP with IGP-TE/BGP-LS. See Section 8.1 for details.

6.2. Capability Advertisement

During PCEP Initialization Phase, PCEP Speakers (PCE or PCC) advertise their support of TED population PCEP extensions. A PCEP Speaker includes the "TED Capability" TLV, described in Section 9, in the OPEN Object to advertise its support for PCEP TED extensions. The presence of the TED Capability TLV in PCC's OPEN Object indicates that the PCC is willing to send TE Reports whenever local TE information changes. The presence of the TED Capability TLV in PCE's OPEN message indicates that the PCE is interested in receiving TE Reports whenever local TE changes.

The PCEP protocol extensions for TED population MUST NOT be used if one or both PCEP Speakers have not included the TED Capability TLV in their respective OPEN message. If the PCE that supports the extensions of this draft but did not advertise this capability, then upon receipt of a PCRpt message from the PCC, it SHOULD generate a PCErr with error-type 19 (Invalid Operation), error-value TBD1 (Attempted TE Report if TED capability was not advertised) and it will terminate the PCEP session.

The TE reports sent by PCC MAY carry the remote TE information learned via existing means like IGP-TE and BGP-LS only if both PCEP Speakers set the R (remote) Flag in the "TED Capability" TLV to 'Remote Allowed (R Flag = 1)'. If this is not the case and TE reports carry remote TE information, then a PCErr with error-type 19 (Invalid Operation) and error-value TBD1 (Attempted TE Report if TED capability was not advertised) and it will terminate the PCEP session.

6.3. Initial TED Synchronization

The purpose of TED Synchronization is to provide a checkpoint-in-time state replica of a PCC's TED in a PCE. State Synchronization is performed immediately after the Initialization phase (see [RFC5440]]). In case of stateful PCE ([I-D.ietf-pce-stateful-pce]) it is RECOMENDED that the TED synchronization should be done before LSP state synchronization.

During TED Synchronization, a PCC first takes a snapshot of the state of its TED, then sends the snapshot to a PCE in a sequence of TE Reports. Each TE Report sent during TE Synchronization has the SYNC

Flag in the TE Object set to 1. The end of synchronization marker is a TERpt message with the SYNC Flag set to 0 for an TE Object with TED-ID equal to the reserved value 0. If the PCC has no TED state to synchronize, it will only send the end of synchronization marker.

Either the PCE or the PCC MAY terminate the session using the PCEP session termination procedures during the synchronization phase. If the session is terminated, the PCE MUST clean up state it received from this PCC. The session re-establishment MUST be re-attempted per the procedures defined in [RFC5440], including use of a back-off timer.

If the PCC encounters a problem which prevents it from completing the TED population, it MUST send a PCErr message with error-type TBD2 (TE Synchronization Error) and error-value 5 (indicating an internal PCC error) to the PCE and terminate the session.

The PCE does not send positive acknowledgements for properly received TED synchronization messages. It MUST respond with a PCErr message with error-type TBD2 (TE Synchronization Error) and error-value 1 (indicating an error in processing the TERpt) if it encounters a problem with the TE Report it received from the PCC and it MUST terminate the session.

The TE reports may carry local as well as remote TED information depending on the R flag in TED capability TLV.

The successful TED Synchronization sequences is shown in Figure 1.

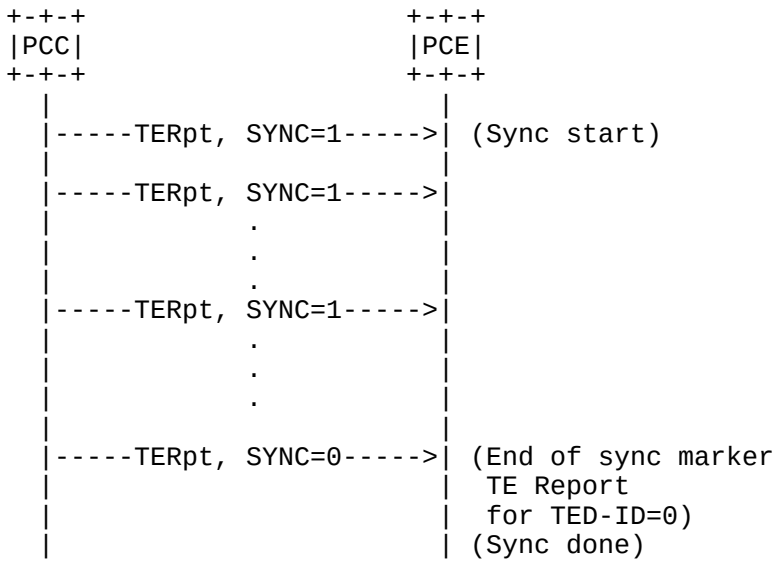


Figure 1: Successful state synchronization

The sequence where the PCE fails during the TED Synchronization phase is shown in Figure 2.

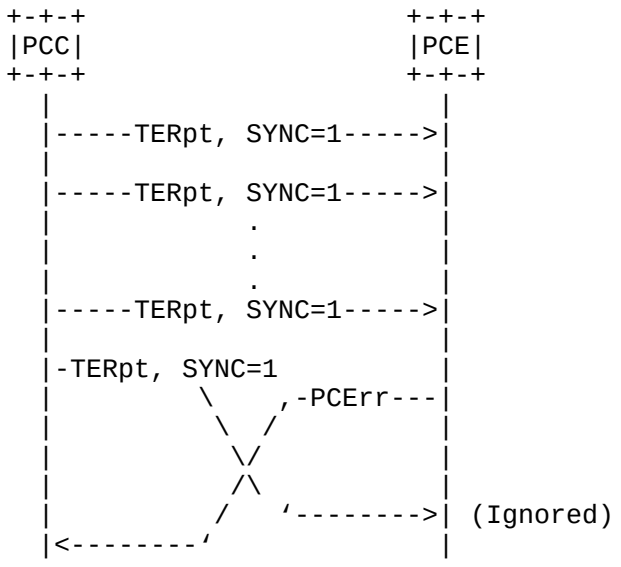


Figure 2: Failed TED synchronization (PCE failure)

The sequence where the PCC fails during the TED Synchronization phase is shown in Figure 3.

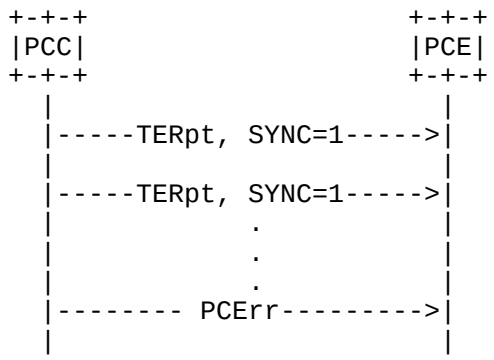


Figure 3: Failed TED synchronization (PCC failure)

6.3.1. Optimizations for TED Synchronization

TBD

6.4. TE Report

The PCC MUST report any changes in the TEDB to the PCE by sending a TE Report carried on a TERpt message to the PCE, indicating that the TE state. Each TE node and TE Link would be uniquely identified by a PCEP TE identifier (TE-ID). The TE reports may carry local as well as remote TED information depending on the R flag in TED capability TLV. In case R flag is set, It MAY also include the mapping of IGP-TE or BGP-LS identifier to map the TED information populated via PCEP with IGP-TE/BGP-LS.

More details about TERpt message are in Section 8.1.

7. Transport

A permanent PCEP session MUST be established between a PCE and PCC supporting TED population via PCEP. In the case of session failure, session re-establishment MUST be re-attempted per the procedures defined in [RFC5440].

8. PCEP Messages

As defined in [RFC5440], a PCEP message consists of a common header followed by a variable-length body made of a set of objects that can be either mandatory or optional. An object is said to be mandatory

in a PCEP message when the object must be included for the message to be considered valid. For each PCEP message type, a set of rules is defined that specify the set of objects that the message can carry. An implementation MUST form the PCEP messages using the object ordering specified in this document.

8.1. TE Report Message

A PCEP TE Report message (also referred to as TERpt message) is a PCEP message sent by a PCC to a PCE to report the TED state. A TERpt message can carry more than one TE Reports. The Message-Type field of the PCEP common header for the PCRpt message is set to [TBD3].

The format of the PCRpt message is as follows:

```
<TERpt Message> ::= <Common Header>
                    <te-report-list>
```

Where:

```
<te-report-list> ::= <TE>[<te-report-list>]
```

The TE object is a mandatory object which carries TE information of a TE node or a TE link. Each TE object has an unique TE-ID as described in Section 9.2. If the TE object is missing, the receiving PCE MUST send a PCErr message with Error-type=6 (Mandatory Object missing) and Error-value=[TBD4] (TE object missing).

A PCE may choose to implement a limit on the TE information a single PCC can populate. If a TERpt is received that causes the PCE to exceed this limit, it MUST send a PCErr message with error-type 19 (invalid operation) and error-value 4 (indicating resource limit exceeded) in response to the TERpt message triggering this condition and MAY terminate the session.

8.2. The PCErr Message

If a PCEP speaker has advertised the TED capability on the PCEP session, the PCErr message MAY include the TE object. If the error reported is the result of an TE report, then the TE-ID number MUST be the one from the TERpt that triggered the error.

The format of a PCErr message from [RFC5440] is extended as follows:

The format of the PCRpt message is as follows:

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

<error-obj-list>::=<PCEP-ERROR>[<error-obj-list>]

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

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

<te-id-list>::=<TE>[<te-id-list>]

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

9. Objects and TLV

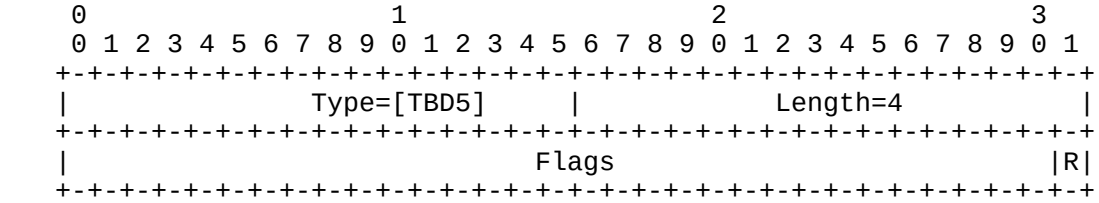
The PCEP objects defined in this document are compliant with the PCEP object format defined in [RFC5440]. The P flag and the I flag of the PCEP objects defined in this document MUST always be set to 0 on transmission and MUST be ignored on receipt since these flags are exclusively related to path computation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

9.1. Open Object

This document defines a new optional TLV for use in the OPEN Object.

9.1.1. TED Capability TLV

The TED-CAPABILITY TLV is an optional TLV for use in the OPEN Object for TED population via PCEP capability advertisement. Its format is shown in the following figure:



The type of the TLV is [TBD5] and it has a fixed length of 4 octets.

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

- R (remote - 1 bit): if set to 1 by a PCC, the R Flag indicates that the PCC allows reporting of remote TED information learned via other means like IGP-TE and BGP-LS; if set to 1 by a PCE, the

R Flag indicates that the PCE is capable of receiving remote TED information (from the PCC point of view). The R Flag must be advertised by both a PCC and a PCE for TERpt messages to report remote as well as local TE information on a PCEP session. The TLVs related to IGP-TE/BGP-LS identifier MUST be encoded when both PCEP speakers have the R Flag set.

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

Advertisement of the TED capability implies support of local TE population, as well as the objects, TLVs and procedures defined in this document.

9.2. TE Object

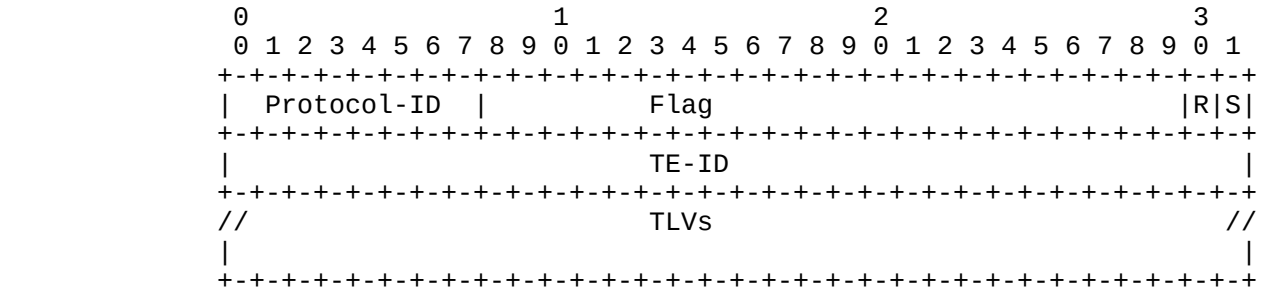
The TE (traffic engineering) object MUST be carried within TERpt messages and MAY be carried within PCErr messages. The TE object contains a set of fields used to specify the target TE node or link. It also contains a flag indicating to a PCE that the TED synchronization is in progress. The TLVs used with the TE object correlate with the IGP-TE/BGP-LS TE encodings.

TE Object-Class is [TBD6].

Two Object-Type values are defined for the TE object:

- TE Node: TE Object-Type is 1.
- TE Link: TE Object-Type is 2.

The format of the TE object body is as follows:



Protocol-ID (8-bit): The field provide the source information. Incase PCC only provides local information (R flag is not set), it MUST use Protocol-ID as Direct. The following values are defined (same as [I-D.ietf-idr-ls-distribution]):

Protocol-ID	Source protocol
1	IS-IS Level 1
2	IS-IS Level 2
3	OSPFv2
4	Direct
5	Static configuration
6	OSPFv3

Flags (32-bit):

- o S (SYNC - 1 bit): the S Flag MUST be set to 1 on each TERpt sent from a PCC during TED Synchronization. The S Flag MUST be set to 0 in other TERpt messages sent from the PCC.

- o R (Remove - 1 bit): On TERpt messages the R Flag indicates that the TE node/link has been removed from the PCC and the PCE SHOULD remove from its database. Upon receiving an TE Report with the R Flag set to 1, the PCE SHOULD remove all state for the TE node/ link identified by the TE Identifiers from its database.

TE-ID(32-bit): A PCEP-specific identifier for the TE node or link. A PCC creates a unique TE-ID for each TE node/link that is constant for the lifetime of a PCEP session. The PCC will advertise the same TE-ID on all PCEP sessions it maintains at a given times. All subsequent PCEP messages then address the TE node/link by the TE-ID. The values of 0 and 0xFFFFFFFF are reserved.

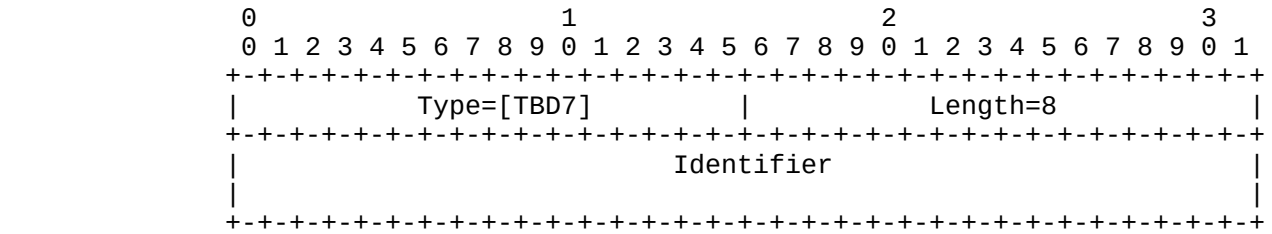
Unassigned bits are considered reserved. They MUST be set to 0 on transmission and MUST be ignored on receipt.

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

9.2.1. Routing Universe TLV

In case of remote TED population when existing IGP-TE/BGP-LS are also used, OSPF and IS-IS may run multiple routing protocol instances over the same link as described in [I-D.ietf-idr-ls-distribution]. See [RFC6822] and [RFC6549]. These instances define independent "routing universes". The 64-Bit 'Identifier' field is used to identify the "routing universe" where the TE object belongs. The TE objects representing IGP objects (nodes or links) from the same routing universe MUST have the same 'Identifier' value; TE objects with different 'Identifier' values MUST be considered to be from different routing universes.

The format of the ROUTING-UNIVERSE TLV is shown in the following figure:



Below table lists the 'Identifier' values that are defined as well-known in this draft (same as [I-D.ietf-idr-ls-distribution]).

Identifier	Routing Universe
0	L3 packet topology
1	L1 optical topology

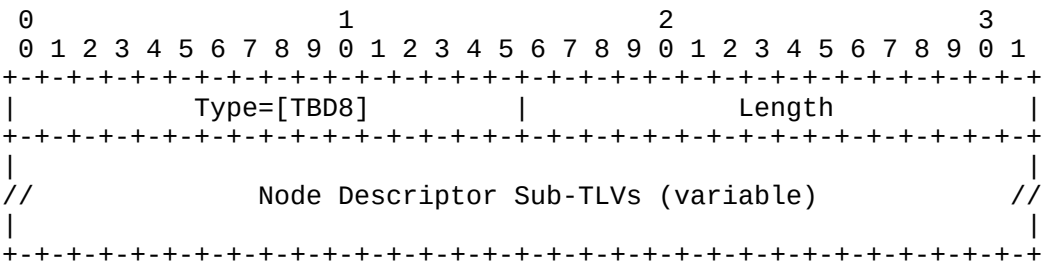
If this TLV is not present the default value 0 is assumed.

9.2.2. Local TE Node Descriptors TLV

As described in [I-D.ietf-idr-ls-distribution], each link is anchored by a pair of Router-IDs that are used by the underlying IGP, namely, 48 Bit ISO System-ID for IS-IS and 32 bit Router-ID for OSPFv2 and OSPFv3. Incase of additional auxiliary Router-IDs used for TE, these MUST also be included in the TE link attribute TLV (see Section 9.2.6).

It is desirable that the Router-ID assignments inside the TE Node Descriptor are globally unique. Autonomous System (AS) Number and PCEP-TED Identifier in order to disambiguate the Router-IDs, as described in [I-D.ietf-idr-ls-distribution].

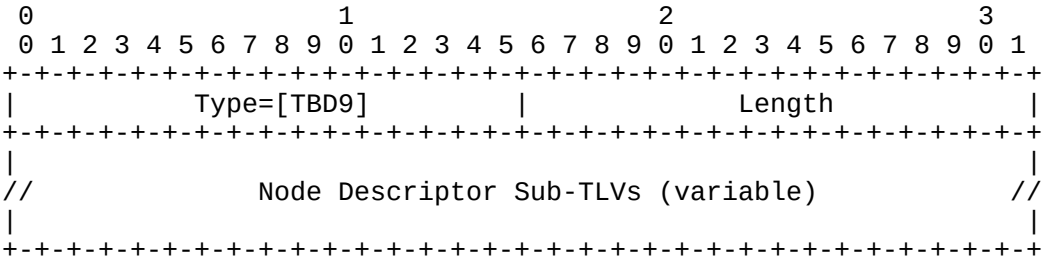
The Local TE Node Descriptors TLV contains Node Descriptors for the node anchoring the local end of the link. This TLV MUST be included in the TE Report when during a given PCEP session a TE node/link is first reported to a PCE. A PCC sends to a PCE the first TE Report either during State Synchronization, or when a new TE node/link is learned at the PCC. The value contains one or more Node Descriptor Sub-TLVs, which allows specification of a flexible key for any given Node/Link information such that global uniqueness of the TE node/link is ensured.



The value contains one or more Node Descriptor Sub-TLVs defined in Section 9.2.4.

9.2.3. Remote TE Node Descriptors TLV

The Remote TE Node Descriptors contains Node Descriptors for the node anchoring the remote end of the link. This TLV MUST be included in the TE Report when during a given PCEP session a TE link is first reported to a PCE. A PCC sends to a PCE the first TE Report either during State Synchronization, or when a new TE link is learned at the PCC. The length of this TLV is variable. The value contains one or more Node Descriptor Sub-TLVs defined in Section 9.2.4.



9.2.4. TE Node Descriptors Sub-TLVs

The Node Descriptor Sub-TLV type Type and lengths are listed in the following table:

Sub-TLV	Description	Length
TBD10	Autonomous System	4
TBD11	BGP-LS Identifier	4
TBD12	OSPF Area-ID	4
TBD13	Router-ID	Variable

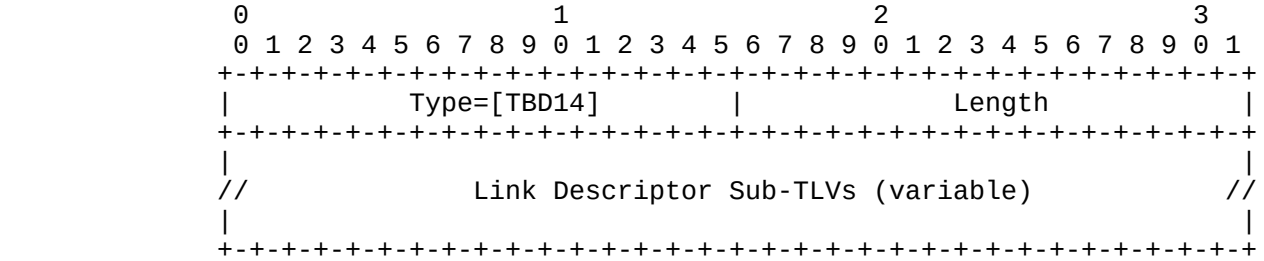
The sub-TLV values in Node Descriptor TLVs are defined as follows (similar to [I-D.ietf-idr-ls-distribution]):

- Autonomous System: opaque value (32 Bit AS Number)
- BGP-LS Identifier: opaque value (32 Bit ID). In conjunction with ASN, uniquely identifies the BGP-LS domain as described in [I-D.ietf-idr-ls-distribution].
- Area ID: It is used to identify the 32 Bit area to which the TE object belongs. Area Identifier allows the different TE objects of the same router to be discriminated.
- Router ID: opaque value. Usage is described in [I-D.ietf-idr-ls-distribution] for IGP Router ID. In case only local TE information is transported and PCE learns TED only from PCEP, it contain the unique local TE IPv4 or IPv6 router ID.
- There can be at most one instance of each sub-TLV type present in any Node Descriptor.

9.2.5. TE Link Descriptors TLV

The TE Link Descriptors TLV contains Link Descriptors for each TE link. This TLV MUST be included in the TE Report when during a given PCEP session a TE link is first reported to a PCE. A PCC sends to a PCE the first TE Report either during State Synchronization, or when a new TE link is learned at the PCC. The length of this TLV is variable. The value contains one or more TE Link Descriptor Sub-TLVs

The 'TE Link descriptor' TLVs uniquely identify a link among multiple parallel links between a pair of anchor routers similar to [I-D.ietf-idr-ls-distribution].



The Link Descriptor Sub-TLV type and lengths are listed in the following table:

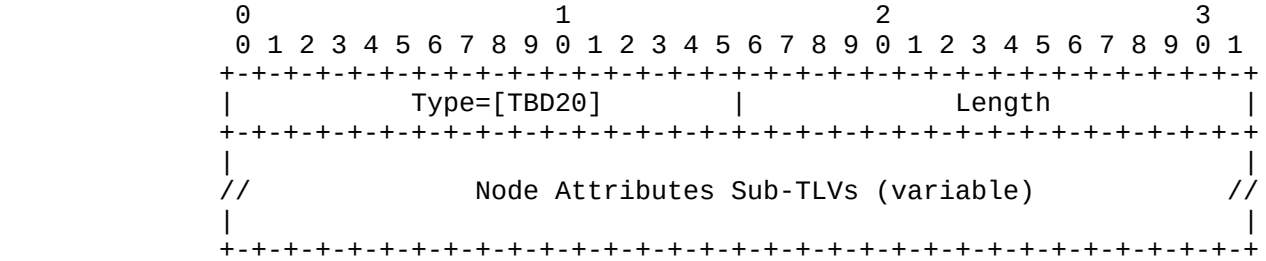
Sub-TLV	Description	IS-IS TLV /Sub-TLV	Value defined in:
TBD15	Link Local/Remote Identifiers	22/4	[RFC5307]/1.1
TBD16	IPv4 interface address	22/6	[RFC5305]/3.2
TBD17	IPv4 neighbor address	22/8	[RFC5305]/3.3
TBD18	IPv6 interface address	22/12	[RFC6119]/4.2
TBD19	IPv6 neighbor address	22/13	[RFC6119]/4.3

The format and semantics of the 'value' fields in most 'Link Descriptor' sub-TLVs correspond to the format and semantics of value fields in IS-IS Extended IS Reachability sub-TLVs, defined in [RFC5305], [RFC5307] and [RFC6119]. Although the encodings for 'Link Descriptor' TLVs were originally defined for IS-IS, the TLVs can carry data sourced either by IS-IS or OSPF or direct.

The information about a link present in the LSA/LSP originated by the local node of the link determines the set of sub-TLVs in the Link Descriptor of the link as described in [I-D.ietf-idr-ls-distribution].

9.2.6. TE Node Attributes TLV

This is an optional, non-transitive attribute that is used to carry TE node attributes. The TE node attribute TLV may be encoded in the TE node Object.

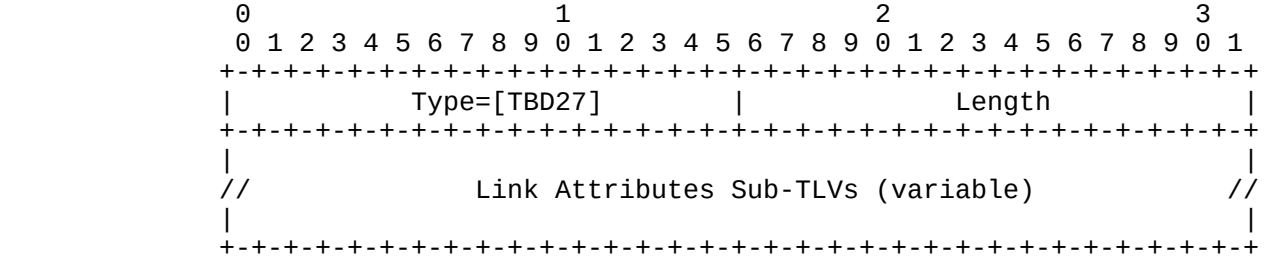


The Node Attributes Sub-TLV type and lengths are listed in the following table:

Sub TLV	Description	Length	Value defined in:
TBD21	Node Flag Bits	1	[I-D.ietf-idr-ls-distribution]/3.3.1.1
TBD22	Opaque Node Properties	variable	[I-D.ietf-idr-ls-distribution]/3.3.1.5
TBD23	Node Name	variable	[I-D.ietf-idr-ls-distribution]/3.3.1.3
TBD24	IS-IS Area Identifier	variable	[I-D.ietf-idr-ls-distribution]/3.3.1.2
TBD25	IPv4 Router-ID of Local Node	4	[RFC5305]/4.3
TBD26	IPv6 Router-ID of Local Node	16	[RFC6119]/4.1

9.2.7. TE Link Attributes TLV

TE Link attribute TLV may be encoded in the TE Link Object. The format and semantics of the 'value' fields in some 'Link Attribute' sub-TLVs correspond to the format and semantics of value fields in IS-IS Extended IS Reachability sub-TLVs, defined in [RFC5305], [RFC5307] and [I-D.ietf-idr-ls-distribution]. Although the encodings for 'Link Attribute' TLVs were originally defined for IS-IS, the TLVs can carry data sourced either by IS-IS or OSPF or direct.



The following 'Link Attribute' sub-TLVs are are valid :

Sub-TLV	Description	IS-IS TLV /Sub-TLV BGP-LS TLV	Defined in:
TBD28	IPv4 Router-ID of Local Node	134/---	[RFC5305]/4.3
TBD29	IPv6 Router-ID of Local Node	140/---	[RFC6119]/4.1
TBD30	IPv4 Router-ID of Remote Node	134/---	[RFC5305]/4.3
TBD31	IPv6 Router-ID of Remote Node	140/---	[RFC6119]/4.1
TBD32	Link Local/Remote Identifiers	22/4	[RFC5307]/1.1
TBD33	Administrative group (color)	22/3	[RFC5305]/3.1
TBD34	Maximum link bandwidth	22/9	[RFC5305]/3.3
TBD35	Max. reservable link bandwidth	22/10	[RFC5305]/3.5
TBD36	Unreserved bandwidth	22/11	[RFC5305]/3.6
TBD37	TE Default Metric	22/18	[I-D.ietf-idr-ls-distribution] /3.3.2.3
TBD38	Link Protection Type	22/20	[RFC5307]/1.2
TBD39	MPLS Protocol Mask	1094	[I-D.ietf-idr-ls-distribution] /3.3.2.2
TBD40	IGP Metric	1095	[I-D.ietf-idr-ls-distribution] /3.3.2.4
TBD41	Shared Risk Link Group	1096	[I-D.ietf-idr-ls-distribution] /3.3.2.5
TBD42	Opaque link attributes	1097	[I-D.ietf-idr-ls-distribution] /3.3.2.6
TBD43	Link Name attribute	1098	[I-D.ietf-idr-ls-distribution] /3.3.2.7

10. Other Considerations

10.1. Inter-AS Links

The main source of TE information is the IGP, which is not active on inter-AS links. In some cases, the IGP may have information of inter-AS links ([RFC5392], [RFC5316]). In other cases, an implementation SHOULD provide a means to inject inter-AS links into PCEP. The exact mechanism used to provision the inter-AS links is outside the scope of this document.

11. Security Considerations

TBD.

12. Manageability Considerations

12.1. Control of Function and Policy

TBD.

12.2. Information and Data Models

TBD.

12.3. Liveness Detection and Monitoring

TBD.

12.4. Verify Correct Operations

TBD.

12.5. Requirements On Other Protocols

TBD.

12.6. Impact On Network Operations

TBD.

13. IANA Considerations

14. Acknowledgments

This document borrows some of the structure and text from the [I-D.ietf-pce-stateful-pce].

15. References

15.1. Normative References

[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.

15.2. Informative References

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

[RFC4203] Kompella, K. and Y. Rekhter, "OSPF Extensions in Support of Generalized Multi-Protocol Label Switching (GMPLS)", RFC 4203, October 2005.

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

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

[RFC5307] Kompella, K. and Y. Rekhter, "IS-IS Extensions in Support of Generalized Multi-Protocol Label Switching (GMPLS)", RFC 5307, October 2008.

[RFC5316] Chen, M., Zhang, R., and X. Duan, "ISIS Extensions in Support of Inter-Autonomous System (AS) MPLS and GMPLS Traffic Engineering", RFC 5316, December 2008.

[RFC5392] Chen, M., Zhang, R., and X. Duan, "OSPF Extensions in Support of Inter-Autonomous System (AS) MPLS and GMPLS Traffic Engineering", RFC 5392, January 2009.

[RFC6119] Harrison, J., Berger, J., and M. Bartlett, "IPv6 Traffic Engineering in IS-IS", RFC 6119, February 2011.

[RFC6549] Lindem, A., Roy, A., and S. Mirtorabi, "OSPFv2 Multi-Instance Extensions", RFC 6549, March 2012.

[RFC6822] Previdi, S., Ginsberg, L., Shand, M., Roy, A., and D. Ward, "IS-IS Multi-Instance", RFC 6822, December 2012.

[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-10 (work in progress), October 2014.

[I-D.ietf-idr-ls-distribution] Gredler, H., Medved, J., Previdi, S., Farrel, A., and S. Ray, "North-Bound Distribution of Link-State and TE Information using BGP", draft-ietf-idr-ls-distribution-06 (work in progress), September 2014.

[I-D.lee-pce-transporting-te-data] Lee, Y. and z. zhenghaomian@huawei.com, "PCE in Support of Transporting Traffic Engineering Data", draft-lee-pce-transporting-te-data-01 (work in progress), October 2014.

Appendix A. Contributor Addresses

Udayasree Palle
Huawei Technologies
Leela Palace
Bangalore, Karnataka 560008
India

EMail: udayasree.palle@huawei.com

Authors' Addresses

Dhruv Dhody
Huawei Technologies
Leela Palace
Bangalore, Karnataka 560008
India

EMail: dhruv.ietf@gmail.com

Young Lee
Huawei Technologies
5340 Legacy Drive, Building 3
Plano, TX 75023
USA

EMail: leeyoung@huawei.com

Daniele Ceccarelli
Ericsson
Torshamnsgatan,48
Stockholm
Sweden

EMail: daniele.ceccarelli@ericsson.com

PCE Working Group
Internet-Draft
Intended status: Experimental
Expires: September 5, 2015

D. Dhody
Y. Lee
Huawei Technologies
D. Ceccarelli
Ericsson
March 4, 2015

PCEP Extension for Transporting TE Data
draft-dhodylee-pce-pcep-te-data-extn-02

Abstract

In order to compute and provide optimal paths, Path Computation Elements (PCEs) require an accurate and timely Traffic Engineering Database (TED). Traditionally this TED has been obtained from a link state routing protocol supporting traffic engineering extensions.

This document extends the Path Computation Element Communication Protocol (PCEP) with TED population capability.

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 September 5, 2015.

Copyright Notice

Table of Contents

1. Introduction 3
1.1. Requirements Language 4
2. Terminology 4
3. Applicability 4
4. Requirements for PCEP extension 4
5. New Functions to Support TED via PCEP 5
6. Overview of Extension to PCEP 5
6.1. New Messages 5
6.2. Capability Advertisement 6
6.3. Initial TED Synchronization 6
6.3.1. Optimizations for TED Synchronization 9
6.4. TE Report 9
7. Transport 9
8. PCEP Messages 9
8.1. TE Report Message 10
8.2. The PCErr Message 10
9. Objects and TLV 11
9.1. Open Object 11
9.1.1. TED Capability TLV 11
9.2. TE Object 12
9.2.1. Routing Universe TLV 13
9.2.2. Local TE Node Descriptors TLV 14
9.2.3. Remote TE Node Descriptors TLV 15
9.2.4. TE Node Descriptors Sub-TLVs 15
9.2.5. TE Link Descriptors TLV 16
9.2.6. TE Node Attributes TLV 17
9.2.7. TE Link Attributes TLV 18
10. Other Considerations 20
10.1. Inter-AS Links 20
11. Security Considerations 20
12. Manageability Considerations 20
12.1. Control of Function and Policy 20
12.2. Information and Data Models 20
12.3. Liveness Detection and Monitoring 21
12.4. Verify Correct Operations 21
12.5. Requirements On Other Protocols 21
12.6. Impact On Network Operations 21
13. IANA Considerations 21
14. Acknowledgments 21
15. References 21
15.1. Normative References 22

15.2. Informative References 22
Appendix A. Contributor Addresses 24
Authors' Addresses 24

1. Introduction

In Multiprotocol Label Switching (MPLS) and Generalized MPLS (GMPLS), a Traffic Engineering Database (TED) is used in computing paths for connection oriented packet services and for circuits. The TED contains all relevant information that a Path Computation Element (PCE) needs to perform its computations. It is important that the TED be complete and accurate each time, the PCE performs a path computation.

In MPLS and GMPLS, interior gateway routing protocols (IGPs) have been used to create and maintain a copy of the TED at each node running the IGP. One of the benefits of the PCE architecture [RFC4655] is the use of computationally more sophisticated path computation algorithms and the realization that these may need enhanced processing power not necessarily available at each node participating in an IGP.

Section 4.3 of [RFC4655] describes the potential load of the TED on a network node and proposes an architecture where the TED is maintained by the PCE rather than the network nodes. However, it does not describe how a PCE would obtain the information needed to populate its TED. PCE may construct its TED by participating in the IGP ([RFC3630] and [RFC5305] for MPLS-TE; [RFC4203] and [RFC5307] for GMPLS). An alternative is offered by BGP-LS [I-D.ietf-idr-ls-distribution] .

[I-D.lee-pce-transporting-te-data] proposes some other approaches for creating and maintaining the TED directly on a PCE as an alternative to IGPs and BGP flooding and investigate the impact from the PCE, routing protocol, and node perspectives.

[RFC5440] describes the specifications for the Path Computation Element Communication Protocol (PCEP). PCEP specifies the communication between a Path Computation Client (PCC) and a Path Computation Element (PCE), or between two PCEs based on the PCE architecture [RFC4655].

This document specifies a PCEP extension for TED population capability to support functionalities described in [I-D.lee-pce-transporting-te-data].

1.1. Requirements Language

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

2. Terminology

The terminology is as per [RFC4655] and [RFC5440].

3. Applicability

As per [I-D.lee-pce-transporting-te-data], the mechanism specified in this draft is applicable to:

- Where there is no IGP-TE or BGP-LS running at the PCE to learn TED.
- Where there is IGP-TE or BGP-LS running but with a need for a faster TED population and convergence at the PCE.
  - A PCE may receive partial information (say basic TE) from IGP-TE and other information (optical and impairment) from PCEP.
  - A PCE may receive full information from both IGP-TE and PCEP.

A PCC may further choose to send only local TE information or both local and remote learned TED information.

How a PCE manages the TED information is implementation specific and thus out of scope of this document.

4. Requirements for PCEP extension

Following key requirements associated with TED population are identified for PCEP:

1. The PCEP speaker supporting this draft MUST be a mechanism to advertise the TED capability.
2. PCC supporting this draft MUST have the capability to report the TED to the PCE. This includes self originated TE information and remote TE information learned via routing protocols. PCC MUST be capable to do the initial bulk sync at the time of session initialization as well as changes to TED after.
3. A PCE MAY learn TED from PCEP as well as from existing mechanism like IGP-TE/BGP-LS. PCEP extension MUST have a mechanism to link

the TED information learned via other means. There MUST NOT be any changes to the existing TED population mechanism via IGP-TE/ BGP-LS. PCEP extension SHOULD keep the TE properties in a routing protocol (IGP-TE or BGP-LS) neutral way, such that an implementation which do want to learn about a Link-state topology do not need to know about any OSPF or IS-IS or BGP protocol specifics.

4. It SHOULD be possible to encode only the changes in TED properties (after the initial sync) in PCEP messages.

5. The same mechanism should be used for both MPLS TE as well as GMPLS, optical and impairment aware properties.

6. The extension in this draft SHOULD be extensible to support various architecture options listed in [I-D.lee-pce-transporting-te-data].

5. New Functions to Support TED via PCEP

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

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

- TE synchronization (C-E): after the session between the PCC and a PCE is initialized, the PCE must learn PCC’s TED before it can perform path computations. In case of stateful PCE it is RECOMENDED that this operation be done before LSP state synchronization.

- TE Report (C-E): a PCC sends a TE report to a PCE whenever the TED changes.

6. Overview of Extension to PCEP

6.1. New Messages

In this document, we define a new PCEP messages called TE Report (TERpt), a PCEP message sent by a PCC to a PCE to report TED. Each TE Report in a TERpt message can contain the TE node or TE Link properties. An unique PCEP specific TE identifier (TE-ID) is also carried in the message to identify the TE node or link and that remains constant for the lifetime of a PCEP session. This identifier on its own is sufficient when no IGP-TE or BGP-LS running in the

network for PCE to learn TED. Incase PCE learns some information from PCEP and some from the existing mechanism, the PCC SHOULD include the mapping of IGP-TE or BGP-LS identifier to map the TED information populated via PCEP with IGP-TE/BGP-LS. See Section 8.1 for details.

6.2. Capability Advertisement

During PCEP Initialization Phase, PCEP Speakers (PCE or PCC) advertise their support of TED population PCEP extensions. A PCEP Speaker includes the "TED Capability" TLV, described in Section 9, in the OPEN Object to advertise its support for PCEP TED extensions. The presence of the TED Capability TLV in PCC's OPEN Object indicates that the PCC is willing to send TE Reports whenever local TE information changes. The presence of the TED Capability TLV in PCE's OPEN message indicates that the PCE is interested in receiving TE Reports whenever local TE changes.

The PCEP protocol extensions for TED population MUST NOT be used if one or both PCEP Speakers have not included the TED Capability TLV in their respective OPEN message. If the PCE that supports the extensions of this draft but did not advertise this capability, then upon receipt of a PCRpt message from the PCC, it SHOULD generate a PCErr with error-type 19 (Invalid Operation), error-value TBD1 (Attempted TE Report if TED capability was not advertised) and it will terminate the PCEP session.

The TE reports sent by PCC MAY carry the remote TE information learned via existing means like IGP-TE and BGP-LS only if both PCEP Speakers set the R (remote) Flag in the "TED Capability" TLV to 'Remote Allowed (R Flag = 1)'. If this is not the case and TE reports carry remote TE information, then a PCErr with error-type 19 (Invalid Operation) and error-value TBD1 (Attempted TE Report if TED capability was not advertised) and it will terminate the PCEP session.

6.3. Initial TED Synchronization

The purpose of TED Synchronization is to provide a checkpoint-in-time state replica of a PCC's TED in a PCE. State Synchronization is performed immediately after the Initialization phase (see [RFC5440]]). In case of stateful PCE ([I-D.ietf-pce-stateful-pce]) it is RECOMENDED that the TED synchronization should be done before LSP state synchronization.

During TED Synchronization, a PCC first takes a snapshot of the state of its TED, then sends the snapshot to a PCE in a sequence of TE Reports. Each TE Report sent during TE Synchronization has the SYNC

Flag in the TE Object set to 1. The end of synchronization marker is a TERpt message with the SYNC Flag set to 0 for an TE Object with TED-ID equal to the reserved value 0. If the PCC has no TED state to synchronize, it will only send the end of synchronization marker.

Either the PCE or the PCC MAY terminate the session using the PCEP session termination procedures during the synchronization phase. If the session is terminated, the PCE MUST clean up state it received from this PCC. The session re-establishment MUST be re-attempted per the procedures defined in [RFC5440], including use of a back-off timer.

If the PCC encounters a problem which prevents it from completing the TED population, it MUST send a PCErr message with error-type TBD2 (TE Synchronization Error) and error-value 5 (indicating an internal PCC error) to the PCE and terminate the session.

The PCE does not send positive acknowledgements for properly received TED synchronization messages. It MUST respond with a PCErr message with error-type TBD2 (TE Synchronization Error) and error-value 1 (indicating an error in processing the TERpt) if it encounters a problem with the TE Report it received from the PCC and it MUST terminate the session.

The TE reports may carry local as well as remote TED information depending on the R flag in TED capability TLV.

The successful TED Synchronization sequences is shown in Figure 1.

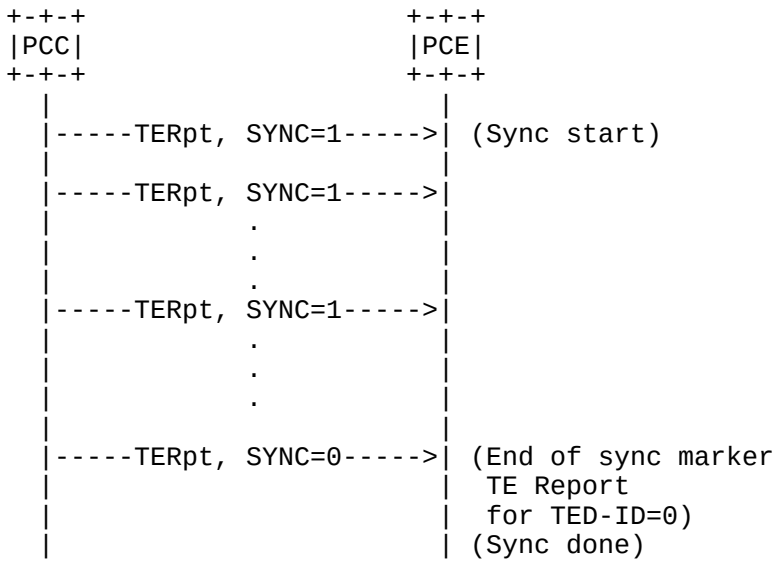


Figure 1: Successful state synchronization

The sequence where the PCE fails during the TED Synchronization phase is shown in Figure 2.

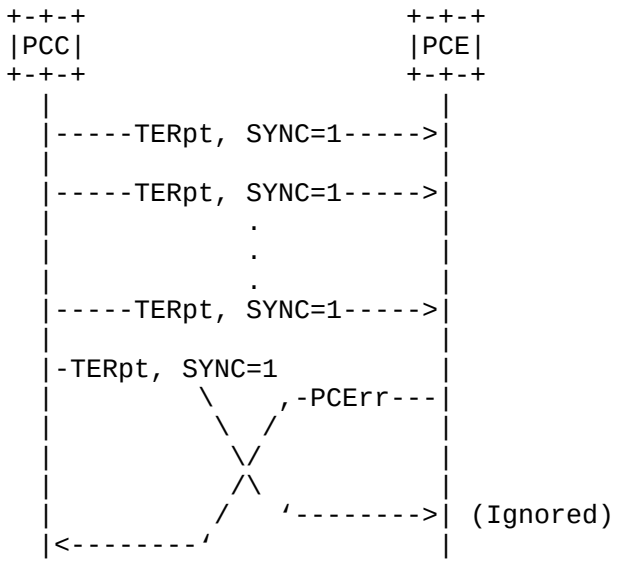


Figure 2: Failed TED synchronization (PCE failure)

The sequence where the PCC fails during the TED Synchronization phase is shown in Figure 3.

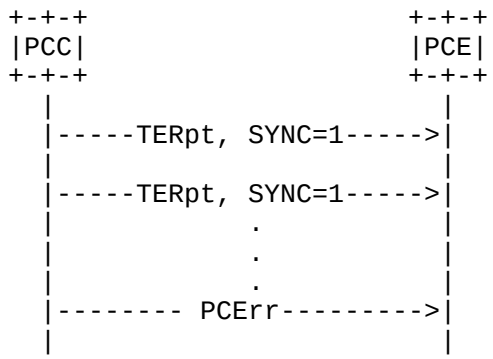


Figure 3: Failed TED synchronization (PCC failure)

6.3.1. Optimizations for TED Synchronization

TBD

6.4. TE Report

The PCC MUST report any changes in the TEDB to the PCE by sending a TE Report carried on a TERpt message to the PCE, indicating that the TE state. Each TE node and TE Link would be uniquely identified by a PCEP TE identifier (TE-ID). The TE reports may carry local as well as remote TED information depending on the R flag in TED capability TLV. In case R flag is set, It MAY also include the mapping of IGP-TE or BGP-LS identifier to map the TED information populated via PCEP with IGP-TE/BGP-LS.

More details about TERpt message are in Section 8.1.

7. Transport

A permanent PCEP session MUST be established between a PCE and PCC supporting TED population via PCEP. In the case of session failure, session re-establishment MUST be re-attempted per the procedures defined in [RFC5440].

8. PCEP Messages

As defined in [RFC5440], a PCEP message consists of a common header followed by a variable-length body made of a set of objects that can be either mandatory or optional. An object is said to be mandatory

in a PCEP message when the object must be included for the message to be considered valid. For each PCEP message type, a set of rules is defined that specify the set of objects that the message can carry. An implementation MUST form the PCEP messages using the object ordering specified in this document.

8.1. TE Report Message

A PCEP TE Report message (also referred to as TERpt message) is a PCEP message sent by a PCC to a PCE to report the TED state. A TERpt message can carry more than one TE Reports. The Message-Type field of the PCEP common header for the PCRpt message is set to [TBD3].

The format of the PCRpt message is as follows:

```
<TERpt Message> ::= <Common Header>
                    <te-report-list>
```

Where:

```
<te-report-list> ::= <TE>[<te-report-list>]
```

The TE object is a mandatory object which carries TE information of a TE node or a TE link. Each TE object has an unique TE-ID as described in Section 9.2. If the TE object is missing, the receiving PCE MUST send a PCErr message with Error-type=6 (Mandatory Object missing) and Error-value=[TBD4] (TE object missing).

A PCE may choose to implement a limit on the TE information a single PCC can populate. If a TERpt is received that causes the PCE to exceed this limit, it MUST send a PCErr message with error-type 19 (invalid operation) and error-value 4 (indicating resource limit exceeded) in response to the TERpt message triggering this condition and MAY terminate the session.

8.2. The PCErr Message

If a PCEP speaker has advertised the TED capability on the PCEP session, the PCErr message MAY include the TE object. If the error reported is the result of an TE report, then the TE-ID number MUST be the one from the TERpt that triggered the error.

The format of a PCErr message from [RFC5440] is extended as follows:

The format of the PCRpt message is as follows:

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

<error-obj-list>::=<PCEP-ERROR>[<error-obj-list>]

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

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

<te-id-list>::=<TE>[<te-id-list>]

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

9. Objects and TLV

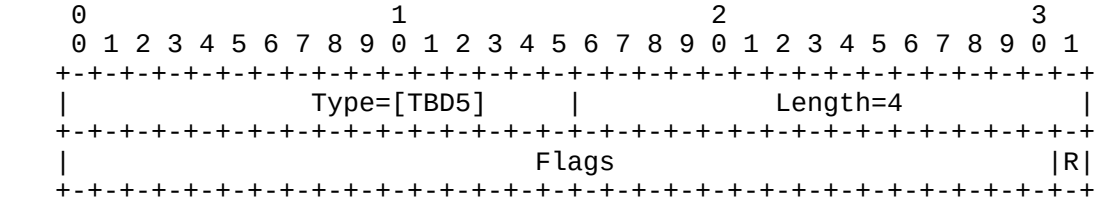
The PCEP objects defined in this document are compliant with the PCEP object format defined in [RFC5440]. The P flag and the I flag of the PCEP objects defined in this document MUST always be set to 0 on transmission and MUST be ignored on receipt since these flags are exclusively related to path computation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

9.1. Open Object

This document defines a new optional TLV for use in the OPEN Object.

9.1.1. TED Capability TLV

The TED-CAPABILITY TLV is an optional TLV for use in the OPEN Object for TED population via PCEP capability advertisement. Its format is shown in the following figure:



The type of the TLV is [TBD5] and it has a fixed length of 4 octets.

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

- R (remote - 1 bit): if set to 1 by a PCC, the R Flag indicates that the PCC allows reporting of remote TED information learned via other means like IGP-TE and BGP-LS; if set to 1 by a PCE, the

R Flag indicates that the PCE is capable of receiving remote TED information (from the PCC point of view). The R Flag must be advertised by both a PCC and a PCE for TERpt messages to report remote as well as local TE information on a PCEP session. The TLVs related to IGP-TE/BGP-LS identifier MUST be encoded when both PCEP speakers have the R Flag set.

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

Advertisement of the TED capability implies support of local TE population, as well as the objects, TLVs and procedures defined in this document.

9.2. TE Object

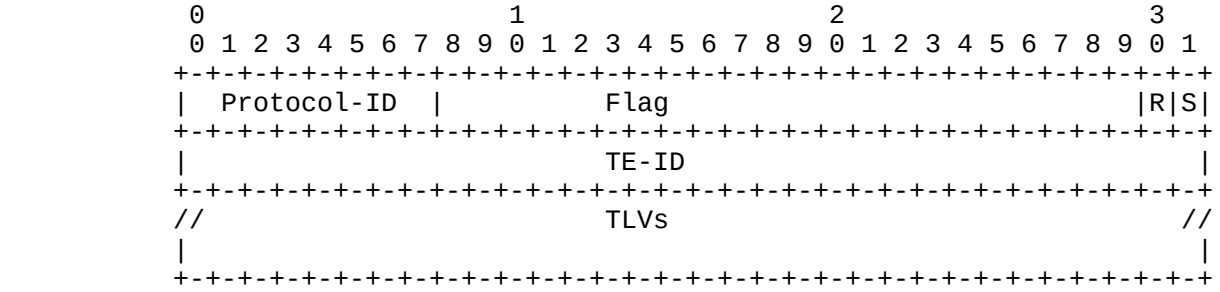
The TE (traffic engineering) object MUST be carried within TERpt messages and MAY be carried within PCErr messages. The TE object contains a set of fields used to specify the target TE node or link. It also contains a flag indicating to a PCE that the TED synchronization is in progress. The TLVs used with the TE object correlate with the IGP-TE/BGP-LS TE encodings.

TE Object-Class is [TBD6].

Two Object-Type values are defined for the TE object:

- TE Node: TE Object-Type is 1.
- TE Link: TE Object-Type is 2.

The format of the TE object body is as follows:



Protocol-ID (8-bit): The field provide the source information. Incase PCC only provides local information (R flag is not set), it MUST use Protocol-ID as Direct. The following values are defined (same as [I-D.ietf-idr-ls-distribution]):

Protocol-ID	Source protocol
1	IS-IS Level 1
2	IS-IS Level 2
3	OSPFv2
4	Direct
5	Static configuration
6	OSPFv3

Flags (32-bit):

- S (SYNC - 1 bit): the S Flag MUST be set to 1 on each TERpt sent from a PCC during TED Synchronization. The S Flag MUST be set to 0 in other TERpt messages sent from the PCC.

- R (Remove - 1 bit): On TERpt messages the R Flag indicates that the TE node/link has been removed from the PCC and the PCE SHOULD remove from its database. Upon receiving an TE Report with the R Flag set to 1, the PCE SHOULD remove all state for the TE node/ link identified by the TE Identifiers from its database.

TE-ID(32-bit): A PCEP-specific identifier for the TE node or link. A PCC creates a unique TE-ID for each TE node/link that is constant for the lifetime of a PCEP session. The PCC will advertise the same TE-ID on all PCEP sessions it maintains at a given times. All subsequent PCEP messages then address the TE node/link by the TE-ID. The values of 0 and 0xFFFFFFFF are reserved.

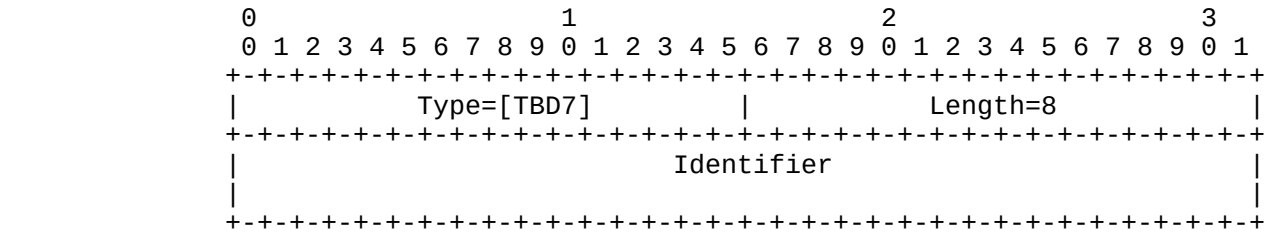
Unassigned bits are considered reserved. They MUST be set to 0 on transmission and MUST be ignored on receipt.

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

9.2.1. Routing Universe TLV

In case of remote TED population when existing IGP-TE/BGP-LS are also used, OSPF and IS-IS may run multiple routing protocol instances over the same link as described in [I-D.ietf-idr-ls-distribution]. See [RFC6822] and [RFC6549]. These instances define independent "routing universes". The 64-Bit 'Identifier' field is used to identify the "routing universe" where the TE object belongs. The TE objects representing IGP objects (nodes or links) from the same routing universe MUST have the same 'Identifier' value; TE objects with different 'Identifier' values MUST be considered to be from different routing universes.

The format of the ROUTING-UNIVERSE TLV is shown in the following figure:



Below table lists the 'Identifier' values that are defined as well-known in this draft (same as [I-D.ietf-idr-ls-distribution]).

Identifier	Routing Universe
0	L3 packet topology
1	L1 optical topology

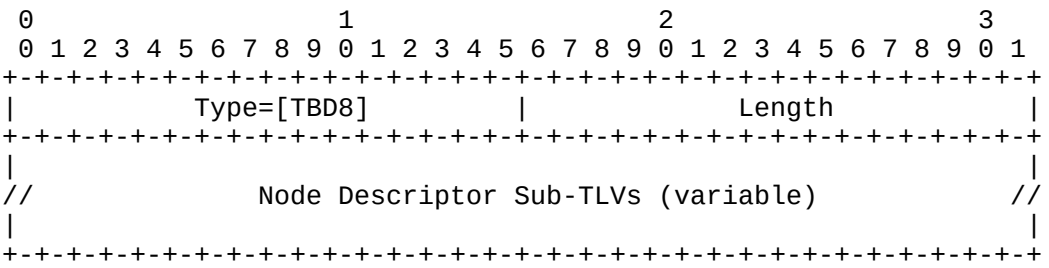
If this TLV is not present the default value 0 is assumed.

9.2.2. Local TE Node Descriptors TLV

As described in [I-D.ietf-idr-ls-distribution], each link is anchored by a pair of Router-IDs that are used by the underlying IGP, namely, 48 Bit ISO System-ID for IS-IS and 32 bit Router-ID for OSPFv2 and OSPFv3. Incase of additional auxiliary Router-IDs used for TE, these MUST also be included in the TE link attribute TLV (see Section 9.2.6).

It is desirable that the Router-ID assignments inside the TE Node Descriptor are globally unique. Autonomous System (AS) Number and PCEP-TED Identifier in order to disambiguate the Router-IDs, as described in [I-D.ietf-idr-ls-distribution].

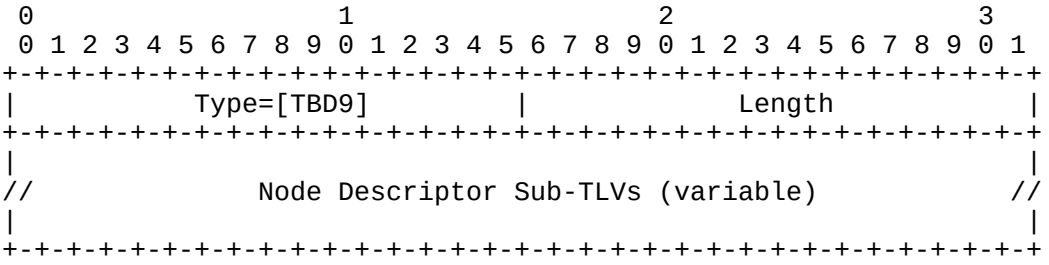
The Local TE Node Descriptors TLV contains Node Descriptors for the node anchoring the local end of the link. This TLV MUST be included in the TE Report when during a given PCEP session a TE node/link is first reported to a PCE. A PCC sends to a PCE the first TE Report either during State Synchronization, or when a new TE node/link is learned at the PCC. The value contains one or more Node Descriptor Sub-TLVs, which allows specification of a flexible key for any given Node/Link information such that global uniqueness of the TE node/link is ensured.



The value contains one or more Node Descriptor Sub-TLVs defined in Section 9.2.4.

9.2.3. Remote TE Node Descriptors TLV

The Remote TE Node Descriptors contains Node Descriptors for the node anchoring the remote end of the link. This TLV MUST be included in the TE Report when during a given PCEP session a TE link is first reported to a PCE. A PCC sends to a PCE the first TE Report either during State Synchronization, or when a new TE link is learned at the PCC. The length of this TLV is variable. The value contains one or more Node Descriptor Sub-TLVs defined in Section 9.2.4.



9.2.4. TE Node Descriptors Sub-TLVs

The Node Descriptor Sub-TLV type Type and lengths are listed in the following table:

Sub-TLV	Description	Length
TBD10	Autonomous System	4
TBD11	BGP-LS Identifier	4
TBD12	OSPF Area-ID	4
TBD13	Router-ID	Variable

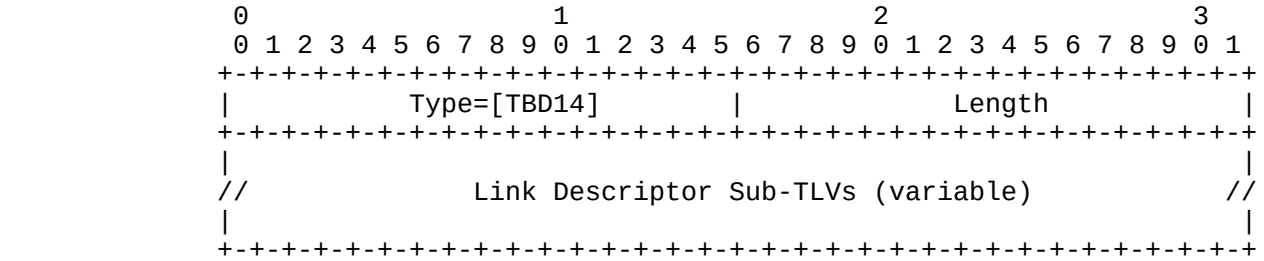
The sub-TLV values in Node Descriptor TLVs are defined as follows (similar to [I-D.ietf-idr-ls-distribution]):

- Autonomous System: opaque value (32 Bit AS Number)
- BGP-LS Identifier: opaque value (32 Bit ID). In conjunction with ASN, uniquely identifies the BGP-LS domain as described in [I-D.ietf-idr-ls-distribution]. This sub-TLV is present only if the node implements BGP-LS and the ID is set by the operator.
- Area ID: It is used to identify the 32 Bit area to which the TE object belongs. Area Identifier allows the different TE objects of the same router to be discriminated.
- Router ID: opaque value. Usage is described in [I-D.ietf-idr-ls-distribution] for IGP Router ID. In case only local TE information is transported and PCE learns TED only from PCEP, it contain the unique local TE IPv4 or IPv6 router ID.
- There can be at most one instance of each sub-TLV type present in any Node Descriptor.

9.2.5. TE Link Descriptors TLV

The TE Link Descriptors TLV contains Link Descriptors for each TE link. This TLV MUST be included in the TE Report when during a given PCEP session a TE link is first reported to a PCE. A PCC sends to a PCE the first TE Report either during State Synchronization, or when a new TE link is learned at the PCC. The length of this TLV is variable. The value contains one or more TE Link Descriptor Sub-TLVs

The 'TE Link descriptor' TLVs uniquely identify a link among multiple parallel links between a pair of anchor routers similar to [I-D.ietf-idr-ls-distribution].



The Link Descriptor Sub-TLV type and lengths are listed in the following table:

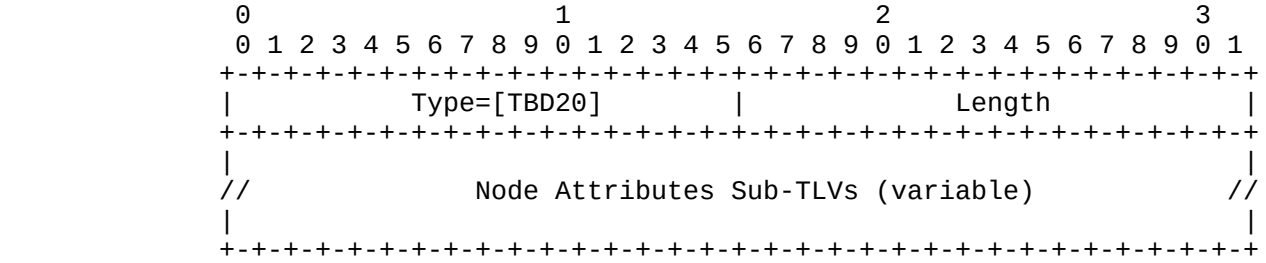
Sub-TLV	Description	IS-IS TLV /Sub-TLV	Value defined in:
TBD15	Link Local/Remote Identifiers	22/4	[RFC5307]/1.1
TBD16	IPv4 interface address	22/6	[RFC5305]/3.2
TBD17	IPv4 neighbor address	22/8	[RFC5305]/3.3
TBD18	IPv6 interface address	22/12	[RFC6119]/4.2
TBD19	IPv6 neighbor address	22/13	[RFC6119]/4.3

The format and semantics of the 'value' fields in most 'Link Descriptor' sub-TLVs correspond to the format and semantics of value fields in IS-IS Extended IS Reachability sub-TLVs, defined in [RFC5305], [RFC5307] and [RFC6119]. Although the encodings for 'Link Descriptor' TLVs were originally defined for IS-IS, the TLVs can carry data sourced either by IS-IS or OSPF or direct.

The information about a link present in the LSA/LSP originated by the local node of the link determines the set of sub-TLVs in the Link Descriptor of the link as described in [I-D.ietf-idr-ls-distribution].

9.2.6. TE Node Attributes TLV

This is an optional, non-transitive attribute that is used to carry TE node attributes. The TE node attribute TLV may be encoded in the TE node Object.

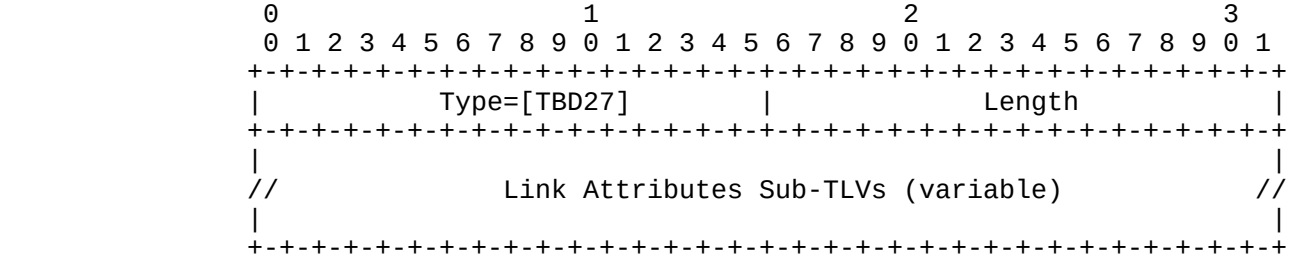


The Node Attributes Sub-TLV type and lengths are listed in the following table:

Sub TLV	Description	Length	Value defined in:
TBD21	Node Flag Bits	1	[I-D.ietf-idr-ls-distribution]/3.3.1.1
TBD22	Opaque Node Properties	variable	[I-D.ietf-idr-ls-distribution]/3.3.1.5
TBD23	Node Name	variable	[I-D.ietf-idr-ls-distribution]/3.3.1.3
TBD24	IS-IS Area Identifier	variable	[I-D.ietf-idr-ls-distribution]/3.3.1.2
TBD25	IPv4 Router-ID of Local Node	4	[RFC5305]/4.3
TBD26	IPv6 Router-ID of Local Node	16	[RFC6119]/4.1

9.2.7. TE Link Attributes TLV

TE Link attribute TLV may be encoded in the TE Link Object. The format and semantics of the 'value' fields in some 'Link Attribute' sub-TLVs correspond to the format and semantics of value fields in IS-IS Extended IS Reachability sub-TLVs, defined in [RFC5305], [RFC5307] and [I-D.ietf-idr-ls-distribution]. Although the encodings for 'Link Attribute' TLVs were originally defined for IS-IS, the TLVs can carry data sourced either by IS-IS or OSPF or direct.



The following 'Link Attribute' sub-TLVs are are valid :

Sub-TLV	Description	IS-IS TLV /Sub-TLV BGP-LS TLV	Defined in:
TBD28	IPv4 Router-ID of Local Node	134/---	[RFC5305]/4.3
TBD29	IPv6 Router-ID of Local Node	140/---	[RFC6119]/4.1
TBD30	IPv4 Router-ID of Remote Node	134/---	[RFC5305]/4.3
TBD31	IPv6 Router-ID of Remote Node	140/---	[RFC6119]/4.1
TBD32	Link Local/Remote Identifiers	22/4	[RFC5307]/1.1
TBD33	Administrative group (color)	22/3	[RFC5305]/3.1
TBD34	Maximum link bandwidth	22/9	[RFC5305]/3.3
TBD35	Max. reservable link bandwidth	22/10	[RFC5305]/3.5
TBD36	Unreserved bandwidth	22/11	[RFC5305]/3.6
TBD37	TE Default Metric	22/18	[I-D.ietf-idr-ls-distribution] /3.3.2.3
TBD38	Link Protection Type	22/20	[RFC5307]/1.2
TBD39	MPLS Protocol Mask	1094	[I-D.ietf-idr-ls-distribution] /3.3.2.2
TBD40	IGP Metric	1095	[I-D.ietf-idr-ls-distribution] /3.3.2.4
TBD41	Shared Risk Link Group	1096	[I-D.ietf-idr-ls-distribution] /3.3.2.5
TBD42	Opaque link attributes	1097	[I-D.ietf-idr-ls-distribution] /3.3.2.6
TBD43	Link Name attribute	1098	[I-D.ietf-idr-ls-distribution] /3.3.2.7

10. Other Considerations

10.1. Inter-AS Links

The main source of TE information is the IGP, which is not active on inter-AS links. In some cases, the IGP may have information of inter-AS links ([RFC5392], [RFC5316]). In other cases, an implementation SHOULD provide a means to inject inter-AS links into PCEP. The exact mechanism used to provision the inter-AS links is outside the scope of this document.

11. Security Considerations

This document extends PCEP to support TED population including a new TERpt message with new object and TLVs. Procedures and protocol extensions defined in this document do not effect the overall PCEP security model. See [RFC5440], [I-D.ietf-pce-pceps]. Tampering with the TERpt message may have an effect on path computations at PCE. It also provides adversaries an opportunity to eavesdrop and learn sensitive information and plan sophisticated attacks on the network infrastructure. The PCE implementation SHOULD provide mechanisms to prevent strains created by network flaps and amount of TED information. Thus it is suggested that any mechanism used for securing the transmission of other PCEP message be applied here as well. As a general precaution, it is RECOMMENDED that these PCEP extensions only be activated on authenticated and encrypted sessions belonging to the same administrative authority.

12. Manageability Considerations

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

12.1. Control of Function and Policy

In addition to configuring specific PCEP session parameters, as specified in section 8.1 of [RFC5440], a PCE or PCC implementation MUST allow configuring the TED PCEP capability. A PCC SHOULD allow the operator to specify an TED population policy where TERpt are sent to which PCE.

12.2. Information and Data Models

PCEP session configuration and information in the PCEP MIB module SHOULD be extended to include advertised TED capabilities, TED synchronization status and TED etc.

12.3. Liveness Detection and Monitoring

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

12.4. Verify Correct Operations

Mechanisms defined in section 8.4 of [RFC5440] also apply to PCEP protocol extensions defined in this document. In addition to monitoring parameters defined in [RFC5440], a PCEP implementation with TED SHOULD provide the following parameters:

- Total number of TE Reports
- Number of TE nodes and links
- Number of dropped TERpt messages

12.5. Requirements On Other Protocols

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

12.6. Impact On Network Operations

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

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

13. IANA Considerations

14. Acknowledgments

This document borrows some of the structure and text from the [I-D.ietf-pce-stateful-pce].

15. References

15.1. Normative References

[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.

15.2. Informative References

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

[RFC4203] Kompella, K. and Y. Rekhter, "OSPF Extensions in Support of Generalized Multi-Protocol Label Switching (GMPLS)", RFC 4203, October 2005.

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

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

[RFC5307] Kompella, K. and Y. Rekhter, "IS-IS Extensions in Support of Generalized Multi-Protocol Label Switching (GMPLS)", RFC 5307, October 2008.

[RFC5316] Chen, M., Zhang, R., and X. Duan, "ISIS Extensions in Support of Inter-Autonomous System (AS) MPLS and GMPLS Traffic Engineering", RFC 5316, December 2008.

[RFC5392] Chen, M., Zhang, R., and X. Duan, "OSPF Extensions in Support of Inter-Autonomous System (AS) MPLS and GMPLS Traffic Engineering", RFC 5392, January 2009.

[RFC6119] Harrison, J., Berger, J., and M. Bartlett, "IPv6 Traffic Engineering in IS-IS", RFC 6119, February 2011.

[RFC6549] Lindem, A., Roy, A., and S. Mirtorabi, "OSPFv2 Multi-Instance Extensions", RFC 6549, March 2012.

[RFC6822] Previdi, S., Ginsberg, L., Shand, M., Roy, A., and D. Ward, "IS-IS Multi-Instance", RFC 6822, December 2012.

[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-10 (work in progress), October 2014.

[I-D.ietf-pce-pceps]
Lopez, D., Dios, O., Wu, W., and D. Dhody, "Secure Transport for PCEP", draft-ietf-pce-pceps-02 (work in progress), October 2014.

[I-D.ietf-idr-ls-distribution]
Gredler, H., Medved, J., Previdi, S., Farrel, A., and S. Ray, "North-Bound Distribution of Link-State and TE Information using BGP", draft-ietf-idr-ls-distribution-10 (work in progress), January 2015.

[I-D.lee-pce-transporting-te-data]
Lee, Y. and z. zhenghaomian@huawei.com, "PCE in Support of Transporting Traffic Engineering Data", draft-lee-pce-transporting-te-data-01 (work in progress), October 2014.

Appendix A. Contributor Addresses

Udayasree Palle
Huawei Technologies
Divyashree Techno Park, Whitefield
Bangalore, Karnataka 560037
India

EMail: udayasree.palle@huawei.com

Sergio Belotti
Alcatel-Lucent
Italy

EMail: sergio.belotti@alcatel-lucent.com

Authors' Addresses

Dhruv Dhody
Huawei Technologies
Divyashree Techno Park, Whitefield
Bangalore, Karnataka 560037
India

EMail: dhruv.ietf@gmail.com

Young Lee
Huawei Technologies
5340 Legacy Drive, Building 3
Plano, TX 75023
USA

EMail: leeyoung@huawei.com

Daniele Ceccarelli
Ericsson
Torshamnsgatan,48
Stockholm
Sweden

EMail: daniele.ceccarelli@ericsson.com

PCE Working Group D. Dhody
Internet-Draft Q. Wu
Intended status: Standards Track Huawei
Expires: December 28, 2014 June 26, 2014

PCEP Extensions for service segment used in Segment Routing
draft-dw-pce-service-segment-routing-00

Abstract

Segment Routing (SR) technology leverages the source routing and tunneling paradigms where a source node can choose a path without relying on hop-by-hop signaling. The same mechanism can also be utilized for Service Function Chaining (SFC) to steer packets through service functions performing specific services such as DPI, Firewall, accounting etc.

This document specifies extensions to the Path Computation Element Protocol (PCEP) that allow a stateful PCE to compute and instantiate SR-TE paths that have service functions (SF) (or service segments) involved.

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 December 28, 2014.

Table of Contents

1. Introduction 2
2. Conventions used in this document 3
2.1. Terminology 3
3. Overview of PCEP Operation in SR Networks for Service Chaining 4
4. Object Formats 4
4.1. The SR-ERO Subobject extension for service segment support 5
4.2. Service Segment SR-ERO Processing 6
5. Backward Compatibility 6
6. Management Considerations 7
7. Security Considerations 7
8. IANA Considerations 7
9. References 7
9.1. Normative References 7
9.2. Informative References 7
Appendix A. Examples 7
Authors' Addresses 8

1. Introduction

Segment Routing (SR) enables Traffic Engineering (TE) without relying on a hop-by-hop signaling. It depends only on "segments" that are advertised by Interior Gateway Protocols (IGPs). These segments made by -

- Node Segment
- Adjacency Segment
- Anycast Segment
- IGP-Prefix Segment

Further to this list, a segment may also be identify a particular service or service function (SF). [I-D.filsfils-spring-segment-routing-use-cases] describes service-segment. A service-segment may also be used to represent a set of SF instances. In a case where it is required to steer the packet

through specific treatment or SF (DPI, firewall..) offered by node(s) in the path, a combination of node-segment and service-segment can be used.

A stateful PCE can be used for computing one or more SR-TE paths taking into account various constraints and objective functions. Once a path is chosen, the stateful PCE can instantiate an SR-TE path on a PCC using PCEP extensions specified in [I-D.sivabalan-pce-segment-routing].

An SR-TE path is defined as a path that consists of one or more SID(s) where each SID is associated with the identifier that represents the node or adjacency corresponding to the SID. This document extends the SR-TE path to use Service-SID(s) in the path as well.

The means by which the PCE learns about the Service-SID (e.g., learnt over a management interface or through a variety of other mechanisms) is beyond scope of this document.

2. 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 RFC2119 [RFC2119].

2.1. Terminology

The following terminologies are used in this document:

ERO: Explicit Route Object

IGP: Interior Gateway Protocol

LSR: Label Switching Router

PCC: Path Computation Client

PCE: Path Computation Element

PCEP: Path Computation Element Protocol

SF: Service Function

SFC: Service Function Chaining

SID: Segment Identifier

SR: Segment Routing

SR-ERO: Segment Routed Explicit Route Object

SR Path: Segment Routed Path

SR-TE: Segment Routed Traffic Engineering

3. Overview of PCEP Operation in SR Networks for Service Chaining

In SR networks, an ingress node of an SR path appends all outgoing packets with an SR header consisting of a list of Segment IDs (SIDs). The header has all necessary information to guide the packets from the ingress node to the egress node of the path, and hence there is no need for any signaling protocol. In the [I-D.sivabalan-pce-segment-routing], an SID represents either a nodal segment representing a path to a node or adjacency segment representing path over a specific adjacency. In this document,we allow SID also can represent a service segment representing a specific treatment or SF.

In a PCEP session, path information is carried in the Explicit Route Object (ERO), which consists of a sequence of subobjects. In this document, a PCE needs to specify EROs containing SID of service segments (or service-SID), and a PCC needs to be capable of processing such ERO sub-objects.

The SR-ERO Subobject defined in the [I-D.sivabalan-pce-segment-routing] can be used to carry SID of service segment. An SR-ERO containing SID of service segment can be included in the same PCEP messages specified in the [I-D.sivabalan-pce-segment-routing].

[Editor’s Note: Another option for [I-D.sivabalan-pce-segment-routing] is to define only the SR related sub-object which can be carried within the existing ERO object with no need to create a new SR-ERO object.]

When a PCEP session between a PCC and a PCE is established, the corresponding PCEP operation is same as defined in the [I-D.sivabalan-pce-segment-routing].

4. Object Formats

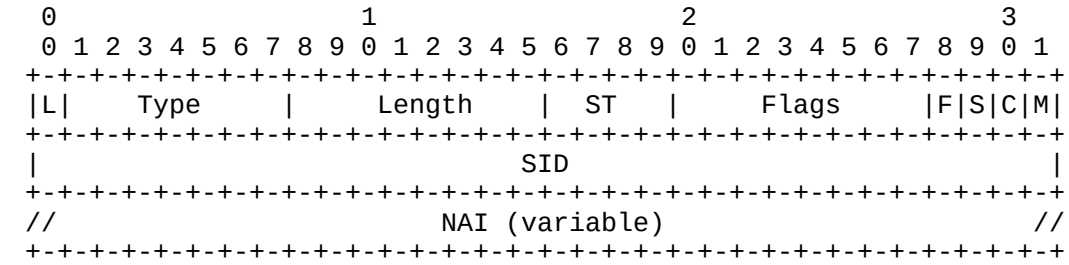
In the [I-D.sivabalan-pce-segment-routing], an SR-TE path is defined as a path that consists of one or more SID(s) where each SID is associated with the identifier that represents the node or adjacency corresponding to the SID. In this document, we allow the SR-TE path

to include one or more SID of service segments (called service-SID) that are inserted along with node segments in SR-TE path. A service-segment may also be used to represent a set of SF instances. The service-SID is local to the node where the service resides, thus a combination of node-segment and service-segment are used together.

4.1. The SR-ERO Subobject extension for service segment support

The SR-ERO Subobject is defined in section 5.3.1 of [I-D.sivabalan-pce-segment-routing] and as an mandatory subobject used to advertise SID and NAI ('Node or Adjacency Identifier') associated with SID. In this document, we extend the existing SR-ERO Subobject as specified in section 5.3.1 of [I-D.sivabalan-pce-segment-routing] to represent service-SID of the service segment.

The SR-ERO Subobject as described in [I-D.sivabalan-pce-segment-routing]:



L

The L bit SHOULD NOT be set, so that the subobject represents a strict hop in the explicit route in case of service-segment.

Type

The Type is as per [I-D.sivabalan-pce-segment-routing].

Length

The Length is as per [I-D.sivabalan-pce-segment-routing].

ST

The ST (SID Type) field is set to specify service-SID. A new SID-Type values is to be assigned.

Flags

All flags (M, C, S, F bit) are as per [I-D.sivabalan-pce-segment-routing].

SID

The SID value represents an service segment as described in [I-D.filsfils-spring-segment-routing-use-cases].

NAI

The NAI for service-segment may be defined in future.

4.2. Service Segment SR-ERO Processing

When the SID represents a service segment (as per the SID Type - ST field), its value is local to node segment offering the service. Thus Service-SID MUST be assosiated with a node-SID preceding it in the SR-ERO. Note that multiple services may be offered by the same node, and in this case node-SID maybe followed by multiple Service-SID. NAI value for service-SID may be defined in future.

If a service segment (or service-SID) cannot be assosiated with a node segment (or node-SID), PCEP speaker MUST send a PCE error with Error-Type = "Reception of an invalid object" and Error-Value ="Segment List Order Error".

The rest of the processing rules are as per [I-D.sivabalan-pce-segment-routing].

5. Backward Compatibility

Backward Compatibility consideration described in section 8 of [I-D.sivabalan-pce-segment-routing] can be applied for service segment support as well.

6. Management Considerations

Management consideration described in section 9 of [I-D.sivabalan-pce-segment-routing] can be applied to service segment support as well.

7. Security Considerations

The security considerations described in [RFC5440] and [I-D.sivabalan-pce-segment-routing] apply.

8. IANA Considerations

TBD.

9. References

9.1. Normative References

[I-D.sivabalan-pce-segment-routing] Sivabalan, S., Medved, J., Filsfils, C., Crabbe, E., and R. Raszuk, "PCEP Extensions for Segment Routing", draft-sivabalan-pce-segment-routing-02 (work in progress), October 2013.

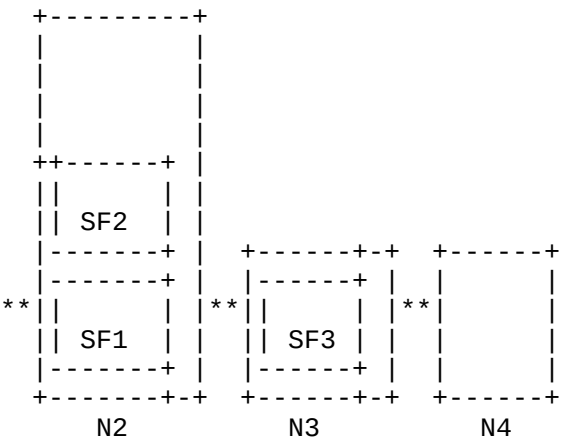
[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

9.2. Informative References

[I-D.filsfils-spring-segment-routing-use-cases] Filsfils, C., Francois, P., Previdi, S., Decraene, B., Litkowski, S., Horneffer, M., Milojevic, I., Shakir, R., Ytti, S., Henderickx, W., Tantsura, J., Kini, S., and E. Crabbe, "Segment Routing Use Cases", draft-filsfils-spring-segment-routing-use-cases-00 (work in progress), March 2014.

Appendix A. Examples

Consider the below example-



- N1 is Ingress;
- N4 is Egress;
- N2 has two services hosted identified as SF1 and SF2;
- N3 hase one service hosted identified as SF3.
- The service chain requires packet to steer through SF1, SF2, SF3.

The SR-ERO for the SR-TE path including the service segment would be -

[{SID_N2, SID_SF1, SID_SF2}, {SID_N3, SID_SF3}, {SID_N4}]

Authors' Addresses

Dhruv Dhody
Huawei
Leela Palace
Bangalore, Karnataka 560008
INDIA

Email: dhruv.ietf@gmail.com

Qin Wu
Huawei
101 Software Avenue, Yuhua District
Nanjing, Jiangsu 210012
China

Email: bill.wu@huawei.com

PCE Working Group D. Dhody
Internet-Draft Q. Wu
Intended status: Standards Track Huawei
Expires: September 7, 2015 March 6, 2015

PCEP Extensions for service segment used in Segment Routing
draft-dw-pce-service-segment-routing-01

Abstract

Segment Routing (SR) technology leverages the source routing and tunneling paradigms where a source node can choose a path without relying on hop-by-hop signaling.

This document specifies extensions to the Path Computation Element Protocol (PCEP) that allow a stateful PCE to compute and instantiate SR-TE paths that also have a local service segments involved.

Table of Contents

1. Introduction 2
2. Conventions used in this document 3
2.1. Terminology 3
3. Overview of PCEP Operation in SR Networks for Service Chaining 4
4. Object Formats 4
4.1. The SR-ERO Subobject extension for service segment support 5
4.2. Service Segment SR-ERO Processing 6
5. Backward Compatibility 6
6. Management Considerations 6
7. Security Considerations 6
8. IANA Considerations 7
9. References 7
9.1. Normative References 7
9.2. Informative References 7
Appendix A. Examples 7
Authors' Addresses 8

1. Introduction

Segment Routing (SR) enables Traffic Engineering (TE) without relying on a hop-by-hop signaling. It depends only on "segments" that are advertised by Interior Gateway Protocols (IGPs). These segments made by -

- Node Segment
- Adjacency Segment
- Anycast Segment
- IGP-Prefix Segment

Further to this list, a segment may also be identify a particular value added service or service function (SF). [I-D.filsfils-spring-segment-routing-use-cases] describes using local Service-Segment to stand for a BGP-VPN service in an example. A service-segment may also be used to represent specific treatment offered by SR enabled node(s) in the path, a combination of node-segment and service-segment can be used. The service segment is local to the SR enabled node.

A stateful PCE can be used for computing one or more SR-TE paths taking into account various constraints and objective functions. Once a path is chosen, the stateful PCE can instantiate an SR-TE path on a PCC using PCEP extensions specified in [I-D.ietf-pce-segment-routing].

An SR-TE path is defined as a path that consists of one or more SID(s) where each SID is associated with the identifier that represents the node or adjacency corresponding to the SID. This document extends the SR-TE path to use Service-SID(s) in the path as well.

The means by which the PCE learns about the Service-SID (e.g., learnt over a management interface or through a variety of other mechanisms) is beyond scope of this document.

2. 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 RFC2119 [RFC2119].

2.1. Terminology

The following terminologies are used in this document:

ERO: Explicit Route Object

IGP: Interior Gateway Protocol

LSR: Label Switching Router

PCC: Path Computation Client

PCE: Path Computation Element

PCEP: Path Computation Element Protocol

SF: Service Function

SFC: Service Function Chaining

SID: Segment Identifier

SR: Segment Routing

SR-ERO: Segment Routed Explicit Route Object

SR Path: Segment Routed Path

SR-TE: Segment Routed Traffic Engineering

3. Overview of PCEP Operation in SR Networks for Service Chaining

In SR networks, an ingress node of an SR path appends all outgoing packets with an SR header consisting of a list of Segment IDs (SIDs). The header has all necessary information to guide the packets from the ingress node to the egress node of the path, and hence there is no need for any signaling protocol. In the [I-D.ietf-pce-segment-routing], an SID represents either a nodal segment representing a path to a node or adjacency segment representing path over a specific adjacency. In this document,we allow SID also can represent a service segment representing a specific treatment or SF.

In a PCEP session, path information is carried in the Explicit Route Object (ERO), which consists of a sequence of subobjects. In this document, a PCE needs to specify EROs containing SID of service segments (or service-SID), and a PCC needs to be capable of processing such ERO sub-objects.

The SR-ERO Subobject defined in the [I-D.ietf-pce-segment-routing] can be used to carry SID of service segment. An SR-ERO containing SID of service segment can be included in the same PCEP messages specified in the [I-D.ietf-pce-segment-routing].

When a PCEP session between a PCC and a PCE is established, the corresponding PCEP operation is same as defined in the [I-D.ietf-pce-segment-routing].

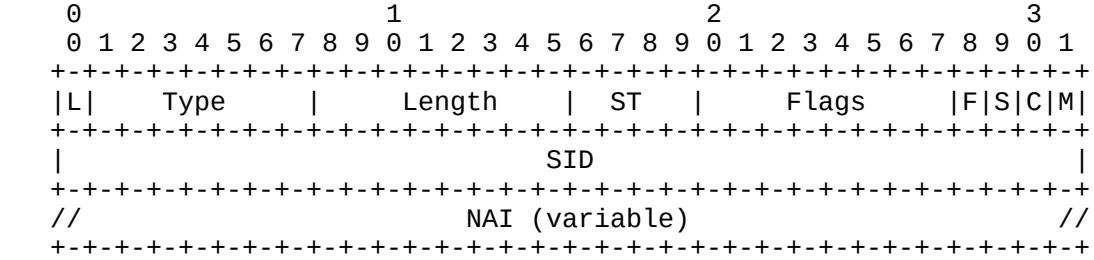
4. Object Formats

In the [I-D.ietf-pce-segment-routing], an SR-TE path is defined as a path that consists of one or more SID(s) where each SID is associated with the identifier that represents the node or adjacency corresponding to the SID. In this document, we allow the SR-TE path to include one or more SID of service segments (called service-SID) that are inserted along with node segments in SR-TE path. A service-segment may also be used to represent a set of SF instances. The service-SID is local to the node where the service resides, thus a combination of node-segment and service-segment are used together.

4.1. The SR-ERO Subobject extension for service segment support

The SR-ERO Subobject is defined in section 5.3.1 of [I-D.ietf-pce-segment-routing] and as an mandatory subobject used to advertise SID and NAI ('Node or Adjacency Identifier') associated with SID. In this document, we extend the existing SR-ERO Subobject as specified in section 5.3.1 of [I-D.ietf-pce-segment-routing] to represent service-SID of the service segment.

The SR-ERO Subobject as described in [I-D.ietf-pce-segment-routing]:



L

The L bit SHOULD NOT be set, so that the subobject represents a strict hop in the explicit route in case of service-segment.

Type

The Type is as per [I-D.ietf-pce-segment-routing].

Length

The Length is as per [I-D.ietf-pce-segment-routing].

ST

The ST (SID Type) field is set to specify service-SID. A new SID-Type values is to be assigned.

Flags

All flags (M, C, S, F bit) are as per [I-D.ietf-pce-segment-routing].

SID

The SID value represents an service segment as described in [I-D.filsfils-spring-segment-routing-use-cases].

NAI

The NAI for service-segment may be defined in future based on the service.

4.2. Service Segment SR-ERO Processing

When the SID represents a service segment (as per the SID Type - ST field), its value is local to node segment offering the service. Thus Service-SID MUST be associated with a node-SID preceding it in the SR-ERO. Note that multiple services may be offered by the same node, and in this case node-SID maybe followed by multiple Service-SID. NAI value for service-SID may be defined in future.

If a service segment (or service-SID) cannot be associated with a node segment (or node-SID), PCEP speaker MUST send a PCE error with Error-Type = "Reception of an invalid object" and Error-Value ="Segment List Order Error".

The rest of the processing rules are as per [I-D.ietf-pce-segment-routing].

5. Backward Compatibility

Backward Compatibility consideration described in section 8 of [I-D.ietf-pce-segment-routing] can be applied for service segment support as well.

6. Management Considerations

Management consideration described in section 9 of [I-D.ietf-pce-segment-routing] can be applied to service segment support as well.

7. Security Considerations

The security considerations described in [RFC5440] and [I-D.ietf-pce-segment-routing] apply.

8. IANA Considerations

TBD.

9. References

9.1. Normative References

[I-D.ietf-pce-segment-routing]
Sivabalan, S., Medved, J., Filsfils, C., Crabbe, E., Raszuk, R., Lopez, V., and J. Tantsura, "PCEP Extensions for Segment Routing", draft-ietf-pce-segment-routing-00 (work in progress), October 2014.

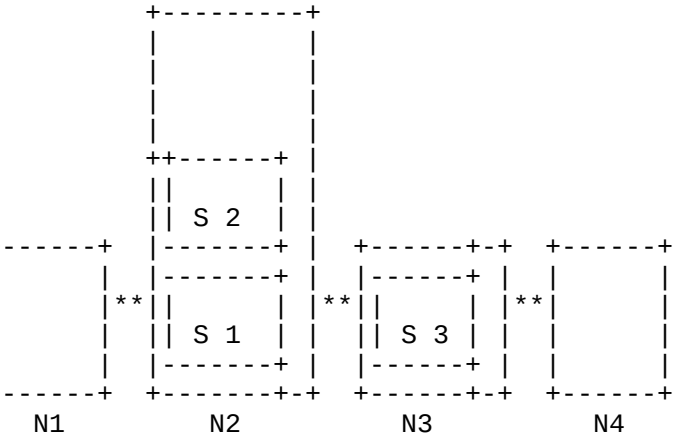
[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

9.2. Informative References

[I-D.filsfils-spring-segment-routing-use-cases]
Filsfils, C., Francois, P., Previdi, S., Decraene, B., Litkowski, S., Horneffer, M., Milojevic, I., Shakir, R., Ytti, S., Henderickx, W., Tantsura, J., Kini, S., and E. Crabbe, "Segment Routing Use Cases", draft-filsfils-spring-segment-routing-use-cases-01 (work in progress), October 2014.

Appendix A. Examples

Consider the below example-



- N1 is Ingress;
- N4 is Egress;
- N2 has two services hosted identified as S1 and S2;
- N3 hase one service hosted identified as S3.

The SR-ERO for the SR-TE path including the service segment would be -

[{SID_N2, SID_S1, SID_S2}, {SID_N3, SID_S3}, {SID_N4}]

Authors' Addresses

Dhruv Dhody
Huawei
Divyashree Techno Park, Whitefield
Bangalore, Karnataka 560037
India

Email: dhruv.ietf@gmail.com

Qin Wu
Huawei
101 Software Avenue, Yuhua District
Nanjing, Jiangsu 210012
China

Email: bill.wu@huawei.com

PCE Working Group
Internet-Draft
Intended status: Experimental
Expires: April 25, 2015

D. Dhody
U. Palle
Huawei Technologies
R. Casellas
CTTC

October 22, 2014

Standard Representation Of Domain-Sequence
draft-ietf-pce-pcep-domain-sequence-06

Abstract

The ability to compute shortest constrained Traffic Engineering Label Switched Paths (TE LSPs) in Multiprotocol Label Switching (MPLS) and Generalized MPLS (GMPLS) networks across multiple domains has been identified as a key requirement. In this context, a domain is a collection of network elements within a common sphere of address management or path computational responsibility such as an Interior Gateway Protocol (IGP) area or an Autonomous Systems (AS). This document specifies a standard representation and encoding of a Domain-Sequence, which is defined as an ordered sequence of domains traversed to reach the destination domain to be used by Path Computation Elements (PCEs) to compute inter-domain shortest constrained paths across a predetermined sequence of domains . This document also defines new subobjects to be used to encode domain identifiers.

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 April 25, 2015.

Table of Contents

1. Introduction 3
1.1. Requirements Language 4
2. Terminology 4
3. Detail Description 5
3.1. Domains 5
3.2. Domain-Sequence 5
3.3. Standard Representation 6
3.4. Include Route Object (IRO) 7
3.4.1. Subobjects 7
3.4.1.1. Autonomous system 8
3.4.1.2. IGP Area 8
3.4.2. Update in IRO specification 9
3.4.3. IRO for domain-sequence 10
3.5. Exclude Route Object (XRO) 11
3.5.1. Subobjects 12
3.5.1.1. Autonomous system 12
3.5.1.2. IGP Area 13
3.6. Explicit Exclusion Route Subobject (EXRS) 14
3.7. Explicit Route Object (ERO) 14
4. Other Considerations 15
4.1. Inter-Area Path Computation 15
4.2. Inter-AS Path Computation 17
4.2.1. Example 1 17
4.2.2. Example 2 19
4.3. Boundary Node and Inter-AS-Link 21
4.4. PCE Serving multiple Domains 21
4.5. P2MP 22
4.6. Hierarchical PCE 22
4.7. Relationship to PCE Sequence 24
4.8. Relationship to RSVP-TE 24
5. IANA Considerations 25

5.1. New Subobjects 25
5.2. Error Object Field Values 25
6. Security Considerations 25
7. Manageability Considerations 26
7.1. Control of Function and Policy 26
7.2. Information and Data Models 26
7.3. Liveness Detection and Monitoring 26
7.4. Verify Correct Operations 26
7.5. Requirements On Other Protocols 26
7.6. Impact On Network Operations 27
8. Acknowledgments 27
9. References 27
9.1. Normative References 27
9.2. Informative References 27

1. Introduction

A PCE may be used to compute end-to-end paths across multi-domain environments using a per-domain path computation technique [RFC5152]. The so called backward recursive path computation (BRPC) mechanism [RFC5441] defines a PCE-based path computation procedure to compute inter-domain constrained (G)MPLS TE LSPs. However, both per-domain and BRPC techniques assume that the sequence of domains to be crossed from source to destination is known, either fixed by the network operator or obtained by other means. Also for inter-domain point-to-multi-point (P2MP) tree computation, [RFC7334] assumes the domain-tree is known in priori.

The list of domains (domain-sequence) in a point-to-point (P2P) path or a point-to-multipoint (P2MP) tree is usually a constraint in the path computation request. A PCE determines the next PCE to forward the request based on the domain-sequence. In a multi-domain path computation, a PCC MAY indicate the sequence of domains to be traversed using the Include Route Object (IRO) defined in [RFC5440].

When the sequence of domains is not known in advance, the Hierarchical PCE (H-PCE) [RFC6805] architecture and mechanisms can be used to determine the end-to-end Domain-Sequence.

This document defines a standard way to represent and encode a Domain-Sequence in various deployment scenarios including P2P, P2MP and H-PCE.

The Domain-Sequence (the set of domains traversed to reach the destination domain) is either administratively predetermined or discovered by some means (H-PCE) that is outside of the scope of this document.

[RFC5440] defines the Include Route Object (IRO) and the Explicit Route Object (ERO); [RFC5521] defines the Exclude Route Object (XRO) and the Explicit Exclusion Route Subobject (EXRS); The use of Autonomous System (AS) (albeit with a 2-Byte AS number) as an abstract node representing domain is defined in [RFC3209], this document specifies new subobjects to include or exclude domains such as an IGP area or an Autonomous Systems (4-Byte as per [RFC4893]).

Further, the domain identifier may simply act as delimiter to specify where the domain boundary starts and ends.

This is a companion document to Resource ReserVation Protocol - Traffic Engineering (RSVP-TE) extensions for the domain identifiers [DOMAIN-SUBOBJ].

1.1. Requirements Language

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

2. Terminology

The following terminology is used in this document.

ABR: OSPF Area Border Router. Routers used to connect two IGP areas.

AS: Autonomous System.

ASBR: Autonomous System Boundary Router.

BN: Boundary Node, Can be an ABR or ASBR.

BRPC: Backward Recursive Path Computation

Domain: As per [RFC4655], any collection of network elements within a common sphere of address management or path computational responsibility. Examples of domains include Interior Gateway Protocol (IGP) areas and Autonomous Systems (ASs).

Domain-Sequence: An ordered sequence of domains traversed to reach the destination domain.

ERO: Explicit Route Object

H-PCE: Hierarchical PCE

IGP: Interior Gateway Protocol. Either of the two routing protocols, Open Shortest Path First (OSPF) or Intermediate System to Intermediate System (IS-IS).

IRO: Include Route Object

IS-IS: Intermediate System to Intermediate System.

OSPF: Open Shortest Path First.

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.

P2MP: Point-to-Multipoint

P2P: Point-to-Point

RSVP: Resource Reservation Protocol

TE LSP: Traffic Engineering Label Switched Path.

XRO: Exclude Route Object

3. Detail Description

3.1. Domains

[RFC4726] and [RFC4655] define domain as a separate administrative or geographic environment within the network. A domain may be further defined as a zone of routing or computational ability. Under these definitions a domain might be categorized as an AS or an IGP area. Each AS can be made of several IGP areas. In order to encode a Domain-Sequence, it is required to uniquely identify a domain in the Domain-Sequence. A domain can be uniquely identified by area-id or AS or both.

3.2. Domain-Sequence

A domain-sequence is an ordered sequence of domains traversed to reach the destination domain.

A domain-sequence can be applied as a constraint and carried in path computation request to PCE(s). A domain-sequence can also be the

result of a path computation. For example, in the case of H-PCE [RFC6805] Parent PCE MAY send the Domain-Sequence as a result in a path computation reply.

In a P2P path, the domains listed appear in the order that they are crossed. In a P2MP path, the domain tree is represented as list of domain sequences.

A domain-sequence enables a PCE to select the next PCE to forward the path computation request based on the domain information.

A PCC or PCE MAY add an additional constraints covering which Boundary Nodes (ABR or ASBR) or Border links (Inter-AS-link) MUST be traversed while defining a Domain-Sequence.

Thus a Domain-Sequence MAY be made up of one or more of -

- AS Number
- Area ID
- Boundary Node ID
- Inter-AS-Link Address

Consequently, a Domain-Sequence can be used:

1. by a PCE in order to discover or select the next PCE in a collaborative path computation, such as in BRPC [RFC5441];
2. by the Parent PCE to return the Domain-Sequence when unknown, this can further be an input to BRPC procedure [RFC6805];
3. by a PCC (or PCE) to constraint the domains used in a H-PCE path computation, explicitly specifying which domains to be expanded;
4. by a PCE in per-domain path computation model [RFC5152] to identify the next domain(s);

3.3. Standard Representation

Domain-Sequence MAY appear in PCEP Messages, notably in -

- Include Route Object (IRO): As per [RFC5440], used to specify set of network elements that MUST be traversed. The subobjects in IRO are used to specify the domain-sequence that MUST be traversed to reach the destination.

- o Exclude Route Object (XRO): As per [RFC5521], used to specify certain abstract nodes that MUST be excluded from whole path. The subobjects in XRO are used to specify certain domains that MUST be avoided to reach the destination.

- o Explicit Exclusion Route Subobject (EXRS): As per [RFC5521], used to specify exclusion of certain abstract nodes between a specific pair of nodes. EXRS are a subobject inside the IRO. These subobjects are used to specify the domains that must be excluded between two abstract nodes.

- o Explicit Route Object (ERO): As per [RFC5440], used to specify a computed path in the network. For example, in the case of H-PCE [RFC6805] Parent PCE MAY send the Domain-Sequence as a result in a path computation reply using ERO.

3.4. Include Route Object (IRO)

As per [RFC5440], IRO (Include Route Object) can be used to specify that the computed path MUST traverse a set of specified network elements or abstract nodes.

3.4.1. Subobjects

Some subobjects are defined in [RFC3209], [RFC3473], [RFC3477] and [RFC4874], but new subobjects related to Domain-Sequence are needed.

The following subobject types are used in IRO.

Type	Subobject
1	IPv4 prefix
2	IPv6 prefix
4	Unnumbered Interface ID
32	Autonomous system number (2 Byte)
33	Explicit Exclusion (EXRS)

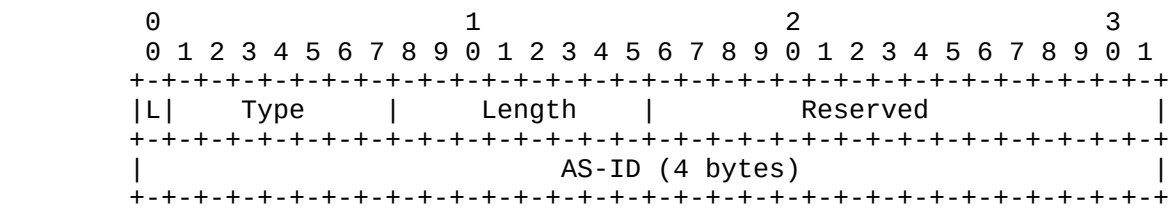
This document extends the above list to support 4-Byte AS numbers and IGP Areas.

Type	Subobject
TBD1	Autonomous system number (4 Byte)
TBD2	OSPF Area id
TBD3	ISIS Area id

3.4.1.1. Autonomous system

[RFC3209] already defines 2 byte AS number.

To support 4 byte AS number as per [RFC4893] following subobject is defined:



L: The L bit is an attribute of the subobject as defined in [RFC3209] and usage in IRO subobject updated in [IRO-UPDATE].

Type: (TBD1 by IANA) indicating a 4-Byte AS Number.

Length: 8 (Total length of the subobject in bytes).

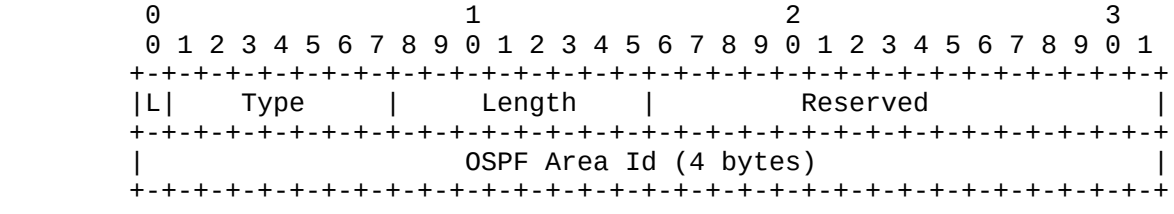
Reserved: Zero at transmission, ignored at receipt.

AS-ID: The 4-Byte AS Number. Note that if 2-Byte AS numbers are in use, the low order bits (16 through 31) should be used and the high order bits (0 through 15) should be set to zero.

3.4.1.2. IGP Area

Since the length and format of Area-id is different for OSPF and ISIS, following two subobjects are defined:

For OSPF, the area-id is a 32 bit number. The subobject is encoded as follows:



L: The L bit is an attribute of the subobject as defined in [RFC3209] and usage in IRO subobject updated in [IRO-UPDATE].

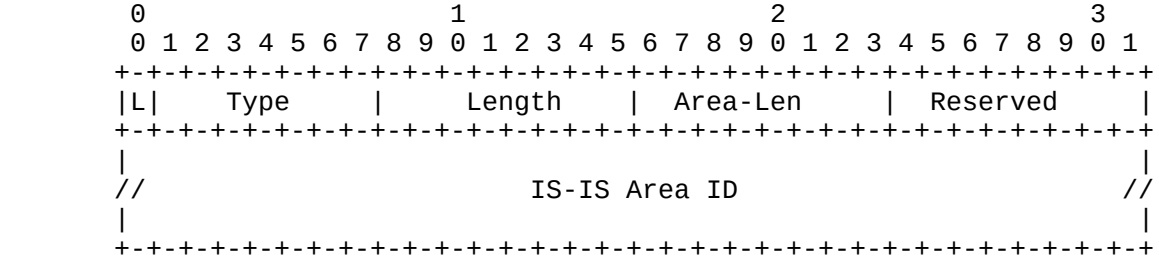
Type: (TBD2 by IANA) indicating a 4-Byte OSPF Area ID.

Length: 8 (Total length of the subobject in bytes).

Reserved: Zero at transmission, ignored at receipt.

OSPF Area Id: The 4-Byte OSPF Area ID.

For IS-IS, the area-id is of variable length and thus the length of the Subobject is variable. The Area-id is as described in IS-IS by ISO standard [ISO10589]. The subobject is encoded as follows:



L: The L bit is an attribute of the subobject as defined in [RFC3209] and usage in IRO subobject updated in [IRO-UPDATE].

Type: (TBD3 by IANA) indicating IS-IS Area ID.

Length: Variable. As per [RFC3209], the total length of the subobject in bytes, including the L, Type and Length fields. The Length MUST be at least 4, and MUST be a multiple of 4.

Area-Len: Variable (Length of the actual (non-padded) IS-IS Area Identifier in octets; Valid values are from 2 to 11 inclusive).

Reserved: Zero at transmission, ignored at receipt.

IS-IS Area Id: The variable-length IS-IS area identifier. Padded with trailing zeroes to a four-byte boundary.

3.4.2. Update in IRO specification

[RFC5440] describes IRO as an optional object used to specify that the computed path MUST traverse a set of specified network elements. It further state that the L bit of such sub-object has no meaning within an IRO. It did not mention if IRO is an ordered or un-ordered list of sub-objects.

An update to IRO specification [IRO-UPDATE] makes IRO as an ordered list as well as support for loose bit (L-bit).

The use IRO for domain-sequence assumes the updated specification for IRO as per [IRO-UPDATE].

3.4.3. IRO for domain-sequence

Some subobjects for IRO are defined in [RFC3209], [RFC3473], [RFC3477] and [RFC4874], further some new subobjects related to Domain-Sequence are also added in this document as mentioned in Section 3.4.

The subobjects for IPv4, IPv6 and unnumbered Interface ID can be used to specify Boundary Node (ABR/ASBR) and Inter-AS-Links. The subobjects for AS Number (2 or 4 Byte) and IGP Area is used to specify the domain identifiers in the domain-sequence.

The IRO MAY have both intra-domain (from the context of the ingress PCC) and inter-domain (domain-sequence) subobjects in a sequence in which they must be traversed in the computed path.

Thus an IRO comprising of subobjects that represents a domain-sequence may constraints or define the domains involved in an inter-domain path computation, typically involving two or more collaborative PCEs.

A Domain-Sequence can have varying degrees of granularity; it is possible to have a Domain-Sequence composed of, uniquely, AS identifiers. It is also possible to list the involved areas for a given AS.

In any case, the mapping between domains and responsible PCEs is not defined in this document. It is assumed that a PCE that needs to obtain a "next PCE" from a Domain-Sequence is able to do so (e.g. via administrative configuration, or discovery).

A PCC builds an IRO to encode the Domain-Sequence, that the cooperating PCEs should compute an inter-domain shortest constrained paths across the specified sequence of domains.

For each inclusion, the PCC clears the L-bit to indicate that the PCE is required to include the domain, or sets the L-bit to indicate that the PCC simply desires that the domain be included in the domain-sequence.

If a PCE encounters a subobject that it does not support or recognize, it MUST act according to the setting of the L-bit in the

subobject. If the L-bit is clear, the PCE MUST respond with a PCErr with Error-Type TBD4 "Unrecognized subobject" and set the Error-Value to the subobject type code. If the L-bit is set, the PCE MAY respond with a PCErr as already stated or MAY ignore the subobject: this choice is a local policy decision.

PCE MUST act according to the requirements expressed in the subobject. That is, if the L-bit is clear, the PCE(s) MUST produce a path that follows domain-sequence nodes in order identified by the subobjects in the path. If the L-bit is set, the PCE(s) SHOULD produce a path along the Domain-Sequence unless it is not possible to construct a path complying with the other constraints expressed in the request.

A successful path computation reported in a PCEP reply message (PCRep) MUST include an ERO to specify the path that has been computed as specified in [RFC5440] following the sequence of domains.

In a PCRep, PCE MAY also supply IRO (with domain sequence information) with the NO-PATH object indicating that the set of elements (domains) of the request's IRO prevented the PCEs from finding a path.

The Subobject types for domains (AS and IGP Area) affect the next domain selection as well as finding the PCE serving that domain.

Note that a particular domain in the domain-sequence can be identified by :-

- A single IGP Area: Only the IGP (OSPF or ISIS) Area subobject is used to identify the next domain. (Refer Figure 1)

- A single AS: Only the AS subobject is used to identify the next domain. (Refer Figure 2)

- Both an AS and an IGP Area: Combination of both AS and Area are used to identify the next domain. In this case the order is AS Subobject followed by Area. (Refer Figure 3)

The Subobjects representing an internal node, a Boundary Node or an Inter-AS-Link MAY influence the selection of the path as well.

3.5. Exclude Route Object (XRO)

The Exclude Route Object (XRO) [RFC5521] is an optional object used to specify exclusion of certain abstract nodes or resources from the whole path.

3.5.1. Subobjects

The following subobject types are defined to be used in XRO as defined in [RFC3209], [RFC3477], [RFC4874], and [RFC5521].

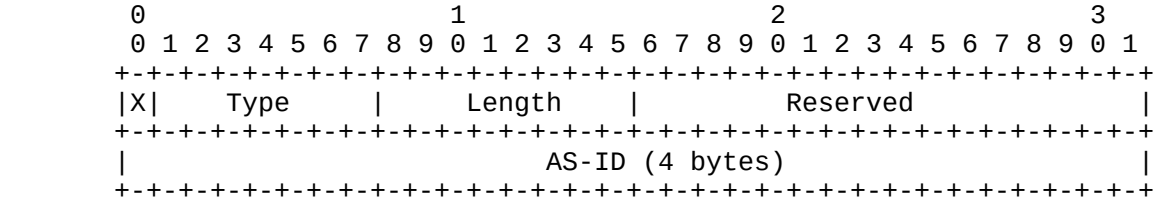
Type	Subobject
1	IPv4 prefix
2	IPv6 prefix
4	Unnumbered Interface ID
32	Autonomous system number (2 Byte)
34	SRLG
64	IPv4 Path Key
65	IPv6 Path Key

This document extends the above list to support 4-Byte AS numbers and IGP Areas.

Type	Subobject
TBD1	Autonomous system number (4 Byte)
TBD2	OSPF Area id
TBD3	ISIS Area id

3.5.1.1. Autonomous system

The new subobjects to support 4 byte AS and IGP (OSPF / ISIS) Area MAY also be used in the XRO to specify exclusion of certain domains in the path computation procedure.



The X-bit indicates whether the exclusion is mandatory or desired.

0: indicates that the AS specified MUST be excluded from the path computed by the PCE(s).

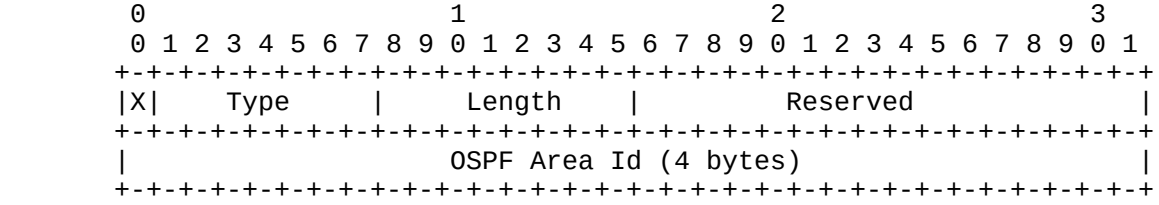
1: indicates that the AS specified SHOULD be avoided from the inter-domain path computed by the PCE(s), but MAY be included subject to PCE policy and the absence of a viable path that meets the other constraints.

All other fields are consistent with the definition in Section 3.4.

3.5.1.2. IGP Area

Since the length and format of Area-id is different for OSPF and ISIS, following two subobjects are defined:

For OSPF, the area-id is a 32 bit number. The subobject is encoded as follows:



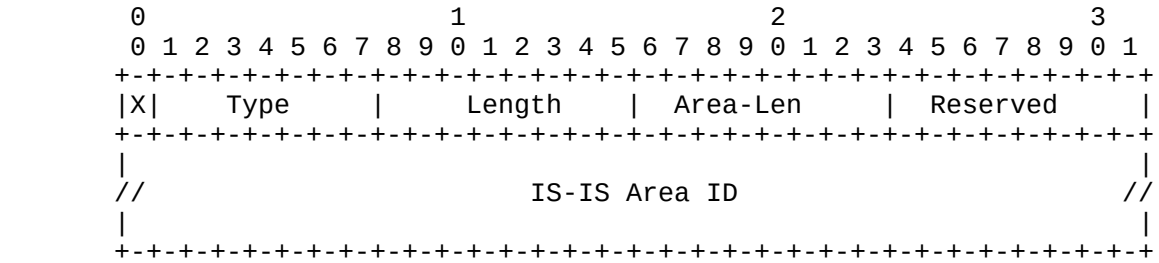
The X-bit indicates whether the exclusion is mandatory or desired.

0: indicates that the OSFF Area specified MUST be excluded from the path computed by the PCE(s).

1: indicates that the OSFF Area specified SHOULD be avoided from the inter-domain path computed by the PCE(s), but MAY be included subject to PCE policy and the absence of a viable path that meets the other constraints.

All other fields are consistent with the definition in Section 3.4.

For IS-IS, the area-id is of variable length and thus the length of the subobject is variable. The Area-id is as described in IS-IS by ISO standard [ISO10589]. The subobject is encoded as follows:



The X-bit indicates whether the exclusion is mandatory or desired.

0: indicates that the ISIS Area specified MUST be excluded from the path computed by the PCE(s).

1: indicates that the ISIS Area specified SHOULD be avoided from the inter-domain path computed by the PCE(s), but MAY be included subject to PCE policy and the absence of a viable path that meets the other constraints.

All other fields are consistent with the definition in Section 3.4.

If a PCE that supports XRO and encounters a subobject that it does not support or recognize, it MUST act according to the setting of the X-bit in the subobject. If the X-bit is clear, the PCE MUST respond with a PCErr with Error-Type TBD4 "Unrecognized subobject" and set the Error-Value to the subobject type code. If the X-bit is set, the PCE MAY respond with a PCErr as already stated or MAY ignore the subobject: this choice is a local policy decision.

All the other processing rules are as per [RFC5521].

3.6. Explicit Exclusion Route Subobject (EXRS)

Explicit Exclusion Route Subobject (EXRS) [RFC5521] is used to specify exclusion of certain abstract nodes between a specific pair of nodes.

The EXRS subobject may carry any of the subobjects defined for inclusion in the XRO, thus the new subobjects to support 4 byte AS and IGP (OSPF / ISIS) Area MAY also be used in the EXRS. The meanings of the fields of the new XRO subobjects are unchanged when the subobjects are included in an EXRS, except that scope of the exclusion is limited to the single hop between the previous and subsequent elements in the IRO.

All the processing rules are as per [RFC5521].

3.7. Explicit Route Object (ERO)

The Explicit Route Object (ERO) [RFC5440] is used to specify a computed path in the network. PCEP ERO subobject types correspond to RSVP-TE ERO subobject types as defined in [RFC3209], [RFC3473], [RFC3477], [RFC4873], [RFC4874], and [RFC5520].

Type	Subobject
1	IPv4 prefix
2	IPv6 prefix
3	Label
4	Unnumbered Interface ID
32	Autonomous system number (2 Byte)
33	Explicit Exclusion (EXRS)
37	Protection
64	IPv4 Path Key
65	IPv6 Path Key

This document extends the above list to support 4-Byte AS numbers and IGP Areas.

Type	Subobject
TBD1	Autonomous system number (4 Byte)
TBD2	OSPF Area id
TBD3	ISIS Area id

The new subobjects to support 4 byte AS and IGP (OSPF / ISIS) Area MAY also be used in the ERO to specify an abstract node (a group of nodes whose internal topology is opaque to the ingress node of the LSP). Using this concept of abstraction, an explicitly routed LSP can be specified as a sequence of domains.

In case of Hierarchical PCE [RFC6805], a Parent PCE MAY be requested to find the domain-sequence. Refer example in Section 4.6.

The format of the new ERO subobjects is similar to new IRO subobjects, refer Section 3.4.

4. Other Considerations

The examples in this section are for illustration purposes only; to show how the new subobjects may be encoded.

4.1. Inter-Area Path Computation

In an inter-area path computation where the ingress and the egress nodes belong to different IGP areas within the same AS, the Domain-Sequence MAY be represented using a ordered list of Area subobjects. The AS number MAY be skipped, as area information is enough to select the next PCE.

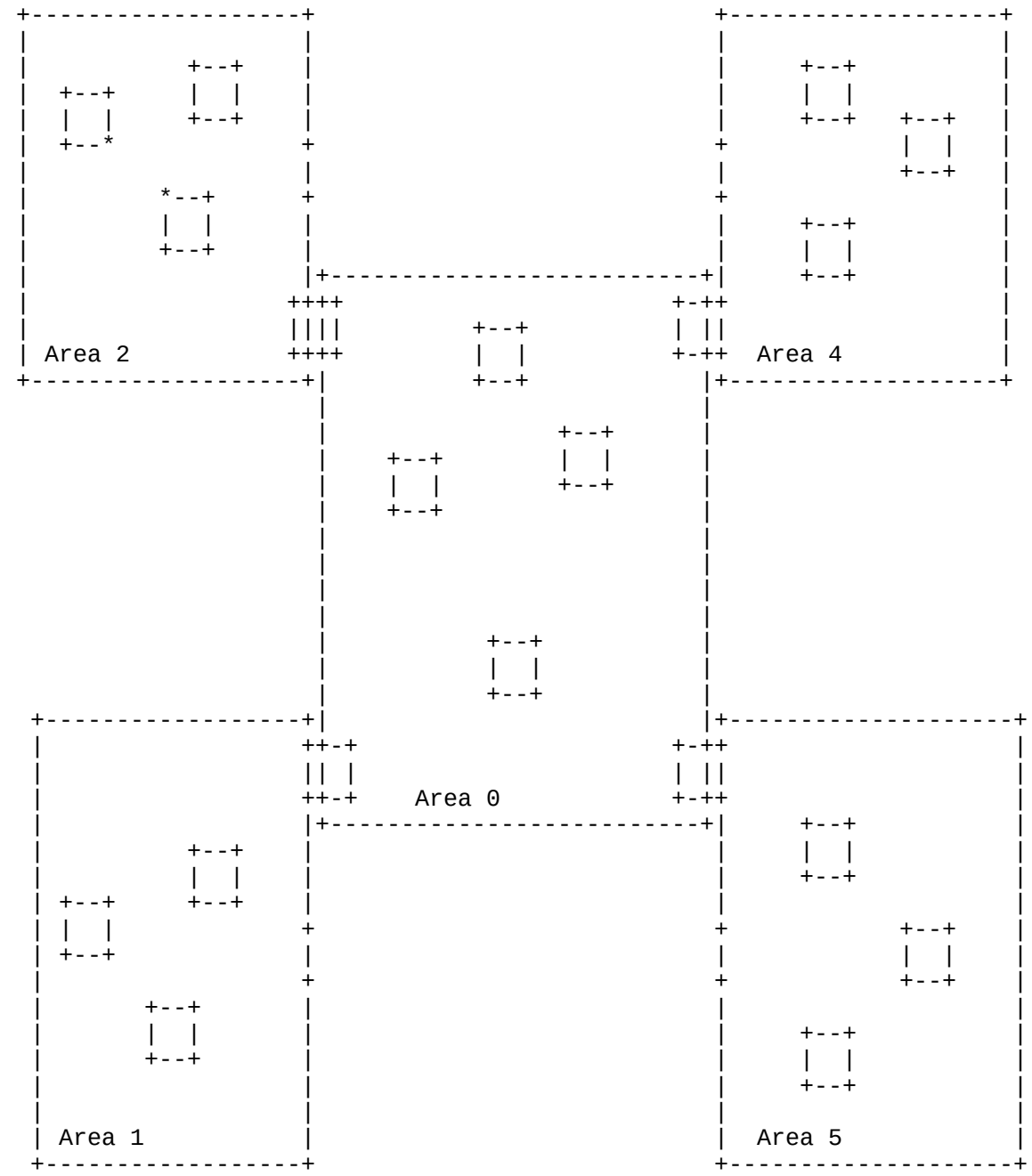
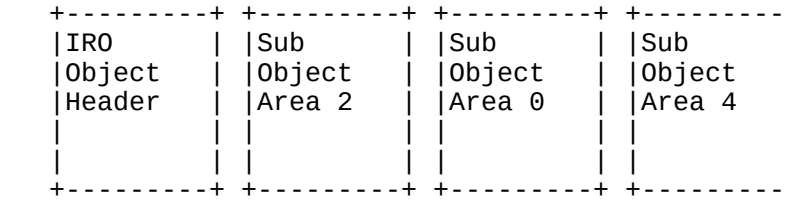
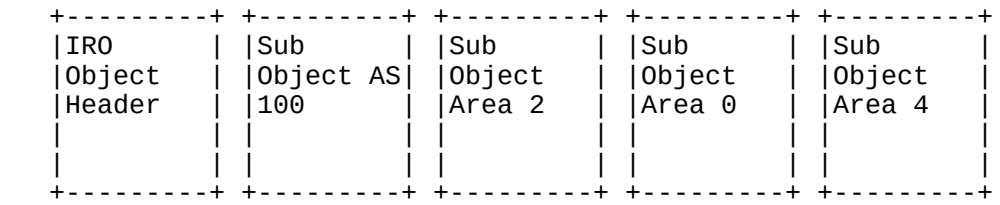


Figure 1: Inter-Area Path Computation

AS Number is 100.

This could be represented in the <IRO> as:





AS is optional and it MAY be skipped. PCE should be able to understand both notations.

4.2. Inter-AS Path Computation

In inter-AS path computation, where ingress and egress belong to different AS, the Domain-Sequence is represented using an ordered list of AS subobjects. The Domain-Sequence MAY further include decomposed area information in Area subobjects.

4.2.1. Example 1

As shown in Figure 2, where AS to be made of a single area, the area subobject MAY be skipped in the Domain-Sequence as AS is enough to uniquely identify the next domain and PCE.

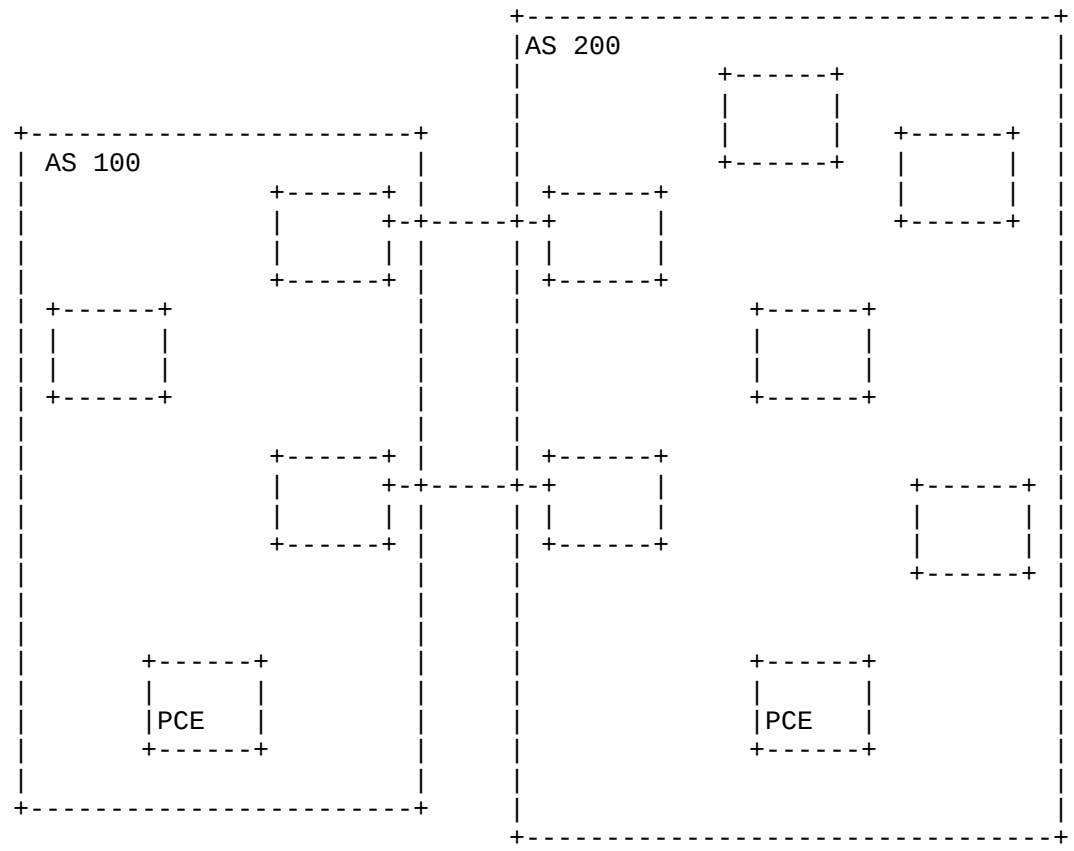
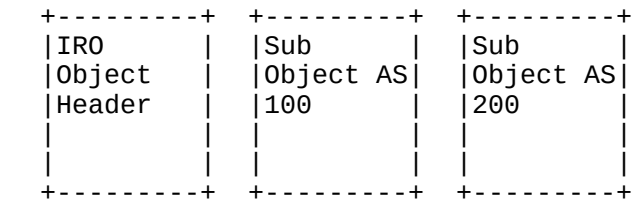
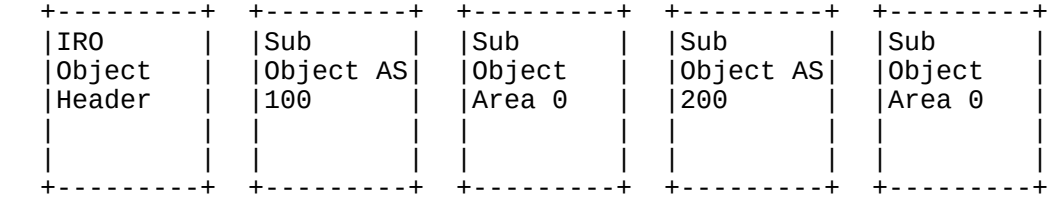


Figure 2: Inter-AS Path Computation

Both AS are made of Area 0.

This could be represented in the <IRO> as:

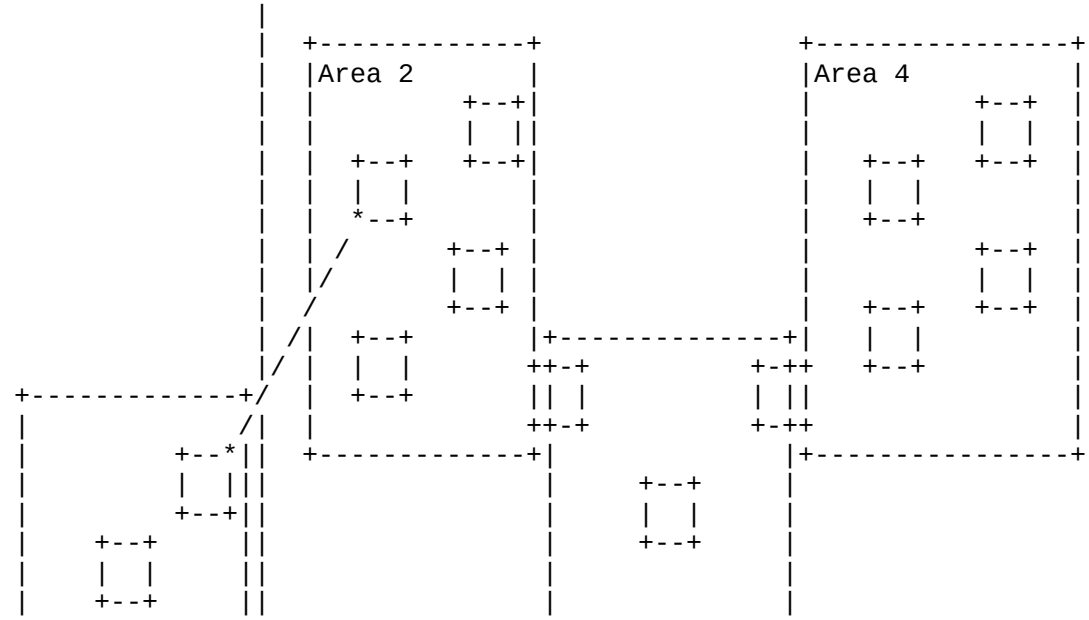




Area subobject is optional and it MAY be skipped. PCE should be able to understand both notations.

4.2.2. Example 2

As shown in Figure 3, where AS 200 is made up of multiple areas and multiple domain-sequence exist, PCE MAY include both AS and Area subobject to uniquely identify the next domain and PCE.



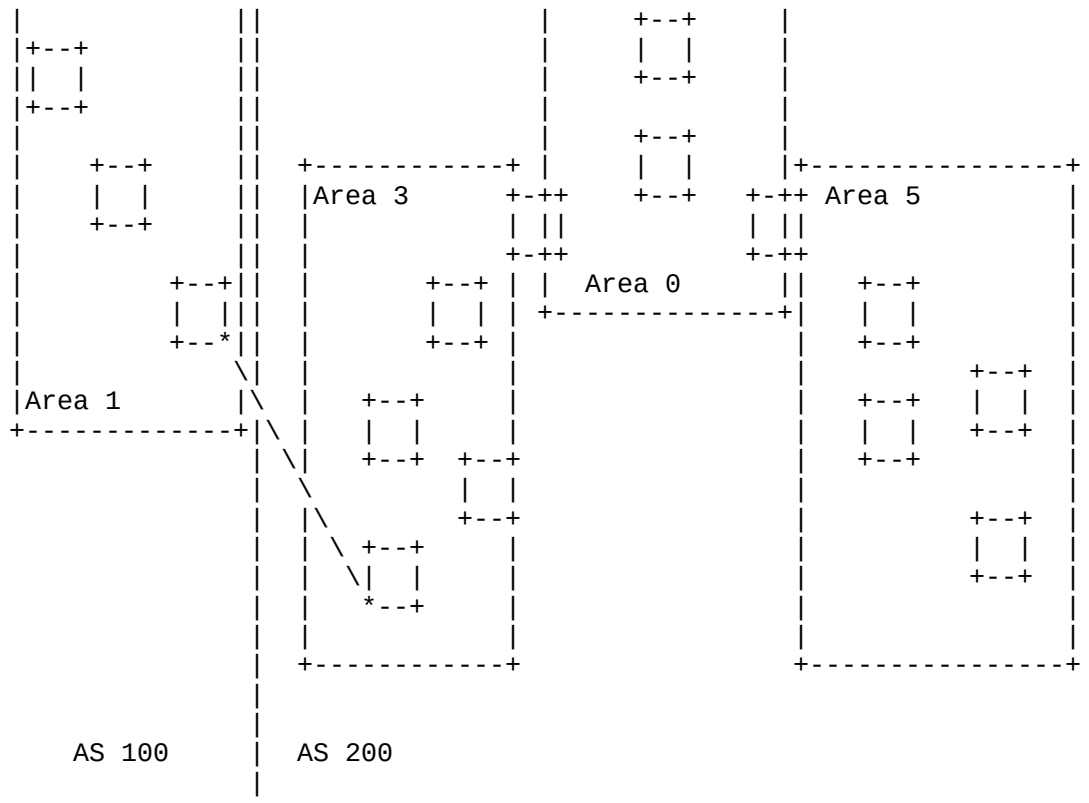


Figure 3: Inter-AS Path Computation

The Domain-Sequence can be carried in the IRO as shown below:

IRO Object Header	Sub Object AS 100	Sub Object Area 1	Sub Object AS 200	Sub Object Area 3	Sub Object Area 0	Sub Object Area 4

The combination of both an AS and an Area uniquely identify a domain in the Domain-Sequence.

Note that an Area domain identifier always belongs to the previous AS that appears before it or, if no AS subobjects are present, it is assumed to be the current AS.

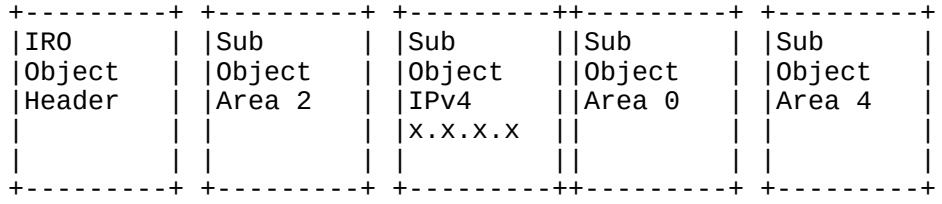
If the area information cannot be provided, PCE MAY forward the path computation request to the next PCE based on AS alone. If multiple PCEs are responsible, PCE MAY apply local policy to select the next PCE.

4.3. Boundary Node and Inter-AS-Link

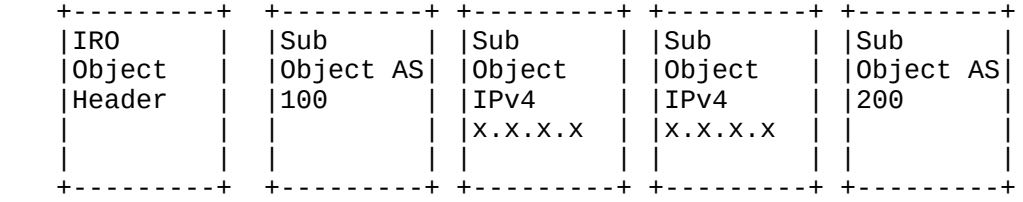
A PCC or PCE MAY add additional constraints covering which Boundary Nodes (ABR or ASBR) or Border links (Inter-AS-link) MUST be traversed while defining a Domain-Sequence. In which case the Boundary Node or Link MAY be encoded as a part of the domain-sequence using the existing subobjects.

Boundary Nodes (ABR / ASBR) can be encoded using the IPv4 or IPv6 prefix subobjects usually the loopback address of 32 and 128 prefix length respectively. An Inter-AS link can be encoded using the IPv4 or IPv6 prefix subobjects or unnumbered interface subobjects.

For Figure 1, an ABR to be traversed can be specified as:



For Figure 2, an inter-AS-link to be traversed can be specified as:



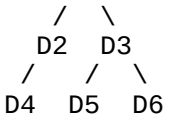
4.4. PCE Serving multiple Domains

A single PCE MAY be responsible for multiple domains; for example PCE function deployed on an ABR. A PCE which can support 2 adjacent domains can internally handle this situation without any impact on the neighbouring domains.

4.5. P2MP

In case of inter-domain P2MP path computation, (Refer [RFC7334]) the path domain tree is nothing but a series of Domain Sequences, as shown in the below figure:

D1-D3-D6, D1-D3-D5 and D1-D2-D4.



All rules of processing as applied to P2P can be applied to P2MP as well.

In case of P2MP, different destinations MAY have different Domain-Sequence within the domain tree, it requires domain-sequence to be attached per destination. (Refer [PCE-P2MP-PER-DEST])

4.6. Hierarchical PCE

As per [RFC6805], consider a case as shown in Figure 4 consisting of multiple child PCEs and a parent PCE.

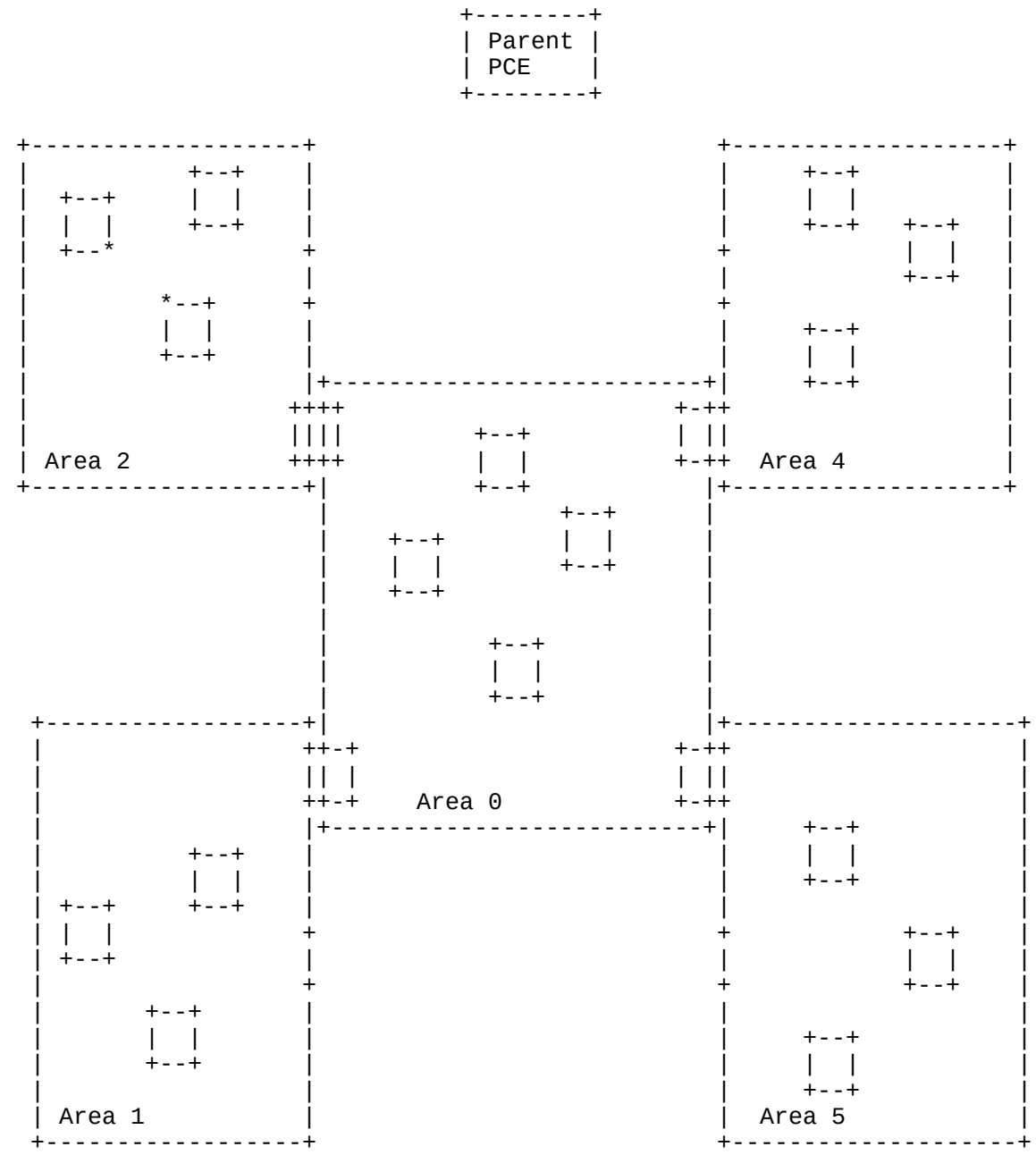
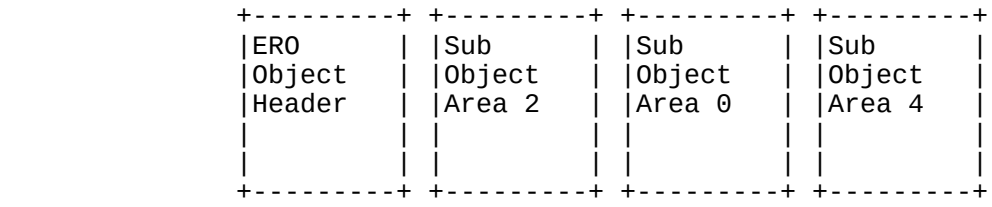
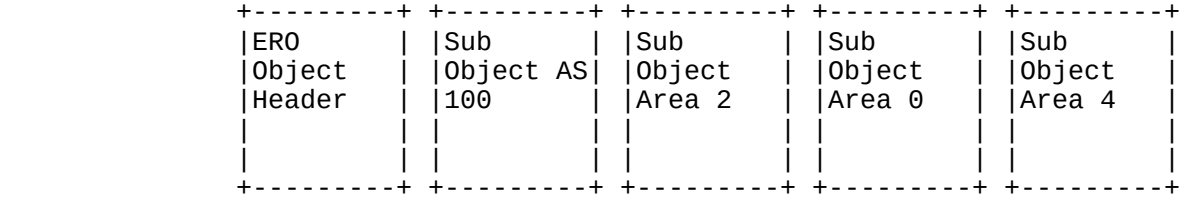


Figure 4: Hierarchical PCE

In H-PCE, the Ingress PCE 'PCE(1)' can request the parent PCE to determine the Domain-Sequence and return it in the PCEP response, using the ERO Object. The ERO can contain an ordered sequence of subobjects such as AS and Area (OSPF/ISIS) subobjects. In this case, the Domain-Sequence appear as:





4.7. Relationship to PCE Sequence

Instead of a domain-sequence, a sequence of PCEs MAY be enforced by policy on the PCC, and this constraint can be carried in the PCReq message (as defined in [RFC5886]).

Note that PCE-Sequence can be used along with domain-sequence in which case PCE-Sequence SHOULD have higher precedence in selecting the next PCE in the inter-domain path computation procedures. Note that Domain-Sequence IRO constraints should still be checked as per the rules of processing IRO.

4.8. Relationship to RSVP-TE

[RFC3209] already describes the notion of abstract nodes, where an abstract node is a group of nodes whose internal topology is opaque to the ingress node of the LSP. It further defines a subobject for AS but with a 2-Byte AS Number.

[DOMAIN-SUBOBJ] extends the notion of abstract nodes by adding new subobjects for IGP Areas and 4-byte AS numbers. These subobjects MAY be included in Explicit Route Object (ERO), Exclude Route object (XRO) or Explicit Exclusion Route Subobject (EXRS) in RSVP-TE.

In any case subobject type defined in RSVP-TE are identical to the subobject type defined in the related documents in PCEP.

5. IANA Considerations

5.1. New Subobjects

The "PCEP Parameters" registry contains a subregistry "PCEP Objects" with an entry for the Include Route Object (IRO), Exclude Route Object (XRO) and Explicit Route Object (ERO). IANA is requested to add further subobjects as follows:

```
7  ERO
10 IRO
17 XRO
```

Subobject Type		Reference
TBD1	4 byte AS number	[This I.D.]
TBD2	OSPF Area ID	[This I.D.]
TBD3	IS-IS Area ID	[This I.D.]

5.2. Error Object Field Values

The "PCEP Parameters" registry contains a subregistry "Error Types and Values". IANA is requested to make the following allocations from this subregistry

ERROR Type	Meaning	Reference
TBD4	"Unrecognized subobject" Error-Value: type code	[This I.D.]

6. Security Considerations

This document specifies a standard representation of Domain-Sequence and new subobjects, which MAY be used in inter-domain PCE scenarios as explained in other RFC and drafts. The new subobjects and Domain-Sequence mechanisms defined in this document allow finer and more specific control of the path computed by a cooperating PCE(s). Such control increases the risk if a PCEP message is intercepted, modified, or spoofed because it allows the attacker to exert control over the path that the PCE will compute or to make the path computation impossible. Therefore, the security techniques described in [RFC5440] are considered more important.

Note, however, that the Domain-Sequence mechanisms also provide the operator with the ability to route around vulnerable parts of the network and may be used to increase overall network security.

7. Manageability Considerations

7.1. Control of Function and Policy

Several local policy decisions should be made at the PCE. Firstly, the exact behavior with regard to desired inclusion and exclusion of domains must be available for examination by an operator and may be configurable. Second, the behavior on receipt of an unrecognized subobjects with the L or X-bit set should be configurable and must be available for inspection. The inspection and control of these local policy choices may be part of the PCEP MIB module.

7.2. Information and Data Models

A MIB module for management of the PCEP is being specified in a separate document [PCEP-MIB]. That MIB module allows examination of individual PCEP messages, in particular requests, responses and errors. The MIB module MUST be extended to include the ability to view the domain-sequence extensions defined in this document.

7.3. Liveness Detection and Monitoring

Mechanisms defined in this document do not imply any new liveness detection and monitoring requirements in addition to those already listed in [RFC5440].

7.4. Verify Correct Operations

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

7.5. Requirements On Other Protocols

In case of per-domain path computation [RFC5152], where the full path of an inter-domain TE LSP cannot be or is not determined at the ingress node, and signaling message may use domain identifiers. The Subobjects defined in this document SHOULD be supported by RSVP-TE. [DOMAIN-SUBOBJ] extends the notion of abstract nodes by adding new subobjects for IGP Areas and 4-byte AS numbers.

Apart from this, mechanisms defined in this document do not imply any requirements on other protocols in addition to those already listed in [RFC5440].

7.6. Impact On Network Operations

Mechanisms defined in this document do not have any impact on network operations in addition to those already listed in [RFC5440].

8. Acknowledgments

We would like to thank Adrian Farrel, Pradeep Shastry, Suresh Babu, Quintin Zhao, Fatai Zhang, Daniel King, Oscar Gonzalez, Chen Huaimo, Venugopal Reddy, Reeja Paul Sandeep Boina and Avantika for their useful comments and suggestions.

9. References

9.1. Normative References

[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.

[RFC5441] Vasseur, JP., Zhang, R., Bitar, N., and JL. Le Roux, "A Backward-Recursive PCE-Based Computation (BRPC) Procedure to Compute Shortest Constrained Inter-Domain Traffic Engineering Label Switched Paths", RFC 5441, 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.

[RFC6805] King, D. and A. Farrel, "The Application of the Path Computation Element Architecture to the Determination of a Sequence of Domains in MPLS and GMPLS", RFC 6805, November 2012.

9.2. Informative References

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

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

[RFC3477] Kompella, K. and Y. Rekhter, "Signalling Unnumbered Links in Resource ReSerVation Protocol - Traffic Engineering (RSVP-TE)", RFC 3477, January 2003.

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

[RFC4726] Farrel, A., Vasseur, J., and A. Ayyangar, "A Framework for Inter-Domain Multiprotocol Label Switching Traffic Engineering", RFC 4726, November 2006.

[RFC4873] Berger, L., Bryskin, I., Papadimitriou, D., and A. Farrel, "GMPLS Segment Recovery", RFC 4873, May 2007.

[RFC4874] Lee, CY., Farrel, A., and S. De Cnodder, "Exclude Routes - Extension to Resource ReserVation Protocol-Traffic Engineering (RSVP-TE)", RFC 4874, April 2007.

[RFC4893] Vohra, Q. and E. Chen, "BGP Support for Four-octet AS Number Space", RFC 4893, May 2007.

[RFC5152] Vasseur, JP., Ayyangar, A., and R. Zhang, "A Per-Domain Path Computation Method for Establishing Inter-Domain Traffic Engineering (TE) Label Switched Paths (LSPs)", RFC 5152, February 2008.

[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.

[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.

[RFC7334] Zhao, Q., Dhody, D., King, D., Ali, Z., and R. Casellas, "PCE-Based Computation Procedure to Compute Shortest Constrained Point-to-Multipoint (P2MP) Inter-Domain Traffic Engineering Label Switched Paths", RFC 7334, August 2014.

[PCEP-MIB] Koushik, A., Emile, S., Zhao, Q., King, D., and J. Hardwick, "PCE communication protocol(PCEP) Management Information Base. (draft-ietf-pce-pcep-mib)", September 2014.

[PCE-P2MP-PER-DEST]
Dhody, D., Palle, U., and V. Kondreddy, "Supporting explicit inclusion or exclusion of abstract nodes for a subset of P2MP destinations in Path Computation Element Communication Protocol (PCEP). (draft-dhody-pce-pcep-p2mp-per-destination)", September 2014.

[DOMAIN-SUBOBJ]
Dhody, D., Palle, U., Kondreddy, V., and R. Casellas, "Domain Subobjects for Resource ReserVation Protocol - Traffic Engineering (RSVP-TE). (draft-dhody-ccamp-rsvp-te-domain-subobjects)", July 2014.

[IRO-SURVEY]
Dhody, D., "Informal Survey into Include Route Object (IRO) Implementations in Path Computation Element communication Protocol (PCEP). (draft-dhody-pce-iro-survey-01)", October 2014.

[IRO-UPDATE]
Dhody, D., "Update to Include Route Object (IRO) specification in Path Computation Element communication Protocol (PCEP. (draft-dhody-pce-iro-update-00)", October 2014.

[ISO10589]
ISO, "Intermediate system to Intermediate system routing information exchange protocol for use in conjunction with the Protocol for providing the Connectionless-mode Network Service (ISO 8473)", ISO/IEC 10589:2002, 1992.

Authors' Addresses

Dhruv Dhody
Huawei Technologies
Leela Palace
Bangalore, Karnataka 560008
INDIA

EMail: dhruv.ietf@gmail.com

Udayasree Palle
Huawei Technologies
Leela Palace
Bangalore, Karnataka 560008
INDIA

EMail: udayasree.palle@huawei.com

Ramon Casellas
CTTC
Av. Carl Friedrich Gauss n7
Castelldefels, Barcelona 08860
SPAIN

EMail: ramon.casellas@cttc.es

PCE Working Group
Internet-Draft
Intended status: Experimental
Expires: June 9, 2016

D. Dhody
U. Palle
Huawei Technologies
R. Casellas
CTTC

December 7, 2015

Domain Subobjects for Path Computation Element (PCE) Communication Protocol (PCEP). draft-ietf-pce-pcep-domain-sequence-12

Abstract

The ability to compute shortest constrained Traffic Engineering Label Switched Paths (TE LSPs) in Multiprotocol Label Switching (MPLS) and Generalized MPLS (GMPLS) networks across multiple domains has been identified as a key requirement. In this context, a domain is a collection of network elements within a common sphere of address management or path computational responsibility such as an Interior Gateway Protocol (IGP) area or an Autonomous System (AS). This document specifies a representation and encoding of a Domain-Sequence, which is defined as an ordered sequence of domains traversed to reach the destination domain to be used by Path Computation Elements (PCEs) to compute inter-domain constrained shortest paths across a predetermined sequence of domains . This document also defines new subobjects to be used to encode domain identifiers.

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 June 9, 2016.

Table of Contents

1. Introduction 3
1.1. Scope 4
1.2. Requirements Language 4
2. Terminology 5
3. Detail Description 6
3.1. Domains 6
3.2. Domain-Sequence 6
3.3. Domain-Sequence Representation 7
3.4. Include Route Object (IRO) 7
3.4.1. Subobjects 8
3.4.1.1. Autonomous system 8
3.4.1.2. IGP Area 9
3.4.2. Update in IRO specification 10
3.4.3. IRO for Domain-Sequence 10
3.4.3.1. PCC Procedures 11
3.4.3.2. PCE Procedures 11
3.5. Exclude Route Object (XRO) 12
3.5.1. Subobjects 13
3.5.1.1. Autonomous system 13
3.5.1.2. IGP Area 14
3.6. Explicit Exclusion Route Subobject (EXRS) 15
3.7. Explicit Route Object (ERO) 16
4. Examples 16
4.1. Inter-Area Path Computation 16
4.2. Inter-AS Path Computation 18
4.2.1. Example 1 19
4.2.2. Example 2 21
4.3. Boundary Node and Inter-AS-Link 23
4.4. PCE Serving multiple Domains 24
4.5. P2MP 24
4.6. Hierarchical PCE 26

5. Other Considerations 26
5.1. Relationship to PCE Sequence 26
5.2. Relationship to RSVP-TE 26
6. IANA Considerations 27
6.1. New Subobjects 27
7. Security Considerations 27
8. Manageability Considerations 28
8.1. Control of Function and Policy 28
8.2. Information and Data Models 28
8.3. Liveness Detection and Monitoring 29
8.4. Verify Correct Operations 29
8.5. Requirements On Other Protocols 29
8.6. Impact On Network Operations 29
9. Acknowledgments 29
10. References 30
10.1. Normative References 30
10.2. Informative References 31
Authors' Addresses 33

1. Introduction

A Path Computation Element (PCE) may be used to compute end-to-end paths across multi-domain environments using a per-domain path computation technique [RFC5152]. The backward recursive path computation (BRPC) mechanism [RFC5441] also defines a PCE-based path computation procedure to compute inter-domain constrained path for (G)MPLS TE LSPs. However, both per-domain and BRPC techniques assume that the sequence of domains to be crossed from source to destination is known, either fixed by the network operator or obtained by other means. Also for inter-domain point-to-multi-point (P2MP) tree computation, [RFC7334] assumes the domain-tree is known in priori.

The list of domains (Domain-Sequence) in point-to-point (P2P) or a domain tree in point-to-multipoint (P2MP) is usually a constraint in inter-domain path computation procedure.

The Domain-Sequence (the set of domains traversed to reach the destination domain) is either administratively predetermined or discovered by some means like H-PCE.

[RFC5440] defines the Include Route Object (IRO) and the Explicit Route Object (ERO). [RFC5521] defines the Exclude Route Object (XRO) and the Explicit Exclusion Route Subobject (EXRS). The use of Autonomous System (AS) (albeit with a 2-Byte AS number) as an abstract node representing a domain is defined in [RFC3209]. In the current document, we specify new subobjects to include or exclude domains including IGP area or an Autonomous Systems (4-Byte as per [RFC6793]).

Further, the domain identifier may simply act as delimiter to specify where the domain boundary starts and ends in some cases.

This is a companion document to Resource ReserVation Protocol - Traffic Engineering (RSVP-TE) extensions for the domain identifiers [DOMAIN-SUBOBJ].

1.1. Scope

The procedures described in this document are experimental. The experiment is intended to enable research for the usage of Domain-Sequence at the PCEs for inter-domain paths. For this purpose this document specifies new domain subobjects as well as how they incorporate with existing subobjects to represent a Domain-Sequence.

The experiment will end two years after the RFC is published. At that point, the RFC authors will attempt to determine how widely this has been implemented and deployed.

This document does not change the procedures for handling existing subobjects in PCEP.

The new subobjects introduced by this document will not be understood by legacy implementations. If a legacy implementation receives one of the subobjects that it does not understand in a PCEP object, the legacy implementation will behave as described in Section 3.4.3. Therefore, it is assumed that this experiment will be conducted only when both the PCE and the PCC form part of the experiment. It is possible that a PCC or PCE can operate with peers some of which form part of the experiment and some that do not. In this case, since no capabilities exchange is used to identify which nodes can use these extensions, manual configuration should be used to determine which peerings form part of the experiment.

When the result of implementation and deployment are available, this document will be updated and refined, and then be moved from Experimental to Standard Track.

1.2. Requirements Language

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

2. Terminology

The following terminology is used in this document.

ABR: OSPF Area Border Router. Routers used to connect two IGP areas.

AS: Autonomous System.

ASBR: Autonomous System Boundary Router.

BN: Boundary Node, Can be an ABR or ASBR.

BRPC: Backward Recursive Path Computation

Domain: As per [RFC4655], any collection of network elements within a common sphere of address management or path computational responsibility. Examples of domains include Interior Gateway Protocol (IGP) area and Autonomous System (AS).

Domain-Sequence: An ordered sequence of domains traversed to reach the destination domain.

ERO: Explicit Route Object

H-PCE: Hierarchical PCE

IGP: Interior Gateway Protocol. Either of the two routing protocols, Open Shortest Path First (OSPF) or Intermediate System to Intermediate System (IS-IS).

IRO: Include Route Object

IS-IS: Intermediate System to Intermediate System.

OSPF: Open Shortest Path First.

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.

P2MP: Point-to-Multipoint

P2P: Point-to-Point

RSVP: Resource Reservation Protocol

TE LSP: Traffic Engineering Label Switched Path.

XRO: Exclude Route Object

3. Detail Description

3.1. Domains

[RFC4726] and [RFC4655] define domain as a separate administrative or geographic environment within the network. A domain could be further defined as a zone of routing or computational ability. Under these definitions a domain might be categorized as an AS or an IGP area. Each AS can be made of several IGP areas. In order to encode a Domain-Sequence, it is required to uniquely identify a domain in the Domain-Sequence. A domain can be uniquely identified by area-id or AS number or both.

3.2. Domain-Sequence

A Domain-Sequence is an ordered sequence of domains traversed to reach the destination domain.

A Domain-Sequence can be applied as a constraint and carried in a path computation request to PCE(s). A Domain-Sequence can also be the result of a path computation. For example, in the case of Hierarchical PCE (H-PCE) [RFC6805], Parent PCE could send the Domain-Sequence as a result in a path computation reply.

In a P2P path, the domains listed appear in the order that they are crossed. In a P2MP path, the domain tree is represented as a list of Domain-Sequences.

A Domain-Sequence enables a PCE to select the next domain and the PCE serving that domain to forward the path computation request based on the domain information.

Domain-Sequence can include Boundary Nodes (ABR or ASBR) or Border links (Inter-AS-links) to be traversed as an additional constraint.

Thus a Domain-Sequence can be made up of one or more of -

- AS Number
- Area ID
- Boundary Node ID

- Inter-AS-Link Address

These are encoded in the new subobjects defined in this document as well as the existing subobjects to represent a Domain-Sequence.

Consequently, a Domain-Sequence can be used:

1. by a PCE in order to discover or select the next PCE in a collaborative path computation, such as in BRPC [RFC5441];

2. by the Parent PCE to return the Domain-Sequence when unknown; this can then be an input to the BRPC procedure [RFC6805];

3. by a Path Computation Client (PCC) or a PCE, to constrain the domains used in inter-domain path computation, explicitly specifying which domains to be expanded or excluded;

4. by a PCE in the per-domain path computation model [RFC5152] to identify the next domain.

3.3. Domain-Sequence Representation

Domain-Sequence appears in PCEP messages, notably in -

- Include Route Object (IRO): As per [RFC5440], IRO can be used to specify a set of network elements to be traversed to reach the destination, which includes subobjects used to specify the Domain-Sequence.

- Exclude Route Object (XRO): As per [RFC5521], XRO can be used to specify certain abstract nodes, to be excluded from whole path, which includes subobjects used to specify the Domain-Sequence.

- Explicit Exclusion Route Subobject (EXRS): As per [RFC5521], EXRS can be used to specify exclusion of certain abstract nodes (including domains) between a specific pair of nodes. EXRS are a subobject inside the IRO.

- Explicit Route Object (ERO): As per [RFC5440], ERO can be used to specify a computed path in the network. For example, in the case of H-PCE [RFC6805], a Parent PCE can send the Domain-Sequence as a result, in a path computation reply using ERO.

3.4. Include Route Object (IRO)

As per [RFC5440], IRO (Include Route Object) can be used to specify that the computed path needs to traverse a set of specified network elements or abstract nodes.

3.4.1. Subobjects

Some subobjects are defined in [RFC3209], [RFC3473], [RFC3477] and [RFC4874], but new subobjects related to Domain-Sequence are needed.

This document extends the support for 4-Byte AS numbers and IGP Areas.

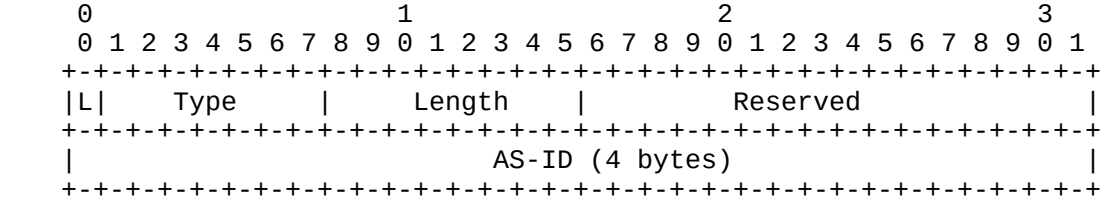
Type	Subobject
TBD1	Autonomous system number (4 Byte)
TBD2	OSPF Area id
TBD3	ISIS Area id

Note: The twins of these subobjects are carried in RSVP-TE messages as defined in [DOMAIN-SUBOBJ].

3.4.1.1. Autonomous system

[RFC3209] already defines 2 byte AS number.

To support 4 byte AS number as per [RFC6793] following subobject is defined:



L: The L bit is an attribute of the subobject as defined in [RFC3209] and usage in IRO subobject updated in [IRO-UPDATE].

Type: (TBD1 by IANA) indicating a 4-Byte AS Number.

Length: 8 (Total length of the subobject in bytes).

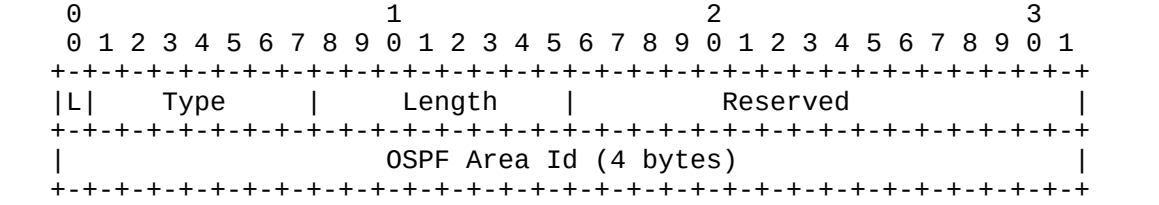
Reserved: Zero at transmission, ignored at receipt.

AS-ID: The 4-Byte AS Number. Note that if 2-Byte AS numbers are in use, the low order bits (16 through 31) MUST be used and the high order bits (0 through 15) MUST be set to zero.

3.4.1.2. IGP Area

Since the length and format of Area-id is different for OSPF and ISIS, following two subobjects are defined:

For OSPF, the area-id is a 32 bit number. The subobject is encoded as follows:



L: The L bit is an attribute of the subobject as defined in [RFC3209] and usage in IRO subobject updated in [IRO-UPDATE].

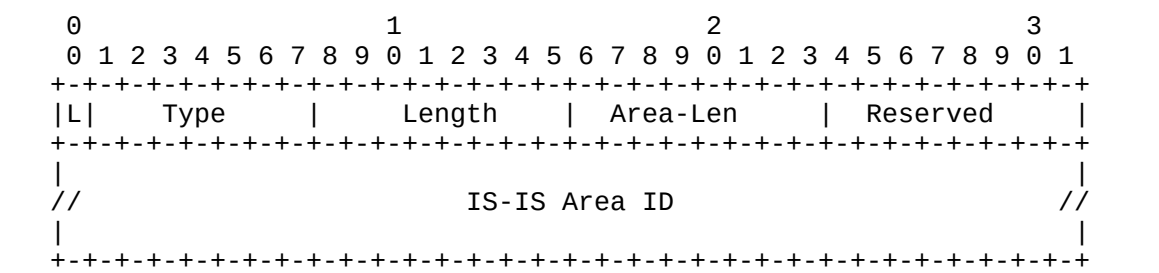
Type: (TBD2 by IANA) indicating a 4-Byte OSPF Area ID.

Length: 8 (Total length of the subobject in bytes).

Reserved: Zero at transmission, ignored at receipt.

OSPF Area Id: The 4-Byte OSPF Area ID.

For IS-IS, the area-id is of variable length and thus the length of the Subobject is variable. The Area-id is as described in IS-IS by ISO standard [ISO10589]. The subobject is encoded as follows:



L: The L bit is an attribute of the subobject as defined in [RFC3209] and usage in IRO subobject updated in [IRO-UPDATE].

Type: (TBD3 by IANA) indicating IS-IS Area ID.

Length: Variable. The Length MUST be at least 8, and MUST be a multiple of 4.

Area-Len: Variable (Length of the actual (non-padded) IS-IS Area Identifier in octets; Valid values are from 1 to 13 inclusive).

Reserved: Zero at transmission, ignored at receipt.

IS-IS Area Id: The variable-length IS-IS area identifier. Padded with trailing zeroes to a four-byte boundary.

3.4.2. Update in IRO specification

[RFC5440] describes IRO as an optional object used to specify network elements to be traversed by the computed path. It further state that the L bit of such subobject has no meaning within an IRO. It also did not mention if IRO is an ordered or un-ordered list of subobjects.

An update to IRO specification [IRO-UPDATE] makes IRO as an ordered list, as well as support for loose bit (L-bit) is added.

The use of IRO for Domain-Sequence, assumes the updated specification for IRO, as per [IRO-UPDATE].

3.4.3. IRO for Domain-Sequence

The subobject type for IPv4, IPv6, and unnumbered Interface ID can be used to specify Boundary Nodes (ABR/ASBR) and Inter-AS-Links. The subobject type for the AS Number (2 or 4 Byte) and the IGP Area are used to specify the domain identifiers in the Domain-Sequence.

The IRO can incorporate the new domain subobjects with the existing subobjects in a sequence of traversal.

Thus an IRO, comprising subobjects, that represents a Domain-Sequence, define the domains involved in an inter-domain path computation, typically involving two or more collaborative PCEs.

A Domain-Sequence can have varying degrees of granularity. It is possible to have a Domain-Sequence composed of, uniquely, AS identifiers. It is also possible to list the involved IGP areas for a given AS.

In any case, the mapping between domains and responsible PCEs is not defined in this document. It is assumed that a PCE that needs to obtain a "next PCE" from a Domain-Sequence is able to do so (e.g. via administrative configuration, or discovery).

3.4.3.1. PCC Procedures

A PCC builds an IRO to encode the Domain-Sequence, so that the cooperating PCEs could compute an inter-domain shortest constrained path across the specified sequence of domains.

A PCC may intersperse Area and AS subobjects with other subobjects without change to the previously specified processing of those subobjects in the IRO.

3.4.3.2. PCE Procedures

If a PCE receives an IRO in a Path Computation request (PCReq) message that contains the subobjects defined in this document, that it does not recognize, it will respond according to the rules for a malformed object as per [RFC5440]. The PCE MAY also include the IRO in the PCErr message as per [RFC5440].

The interpretation of Loose bit (L bit) is as per section 4.3.3.1 of [RFC3209] (as per [IRO-UPDATE]).

In a Path Computation reply (PCRep), PCE MAY also supply IRO (with Domain-Sequence information) with the NO-PATH object indicating that the set of elements (domains) of the request’s IRO prevented the PCEs from finding a path.

The following processing rules apply for Domain-Sequence in IRO -

- When a PCE parses an IRO, it interprets each subobject according to the AS number associated with the preceding subobject. We call this the "current AS". Certain subobjects modify the current AS, as follows.
  - The current AS is initialized to the AS number of the PCC.
  - If the PCE encounters an AS subobject, then it updates the current AS to this new AS number.
  - If the PCE encounters an Area subobject, then it assumes that the area belongs to the current AS.
  - If the PCE encounters an IP address that is globally routable, then it updates the current AS to the AS that owns this IP address. This document does not define how the PCE learns which AS owns the IP address.
  - If the PCE encounters an IP address that is not globally routable, then it assumes that it belongs to the current AS.

    * If the PCE encounters an unnumbered link, then it assumes that it belongs to the current AS.

  o When a PCE parses an IRO, it interprets each subobject according to the Area ID associated with the preceding subobject. We call this the "current Area". Certain subobjects modify the current Area, as follows.

    * The current Area is initialized to the Area ID of the PCC.

    * If the current AS is changed, the current Area is reset and need to be determined again by current or subsequent subobject.

    * If the PCE encounters an Area subobject, then it updates the current Area to this new Area ID.

    * If the PCE encounters an IP address that belongs to a different area, then it updates the current Area to the Area that has this IP address. This document does not define how the PCE learns which Area has the IP address.

    * If the PCE encounters an unnumbered link that belongs to a different area, then it updates the current Area to the Area that has this link.

    * Otherwise, it assumes that the subobject belongs to the current Area.

  o In case the current PCE is not responsible for the path computation in the current AS or Area, then the PCE selects the "next PCE" in the domain-sequence based on the current AS and Area.

Note that it is advised that, PCC should use AS and Area subobject while building the domain-sequence in IRO and avoid using other mechanism to change the "current AS" and "current Area" as described above.

3.5. Exclude Route Object (XRO)

The Exclude Route Object (XRO) [RFC5521] is an optional object used to specify exclusion of certain abstract nodes or resources from the whole path.

3.5.1. Subobjects

Some subobjects to be used in XRO as defined in [RFC3209], [RFC3477], [RFC4874], and [RFC5520], but new subobjects related to Domain-Sequence are needed.

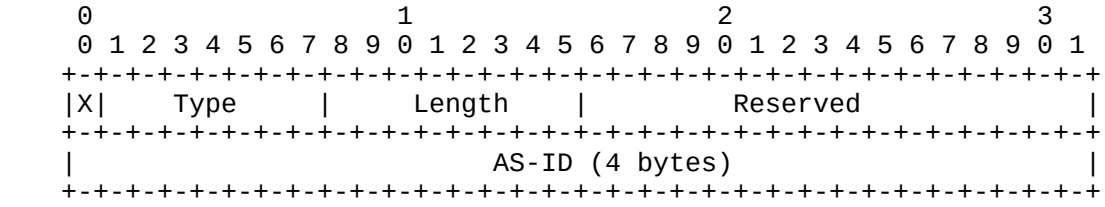
This document extends the support for 4-Byte AS numbers and IGP Areas.

Type	Subobject
TBD1	Autonomous system number (4 Byte)
TBD2	OSPF Area id
TBD3	ISIS Area id

Note: The twins of these subobjects are carried in RSVP-TE messages as defined in [DOMAIN-SUBOBJ].

3.5.1.1. Autonomous system

The new subobjects to support 4 byte AS and IGP (OSPF / ISIS) Area MAY also be used in the XRO to specify exclusion of certain domains in the path computation procedure.



The X-bit indicates whether the exclusion is mandatory or desired.

0: indicates that the AS specified MUST be excluded from the path computed by the PCE(s).

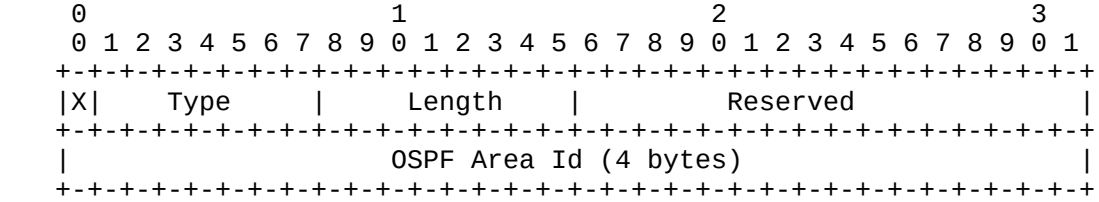
1: indicates that the AS specified SHOULD be avoided from the inter-domain path computed by the PCE(s), but MAY be included subject to PCE policy and the absence of a viable path that meets the other constraints.

All other fields are consistent with the definition in Section 3.4.

3.5.1.2. IGP Area

Since the length and format of Area-id is different for OSPF and ISIS, following two subobjects are defined:

For OSPF, the area-id is a 32 bit number. The subobject is encoded as follows:



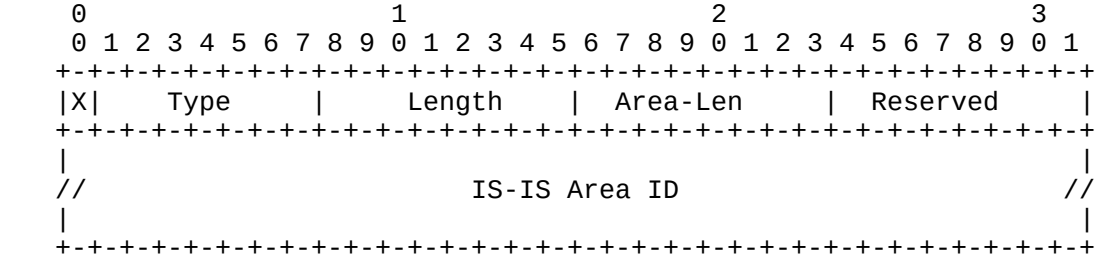
The X-bit indicates whether the exclusion is mandatory or desired.

0: indicates that the OSFF Area specified MUST be excluded from the path computed by the PCE(s).

1: indicates that the OSFF Area specified SHOULD be avoided from the inter-domain path computed by the PCE(s), but MAY be included subject to PCE policy and the absence of a viable path that meets the other constraints.

All other fields are consistent with the definition in Section 3.4.

For IS-IS, the area-id is of variable length and thus the length of the subobject is variable. The Area-id is as described in IS-IS by ISO standard [ISO10589]. The subobject is encoded as follows:



The X-bit indicates whether the exclusion is mandatory or desired.

0: indicates that the ISIS Area specified MUST be excluded from the path computed by the PCE(s).

1: indicates that the ISIS Area specified SHOULD be avoided from the inter-domain path computed by the PCE(s), but MAY be included subject to PCE policy and the absence of a viable path that meets the other constraints.

All other fields are consistent with the definition in Section 3.4.

All the processing rules are as per [RFC5521].

Note that, if a PCE receives an XRO in a PCReq message that contains subobjects defined in this document, that it does not recognize, it will respond according to the rules for a malformed object as per [RFC5440].

IGP Area subobjects in the XRO are local to the current AS. In case of multi-AS path computation to exclude an IGP area in a different AS, IGP Area subobject should be part of Explicit Exclusion Route Subobject (EXRS) in the IRO to specify the AS in which the IGP area is to be excluded. Further policy may be applied to prune/ignore Area subobjects in XRO after "current AS" change during path computation.

3.6. Explicit Exclusion Route Subobject (EXRS)

EXRS [RFC5521] is used to specify exclusion of certain abstract nodes between a specific pair of nodes.

The EXRS subobject can carry any of the subobjects defined for inclusion in the XRO, thus the new subobjects to support 4 byte AS and IGP (OSPF / ISIS) Area can also be used in the EXRS. The meanings of the fields of the new XRO subobjects are unchanged when the subobjects are included in an EXRS, except that scope of the exclusion is limited to the single hop between the previous and subsequent elements in the IRO.

The EXRS subobject should be interpreted in the context of the current AS and current Area of the preceding subobject in the IRO. The EXRS subobject does not change the current AS or current Area. All other processing rules are as per [RFC5521].

Note that, if a PCE that supports the EXRS in an IRO, parses an IRO, and encounters an EXRS that contains subobjects defined in this document, that it does not recognize, it will act according to the setting of the X-bit in the subobject as per [RFC5521].

3.7. Explicit Route Object (ERO)

The Explicit Route Object (ERO) [RFC5440] is used to specify a computed path in the network. PCEP ERO subobject types correspond to RSVP-TE ERO subobject types as defined in [RFC3209], [RFC3473], [RFC3477], [RFC4873], [RFC4874], and [RFC5520]. The subobjects related to Domain-Sequence are further defined in [DOMAIN-SUBOBJ].

The new subobjects to support 4 byte AS and IGP (OSPF / ISIS) Area can also be used in the ERO to specify an abstract node (a group of nodes whose internal topology is opaque to the ingress node of the LSP). Using this concept of abstraction, an explicitly routed LSP can be specified as a sequence of domains.

In case of Hierarchical PCE [RFC6805], a Parent PCE can be requested to find the Domain-Sequence. Refer example in Section 4.6. The ERO in reply from parent PCE can then be used in Per-Domain path computation or BRPC.

If a PCC receives an ERO in a PCRep message that contains subobject defined in this document, that it does not recognize, it will respond according to the rules for a malformed object as per [RFC5440].

4. Examples

The examples in this section are for illustration purposes only; to highlight how the new subobjects could be encoded. They are not meant to be an exhaustive list of all possible usecases and combinations.

4.1. Inter-Area Path Computation

In an inter-area path computation where the ingress and the egress nodes belong to different IGP areas within the same AS, the Domain-Sequence could be represented using a ordered list of Area subobjects.

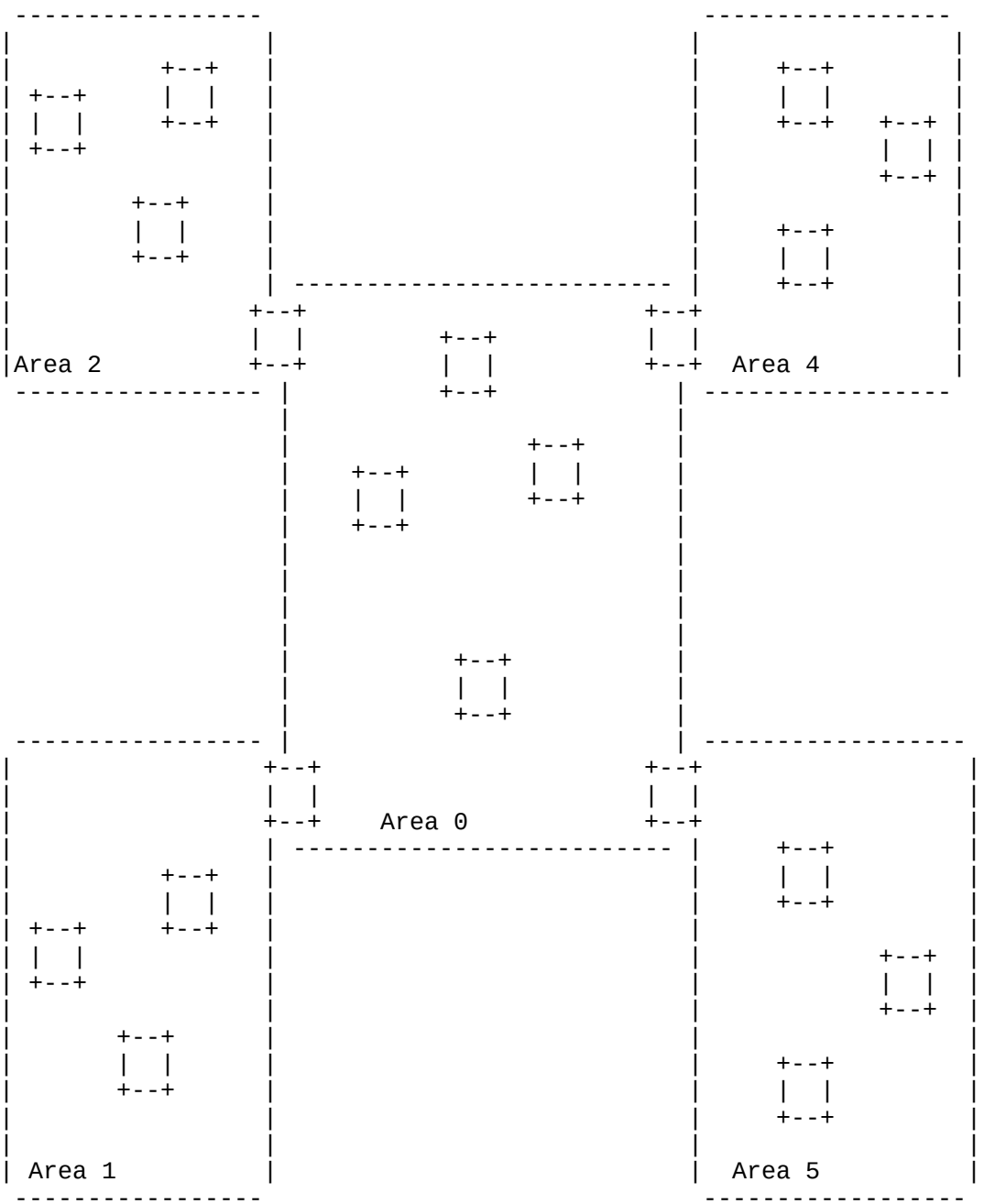
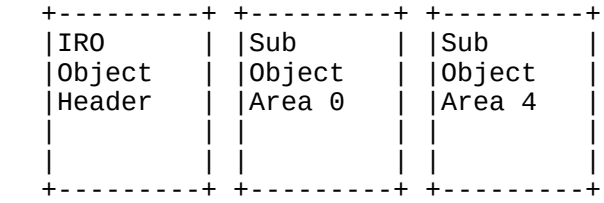


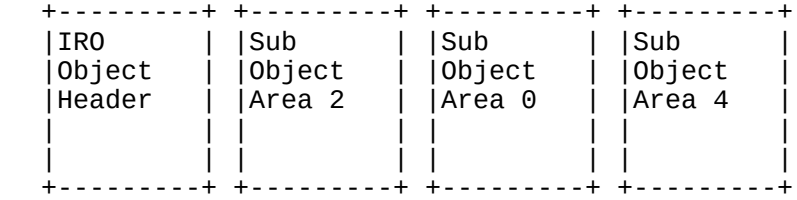
Figure 1: Inter-Area Path Computation

AS Number is 100.

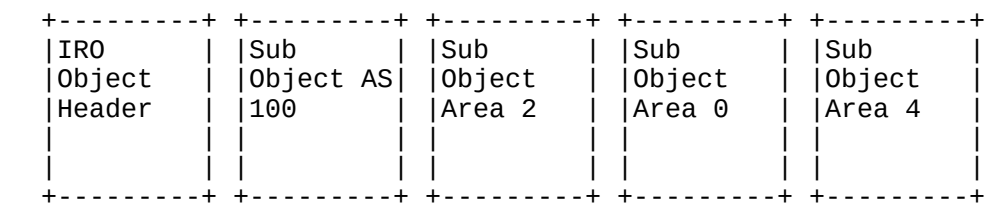
If the ingress is in Area 2, egress in Area 4 and transit through Area 0. Some possible way a PCC can encode the IRO:



or



or



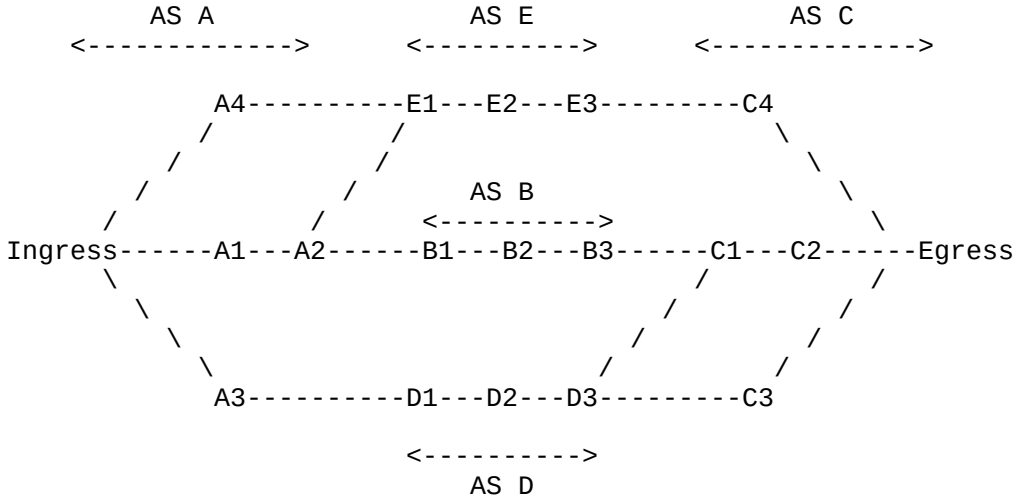
The Domain-Sequence can further include encompassing AS information in the AS subobject.

4.2. Inter-AS Path Computation

In inter-AS path computation, where ingress and egress belong to different AS, the Domain-Sequence could be represented using an ordered list of AS subobjects. The Domain-Sequence can further include decomposed area information in the Area subobject.

4.2.1. Example 1

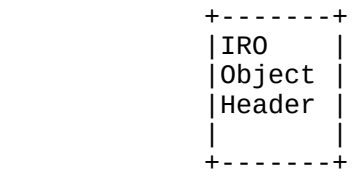
As shown in Figure 2, where AS has a single area, AS subobject in the domain-sequence can uniquely identify the next domain and PCE.



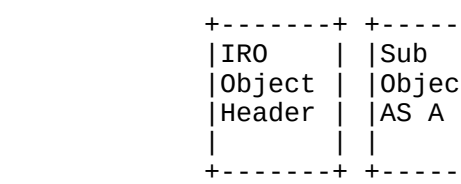
* All AS have one area (area 0)

Figure 2: Inter-AS Path Computation

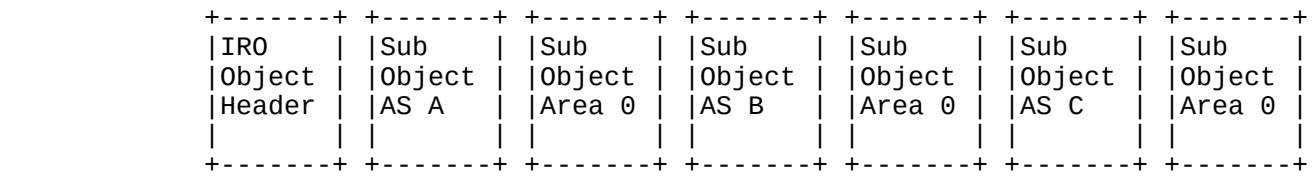
If the ingress is in AS A, egress in AS C and transit through AS B. Some possible way a PCC can encode the IRO:



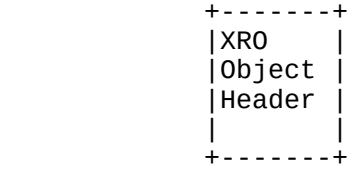
or



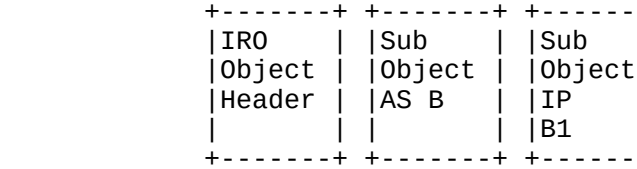
or



Note that to get a domain disjoint path, the ingress could also request the backup path with -

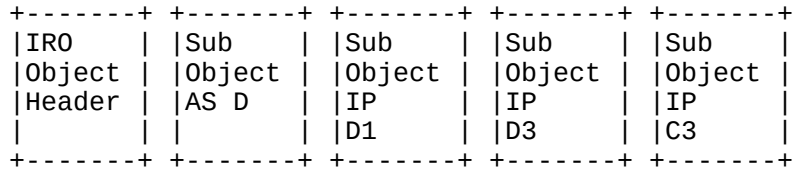


As described in Section 3.4.3, domain subobject in IRO changes the domain information associated with the next set of subobjects; till you encounter a subobject that changes the domain too. Consider the following IRO:



On processing subobject "AS B", it changes the AS of the subsequent subobjects till we encounter another subobject "AS C" which changes the AS for its subsequent subobjects.

Consider another IRO:

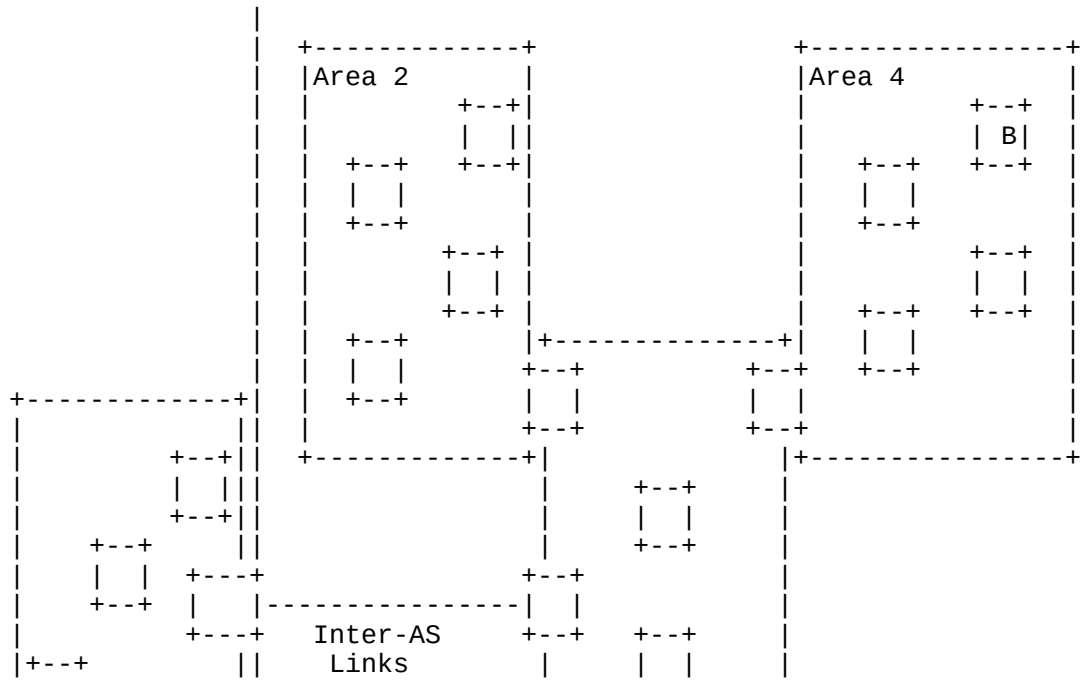


Here as well, on processing "AS D", it changes the AS of the subsequent subobjects till you encounter another subobject "C3" which belong in another AS and changes the AS for its subsequent subobjects.

Further description for the Boundary Node and Inter-AS-Link can be found in Section 4.3.

4.2.2. Example 2

In Figure 3, AS 200 is made up of multiple areas.



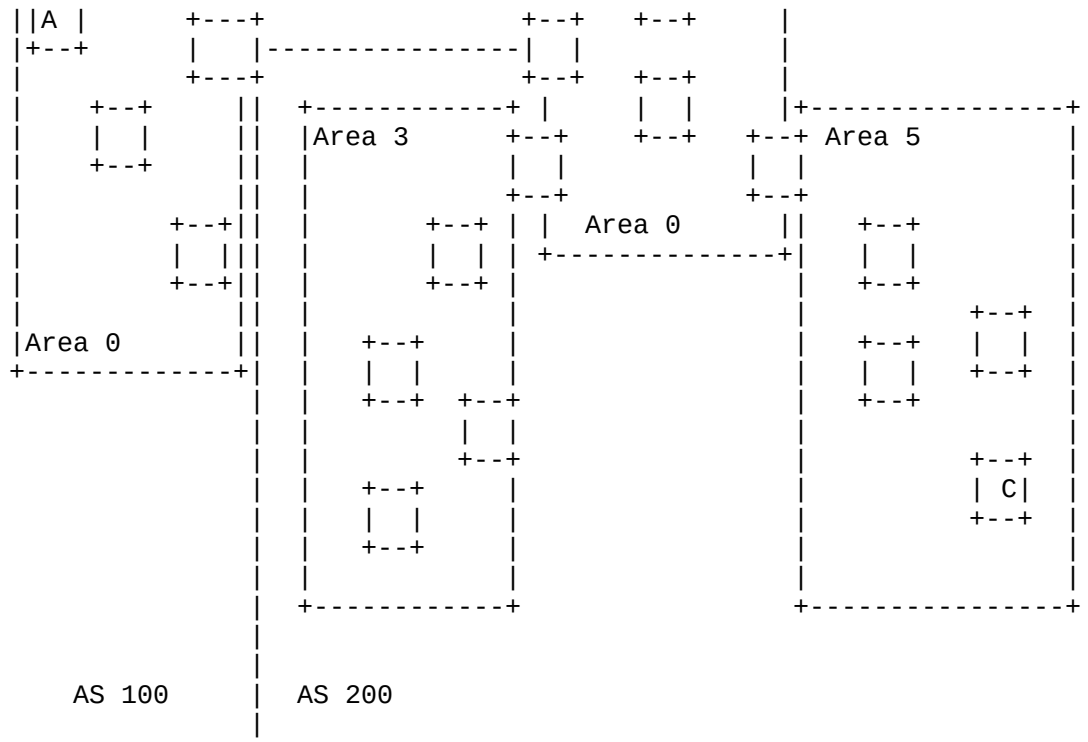
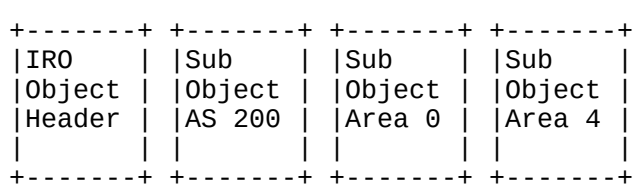
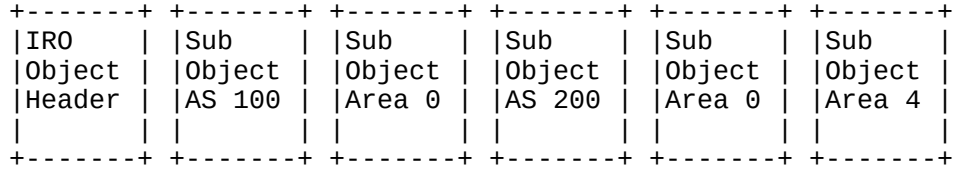


Figure 3: Inter-AS Path Computation

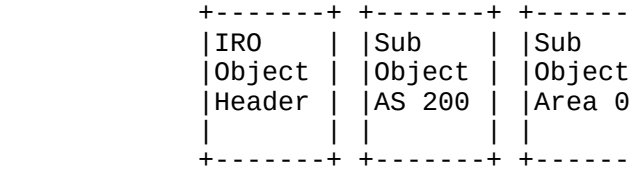
For LSP (A-B), where ingress A is in (AS 100, Area 0), egress B in (AS 200, Area 4) and transit through (AS 200, Area 0). Some possible way a PCC can encode the IRO:



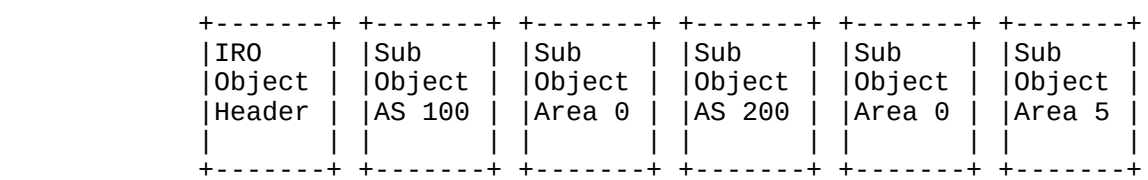
or



For LSP (A-C), where ingress A is in (AS 100, Area 0), egress C in (AS 200, Area 5) and transit through (AS 200, Area 0). Some possible way a PCC can encode the IRO:



or

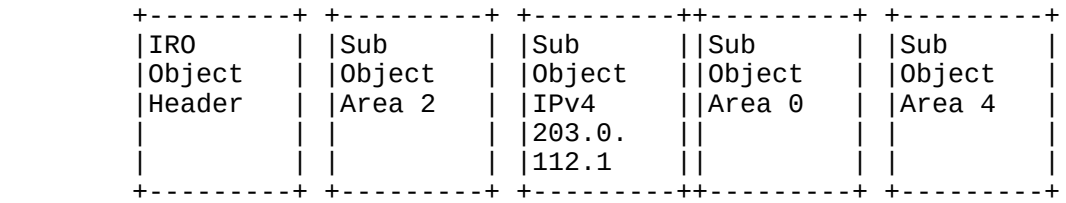


4.3. Boundary Node and Inter-AS-Link

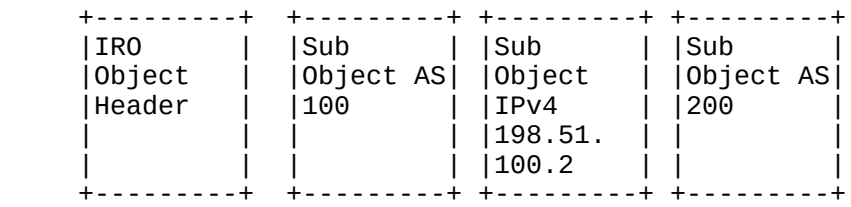
A PCC or PCE can include additional constraints covering which Boundary Nodes (ABR or ASBR) or Border links (Inter-AS-link) to be traversed while defining a Domain-Sequence. In which case the Boundary Node or Link can be encoded as a part of the Domain-Sequence.

Boundary Nodes (ABR / ASBR) can be encoded using the IPv4 or IPv6 prefix subobjects usually the loopback address of 32 and 128 prefix length respectively. An Inter-AS link can be encoded using the IPv4 or IPv6 prefix subobjects or unnumbered interface subobjects.

For Figure 1, an ABR (say 203.0.113.1) to be traversed can be specified in IRO as:



For Figure 3, an inter-AS-link (say 198.51.100.1 - 198.51.100.2) to be traversed can be specified as:

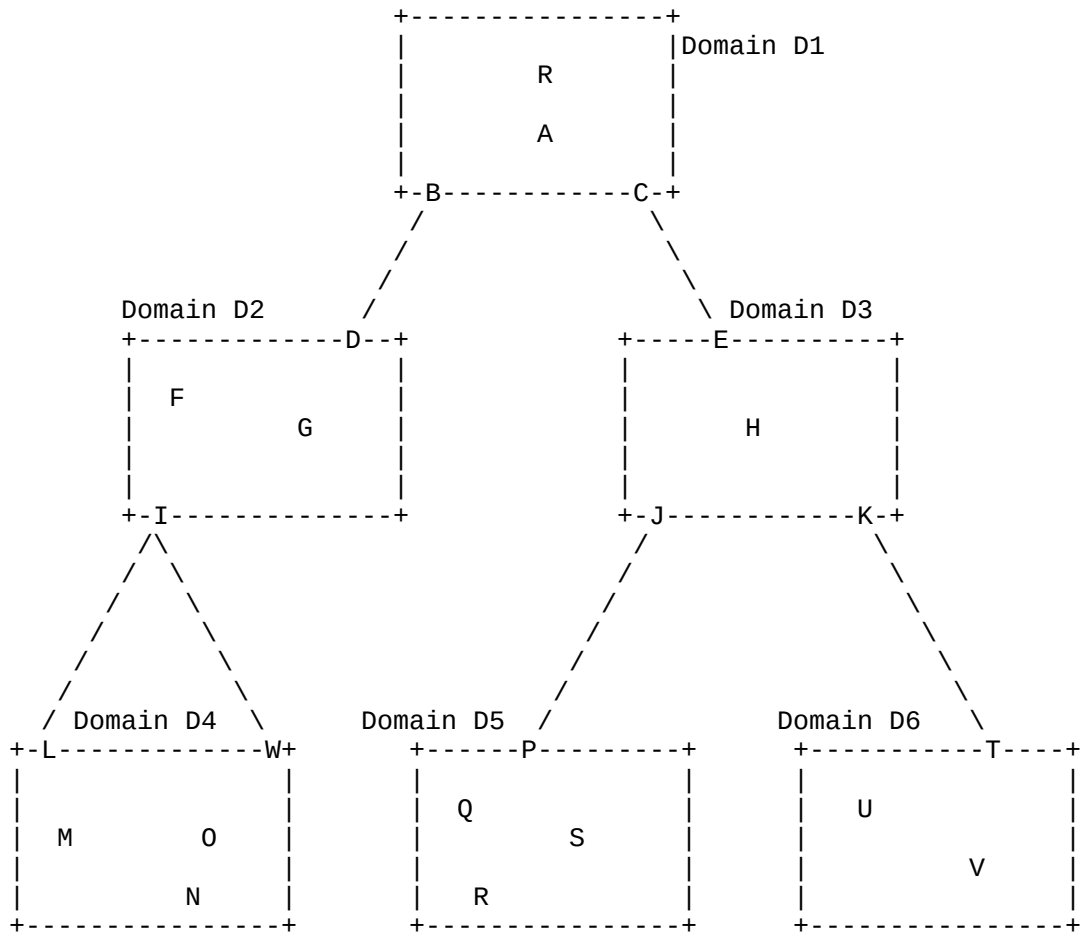


4.4. PCE Serving multiple Domains

A single PCE can be responsible for multiple domains; for example PCE function deployed on an ABR could be responsible for multiple areas. A PCE which can support adjacent domains can internally handle those domains in the Domain-Sequence without any impact on the other domains in the Domain-Sequence.

4.5. P2MP

[RFC7334] describes an experimental inter-domain P2MP path computation mechanism where the path domain tree is described as a series of Domain-Sequences, an example is shown in the below figure:



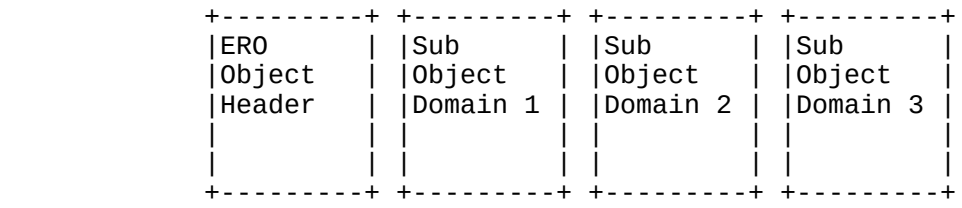
The domain tree can be represented as a series of domain-sequence -

- Domain D1, Domain D3, Domain D6
- Domain D1, Domain D3, Domain D5
- Domain D1, Domain D2, Domain D4

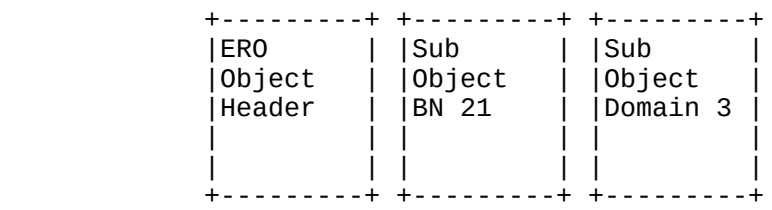
The domain sequence handling described in this document could be applied to P2MP path domain tree.

4.6. Hierarchical PCE

In case of H-PCE [RFC6805], the parent PCE can be requested to determine the Domain-Sequence and return it in the path computation reply, using the ERO. . For the example in section 4.6 of [RFC6805], the Domain-Sequence can possibly appear as:



or



5. Other Considerations

5.1. Relationship to PCE Sequence

Instead of a Domain-Sequence, a sequence of PCEs MAY be enforced by policy on the PCC, and this constraint can be carried in the PCReq message (as defined in [RFC5886]).

Note that PCE-Sequence can be used along with Domain-Sequence in which case PCE-Sequence MUST have higher precedence in selecting the next PCE in the inter-domain path computation procedures.

5.2. Relationship to RSVP-TE

[RFC3209] already describes the notion of abstract nodes, where an abstract node is a group of nodes whose internal topology is opaque to the ingress node of the LSP. It further defines a subobject for AS but with a 2-Byte AS Number.

[DOMAIN-SUBOBJ] extends the notion of abstract nodes by adding new subobjects for IGP Areas and 4-byte AS numbers. These subobjects can

be included in Explicit Route Object (ERO), Exclude Route object (XRO) or Explicit Exclusion Route Subobject (EXRS) in RSVP-TE.

In any case subobject type defined in RSVP-TE are identical to the subobject type defined in the related documents in PCEP.

6. IANA Considerations

6.1. New Subobjects

IANA maintains the "Path Computation Element Protocol (PCEP) Numbers" at <http://www.iana.org/assignments/pcep>. Within this registry IANA maintains two sub-registries:

- IRO Subobjects (see IRO Subobjects at http://www.iana.org/assignments/pcep)
- XRO Subobjects (see XRO Subobjects at http://www.iana.org/assignments/pcep)

Upon approval of this document, IANA is requested to make identical additions to these registries as follows:

Subobject Type		Reference
TBD1	4 byte AS number	[This I.D.][DOMAIN-SUBOBJ]
TBD2	OSPF Area ID	[This I.D.][DOMAIN-SUBOBJ]
TBD3	IS-IS Area ID	[This I.D.][DOMAIN-SUBOBJ]

Further upon approval of this document, IANA is requested to add a reference to this document to the new RSVP numbers that are registered by [DOMAIN-SUBOBJ].

7. Security Considerations

The protocol extensions defined in this document do not substantially change the nature of PCEP. Therefore, the security considerations set out in [RFC5440] apply unchanged. Note that further security considerations for the use of PCEP over TCP are presented in [RFC6952].

This document specifies a representation of Domain-Sequence and new subobjects, which could be used in inter-domain PCE scenarios as explained in [RFC5152], [RFC5441], [RFC6805], [RFC7334] etc. The security considerations set out in each of these mechanisms remain unchanged by the new subobjects and Domain-Sequence representation in this document.

But the new subobjects do allow finer and more specific control of the path computed by a cooperating PCE(s). Such control increases the risk if a PCEP message is intercepted, modified, or spoofed because it allows the attacker to exert control over the path that the PCE will compute or to make the path computation impossible. Consequently, it is important that implementations conform to the relevant security requirements of [RFC5440]. These mechanisms include:

- Securing the PCEP session messages using TCP security techniques (Section 10.2 of [RFC5440]). PCEP implementations SHOULD also consider the additional security provided by the TCP Authentication Option (TCP-AO) [RFC5925] or [PCEPS].

- Authenticating the PCEP messages to ensure the message is intact and sent from an authorized node (Section 10.3 of [RFC5440]).

- PCEP operates over TCP, so it is also important to secure the PCE and PCC against TCP denial-of-service attacks. Section 10.7.1 of [RFC5440] outlines a number of mechanisms for minimizing the risk of TCP-based denial-of-service attacks against PCEs and PCCs.

- In inter-AS scenarios, attacks may be particularly significant with commercial as well as service-level implications.

Note, however, that the Domain-Sequence mechanisms also provide the operator with the ability to route around vulnerable parts of the network and may be used to increase overall network security.

8. Manageability Considerations

8.1. Control of Function and Policy

The exact behaviour with regards to desired inclusion and exclusion of domains MUST be available for examination by an operator and MAY be configurable. Manual configurations is needed to identify which PCEP peers understand the new domain subobjects defined in this document.

8.2. Information and Data Models

A MIB module for management of the PCEP is being specified in a separate document [RFC7420]. This document does not imply any new extension to the current MIB module.

8.3. Liveness Detection and Monitoring

Mechanisms defined in this document do not imply any new liveness detection and monitoring requirements in addition to those already listed in [RFC5440].

8.4. Verify Correct Operations

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

8.5. Requirements On Other Protocols

In case of per-domain path computation [RFC5152], where the full path of an inter-domain TE LSP cannot be, or is not determined at the ingress node, a signaling message can use the domain identifiers. The Subobjects defined in this document SHOULD be supported by RSVP-TE. [DOMAIN-SUBOBJ] extends the notion of abstract nodes by adding new subobjects for IGP Areas and 4-byte AS numbers.

Apart from this, mechanisms defined in this document do not imply any requirements on other protocols in addition to those already listed in [RFC5440].

8.6. Impact On Network Operations

The mechanisms described in this document can provide the operator with the ability to exert finer and more specific control of the path computation by inclusion or exclusion of domain subobjects. There may be some scaling benefit when a single domain subobject may substitute for many subobjects and can reduce the overall message size and processing.

Backward compatibility issues associated with the new subobjects arise when a PCE does not recognize them, in which case PCE responds according to the rules for a malformed object as per [RFC5440]. For successful operations the PCEs in the network would need to be upgraded.

9. Acknowledgments

Authors would like to especially thank Adrian Farrel for his detailed reviews as well as providing text to be included in the document.

Further, we would like to thank Pradeep Shastry, Suresh Babu, Quintin Zhao, Fatai Zhang, Daniel King, Oscar Gonzalez, Chen Huaimo,

Venugopal Reddy, Reeja Paul, Sandeep Boina, Avantika Sergio Belotti and Jonathan Hardwick for their useful comments and suggestions.

Thanks to Jonathan Hardwick for shepherding this document.

Thanks to Deborah Brungard for being the Responsible AD.

Thanks to Amanda Baber for IANA Review.

Thanks to Joel Halpern for Gen-ART Review.

Thanks to Klaas Wierenga for SecDir Review.

Thanks to Spencer Dawkins and Barry Leiba for comments during the IESG Review.

10. References

10.1. Normative References

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

[RFC3209] Awduche, D., Berger, L., Gan, D., Li, T., Srinivasan, V., and G. Swallow, "RSVP-TE: Extensions to RSVP for LSP Tunnels", RFC 3209, DOI 10.17487/RFC3209, December 2001, <http://www.rfc-editor.org/info/rfc3209>.

[RFC3473] Berger, L., Ed., "Generalized Multi-Protocol Label Switching (GMPLS) Signaling Resource ReserVation Protocol-Traffic Engineering (RSVP-TE) Extensions", RFC 3473, DOI 10.17487/RFC3473, January 2003, <http://www.rfc-editor.org/info/rfc3473>.

[RFC3477] Kompella, K. and Y. Rekhter, "Signalling Unnumbered Links in Resource ReSerVation Protocol - Traffic Engineering (RSVP-TE)", RFC 3477, DOI 10.17487/RFC3477, January 2003, <http://www.rfc-editor.org/info/rfc3477>.

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

[RFC5441] Vasseur, JP., Ed., Zhang, R., Bitar, N., and JL. Le Roux, "A Backward-Recursive PCE-Based Computation (BRPC) Procedure to Compute Shortest Constrained Inter-Domain Traffic Engineering Label Switched Paths", RFC 5441, DOI 10.17487/RFC5441, April 2009, <http://www.rfc-editor.org/info/rfc5441>.

[RFC5521] Oki, E., Takeda, T., and A. Farrel, "Extensions to the Path Computation Element Communication Protocol (PCEP) for Route Exclusions", RFC 5521, DOI 10.17487/RFC5521, April 2009, <http://www.rfc-editor.org/info/rfc5521>.

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

[ISO10589] ISO, "Intermediate system to Intermediate system routing information exchange protocol for use in conjunction with the Protocol for providing the Connectionless-mode Network Service (ISO 8473)", ISO/IEC 10589:2002, 1992.

[IRO-UPDATE] Dhody, D., "Update to Include Route Object (IRO) specification in Path Computation Element communication Protocol (PCEP. (draft-ietf-pce-iro-update-02)", May 2015.

[DOMAIN-SUBOBJ] Dhody, D., Palle, U., Kondreddy, V., and R. Casellas, "Domain Subobjects for Resource ReserVation Protocol - Traffic Engineering (RSVP-TE). (draft-ietf-teas-rsvp-te-domain-subobjects-05)", November 2015.

10.2. Informative References

[RFC4655] Farrel, A., Vasseur, J., and J. Ash, "A Path Computation Element (PCE)-Based Architecture", RFC 4655, DOI 10.17487/RFC4655, August 2006, <http://www.rfc-editor.org/info/rfc4655>.

[RFC4726] Farrel, A., Vasseur, J., and A. Ayyangar, "A Framework for Inter-Domain Multiprotocol Label Switching Traffic Engineering", RFC 4726, DOI 10.17487/RFC4726, November 2006, <http://www.rfc-editor.org/info/rfc4726>.

[RFC4873] Berger, L., Bryskin, I., Papadimitriou, D., and A. Farrel, "GMPLS Segment Recovery", RFC 4873, DOI 10.17487/RFC4873, May 2007, <http://www.rfc-editor.org/info/rfc4873>.

[RFC4874] Lee, CY., Farrel, A., and S. De Cnodder, "Exclude Routes - Extension to Resource ReserVation Protocol-Traffic Engineering (RSVP-TE)", RFC 4874, DOI 10.17487/RFC4874, April 2007, <http://www.rfc-editor.org/info/rfc4874>.

[RFC5152] Vasseur, JP., Ed., Ayyangar, A., Ed., and R. Zhang, "A Per-Domain Path Computation Method for Establishing Inter-Domain Traffic Engineering (TE) Label Switched Paths (LSPs)", RFC 5152, DOI 10.17487/RFC5152, February 2008, <http://www.rfc-editor.org/info/rfc5152>.

[RFC5520] Bradford, R., Ed., Vasseur, JP., and A. Farrel, "Preserving Topology Confidentiality in Inter-Domain Path Computation Using a Path-Key-Based Mechanism", RFC 5520, DOI 10.17487/RFC5520, April 2009, <http://www.rfc-editor.org/info/rfc5520>.

[RFC5886] Vasseur, JP., Ed., Le Roux, JL., and Y. Ikejiri, "A Set of Monitoring Tools for Path Computation Element (PCE)-Based Architecture", RFC 5886, DOI 10.17487/RFC5886, June 2010, <http://www.rfc-editor.org/info/rfc5886>.

[RFC5925] Touch, J., Mankin, A., and R. Bonica, "The TCP Authentication Option", RFC 5925, DOI 10.17487/RFC5925, June 2010, <http://www.rfc-editor.org/info/rfc5925>.

[RFC6793] Vohra, Q. and E. Chen, "BGP Support for Four-Octet Autonomous System (AS) Number Space", RFC 6793, DOI 10.17487/RFC6793, December 2012, <http://www.rfc-editor.org/info/rfc6793>.

[RFC6952] Jethanandani, M., Patel, K., and L. Zheng, "Analysis of BGP, LDP, PCEP, and MSDP Issues According to the Keying and Authentication for Routing Protocols (KARP) Design Guide", RFC 6952, DOI 10.17487/RFC6952, May 2013, <http://www.rfc-editor.org/info/rfc6952>.

[RFC7334] Zhao, Q., Dhody, D., King, D., Ali, Z., and R. Casellas, "PCE-Based Computation Procedure to Compute Shortest Constrained Point-to-Multipoint (P2MP) Inter-Domain Traffic Engineering Label Switched Paths", RFC 7334, DOI 10.17487/RFC7334, August 2014, <http://www.rfc-editor.org/info/rfc7334>.

[RFC7420] Koushik, A., Stephan, E., Zhao, Q., King, D., and J. Hardwick, "Path Computation Element Communication Protocol (PCEP) Management Information Base (MIB) Module", RFC 7420, DOI 10.17487/RFC7420, December 2014, <http://www.rfc-editor.org/info/rfc7420>.

[PCEPS] Lopez, D., Dios, O., Wu, W., and D. Dhody, "Secure Transport for PCEP", draft-ietf-pce-pceps-06 (work in progress), November 2015.

Authors' Addresses

Dhruv Dhody
Huawei Technologies
Divyashree Techno Park, Whitefield
Bangalore, Karnataka 560037
India

EMail: dhruv.ietf@gmail.com

Udayasree Palle
Huawei Technologies
Divyashree Techno Park, Whitefield
Bangalore, Karnataka 560037
India

EMail: udayasree.palle@huawei.com

Ramon Casellas
CTTC
Av. Carl Friedrich Gauss n7
Castelldefels, Barcelona 08860
Spain

EMail: ramon.casellas@cttc.es

Path Computation Element
Internet-Draft
Intended status: Experimental
Expires: April 5, 2015

D. Lopez
O. Gonzalez de Dios
Telefonica I+D
Q. Wu
D. Dhody
Huawei

October 2, 2014

Secure Transport for PCEP
draft-ietf-pce-pceps-02

Abstract

The Path Computation Element Communication Protocol (PCEP) defines the mechanisms for the communication between a Path Computation Client (PCC) and a Path Computation Element (PCE), or among PCEs. This document describe the usage of Transport Layer Security (TLS) to enhance PCEP security, hence the PCEPS acronym proposed for it. The additional security mechanisms are provided by the transport protocol supporting PCEP, and therefore they do not affect its flexibility and extensibility.

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 April 5, 2015.

Copyright Notice

Table of Contents

1. Introduction 2
2. Requirements Language 3
3. Applying PCEPS 3
3.1. Overview 3
3.2. Initiating the TLS Procedures 4
3.3. The StartTLS Message 5
3.4. TLS Connection Establishment 7
3.5. Peer Identity 9
3.6. Connection Establishment Failure 10
4. Discovery Mechanisms 10
4.1. DANE Applicability 11
5. Backward Compatibility 11
6. IANA Considerations 11
6.1. New PCEP Message 11
6.2. New Error-Values 11
7. Security Considerations 12
8. Acknowledgements 13
9. References 13
9.1. Normative References 13
9.2. Informative References 14
Authors' Addresses 14

1. Introduction

PCEP [RFC5440] defines the mechanisms for the communication between a Path Computation Client (PCC) and a Path Computation Element (PCE), or between two PCEs. These interactions include requests and replies that can be critical for a sustainable network operation and adequate resource allocation, and therefore appropriate security becomes a key element in the PCE infrastructure. As the applications of the PCE framework evolves, and more complex service patterns emerge, the definition of a secure mode of operation becomes more relevant.

[RFC5440] analyzes in its section on security considerations the potential threats to PCEP and their consequences, and discusses several mechanisms for protecting PCEP against security attacks, without making a specific recommendation on a particular one or defining their application in depth. Moreover, [RFC6952] remarks the importance of ensuring PCEP communication privacy, especially when

PCEP communication endpoints do not reside in the same AS, as the interception of PCEP messages could leak sensitive information related to computed paths and resources.

Among the possible solutions mentioned in these documents, Transport Layer Security (TLS) [RFC5246] provides support for peer authentication, and message encryption and integrity. TLS supports the usage of well-know mechanisms to support key configuration and exchange, and means to perform security checks on the results of PCE discovery procedures via IGP ([RFC5088] and [RFC5089]).

This document describes a security container for the transport of PCEP requests and replies, and therefore it will not interfere with the protocol flexibility and extensibility.

This document describes how to apply TLS in securing PCE interactions, including initiation of the TLS procedures, the TLS handshake mechanisms, the TLS methods for peer authentication, the applicable TLS ciphersuites for data exchange, and the handling of errors in the security checks. In the rest of the document we will refer to this usage of TLS to provide a secure transport for PCEP as "PCEPS".

2. Requirements Language

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

3. Applying PCEPS

3.1. Overview

The steps involved in the PCEPS establishment consists of following successive steps:

1. Establishment of a TCP connection.

2. Initiating the TLS Procedures by StartTLS message.

3. Establishment of TLS connection.

4. Start exchanging PCEP messages as per [RFC5440].

It should be noted that this procedure update what is defined in section 6.7 of [RFC5440] regarding the processing of messages prior to Open message. The details of processing including backward comaptibility is discussed below.

3.2. Initiating the TLS Procedures

Since PCEP can operate either with or without TLS, it is necessary for the PCEP speaker to indicate whether it wants to set up a TLS connection or not. For this purpose, this document proposes a new PCEP message, StartTLS, that MUST be issued by the party willing to use TLS prior to any other PCEP message. PCEP speaker MAY discover that the PCEP peer supports PCEPS or can be preconfigured to use PCEPS for a given peer (see Section 4 for more details). Thus the PCEP session is secured via TLS from the start before exchange of any other PCEP message including open message. Securing via TLS an existing PCEP session is not permitted, session must be closed and restablished with TLS as per the procedure described in this document.

The StartTLS message is a PCEP message sent by a PCC to a PCE and by a PCE to a PCC in order to initiate the TLS procedure for PCEP. The Message-Type field of the PCEP common header for the StartTLS message is set to [TBA].

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 a StartTLS message for the PCEPS. Note this is a significant change from [RFC5440] where the first PCEP message is Open.

A PCEP speaker receiving a StartTLS message after any other PCEP exchange has taken place (by receiving or sending any other messages from either side) MUST treat it as an unexpected message and reply with a PCErr message with Error-Type set to xx (TBA by IANA)(PCEP StartTLS failure) and Error-value set to 1 (reception of StartTLS after any PCEP exchange). A PCEP speaker receives any other message apart from StartTLS or PCErr MUST treat it as an unexpected message and reply with a PCErr message with Error-Type set to xx (TBA by IANA)(PCEP StartTLS failure) and Error-value set to 2 (reception of non-StartTLS or non-PCErr message).

If the PCEP speaker does not support PCEPS and receives a StartTLS message it MUST behave as described in section 6.2 of [RFC5440] in case message is received prior to an Open message or as described in section 6.9 of [RFC5440] for the case of reception of unknown message.

If the PCEP speaker supports PCEPS but cannot establish a TLS connection for some reason (e.g. the certificate server is not responding) it MUST return a PCErr message with Error-Type set to xx (TBA by IANA) (PCEP StartTLS failure) and Error-value set to:

- 3 (not without TLS) if it is not willing to exchange PCEP messages without the solicited TLS connection
- 4 (ok without TLS) if it is willing to exchange PCEP messages without the solicited TLS connection

If the PCEP speaker supports PCEPS and can establish a TLS connection it MUST start the TLS connection establishment steps described in Section 3.4 below before PCEP initialization procedure listed in section 4.2.1 of [RFC5440].

These procedures minimize the impact of PCEPS support in PCEP implementations without requiring additional dedicated ports for running PCEP on TLS.

3.3. The StartTLS Message

The StartTLS message is used to initiate the TLS procedure for a PCEP session between the PCEP peers. A PCEP speaker sends the StartTLS message to request negotiation and establishment of TLS connection for PCEP. On receiving a StartTLS message form the PCEP peer (i.e. when PCEP speaker has sent and received StartTLS message) it is ready to start TLS negotiation and establishment and move to steps described in Section 3.4.

The format of a StartTLS message is as follows:

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

The StartTLS message MUST contain only the PCEP common header with Message-Type field set to [TBA].

Once the TCP connection has been successfully established, the sender MUST start a timer called StartTLSWait after the expiration of which, if no StartTLS message has been received, it sends a PCErr message and releases the TCP connection with Error-Type set to xx (TBA by IANA) and Error-value set to 5 (no StartTLS message received before the expiration of the StartTLSWait timer).

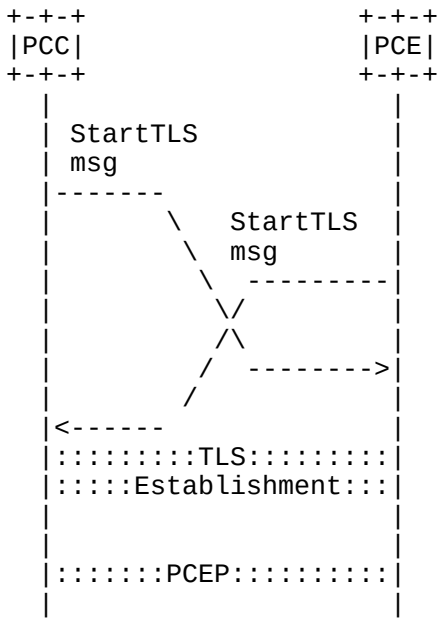


Figure 1: Both PCEP Speaker supports PCEPS

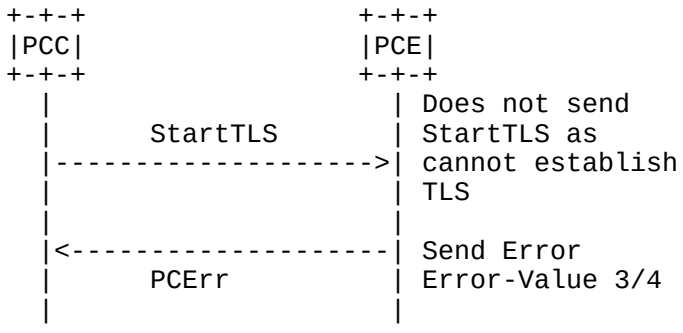


Figure 2: Both PCEP Speaker supports PCEPS, But cannot establish TLS

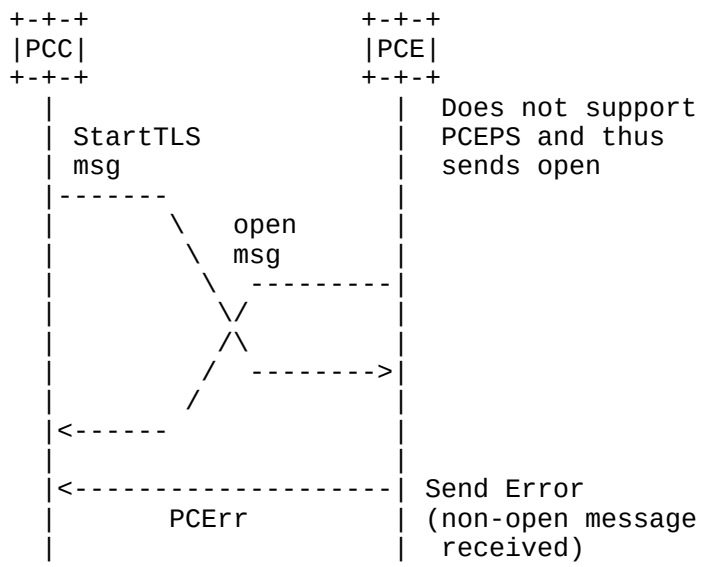


Figure 3: One PCEP Speaker does not support PCEPS

3.4. TLS Connection Establishment

Once the establishment of TLS has been agreed by the PCEP peers, the connection establishment SHALL follow the following steps:

1. Immediately negotiate TLS sessions according to [RFC5246]. The following restrictions apply:

   * Support for TLS v1.2 [RFC5246] or later is REQUIRED.

   * Support for certificate-based mutual authentication is REQUIRED.

   * Negotiation of mutual authentication is REQUIRED.

   * Negotiation of a ciphersuite providing for integrity protection is REQUIRED.

   * Negotiation of a ciphersuite providing for confidentiality is RECOMMENDED.

   * Support for and negotiation of compression is OPTIONAL.

* PCEPS implementations MUST, at a minimum, support negotiation of the TLS_RSA_WITH_3DES_EDE_CBC_SHA, and SHOULD support TLS_RSA_WITH_RC4_128_SHA and TLS_RSA_WITH_AES_128_CBC_SHA as well. In addition, PCEPS implementations MUST support negotiation of the mandatory-to-implement ciphersuites required by the versions of TLS that they support.

2. Peer authentication can be performed in any of the following two REQUIRED operation models:

   * TLS with X.509 certificates using PKIX trust models:

     + Implementations MUST allow the configuration of a list of trusted Certification Authorities (CAs) for incoming connections.

     + Certificate validation MUST include the verification rules as per [RFC5280].

     + Implementations SHOULD indicate their trusted CAs. For TLS 1.2, this is done using [RFC5246], Section 7.4.4, "certificate_authorities" (server side) and [RFC6066], Section 6 "Trusted CA Indication" (client side).

     + Peer validation always SHOULD include a check on whether the locally configured expected DNS name or IP address of the peer that is contacted matches its presented certificate. DNS names and IP addresses can be contained in the Common Name (CN) or subjectAltName entries. For verification, only one of these entries is to be considered. The following precedence applies: for DNS name validation, subjectAltName:DNS has precedence over CN; for IP address validation, subjectAltName:iPAddr has precedence over CN.

     + Implementations MAY allow the configuration of a set of additional properties of the certificate to check for a peer’s authorization to communicate (e.g., a set of allowed values in subjectAltName:URI or a set of allowed X509v3 Certificate Policies)

   * TLS with X.509 certificates using certificate fingerprints: Implementations MUST allow the configuration of a list of trusted certificates, identified via fingerprint of the Distinguished Encoding Rules (DER) encoded certificate octets. Implementations MUST support SHA-256 as the hash algorithm for the fingerprint.

3. Start exchanging PCEP messages.

To support TLS re-negotiation both peers MUST support the mechanism described in [RFC5746]. Any attempt of initiate a TLS handshake to establish new cryptographic parameters not aligned with [RFC5746] SHALL be considered a TLS negotiation failure.

3.5. Peer Identity

Depending on the peer authentication method in use, PCEPS supports different operation modes to establish peer's identity and whether it is entitled to perform requests or can be considered authoritative in its replies. PCEPS implementations SHOULD provide mechanisms for associating peer identities with different levels of access and/or authoritativeness, and they MUST provide a mechanism for establish a default level for properly identified peers. Any connection established with a peer that cannot be properly identified SHALL be terminated before any PCEP exchange takes place.

In TLS-X.509 mode using fingerprints, a peer is uniquely identified by the fingerprint of the presented client certificate.

There are numerous trust models in Public-Key Infrastructure (PKI) environments, and it is beyond the scope of this document to define how a particular deployment determines whether a client is trustworthy. Implementations that want to support a wide variety of trust models should expose as many details of the presented certificate to the administrator as possible so that the trust model can be implemented by the administrator. As a suggestion, at least the following parameters of the X.509 client certificate should be exposed:

- Peer's IP address
- Peer's fully qualified domain name (FQDN)
- Certificate Fingerprint
- Issuer
- Subject
- All X509v3 Extended Key Usage
- All X509v3 Subject Alternative Name
- All X509v3 Certificate Policies

In addition, a PCC MAY apply the procedures described in [RFC6698] (DANE) to verify its peer identity when using DNS discovery. See section Section 4.1 for further details.

3.6. Connection Establishment Failure

In case the initial TLS negotiation or the peer identity check fail according to the procedures listed in this document, the peer MUST immediately terminate the session. It SHOULD follow the procedure listed in [RFC5440] to retry session setup along with an exponential back-off session establishment retry procedure.

4. Discovery Mechanisms

A PCE can advertise its capability to support PCEPS using the IGP advertisement and discovery mechanism. The PCE-CAP-FLAGS sub-TLV is an optional sub-TLV used to advertise PCE capabilities. It MAY be present within the PCED sub-TLV carried by OSPF or IS-IS. [RFC5088] and [RFC5089] provide the description and processing rules for this sub-TLV when carried within OSPF and IS-IS, respectively. PCE capability bits are defined in [RFC5088]. A new capability flag bit for the PCE-CAP-FLAGS sub-TLV that can be announced as attribute to distribute PCEP security support information is proposed in [I-D.wu-pce-discovery-pceps-support]

When DNS is used by a PCC (or a PCE acting as a client, for the rest of the section, PCC refers to both) willing to use PCEPS to locate an appropriate PCE [I-D.wu-pce-dns-pce-discovery], the PCC as initiating entity chooses at least one of the returned FQDNs to resolve, which it does by performing DNS "A" or "AAAA" lookups on the FDQN. This will eventually result in an IPv4 or IPv6 address. The PCC SHALL use the IP address(es) from the successfully resolved FDQN (with the corresponding port number returned by the DNS SRV lookup) as the connection address(es) for the receiving entity.

If the PCC fails to connect using an IP address but the "A" or "AAAA" lookups returned more than one IP address, then the PCC SHOULD use the next resolved IP address for that FDQN as the connection address. If the PCC fails to connect using all resolved IP addresses for a given FDQN, then it SHOULD repeat the process of resolution and connection for the next FQDN returned by the SRV lookup based on the priority and weight.

If the PCC receives a response to its SRV query but it is not able to establish a PCEPS connection using the data received in the response, as initiating entity it MAY fall back to lookup a PCE that uses TCP as transport.

4.1. DANE Applicability

DANE [RFC6698] defines a secure method to associate the certificate that is obtained from a TLS server with a domain name using DNS, i.e., using the TLSA DNS resource record (RR) to associate a TLS server certificate or public key with the domain name where the record is found, thus forming a "TLSA certificate association". The DNS information needs to be protected by DNSSEC. A PCC willing to apply DANE to verify server identity MUST conform to the rules defined in section 4 of [RFC6698].

5. Backward Compatibility

The procedures described in this document define a security container for the transport of PCEP requests and replies carried by a TLS connection initiated by means of a specific extended message (StartTLS) that does not interfere with PCEP speaker implementations not supporting it.

6. IANA Considerations

6.1. New PCEP Message

Each PCEP message has a message type value.

One new PCEP messages is defined in this document:

Value	Description	Reference
TBA	The Start TLS Message (StartTLS)	This document

6.2. New Error-Values

A registry was created for the Error-type and Error-value of the PCEP Error Object. Following new Error-Types and Error-Values are defined:

Error-Type	Meaning	Reference
TBA	StartTLS Failure	This document
	Error-value=1: Reception of StartTLS after any PCEP exchange	This document
	Error-value=2: Reception of non-StartTLS or non-PCErr message	This document
	Error-value=3: Failure, connection without TLS not possible	This document
	Error-value=4: Failure, connection without TLS possible	This document
	Error-value=5: No StartTLS message before StartTLSWait timer expiry	This document

7. Security Considerations

While the application of TLS satisfies the requirement on privacy as well as fine-grained, policy-based peer authentication, there are security threats that it cannot address. It is advisable to apply additional protection measures, in particular in what relates to attacks specifically addressed to forging the TCP connection underpinning TLS. TCP-AO (TCP Authentication Option [RFC5925]) is fully compatible with and deemed as complementary to TLS, so its usage is to be considered as a security enhancement whenever any of the PCEPS peers require it, especially in the case of long-lived connections. The mechanisms to configure the requirements to use TCP-AO and other lower-layer protection measures, as well as the association of the required crypto material (MKT in the case of TCP-AO) with a particular peer are outside the scope of this document. [I-D.chunduri-karp-using-ikev2-with-tcp-ao] defines a method to perform such association.

Since computational resources required by TLS handshake and ciphersuite are higher than unencrypted TCP, clients connecting to a PCEPS server can more easily create high load conditions and a malicious client might create a Denial-of-Service attack more easily.

Some TLS ciphersuites only provide integrity validation of their payload, and provide no encryption. This specification does not forbid the use of such ciphersuites, but administrators must weight carefully the risk of relevant internal data leakage that can occur in such a case, as explicitly stated by [RFC6952].

When using certificate fingerprints to identify PCEPS peers, any two certificates that produce the same hash value will be considered the same peer. Therefore, it is important to make sure that the hash function used is cryptographically uncompromised so that attackers are very unlikely to be able to produce a hash collision with a certificate of their choice. This document mandates support for SHA-256, but a later revision may demand support for stronger functions if suitable attacks on it are known.

8. Acknowledgements

This specification relies on the analysis and profiling of TLS included in [RFC6614] and the procedures described for the STARTTLS command in [RFC2830].

We would like to thank Joe Touch for his suggestions and support regarding the TLS start mechanisms.

9. References

9.1. Normative References

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

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

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

[RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, August 2008.

[RFC5280] Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., and W. Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 5280, May 2008.

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

[RFC5746] Rescorla, E., Ray, M., Dispensa, S., and N. Oskov, "Transport Layer Security (TLS) Renegotiation Indication Extension", RFC 5746, February 2010.

[RFC5925] Touch, J., Mankin, A., and R. Bonica, "The TCP Authentication Option", RFC 5925, June 2010.

[RFC6066] Eastlake, D., "Transport Layer Security (TLS) Extensions: Extension Definitions", RFC 6066, January 2011.

[RFC6698] Hoffman, P. and J. Schlyter, "The DNS-Based Authentication of Named Entities (DANE) Transport Layer Security (TLS) Protocol: TLSA", RFC 6698, August 2012.

9.2. Informative References

[I-D.chunduri-karp-using-ikev2-with-tcp-ao] Chunduri, U., Tian, A., and J. Touch, "A framework for RPs to use IKEv2 KMP", draft-chunduri-karp-using-ikev2-with-tcp-ao-06 (work in progress), February 2014.

[I-D.wu-pce-discovery-pceps-support] Lopez, D., Wu, Q., Dhody, D., and D. King, "IGP extension for PCEP security capability support in the PCE discovery", draft-wu-pce-discovery-pceps-support-01 (work in progress), August 2014.

[I-D.wu-pce-dns-pce-discovery] Wu, W., Dhody, D., King, D., Lopez, D., and J. Tantsura, "Path Computation Element (PCE) Discovery using Domain Name System(DNS)", draft-wu-pce-dns-pce-discovery-06 (work in progress), May 2014.

[RFC2830] Hodges, J., Morgan, R., and M. Wahl, "Lightweight Directory Access Protocol (v3): Extension for Transport Layer Security", RFC 2830, May 2000.

[RFC6614] Winter, S., McCauley, M., Venaas, S., and K. Wierenga, "Transport Layer Security (TLS) Encryption for RADIUS", RFC 6614, May 2012.

[RFC6952] Jethanandani, M., Patel, K., and L. Zheng, "Analysis of BGP, LDP, PCEP, and MSDP Issues According to the Keying and Authentication for Routing Protocols (KARP) Design Guide", RFC 6952, May 2013.

Authors' Addresses

Diego R. Lopez
Telefonica I+D
Don Ramon de la Cruz, 82
Madrid 28006
Spain

Phone: +34 913 129 041
Email: diego.r.lopez@telefonica.com

Oscar Gonzalez de Dios
Telefonica I+D
Don Ramon de la Cruz, 82
Madrid 28006
Spain

Phone: +34 913 129 041
Email: oscar.gonzalezdedios@telefonica.com

Qin Wu
Huawei
101 Software Avenue, Yuhua District
Nanjing, Jiangsu 210012
China

Email: sunseawq@huawei.com

Dhruv Dhody
Huawei
Leela Palace
Bangalore, KA 560008
India

Email: dhruv.ietf@gmail.com

PCE Working Group
Internet-Draft
Updates: 5440 (if approved)
Intended status: Standards Track
Expires: March 8, 2018

D. Lopez
O. Gonzalez de Dios
Telefonica I+D
Q. Wu
D. Dhody
Huawei

September 4, 2017

Secure Transport for PCEP
draft-ietf-pce-pceps-18

Abstract

The Path Computation Element Communication Protocol (PCEP) defines the mechanisms for the communication between a Path Computation Client (PCC) and a Path Computation Element (PCE), or among PCEs. This document describes the usage of Transport Layer Security (TLS) to enhance PCEP security, hence the PCEPS acronym proposed for it. The additional security mechanisms are provided by the transport protocol supporting PCEP, and therefore they do not affect the flexibility and extensibility of PCEP.

This document updates RFC 5440 in regards to the PCEP initialization phase procedures.

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 March 8, 2018.

Copyright Notice

Table of Contents

1. Introduction 3
2. Requirements Language 4
3. Applying PCEPS 4
3.1. Overview 4
3.2. Initiating the TLS Procedures 4
3.3. The StartTLS Message 7
3.4. TLS Connection Establishment 12
3.5. Peer Identity 14
3.6. Connection Establishment Failure 15
4. Discovery Mechanisms 15
4.1. DANE Applicability 16
5. Backward Compatibility 16
6. IANA Considerations 17
6.1. New PCEP Message 17
6.2. New Error-Values 17
7. Security Considerations 18
8. Manageability Considerations 19
8.1. Control of Function and Policy 19
8.2. Information and Data Models 20
8.3. Liveness Detection and Monitoring 20
8.4. Verifying Correct Operations 20
8.5. Requirements on Other Protocols 20
8.6. Impact on Network Operation 21
9. Acknowledgements 21

10. References 21
10.1. Normative References 21
10.2. Informative References 23
Authors' Addresses 24

1. Introduction

The Path Computation Element Communication Protocol (PCEP) [RFC5440] defines the mechanisms for the communication between a Path Computation Client (PCC) and a Path Computation Element (PCE), or between two PCEs. These interactions include requests and replies that can be critical for a sustainable network operation and adequate resource allocation, and therefore appropriate security becomes a key element in the PCE infrastructure. As the applications of the PCE framework evolves, and more complex service patterns emerge, the definition of a secure mode of operation becomes more relevant.

[RFC5440] analyzes in its section on security considerations the potential threats to PCEP and their consequences, and discusses several mechanisms for protecting PCEP against security attacks, without making a specific recommendation on a particular one or defining their application in depth. Moreover, [RFC6952] remarks the importance of ensuring PCEP communication confidentiality, especially when PCEP communication endpoints do not reside in the same Autonomous System (AS), as the interception of PCEP messages could leak sensitive information related to computed paths and resources.

Among the possible solutions mentioned in these documents, Transport Layer Security (TLS) [RFC5246] provides support for peer authentication, message encryption and integrity. TLS provides well-known mechanisms to support key configuration and exchange, as well as means to perform security checks on the results of PCE discovery procedures via Interior Gateway Protocol (IGP) ([RFC5088] and [RFC5089]).

This document describes a security container for the transport of PCEP messages, and therefore it does not affect the flexibility and extensibility of PCEP.

This document describes how to apply TLS to secure interactions with PCE, including initiation of the TLS procedures, the TLS handshake mechanism, the TLS methods for peer authentication, the applicable TLS ciphersuites for data exchange, and the handling of errors in the security checks. In the rest of the document we will refer to this usage of TLS to provide a secure transport for PCEP as "PCEPS".

Within this document, PCEP communications are described through PCC-PCE relationship. The PCE architecture also supports the PCE-PCE

communication, this is achieved by requesting PCE fill the role of a PCC, as usual. Thus, in this document, the PCC refers to a PCC or a PCE initiating the PCEP session and acting as a client.

2. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

3. Applying PCEPS

3.1. Overview

The steps involved in establishing a PCEPS session are as follows:

1. Establishment of a TCP connection.

2. Initiating the TLS procedures by the StartTLS message from PCE to PCC and from PCC to PCE.

3. Negotiation and establishment of TLS connection.

4. Start exchanging PCEP messages as per [RFC5440].

This document uses the standard StartTLS procedure in PCEP, instead of using a different port for the secured session. This is done to avoid requesting allocation of another port number for the PCEPS. The StartTLS procedure makes more efficient use of scarce port numbers and allow simpler configuration of PCEP.

Implementations SHOULD follow the best practices and recommendations for using TLS, as per [RFC7525].

It should be noted that this procedure updates what is defined in section 4.2.1 and section 6.7 of [RFC5440] regarding the initialization phase and the processing of messages prior to the Open message. The details of processing, including backward compatibility, are discussed in the following sections.

3.2. Initiating the TLS Procedures

Since PCEP can operate either with or without TLS, it is necessary for a PCEP speaker to indicate whether it wants to set up a TLS connection or not. For this purpose, this document specifies a new PCEP message called StartTLS. Thus, the PCEP session is secured via

TLS from the start before exchange of any other PCEP message (that includes the Open message). This document thus updates [RFC5440], which required the Open message to be the first PCEP message that is exchanged. In the case of a PCEP session using TLS, the StartTLS message will be sent first. Also a PCEP speaker that supports PCEPS MUST NOT start the OpenWait timer after the TCP establishment, instead it starts a StartTLSWait timer as described in Section 3.3.

The PCEP speaker MAY discover that the PCEP peer supports PCEPS or can be preconfigured to use PCEPS for a given peer (see Section 4 for more details). An existing PCEP session cannot be secured via TLS, the session MUST be closed and re-established with TLS as per the procedure described in this document.

The StartTLS message is a PCEP message sent by a PCC to a PCE and by a PCE to a PCC in order to initiate the TLS procedure for PCEP. The PCC initiates the use of TLS by sending a StartTLS message. The PCE agrees to the use of TLS by responding with its own StartTLS message. If the PCE is configured to only support TLS, it may send the StartTLS message immediately upon TCP connection establishment; otherwise it MUST wait for the PCC's first message to see whether it is an Open or a StartTLS message. The TLS negotiation and establishment procedures are triggered once the PCEP speaker has sent and received the StartTLS message. The Message-Type field of the PCEP common header for the StartTLS message is set to [TBA1 by IANA].

Once the TCP connection has been successfully established, the first message sent by the PCC to the PCE and by the PCE to the PCC MUST be a StartTLS message for the PCEPS. Note that, this is a significant change from [RFC5440], where the first PCEP message is the Open message.

A PCEP speaker receiving a StartTLS message, after any other PCEP exchange has taken place (by receiving or sending any other messages from either side) MUST treat it as an unexpected message and reply with a PCErr message with Error-Type set to [TBA2 by IANA] (PCEP StartTLS failure) and Error-value set to 1 (reception of StartTLS after any PCEP exchange), and MUST close the TCP connection.

Any message received prior to StartTLS or Open message MUST trigger a protocol error condition causing a PCErr message to be sent with Error-Type set to [TBA2 by IANA] (PCEP StartTLS failure) and Error-value set to 2 (reception of a message apart from StartTLS or Open) and MUST close the TCP connection.

If the PCEP speaker that does not support PCEPS, receives a StartTLS message, it will behave according to the existing error mechanism described in section 6.2 of [RFC5440] (in case message is received

prior to an Open message) or section 6.9 of [RFC5440] (for the case of reception of unknown message). See Section 5 for more details.

If the PCEP speaker that only supports PCEPS connection (as a local policy), receives an Open message, it MUST treat it as an unexpected message and reply with a PCErr message with Error-Type set to 1 (PCEP session establishment failure) and Error-value set to 1 (reception of an invalid Open message or a non Open message), and MUST close the TCP connection.

If a PCC supports PCEPS connections as well as allow non-PCEPS connection (as a local policy), it MUST first try to establish PCEPS, by sending StartTLS message and in case it receives an PCErr message from the PCE, it MAY retry to establish connection without PCEPS by sending an Open message. If a PCE supports PCEPS connections as well as allow non-PCEPS connection (as a local policy), it MUST wait to respond after TCP establishment, based on the message received from the PCC. In case of StartTLS message, PCE MUST respond with sending a StartTLS message and moving to TLS establishment procedures as described in this document. In case of Open message, PCE MUST respond with Open message and move to PCEP session establishment procedure as per [RFC5440]. If a PCE supports PCEPS connections only (as a local policy), it MAY send StartTLS message to PCC without waiting to receive a StartTLS message from PCC.

If a PCEP speaker that is unwilling or unable to negotiate TLS receives a StartTLS messages, it MUST return a PCErr message (in clear) with Error-Type set to [TBA2 by IANA] (PCEP StartTLS failure) and Error-value set to:

- 3 (Failure, connection without TLS is not possible) if it is not willing to exchange PCEP messages without the solicited TLS connection, and it MUST close the TCP session.

- 4 (Failure, connection without TLS is possible) if it is willing to exchange PCEP messages without the solicited TLS connection, and it MUST close the TCP session. The receiver MAY choose to attempt to re-establish the PCEP session without TLS next. The attempt to re-establish the PCEP session without TLS SHOULD be limited to only once.

If the PCEP speaker supports PCEPS and can establish a TLS connection it MUST start the TLS connection negotiation and establishment steps described in Section 3.4 before the PCEP initialization procedure (section 4.2.1 of [RFC5440]).

After the exchange of StartTLS messages, if the TLS negotiation fails for some reason (e.g. the required mechanisms for certificate

revocation checking are not available), both peers MUST immediately close the connection.

A PCEP speaker that does not support PCEPS sends the Open message directly, as per [RFC5440]. A PCEP speaker that supports PCEPS, but has learned in the last exchange the peer's willingness to reestablish session without TLS, MAY send the Open message directly, as per [RFC5440]. The attempt to re-establish the PCEP session without TLS SHOULD be limited to only once.

Given the asymmetric nature of TLS for connection establishment, it is relevant to identify the roles of each of the PCEP peers in it. The PCC SHALL act as TLS client, and the PCE SHALL act as TLS server as per [RFC5246].

As per the recommendation from [RFC7525] to avoid downgrade attacks, PCEP peers that support PCEPS, SHOULD default to strict TLS configuration i.e. do not allow non-TLS PCEP sessions to be established. PCEPS implementations MAY provide an option to allow the operator to manually override strict TLS configuration and allow unsecured connections. Execution of this override SHOULD trigger a warning about the security implications of permitting unsecured connections.

3.3. The StartTLS Message

The StartTLS message is used to initiate the TLS procedure for a PCEPS session between the PCEP peers. A PCEP speaker sends the StartTLS message to request negotiation and establishment of TLS connection for PCEP. On receiving a StartTLS message from the PCEP peer (i.e. when the PCEP speaker has sent and received StartTLS message) it is ready to start the negotiation and establishment of TLS and move to steps described in Section 3.4.

The collision resolution procedures described in [RFC5440] for the exchange of Open messages MUST be applied by the PCEP peers during the exchange of StartTLS messages.

The format of a StartTLS message is as follows:

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

The StartTLS message MUST contain only the PCEP common header with Message-Type field set to [TBA1 by IANA].

Once the TCP connection has been successfully established, the PCEP speaker MUST start a timer called StartTLSWait timer, after the expiration of which, if neither StartTLS message has been received, nor a PCErr/Open message (in case of failure and PCEPS not supported by the peer, respectively), it MUST send a PCErr message with Error-Type set to [TBA2 by IANA] and Error-value set to 5 (no StartTLS (nor PCErr/Open) message received before the expiration of the StartTLSWait timer) and it MUST release the TCP connection . A RECOMMENDED value for StartTLSWait timer is 60 seconds. The value of StartTLSWait timer MUST NOT be less than OpenWait timer.

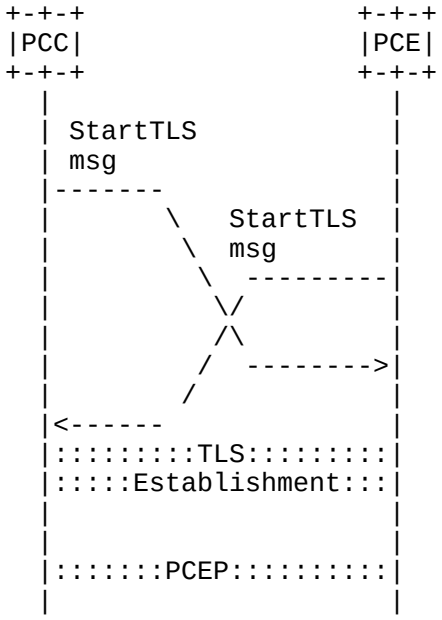


Figure 1: Both PCEP Speaker supports PCEPS (strict)

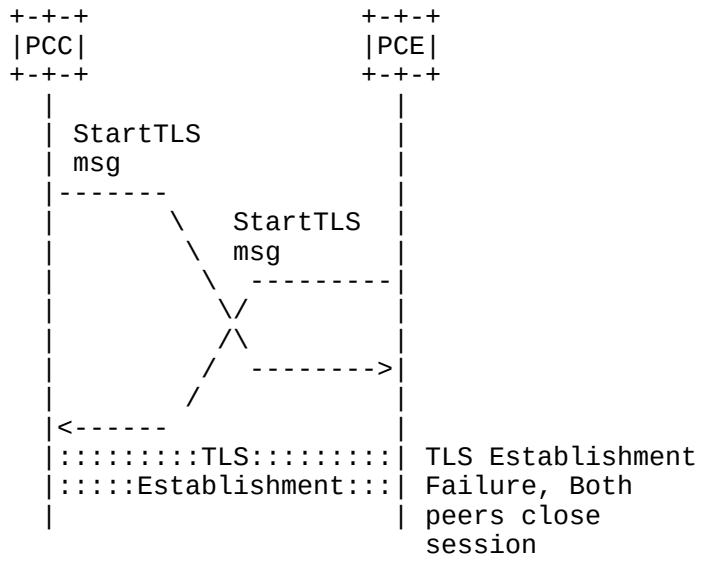


Figure 2: Both PCEP Speaker supports PCEPS (strict), but cannot establish TLS

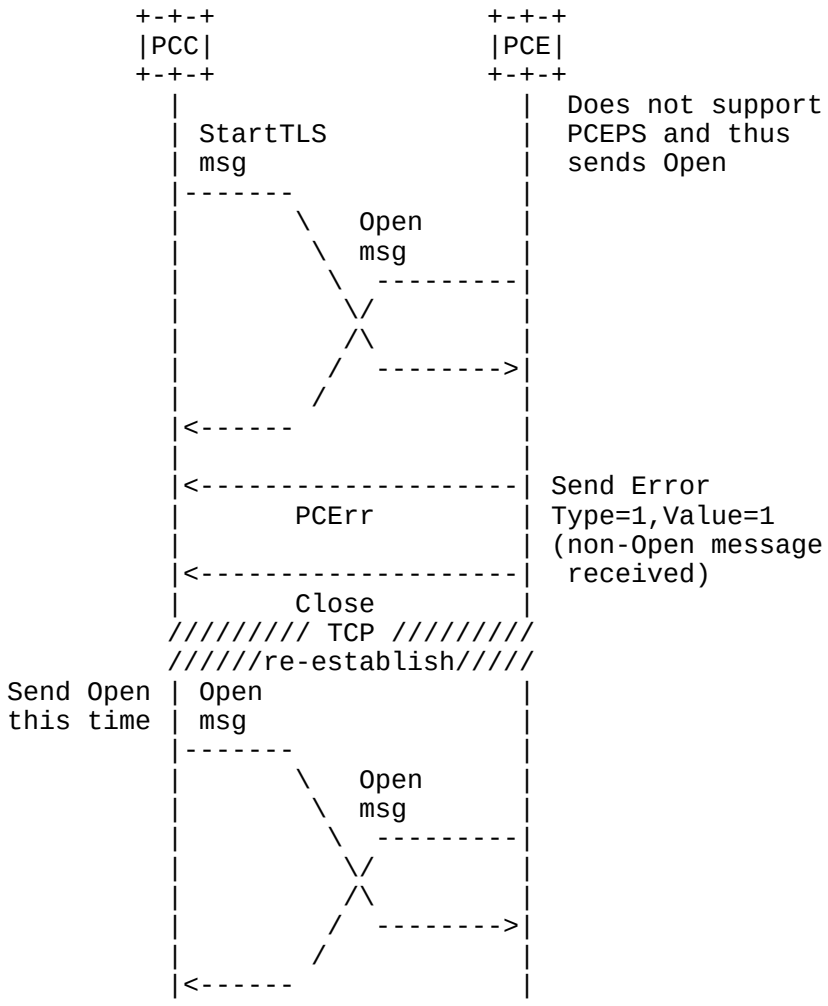


Figure 3: One PCEP Speaker (PCE) does not support PCEPS, while PCC supports both with or without PCEPS

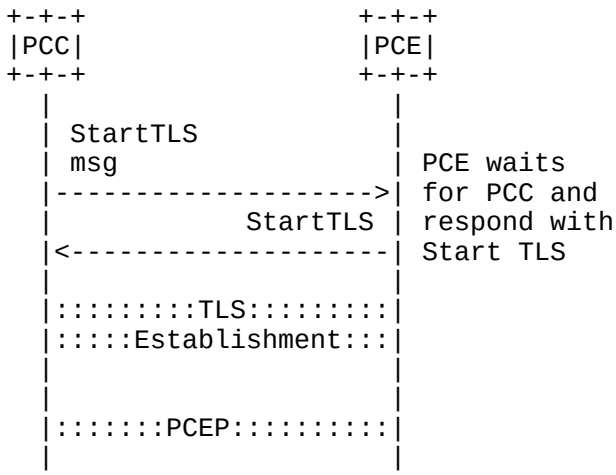


Figure 4: Both PCEP Speaker supports PCEPS as well as without PCEPS

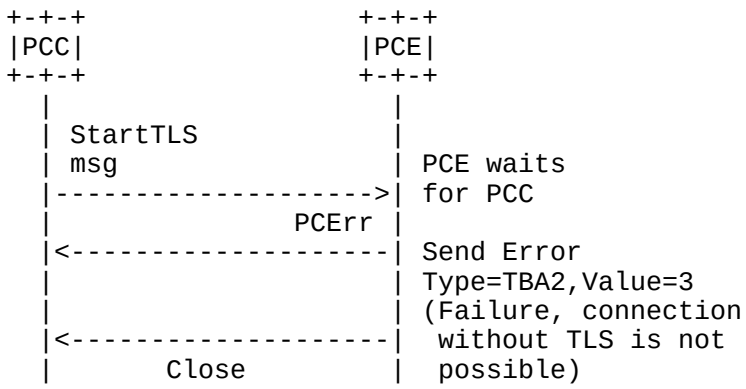


Figure 5: Both PCEP Speaker supports PCEPS as well as without PCEPS, but PCE cannot start TLS negotiation

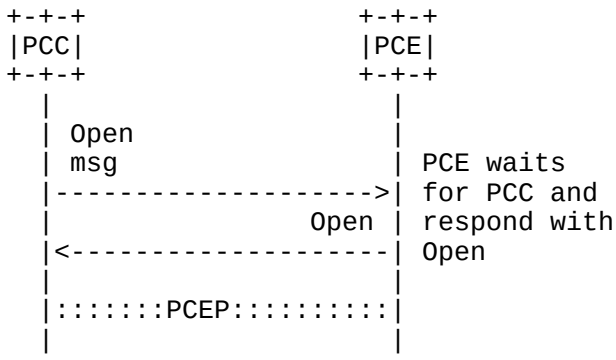


Figure 6: PCE supports PCEPS as well as without PCEPS, while PCC does not support PCEPS

3.4. TLS Connection Establishment

Once the establishment of TLS has been agreed by the PCEP peers, the connection establishment SHALL follow the following steps:

1. Immediately negotiate a TLS session according to [RFC5246]. The following restrictions apply:

   * Support for TLS v1.2 [RFC5246] or later is REQUIRED.

   * Support for certificate-based mutual authentication is REQUIRED.

   * Negotiation of a ciphersuite providing for integrity protection is REQUIRED.

   * Negotiation of a ciphersuite providing for confidentiality is RECOMMENDED.

   * Support for and negotiation of compression is OPTIONAL.

   * PCEPS implementations MUST, at a minimum, support negotiation of the TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 [RFC6460], and SHOULD support TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as well. Implementations SHOULD support the NIST P-256 (secp256r1) curve [RFC4492]. In addition, PCEPS implementations MUST support negotiation of the mandatory-to-implement ciphersuites required by the versions of TLS that they support from TLS 1.3 onwards.

2. Peer authentication can be performed in any of the following two REQUIRED operation models:

* TLS with X.509 certificates using Public-Key Infrastructure Exchange (PKIX) trust models:

  + Implementations MUST allow the configuration of a list of trusted Certification Authorities (CAs) for incoming connections.

  + Certificate validation MUST include the verification rules as per [RFC5280].

  + PCEPS implementations SHOULD incorporate revocation methods (CRL downloading, OCSP...) according to the trusted CA policies.

  + Implementations SHOULD indicate their trusted CAs. For TLS 1.2, this is done using [RFC5246], Section 7.4.4, "certificate_authorities" (server side) and [RFC6066], Section 6 "Trusted CA Indication" (client side).

  + Implementations MUST follow the rules and guidelines for peer validation as defined in [RFC6125]. If an expected DNS name or IP address for the peer is configured, then the implementations MUST check them against the values in the presented certificate. The DNS names and the IP addresses can be contained in the CN-ID [RFC6125] (Common Name Identifier) or the subjectAltName entries. For verification, only one of these entries is considered. The following precedence applies: for DNS name validation, DNS-ID [RFC6125] has precedence over CN-ID; for IP address validation, subjectAltName:iPAddr has precedence over CN-ID.

  + Implementations MAY allow the configuration of a set of additional properties of the certificate to check for a peer’s authorization to communicate (e.g., a set of allowed values in URI-ID [RFC6125] or a set of allowed X509v3 Certificate Policies). The definition of these properties are out of scope of this document.

* TLS with X.509 certificates using certificate fingerprints: Implementations MUST allow the configuration of a list of certificates that are trusted to identify peers, identified via fingerprint of the Distinguished Encoding Rules (DER) encoded certificate octets. Implementations MUST support SHA-256 as defined by [SHS] as the hash algorithm for the fingerprint, but a later revision may demand support for a stronger hash function.

3. Start exchanging PCEP messages.

   * Once the TLS connection has been successfully established, the PCEP speaker MUST start the OpenWait timer [RFC5440], after the expiration of which, if no Open message has been received, it sends a PCErr message and releases the TCP/TLS connection.

3.5. Peer Identity

Depending on the peer authentication method in use, PCEPS supports different operation modes to establish peer's identity and whether it is entitled to perform requests or can be considered authoritative in its replies. PCEPS implementations SHOULD provide mechanisms for associating peer identities with different levels of access and/or authoritativeness, and they MUST provide a mechanism for establishing a default level for properly identified peers. Any connection established with a peer that cannot be properly identified SHALL be terminated before any PCEP exchange takes place.

In TLS-X.509 mode using fingerprints, a peer is uniquely identified by the fingerprint of the presented certificate.

There are numerous trust models in PKIX environments, and it is beyond the scope of this document to define how a particular deployment determines whether a peer is trustworthy. Implementations that want to support a wide variety of trust models should expose as many details of the presented certificate to the administrator as possible so that the trust model can be implemented by the administrator. At least the following parameters of the X.509 certificate SHOULD be exposed:

- Peer's IP address
- Peer's fully qualified domain name (FQDN)
- Certificate Fingerprint
- Issuer
- Subject
- All X509v3 Extended Key Usage
- All X509v3 Subject Alternative Name
- All X509v3 Certificate Policies

Note that the remote IP address used for the TCP session establishment is also exposed.

[I-D.ietf-pce-stateful-sync-optimizations] specify a Speaker Entity Identifier TLV (SPEAKER-ENTITY-ID), as an optional TLV that is included in the OPEN Object. It contains a unique identifier for the node that does not change during the lifetime of the PCEP speaker. An implementation would thus expose the speaker entity identifier as part of the X509v3 certificate's subjectAltName:otherName, so that an implementation could use this identifier for the peer identification trust model.

In addition, a PCC MAY apply the procedures described in [RFC6698] DNS-Based Authentication of Named Entities (DANE) to verify its peer identity when using DNS discovery. See section Section 4.1 for further details.

3.6. Connection Establishment Failure

In case the initial TLS negotiation or the peer identity check fails, according to the procedures listed in this document, both peers MUST immediately close the connection.

The initiator SHOULD follow the procedure listed in [RFC5440] to retry session setup as per the exponential back-off session establishment retry procedure.

4. Discovery Mechanisms

This document does not specify any discovery mechanism for support of PCEPS. Other documents, [I-D.wu-pce-discovery-pceps-support] and [I-D.wu-pce-dns-pce-discovery] have made proposals:

- A PCE can advertise its capability to support PCEPS using the IGP's advertisement mechanism of the PCE discovery information. The PCE-CAP-FLAGS sub-TLV is an optional sub-TLV used to advertise PCE capabilities. It is present within the PCE Discovery (PCED) sub-TLV carried by OSPF or IS-IS. [RFC5088] and [RFC5089] provide the description and processing rules for this sub-TLV when carried within OSPF and IS-IS, respectively. PCE capability bits are defined in [RFC5088]. A new capability flag bit for the PCE-CAP-FLAGS sub-TLV that can be announced as an attribute to distribute PCEP security support information is proposed in [I-D.wu-pce-discovery-pceps-support].

- A PCE can advertise its capability to support PCEPS using the DNS [I-D.wu-pce-dns-pce-discovery] by identifying the support of TLS.

4.1. DANE Applicability

DANE [RFC6698] defines a secure method to associate the certificate that is obtained from a TLS server with a domain name using DNS, i.e., using the TLSA DNS resource record (RR) to associate a TLS server certificate or public key with the domain name where the record is found, thus forming a "TLSA certificate association". The DNS information needs to be protected by DNS Security (DNSSEC). A PCC willing to apply DANE to verify server identity MUST conform to the rules defined in section 4 of [RFC6698]. The implementation MUST support Service certificate constraint (TLSA Certificate Usages type 1) with Matching type 2 (SHA2-256) as described in [RFC6698][RFC7671]. The server’s domain name must be authorized separately, as TLSA does not provide any useful authorization guarantees.

5. Backward Compatibility

The procedures described in this document define a security container for the transport of PCEP requests and replies carried by a TLS connection initiated by means of a specific extended message (StartTLS) that does not interfere with PCEP speaker implementations not supporting it.

A PCC that does not support PCEPS will send Open message as the first message on TCP establishment. A PCE that supports PCEPS only, will send StartTLS message on TCP establishment. On receiving StartTLS message, PCC would consider it as an error and behave according to the existing error mechanism of [RFC5440] and send PCErr message with Error-Type 1 (PCEP session establishment failure) and Error-Value 1 (reception of an invalid Open message or a non Open message) and close the session.

A PCC that support PCEPS will send StartTLS message as the first message on TCP establishment. A PCE that does not supports PCEPS, would consider receiving StartTLS message as an error and respond with PCErr message (with Error-Type 1 and Error-Value 1) and close the session.

If a StartTLS message is received at any other time by a PCEP speaker that does not implement PCEPS, it would consider it as an unknown message and would behave according to the existing error mechanism of [RFC5440] and send PCErr message with Error-Type 2 (Capability not supported) and close the session.

An existing PCEP session cannot be upgraded to PCEPS, the session needs to be terminated and reestablished as per the procedure described in this document. During the incremental upgrade, the PCEP

speaker SHOULD allow session establishment with and without TLS. Once both PCEP speakers are upgraded to support PCEPS, the PCEP session is re-established with TLS, otherwise PCEP session without TLS is setup. A redundant PCE MAY also be used during the incremental deployment to take over the PCE undergoing upgrade. Once the upgrade is completed, support for unsecured version SHOULD be removed.

A PCE that accepts connections with or without PCEPS, it would respond based on the message received from PCC. A PCC that supports connection with or without PCEPS, it would first attempt to connect with PCEPS and in case of error, it MAY retry to establish connection without PCEPS. For successful TLS operations with PCEP, both PCEP peers in the network would need to be upgraded to support this document.

Note that, a PCEP implementation that support PCEPS would respond with PCErr message with Error-Type set to [TBA2 by IANA] (PCEP StartTLS failure) and Error-value set to 2 if any other message is sent before StartTLS or Open. If the sender of the invalid message is a PCEP implementation that does not support PCEPS, it will not be able to understand this error. A PCEPS implementation could also send the PCErr message as per [RFC5440] with Error-Type "PCEP session establishment failure" and Error-value "reception of an invalid Open message or a non Open message" before closing the session.

6. IANA Considerations

6.1. New PCEP Message

IANA is requested to allocate new message types within the "PCEP Messages" sub-registry of the PCEP Numbers registry, as follows:

Value	Description	Reference
TBA1	The Start TLS Message (StartTLS)	This document

6.2. New Error-Values

IANA is requested to allocate new Error Types and Error Values within the " PCEP-ERROR Object Error Types and Values" sub-registry of the PCEP Numbers registry, as follows:

Error-Type	Meaning	Error-value	Reference
TBA2	PCEP StartTLS failure	0:Unassigned	This document
		1:Reception of StartTLS after any PCEP exchange	This document
		2:Reception of any other message apart from StartTLS, Open or PCErr	This document
		3:Failure, connection without TLS is not possible	This document
		4:Failure, connection without TLS is possible	This document
		5:No StartTLS message (nor PCErr/Open) before StartTLSWait timer expiry	This document

7. Security Considerations

While the application of TLS satisfies the requirement on confidentiality as well as fine-grained, policy-based peer authentication, there are security threats that it cannot address. It may be advisable to apply additional protection measures, in particular in what relates to attacks specifically addressed to forging the TCP connection underpinning TLS, especially in the case of long-lived connections. One of these measures is the application of TCP-AO (TCP Authentication Option [RFC5925]), which is fully compatible with and deemed as complementary to TLS. The mechanisms to configure the requirements to use TCP-AO and other lower-layer protection measures with a particular peer are outside the scope of this document.

Since computational resources required by TLS handshake and ciphersuite are higher than unencrypted TCP, clients connecting to a PCEPS server can more easily create high load conditions and a malicious client might create a Denial-of-Service attack more easily.

Some TLS ciphersuites only provide integrity validation of their payload, and provide no encryption, such ciphersuites SHOULD NOT be used by default. Administrators MAY allow the usage of these ciphersuites after careful weighting of the risk of relevant internal data leakage, that can occur in such a case, as explicitly stated by [RFC6952].

When using certificate fingerprints to identify PCEPS peers, any two certificates that produce the same hash value will be considered the same peer. Therefore, it is important to make sure that the hash function used is cryptographically uncompromised, so that attackers are very unlikely to be able to produce a hash collision with a certificate of their choice. This document mandates support for SHA-256 as defined by [SHS], but a later revision may demand support for stronger functions if suitable attacks on it are known.

PCEPS implementations that continue to accept connections without TLS are susceptible to downgrade attacks as described in [RFC7457]. An attacker could attempt to remove the use of StartTLS message that request the use of TLS as it pass on the wire in clear, and further inject a PCErr message that suggest to attempt PCEP connection without TLS.

The guidance given in [RFC7525] SHOULD be followed to avoid attacks on TLS.

8. Manageability Considerations

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

8.1. Control of Function and Policy

A PCE or PCC implementation SHOULD allow configuring the PCEP security via TLS capabilities as described in this document.

A PCE or PCC implementation supporting PCEP security via TLS MUST support general TLS configuration as per [RFC5246]. At least the configuration of one of the trust models and its corresponding parameters, as described in Section 3.4 and Section 3.5, MUST be supported by the implementation.

A PCEP implementation SHOULD allow configuring the StartTLSWait timer value.

PCEPS implementations MAY provide an option to allow the operator to manually override strict TLS configuration and allow unsecure connections. Execution of this override SHOULD trigger a warning about the security implications of permitting unsecure connections.

Further, the operator needs to develop suitable security policies around PCEP within his network. The PCEP peers SHOULD provide ways

for the operator to complete the following tasks in regards to a PCEP session:

- Determine if a session is protected via PCEPS.
- Determine the version of TLS, the mechanism used for authentication, and the ciphersuite in use.
- Determine if the certificate could not be verified, and the reason for this circumstance.
- Inspect the certificate offered by the PCEP peer.
- Be warned if StartTLS procedure fails for the PCEP peers, that are known to support PCEPS via configurations or capability advertisements.

8.2. Information and Data Models

The PCEP MIB module is defined in [RFC7420]. The MIB module could be extended to include the ability to view the PCEPS capability, TLS related information as well as TLS status for each PCEP peer.

Further, to allow the operator to configure the PCEPS capability and various TLS related parameters as well as to view the current TLS status for a PCEP session, the PCEP YANG module [I-D.ietf-pce-pcep-yang] is extended to include TLS related information.

8.3. Liveness Detection and Monitoring

Mechanisms defined in this document do not imply any new liveness detection and monitoring requirements in addition to those already listed in [RFC5440] and [RFC5246].

8.4. Verifying Correct Operations

A PCEPS implementation SHOULD log error events and provide PCEPS failure statistics with reasons.

8.5. Requirements on Other Protocols

Mechanisms defined in this document do not imply any new requirements on other protocols. Note that, Section 4 list possible discovery mechanism for support of PCEPS.

8.6. Impact on Network Operation

Mechanisms defined in this document do not have any significant impact on network operations in addition to those already listed in [RFC5440], and the policy and management implications discussed above.

9. Acknowledgements

This specification relies on the analysis and profiling of TLS included in [RFC6614] and the procedures described for the STARTTLS command in [RFC4513].

We would like to thank Joe Touch for his suggestions and support regarding the StartTLS mechanisms.

Thanks to Daniel King for reminding the authors about manageability considerations.

Thanks to Cyril Margaria for shepherding this document.

Thanks to David Mandelberg for early SECDIR review comments as well as re-reviewing during IETF last call.

Thanks to Dan Frost for the RTGDIR review and comments.

Thanks to Dale Worley for the Gen-ART review and comments.

Also thanks to Tianran Zhou for OPSDIR review.

Thanks to Deborah Brungard for being the responsible AD and guiding the authors as needed.

Thanks to Mirja Kuhlewind, Eric Rescorla, Warren Kumari, Kathleen Moriarty, Suresh Krishnan, Ben Campbell and Alexey Melnikov for the IESG review and comments.

10. References

10.1. Normative References

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

[RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, DOI 10.17487/RFC5246, August 2008, <https://www.rfc-editor.org/info/rfc5246>.

[RFC5280] Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., and W. Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 5280, DOI 10.17487/RFC5280, May 2008, <https://www.rfc-editor.org/info/rfc5280>.

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

[RFC6066] Eastlake 3rd, D., "Transport Layer Security (TLS) Extensions: Extension Definitions", RFC 6066, DOI 10.17487/RFC6066, January 2011, <https://www.rfc-editor.org/info/rfc6066>.

[RFC6125] Saint-Andre, P. and J. Hodges, "Representation and Verification of Domain-Based Application Service Identity within Internet Public Key Infrastructure Using X.509 (PKIX) Certificates in the Context of Transport Layer Security (TLS)", RFC 6125, DOI 10.17487/RFC6125, March 2011, <https://www.rfc-editor.org/info/rfc6125>.

[RFC6698] Hoffman, P. and J. Schlyter, "The DNS-Based Authentication of Named Entities (DANE) Transport Layer Security (TLS) Protocol: TLSA", RFC 6698, DOI 10.17487/RFC6698, August 2012, <https://www.rfc-editor.org/info/rfc6698>.

[RFC7525] Sheffer, Y., Holz, R., and P. Saint-Andre, "Recommendations for Secure Use of Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS)", BCP 195, RFC 7525, DOI 10.17487/RFC7525, May 2015, <https://www.rfc-editor.org/info/rfc7525>.

[RFC7671] Dukhovni, V. and W. Hardaker, "The DNS-Based Authentication of Named Entities (DANE) Protocol: Updates and Operational Guidance", RFC 7671, DOI 10.17487/RFC7671, October 2015, <https://www.rfc-editor.org/info/rfc7671>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[SHS] National Institute of Standards and Technology, "Secure Hash Standard (SHS), FIPS PUB 180-4", DOI 10.6028/NIST.FIPS.180-4, August 2015, <http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.180-4.pdf>.

10.2. Informative References

[RFC4492] Blake-Wilson, S., Bolyard, N., Gupta, V., Hawk, C., and B. Moeller, "Elliptic Curve Cryptography (ECC) Cipher Suites for Transport Layer Security (TLS)", RFC 4492, DOI 10.17487/RFC4492, May 2006, <https://www.rfc-editor.org/info/rfc4492>.

[RFC4513] Harrison, R., Ed., "Lightweight Directory Access Protocol (LDAP): Authentication Methods and Security Mechanisms", RFC 4513, DOI 10.17487/RFC4513, June 2006, <https://www.rfc-editor.org/info/rfc4513>.

[RFC5088] Le Roux, JL., Ed., Vasseur, JP., Ed., Ikejiri, Y., and R. Zhang, "OSPF Protocol Extensions for Path Computation Element (PCE) Discovery", RFC 5088, DOI 10.17487/RFC5088, January 2008, <https://www.rfc-editor.org/info/rfc5088>.

[RFC5089] Le Roux, JL., Ed., Vasseur, JP., Ed., Ikejiri, Y., and R. Zhang, "IS-IS Protocol Extensions for Path Computation Element (PCE) Discovery", RFC 5089, DOI 10.17487/RFC5089, January 2008, <https://www.rfc-editor.org/info/rfc5089>.

[RFC5925] Touch, J., Mankin, A., and R. Bonica, "The TCP Authentication Option", RFC 5925, DOI 10.17487/RFC5925, June 2010, <https://www.rfc-editor.org/info/rfc5925>.

[RFC6460] Salter, M. and R. Housley, "Suite B Profile for Transport Layer Security (TLS)", RFC 6460, DOI 10.17487/RFC6460, January 2012, <https://www.rfc-editor.org/info/rfc6460>.

[RFC6614] Winter, S., McCauley, M., Venaas, S., and K. Wierenga, "Transport Layer Security (TLS) Encryption for RADIUS", RFC 6614, DOI 10.17487/RFC6614, May 2012, <https://www.rfc-editor.org/info/rfc6614>.

[RFC6952] Jethanandani, M., Patel, K., and L. Zheng, "Analysis of BGP, LDP, PCEP, and MSDP Issues According to the Keying and Authentication for Routing Protocols (KARP) Design Guide", RFC 6952, DOI 10.17487/RFC6952, May 2013, <https://www.rfc-editor.org/info/rfc6952>.

[RFC7420] Koushik, A., Stephan, E., Zhao, Q., King, D., and J. Hardwick, "Path Computation Element Communication Protocol (PCEP) Management Information Base (MIB) Module", RFC 7420, DOI 10.17487/RFC7420, December 2014, <https://www.rfc-editor.org/info/rfc7420>.

[RFC7457] Sheffer, Y., Holz, R., and P. Saint-Andre, "Summarizing Known Attacks on Transport Layer Security (TLS) and Datagram TLS (DTLS)", RFC 7457, DOI 10.17487/RFC7457, February 2015, <https://www.rfc-editor.org/info/rfc7457>.

[I-D.ietf-pce-stateful-sync-optimizations] Crabbe, E., Minei, I., Medved, J., Varga, R., Zhang, X., and D. Dhody, "Optimizations of Label Switched Path State Synchronization Procedures for a Stateful PCE", draft-ietf-pce-stateful-sync-optimizations-10 (work in progress), March 2017.

[I-D.ietf-pce-pcep-yang] Dhody, D., Hardwick, J., Beeram, V., and j. jefftant@gmail.com, "A YANG Data Model for Path Computation Element Communications Protocol (PCEP)", draft-ietf-pce-pcep-yang-05 (work in progress), June 2017.

[I-D.wu-pce-dns-pce-discovery] Wu, Q., Dhody, D., King, D., Lopez, D., and J. Tantsura, "Path Computation Element (PCE) Discovery using Domain Name System(DNS)", draft-wu-pce-dns-pce-discovery-10 (work in progress), March 2017.

[I-D.wu-pce-discovery-pceps-support] Lopez, D., Wu, Q., Dhody, D., and D. King, "IGP extension for PCEP security capability support in the PCE discovery", draft-wu-pce-discovery-pceps-support-07 (work in progress), March 2017.

Authors' Addresses

Diego R. Lopez
Telefonica I+D
Don Ramon de la Cruz, 82
Madrid 28006
Spain

Phone: +34 913 129 041
EMail: diego.r.lopez@telefonica.com

Oscar Gonzalez de Dios
Telefonica I+D
Don Ramon de la Cruz, 82
Madrid 28006
Spain

Phone: +34 913 129 041
EMail: oscar.gonzalezdedios@telefonica.com

Qin Wu
Huawei
101 Software Avenue, Yuhua District
Nanjing, Jiangsu 210012
China

EMail: sunseawq@huawei.com

Dhruv Dhody
Huawei
Divyashree Techno Park, Whitefield
Bangalore, KA 560066
India

EMail: dhruv.ietf@gmail.com

PCE Working Group
Internet Draft
Intended Status: Informational

Y. Lee
Huawei

H. Zheng
Huawei

October 24, 2014

Expires: January 2015

PCE in Support of Transporting Traffic Engineering Data

draft-lee-pce-transporting-te-data-01.txt

Abstract

In order to compute and provide optimal paths, Path Computation Elements (PCEs) require an accurate and timely Traffic Engineering Database (TED). Traditionally this TED has been obtained from a link state routing protocol supporting traffic engineering extensions. This document discusses possible alternatives and enhancements to the existing approach to TED creation. This document gives architectural alternatives for these alternatives and their potential impacts on network nodes, routing protocols, and PCEP.

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 April 24, 2015.

Table of Contents

1. Introduction.....2
1.1. TED Creation and Maintenance via IGP-TEs.....5
2. Alternative TED Creation & Maintenance for a PCE.....6
2.1. Architecture Options.....8
2.1.1. Nodes Send TE Info to all PCEs.....12
2.1.2. Nodes Send TE Info via an Intermediate System.....12
2.1.3. Nodes Send TE Info to At Least One PCE.....13
2.2. Nodes Finding PCEs.....13
2.3. Node TE Information Update Procedures.....14
2.4. PCE TED Maintenance Procedures.....14
3. Standardization and Protocol Considerations.....15
3.1. Architecture Specific Standardization Aspects.....16
4. Security Considerations.....16
5. IANA Considerations.....17
6. Conclusions.....17
7. Acknowledgments.....17
8. References.....17
8.1. Normative References.....17
8.2. Informative References.....18
Author's Addresses.....19
Disclaimer of Validity.....20

1. Introduction

In Multiprotocol Label Switching (MPLS) and Generalized MPLS (GMPLS), a Traffic Engineering Database (TED) is used in computing paths for connection oriented packet services and for circuits. The TED contains all relevant information that a Path Computation

Element (PCE) needs to perform its computations. It is important that the TED should be complete and accurate anytime so that the PCE can perform path computations.

In MPLS and GMPLS networks, Interior Gateway routing Protocols (IGPs) have been used to create and maintain a copy of the TED at each node. One of the benefits of the PCE architecture [RFC4655] is the use of computationally more sophisticated path computation algorithms and the realization that these may need enhanced processing power not necessarily available at each node participating in an IGP.

Section 4.3 of [RFC4655] describes the potential load of the TED on a network node and proposes an architecture where the TED is maintained by the PCE rather than the network nodes. However it did not describe how a PCE would obtain the information needed to populate its TED. PCE may construct its TED by participating in the IGP ([RFC3630] and [RFC5305] for MPLS-TE; [RFC4203] and [RFC5307] for GMPLS). An alternative is offered by BGP-LS [I-D.ietf-idr-ls-distribution].

In this document we propose approaches for creating and maintaining the TED directly on a PCE as an alternative to IGPs and BGP transport and investigate on the impact from the PCE, routing protocol, and node perspective.

There are two main applicability of this alternative proposed by this draft:

- Where there is no IGP-TE or BGP-LS running at the PCE to learn TED.
- Where there is IGP-TE or BGP-LS running but with a need for a faster TED population and convergence at the PCE.
    - A PCE may receive partial information (say basic TE) from IGP-TE and other information (optical and impairment) from PCEP.
    - A PCE may receive full information from both IGP-TE and PCEP.

A PCC may further choose to send only local TE information or both local and remote learned TED information. How a PCE manages the TED

information is implementation specific and thus out of scope of this document. PCEP extensions to support this idea is pursued in a separate draft [PCEP-TE].

New application areas for GMPLS and PCE in optical transport networks include Wavelength Switched Optical Networking (WSON) and Optical Transport Networks (OTN). WSON scenarios can be divided into routing wavelength assignment (RWA) problems where a PCE requires detailed information about switching node asymmetries and wavelength constraints as well as detailed up to date information on wavelength usage per link [WSON-Frame]. As more data is anticipated to be made available to PCE with addition of OTN [Reference] and Flex-grid [Reference] and possible with some optical impairment data [WSON-IMP-Info] even with the minimum set specified in [G.680], the total amount of data requires significantly more information to be held in the TED than is required for other traffic engineered networks. Related to this issue published by [HWANG] indicated that long convergence time and large number of LSAs flooded in the network might cause scalability problems in OSPF-TE and impose limitations on OSPF-TE applications.

In some circumstances such additional information could "bog down" the routing protocols on the nodes from a data processing, a storage, or communications perspective. In environments where PCEs are external to the nodes running the routing protocol, and where the information in the TED is not used by the switching nodes it makes sense to investigate alternative methods to create and maintain the TED at its place of use, i.e., the PCE.

Recent development of a stateful PCE Model [PCE-Initiated] changes the PCE operation from path computation alone to include the support of PCE-initiated LSPs. With a stateful PCE model, it is also noted that LSP-DB is maintained by the PCE. For LSP state synchronization of stateful PCEs in GMPLS networks, the LSP attributes, such as its bandwidth, associated route as well as protection information etc, should be updated by PCCs to PCE LSP database (LSP-DB) [S-PCE-GMPLS]. To support all these recent changes in a stateful PCE model, a direct PCE interface to each PCC has to be supported. Relevant TED information can also be transported from each node to PCE using this PCC-PCE interface. Any resource changes in the node and links can also be quickly updated to PCE using this interface. Convergence time of IGP in GMPLS networks may not be quick enough to support on-line dynamic connectivity required for some applications.

This draft does not advocate that the alternative methods specified in this draft should completely replace the IGP-TE as the method of creating the TED. The split between the data to be distributed via an IGP and the information conveyed via one of the alternatives in this document depends on the nature of the network situation. One could potentially choose to have some traffic engineering information distributed via an IGP while other more specialized traffic information is only conveyed to the PCEs via an alternative interface discussed here. In addition, the methods specified in this draft is only relevant to a set of architecture options where routing decisions are wholly or partially made in the PCE.

However, the networks that do not support IGP-TE/BGP-LS, the method proposed by this draft may be very relevant.

1.1. TED Creation and Maintenance via IGP-TEs

Routing protocols, in particular, IGP-TEs such as Open Shortest Path First (OSPF) and Intermediate system to intermediate system (IS-IS), take on a number of roles with respect to the control and data planes for IP, MPLS, and GMPLS. In all three technology families the underlying control plane communications technology is IP and hence all utilize the IGPs ability to control and run the IP data plane.

For the IP layer, the IGP directly establishes data plane connectivity. In the MPLS and GMPLS cases separate signaling protocols are used to directly control the data plane connectivity and in these cases the prime purpose of the routing protocol is to furnish network topology and resource status information used by path computation algorithms on the nodes or PCEs. Hence in the IP case the IGP is directly service impacting, while in the MPLS/GMPLS case it is only indirectly service impacting.

The IP layer information and the MPLS/GMPLS data plane layer information may be kept by the IGPs in two different information stores. These are referred to as databases but are not necessarily relational databases. In OSPF the information directly related to IP connectivity (and hence the control communications plane for all three technologies) and non-IP advertisements are kept in the link state database (LSDB), while information related to traffic engineering used by MPLS and GMPLS is kept in a (conceptually) separate TED which can be considered a subset of the LSDB. This TED information is distributed in a different data structure (Opaque LSA [RFC5250]). When we talk about adding additional technology-specific GMPLS information used for path computation we are only talking about adding to the TED and not the IP portion of the LSDB.

There are three main functions performed by an IGP: (a) hello protocol, (b) database synchronization (with neighbors), (c) database updates.

Data Plane Technologies	Hello Protocol	Database Sync & Updates
IP	Establish Control & Data Plane Adjacencies	LSDB
MPLS	Establish Control & Data Plane Adjacencies	LSDB & TED
GMPLS	Establish Control Plane Adjacencies (only)	LSDB & TED

Table 1 Main Functions of an IGP for various technologies

The procedures for maintaining LSDBs and TEDs in IGP-TEs have been very successful and well proven over time. These consist of:

1. Ageing the individual pieces of information in the TED (including discarding them when the information gets too old) to remove stale information from the TED.

2. Originator of the information being required to periodically resend TED information to prevent it from being discarded.

3. Originator of the information sending updates of information as needed, but subject to limits on how many/often these can be sent to keep the TED up-to-date, but to avoid swamping the network.

4. Reliable method for getting this information to other peers (flooding) to ensure that the information is delivered to all participants.

5. An efficient database synchronization mechanism for sharing info with a newly established peer.

2. Alternative TED Creation & Maintenance for a PCE

Given that nodes, by their position and role in the network, have accurate traffic engineering information concerning their local link ends and switching properties, it seems natural that, if other nodes in the network cannot make use of this information or do not want

it, the information should only be conveyed to interested PCEs. In such case the flooding of TE information to all nodes may not be very efficient in terms of memory, CPU, bandwidth, etc.

In addition, one could potentially choose to have some traffic engineering information distributed via an IGP-TE protocol while other more specialized traffic information is only conveyed to the PCEs. For example, it makes sense to distribute "static" (rarely modified) and sizable data (e.g., NE switching asymmetry structure) via methods other than IGP-TE while more frequently changed data via IGP-TE. This could significantly decrease the IGP-TE information and its footprint on all nodes.

The benefits of such an approach include:

- Node: reduced storage demands (doesn’t keep the entire TED)
- Node: reduced processing demands for TED updates and synchronization
- Control Plane: reduced overall communication demands since the TED is not being updated and maintained on all nodes in the network.
- PCE: More timely TED updates are possible.
- Information distribution constraints, such as seen in [Imp-Frame] can be met.

To quantify the previous advantages requires a bit more detail on how such an approach could actually be accomplished. The key pieces needed to implement such an approach include:

- Multiple PCEs must be supported for robustness and load sharing.
- Nodes must be able to find a PCE to which to send their traffic engineering information.
- Nodes must have procedures and a mechanism (protocols) with which to communicate their TE information to a PCE. PCEs must have procedures and a mechanism (protocols) with which to receive this TE information from nodes.
- Efficient mechanisms must exist in the multi-PCE case to ensure all PCEs have the same TED.

The advantages of using an alternative to IGP-TE comes at the cost of:

- Additional protocols to be configured and secured. Recall that we still must have an IP IGP for control plane communications.

- Any new protocols/implementations for alternative TED creation still must support many IGP-TE like features such as removal of stale information, reliable delivery of updates to all participants, recovery after reboots/crashes/upgrades, etc. It should also work along with IGP-TE/BGP-LS TED mechanism with some information in the TED received from existing mechanisms.

- Mechanisms to discover PCEs that are capable and willing to accept direct TED updates.

2.1. Architecture Options

There are three general architectural alternatives based on how nodes get their local TED information to the PCEs: (1) Nodes send local information to all PCEs; (2) Nodes send local information to an intermediate server that will send to all PCEs; (3) Nodes send local information to at least one PCE and have the PCEs share this information with each other. An important functionality that needs to be addressed in each of these approaches is how a new PCE gets initialized in a reasonably timely fashion.

Figures 1-3 show examples of three options for nodes to share local TED information with multiple PCEs. As in the IGP case we assume that switching nodes know their local properties and state including the state of all their local links. In these figures the data plane links are shown with the character "o"; TE information flow from nodes to PCE by the characters "|", "-", "/", or "\"; and PCE to PCE TE information, if any, by the character "i".

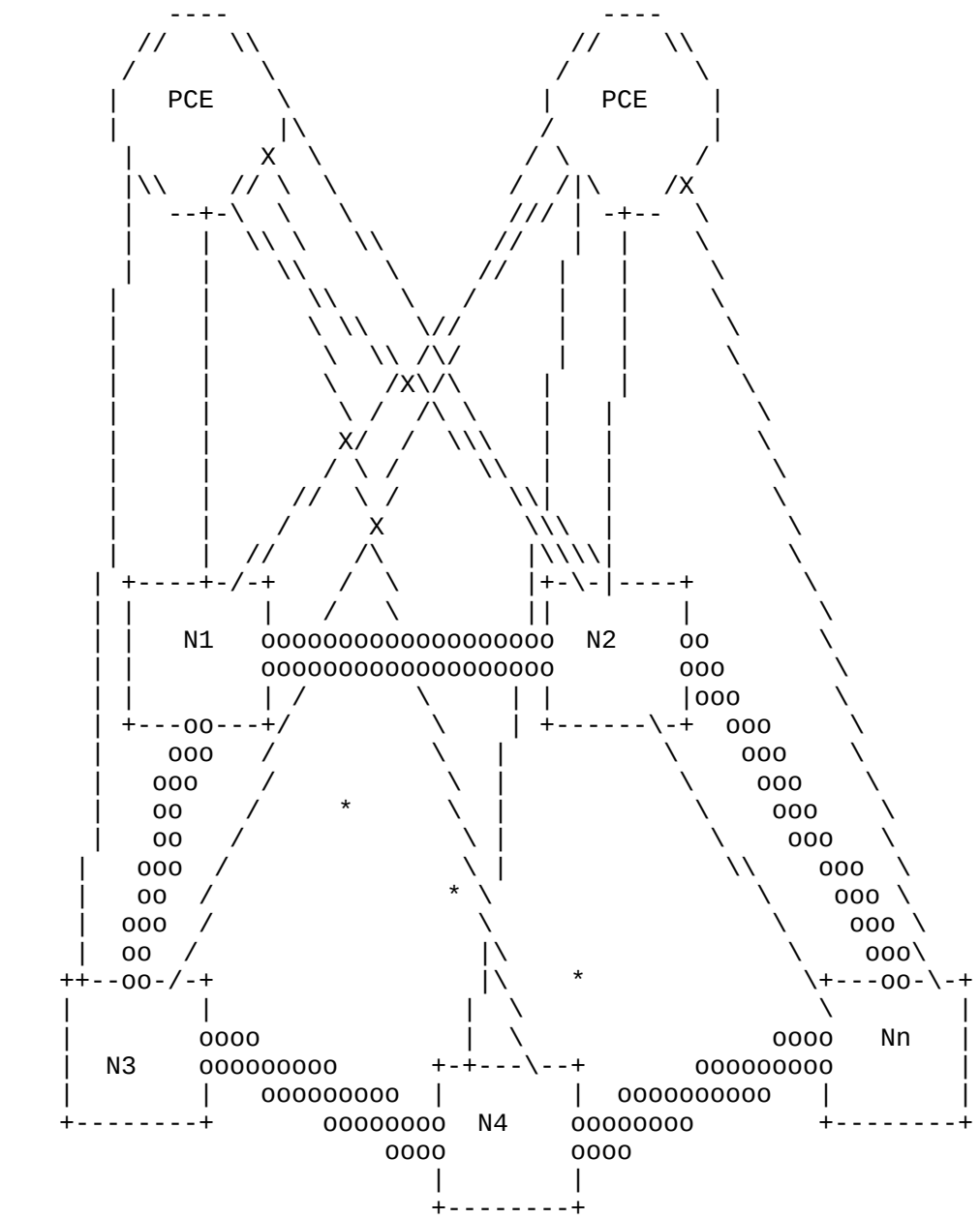


Figure 1 . Nodes send local TE information directly to all PCEs

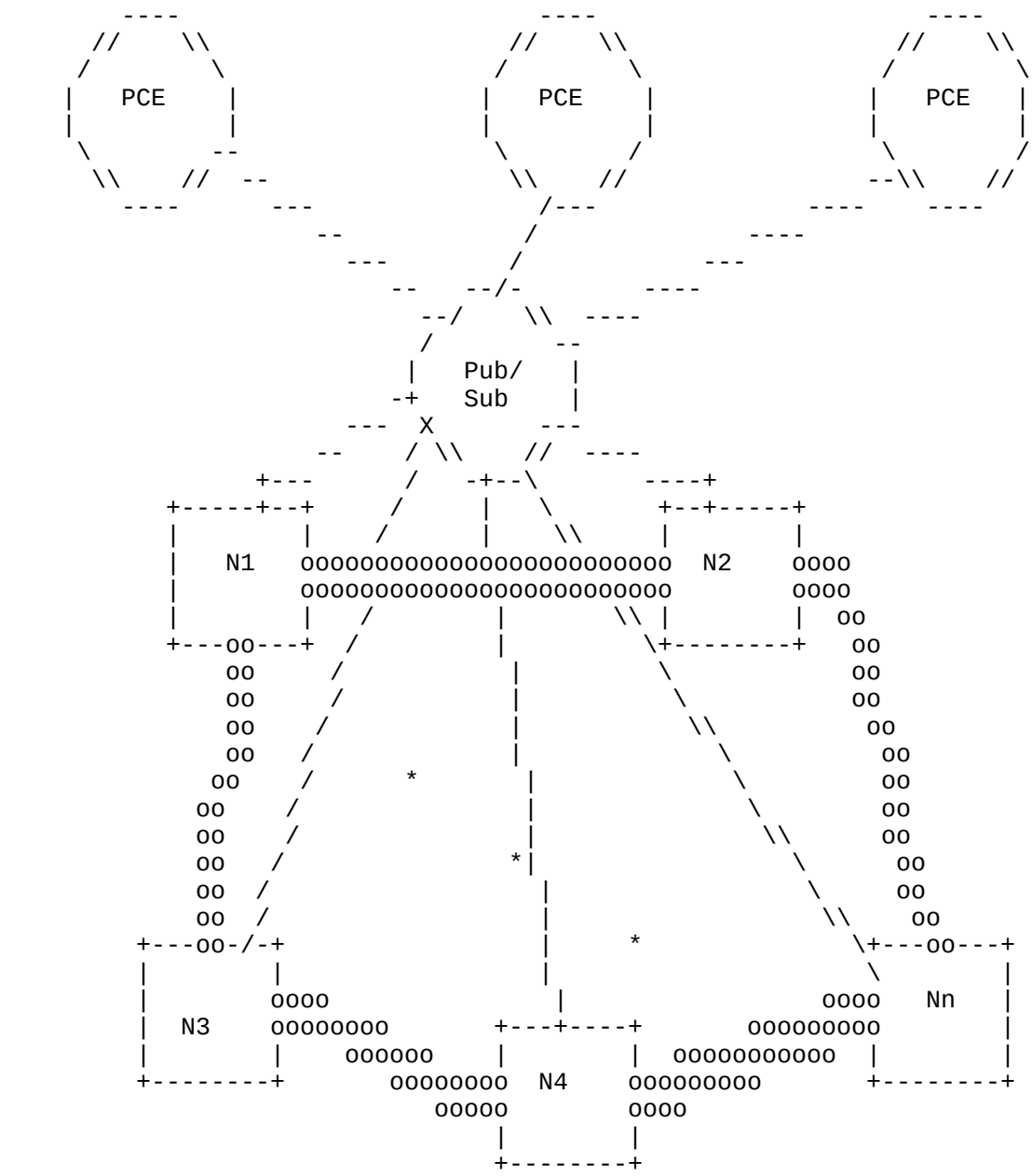


Figure 2 . Nodes send local TE information to PCEs via an intermediary (publish/subscribe)server

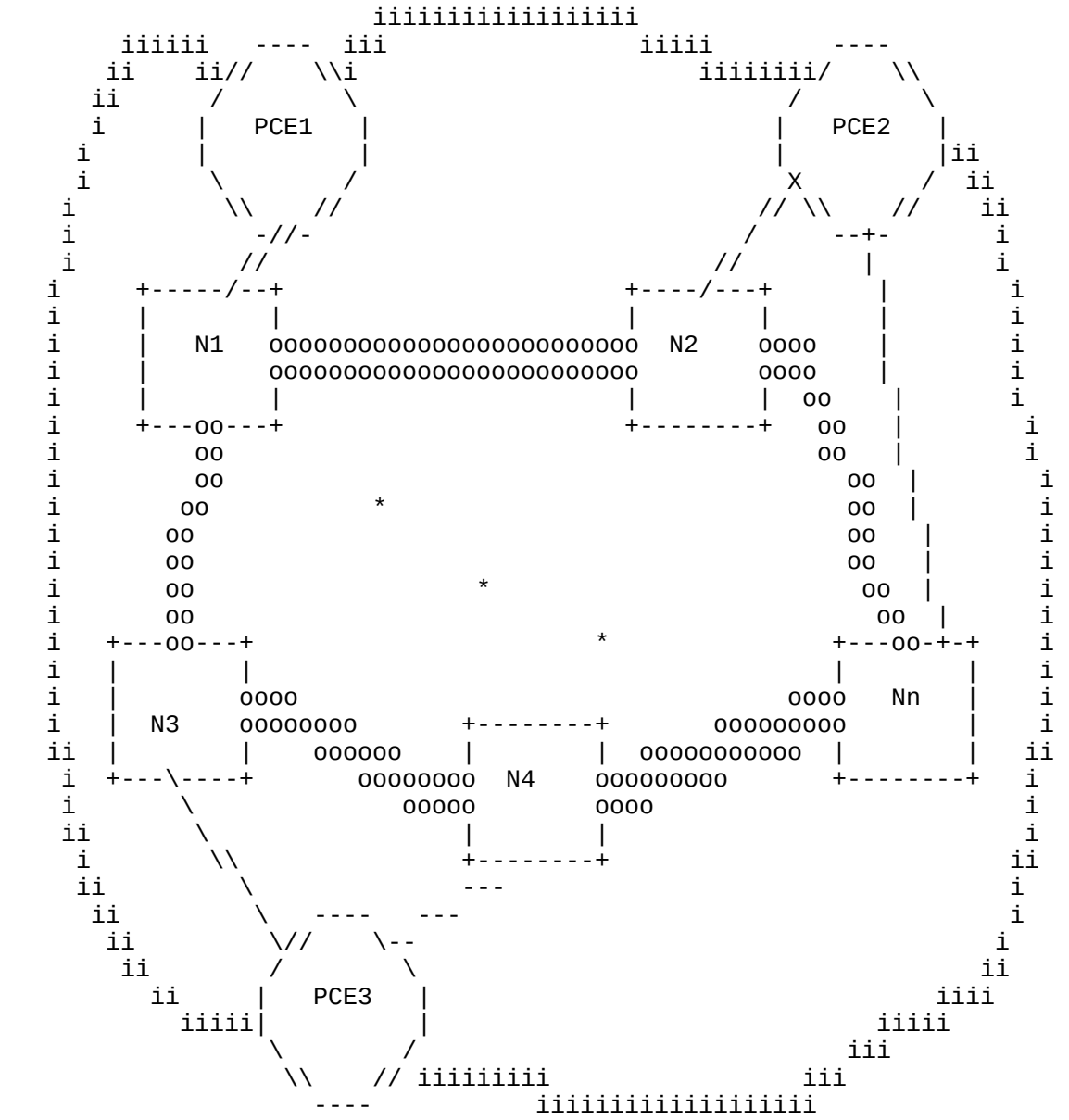


Figure 3 . Nodes send local TE information to at least one PCE and have the PCEs share TED information

2.1.1. Nodes Send TE Info to all PCEs

Architectural alternative 1 shown in Figure 1, illustrates nodes sending their local TE information to all PCEs within their domain. As the number of PCEs grows we have scalability concerns. However, if we are only talking about 2-3 PCEs, then we do not have this scalability concern. In particular each node needs to keep track of which PCE it has sent information to and update that information periodically.

If a new PCE is added to the domain the node must send all its local TED information to that PCE rather than just sending status updates.

2.1.2. Nodes Send TE Info via an Intermediate System

Architecture alternative 2 is shown in Figure 2. This architecture reduces the burden on switching nodes by having the nodes send TE information to an intermediate system. This general approach is typically described in the software literature as a publish/subscribe paradigm. Here the nodes send their local TED information to an intermediate entity whose job is to insure that all PCEs receive this information. The nodes in this case being the publishers of the information and the PCEs the subscribers of the information. Publish/subscribe functionality can be found in general messaging oriented middleware such as the Java Messaging Service [JMS] and many others. A routing specific example of this approach is seen in BGP route reflectors [RFC4456].

Note that the publish/subscribe entity can be collocated with a PCE. This would then looks like a master/slave type system architecture.

If a new PCE is added then the intermediate server will need to work with this new PCE to initialize its TED. Hence the publish/subscribe entity will need to also keep a copy of the entire TED and for reliability purposes a redundant server would be required. The publish/subscribe entity itself can be a PCE.

Architecture alternative 2 could be useful when there are a number of PCEs in the network and as such there is the scaling issue with each of the NEs talking to all the PCEs. The advantage of this alternative would diminish when we are dealing only with only a few PCEs.

2.1.3. Nodes Send TE Info to At Least One PCE

In this architectural alternative, shown in Figure 3, each node would be associated with at least one PCE. This implies that each PCE will only have partial TED information directly from the nodes. It would be the responsibility of a node to get its local TED information to its associated PCE, then the PCEs within a domain would then need to share the partial TED information they learned from their associated nodes with each other so that they can create and maintain the complete TED. As we have seen in section 1.1. this is very similar to part of the functionality provided by a link state protocol, but in this case the protocol would be used between PCEs so that they can share the information they have obtained from their associated switching nodes (rather than from attached links as in a regular link state protocol). To allow for this sharing of information PCEs would need to peer with each other. PCE discovery extensions [RFC4674] could be used to allow PCEs to find other PCEs. If a new PCE is added to the domain it would need to peer with at least one other PCE and then link state protocol procedures for TED synchronization could then be used to initialize the new PCEs TED.

A number of approaches can be used to ensure control plane resilience in this architecture. (1) Each node can be configured with a primary and a secondary PCE to send its information to; In case of failure of communications with the primary PCE the node would send its information to a secondary PCE (warm standby). (2) Each node could be configured to send its information to two different PCEs (hot standby).

2.2. Nodes Finding PCEs

In cases 1 and 3 nodes need to send TE information directly to PCEs. Path Computation Clients (PCCs) and network nodes participating in an IGP (with or without TE extensions) have a mechanism to discover a PCE and its capabilities. [RFC4674] outlines the general requirements for this mechanism and extensions have been defined to provide information so that PCCs can obtain key details about available PCEs in OSPF [RFC5088] and in IS-IS [RFC5089].

After finding candidate PCEs, a node would need to see which if any of the PCEs actually want to receive TE information directly from this node.

In architectural alternative 2 (publish/subscribe) the location of intermediate system would either need to be configured or PCE discovery could be extended so that a when a node asks a PCE if it

wants to hear TE info the PCE points it to the intermediate publish/subscribe system.

2.3. Node TE Information Update Procedures

First a node must establish an association between itself and a PCE or intermediate system that will be maintaining a TED. It is the responsibility of the node to share TE information concerning its local environment, e.g., links and node properties. General and technology specific information models would specify the content of this information while the specific protocols would determine the format. Note that a node would not be sending to the PCE information it might be passed from neighbor nodes. Note that data plane neighbor information would be passed to the PCE embedded in TE link information.

There will be cases where the node would have to send to the PCE only a subset of TE link information depending on the path computation option. For instance, if the node is responsible for routing while the PCE is responsible for wavelength assignment for the route, the node would only need to send the PCE the WSON link usage information. This path computation option is referred to as separate Fouting (R) and Wavelength Assignment (WA) option in [PCE-WSON].

2.4. PCE TED Maintenance Procedures

The PCE is responsible for creating and maintaining the TED that it will use. Key functions include:

1. Establishing and authenticating communications between the PCE and sources of TED information.

2. Timely updates of the TED with information received from nodes, peers or other entities.

3. Verifying the validity of information in the TED,i.e., ensure that the network information obtained from nodes or elsewhere is relatively timely, or not stale. By analogy with similar functionality provided by IGPs this can be done via a process where discrete "chunks" of TED information are "aged" and discard when expired. This combined with nodes periodically resending their local TE information leads to a timely TED.

3. Standardization and Protocol Considerations

In the previous section we examined a number of architectural alternatives for TED creation and maintenance between PCE(s) and the network. Here we examine aspects of these alternatives that could be suitable for standardization. First there are a number of functions which are independent of the particular architectural alternatives, these include:

- An information model for the TED
- Basic PCE TED creation and maintenance procedures
- Information packaging for use in TED creation, maintenance and exchange
- NE to PCE (or Pub/Sub) communication of TED information --- interface and protocol (e.g. PCEP)
- NEs discovering PCE (or Pub/Sub) for TED creation and maintenance purposes

By the "information model" for the TED we mean the raw information that a path computation algorithm would work with somewhat independent of how it might be packaged for TED maintenance and creation. Initial efforts along these lines have started at CCAMP for wavelength switched optical networks for non-impairment RWA [WSON-Info] and impairment aware RWA [WSON-IMP-Info].

Given a TED information model if we can agree on basic PCE TED creation and maintenance procedures we can then come up with a standardized way to package the information for use in such procedures. The analogy here is with an IGPs database maintenance procedures such as aging and the packaging of link state information information into LSA (link state advertisements). LSAs form the basic chunks of an IGP’s database. OSPF LSAs include an age field to assist in the ageing procedure and also has an advertising router field that aids in redistribution decisions, i.e., flooding. However the detailed TE information is encoded in LSAs via type length value (TLV) structures and it is this information that is used in path computation.

From there we could standardize the interface between a NE and a PCE for communication of TE information. This interface includes NE and PCE behaviors as well as a communications protocol.

Finally for the common behaviors we need a way for the NEs to find the PCEs or an intermediate publish/subscribe system to which they will send their TE information. As was previously pointed out this could be based on small enhancements to existing PCE discovery mechanisms.

3.1. Architecture Specific Standardization Aspects

Case 1: NEs send to all PCEs

This case has commonalities with both cases 2 and 3 and does not appear to have unique standardization aspects. As pointed out in section 2.1. We do need to consider when a new PCE comes online.

Case 2: Publish/Subscribe Server

In this case we would need to additionally standardize

1. how a new PCE coming online synchronizes with the publish/subscribe server

1. how PCEs and publish subscribe server communicate

2. Redundancy for publish subscribe server

Case 3: PCE to PCE sharing TE information learned from NEs

Here we would need the following additional mechanisms standardized:

1. The PCE to PCE interface and protocol

2. The method for PCEs to discover PCEs for the purpose of TE information sharing

3. PCE to PCE association for information sharing, in particular sharing update information.

4. Security Considerations

This draft discusses an alternative technique for PCEs to build and maintain a traffic engineering database. In this approach network nodes would directly send traffic engineering information to a PCE. It may be desirable to protect such information from disclosure to unauthorized parties in addition it may be desirable to protect such communications from interference (modification) since they can be critical to the operation of the network. In particular, this information is the same or similar to that which would be

disseminated via a link state routing protocol with traffic engineering extensions.

5. IANA Considerations

This version of this document does not introduce any items for IANA to consider.

6. Conclusions

This document introduced several alternative architectures for PCEs to create and maintain a traffic engineering database (TED) via information directly or indirectly received from network elements and identified common aspects of these approaches. The TED is a critical piece of the overall PCE architecture since without it path computations cannot proceed. Though not explicitly out of scope the PCE working group does not have a work item or study item devoted to TED creation and maintenance. Such a work item can lead to enhanced interoperability and simplicity of PCE implementations. This document identified several common areas within these alternatives that could be standardized. In addition, the alternative approaches to TED creation and maintenance discussed here offloads both the network nodes and routing protocols from either some or all TED creation and maintenance duties at the same time it does not add significant new processing to a PCE that has already been participating in IGP based TED creation and maintenance.

7. Acknowledgments

We would like to thank Adrian Farrel for his useful comments and suggestions.

8. References

8.1. Normative References

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

[RFC4674] Le Roux, J., Ed., "Requirements for Path Computation Element (PCE) Discovery", RFC 4674, October 2006.

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

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

[RFC5250] Berger, L., Bryskin, I., Zinin, A., and R. Coltun, "The OSPF Opaque LSA Option", RFC 5250, July 2008.

8.2. Informative References

[JMS] Java Message Service, Version 1.1, April 2002, Sun Microsystems.

[PCE-Initiated] E. Crabbe, et. al., "PCEP Extensions for PCE-initiated LSP Setup in a Stateful PCE Model", draft-ietf-pce-pce-initiated-lsp, work in progress.

[S-PCE-GMPLS] X. Zhang, et. al, "Path Computation Element (PCE) Protocol Extensions for Stateful PCE Usage in GMPLS-controlled Networks", draft-ietf-pce-pcep-stateful-pce-gmpls, work in progress.

[PCE-WSON] Y. Lee, G. Bernstein, "PCEP Requirements for the support of Wavelength Switched Optical Networks (WSON)", work in progress, draft-lee-pce-wson-routing-wavelength-05.txt, February 2009.

[RFC4456] Bates, T., Chen, E., and R. Chandra, "BGP Route Reflection: An Alternative to Full Mesh Internal BGP (IBGP)", RFC 4456, April 2006.

[Imp-Frame] G. Bernstein, Y. Lee, D. Li, A Framework for the Control and Measurement of Wavelength Switched Optical Networks (WSON) with Impairments, Work in Progress, October 2008.

[WSON-Frame] Y. Lee, G. Bernstein, W. Imajuku, "Framework for GMPLS and PCE Control of Wavelength Switched Optical Networks", work in progress: draft-ietf-ccamp-wavelength-switched-framework.

[PCEP-TE] D. Dhody, Y. Lee, "PCEP Extension for Transporting TE Data", work in progress: draft-dhodylee-pce-pcep-te-data-extn.

[WSON-IMP-Info] Y. Lee, G. Bernstein, "Information Model for Impaired Optical Path Validation", work in progress: draft-bernstein-wson-impairment-info-02.txt, March 2009.

[HWANG] S. Hwang, et al, "An Experimental Analysis on OSPF-TE Convergence Time", Proc. SPIE 7137, Network Architectures, Management, and Applications, November 19, 2008.

Author's Addresses

Young Lee
Huawei Technologies
5340 Legacy Drive, Building 3
Plano, TX 75023, USA

Phone: (469) 277-5838
Email: leeyoung@huawei.com

Haomian Zheng
Huawei Technologies Co., Ltd.
F3-1-B R&D Center, Huawei Base,
Bantian, Longgang District
Shenzhen 518129 P.R.China

Phone: +86-755-28979835
Email: zhenghaomian@huawei.com

Contributor's Addresses

Dhruv Dhody
Huawei Technologies
Leela Palace
Bangalore, Karnataka 560008
India

EMail: dhruv.ietf@gmail.com

Xian Zhang
Huawei Technologies Co., Ltd.
F3-1-B R&D Center, Huawei Base,
Bantian, Longgang District
Shenzhen 518129 P.R.China

Phone: +86-755-28979835
Email: zhangxian@huawei.com

Disclaimer of Validity

Acknowledgment

Funding for the RFC Editor function is currently provided by the Internet Society.

PCE Working Group
Internet-Draft
Intended status: Best Current Practice
Expires: April 26, 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
October 23, 2014

PCEP Best Current Practices - Message formats and extensions
draft-many-pce-pcep-bcp-01

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.

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 capabilites 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 appropiate, 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 specific 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>
                       <udpate-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

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.

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 capabilites 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 appropiate, 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 specific 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>
                    <udpate-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

PCE Working Group
Internet-Draft
Intended status: Standards Track
Expires: January 8, 2017

D. Dhody, Ed.
Huawei Technologies
J. Hardwick
Metaswitch
V. Beeram
Juniper Networks
J. Tantsura
July 7, 2016

A YANG Data Model for Path Computation Element Communications Protocol (PCEP)
draft-pkd-pce-pcep-yang-06

Abstract

This document defines a YANG data model for the management of Path Computation Element communications Protocol (PCEP) for communications between a Path Computation Client (PCC) and a Path Computation Element (PCE), or between two PCEs. The data model includes configuration data and state data (status information and counters for the collection of statistics).

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 January 8, 2017.

Copyright Notice

Table of Contents

1. Introduction 2
2. Requirements Language 3
3. Terminology and Notation 3
3.1. Tree Diagrams 4
3.2. Prefixes in Data Node Names 5
4. Objectives 5
5. The Design of PCEP Data Model 6
5.1. The Entity 17
5.2. The Peer Lists 18
5.3. The Session Lists 18
5.4. Notifications 19
6. Advanced PCE Features 19
6.1. Stateful PCE's LSP-DB 19
7. Open Issues and Next Step 20
7.1. The PCE-Initiated LSP 20
7.2. PCEP over TLS (PCEPS) 20
8. PCEP YANG Module 20
9. Security Considerations 83
10. Manageability Considerations 84
10.1. Control of Function and Policy 84
10.2. Information and Data Models 84
10.3. Liveness Detection and Monitoring 84
10.4. Verify Correct Operations 84
10.5. Requirements On Other Protocols 84
10.6. Impact On Network Operations 84
11. IANA Considerations 84
12. Acknowledgements 85
13. References 85
13.1. Normative References 85
13.2. Informative References 86
Appendix A. Contributor Addresses 88
Authors' Addresses 89

1. Introduction

The Path Computation Element (PCE) defined in [RFC4655] is an entity that is capable of computing a network path or route based on a network graph, and applying computational constraints. A Path

Computation Client (PCC) may make requests to a PCE for paths to be computed.

PCEP is the communication protocol between a PCC and PCE and is defined in [RFC5440]. PCEP 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 Multiprotocol Label Switching (MPLS) and Generalized MPLS (GMPLS) Traffic Engineering (TE). [I-D.ietf-pce-stateful-pce] specifies extensions to PCEP to enable stateful control of MPLS TE LSPs.

This document defines a YANG [RFC6020] data model for the management of PCEP speakers. It is important to establish a common data model for how PCEP speakers are identified, configured, and monitored. The data model includes configuration data and state data (status information and counters for the collection of statistics).

This document contains a specification of the PCEP YANG module, "ietf-pcep" which provides the PCEP [RFC5440] data model.

2. Requirements Language

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

3. Terminology and Notation

This document uses the terminology defined in [RFC4655] and [RFC5440]. In particular, it uses the following acronyms.

- Path Computation Request message (PCReq).
- Path Computation Reply message (PCRep).
- Notification message (PCNtf).
- Error message (PCErr).
- Request Parameters object (RP).
- Synchronization Vector object (SVEC).
- Explicit Route object (ERO).

This document also uses the following terms defined in [RFC7420]:

- PCEP entity: a local PCEP speaker.

- PCEP peer: to refer to a remote PCEP speaker.
- PCEP speaker: where it is not necessary to distinguish between local and remote.

Further, this document also uses the following terms defined in [I-D.ietf-pce-stateful-pce] :

- Stateful PCE, Passive Stateful PCE, Active Stateful PCE
- Delegation, Revocation, Redelegation
- LSP State Report, Path Computation Report message (PCRpt).
- LSP State Update, Path Computation Update message (PCUpd).

[I-D.ietf-pce-pce-initiated-lsp] :

- PCE-initiated LSP, Path Computation LSP Initiate Message (PCInitiate).

[I-D.ietf-pce-lsp-setup-type] :

- Path Setup Type (PST).

[I-D.ietf-pce-segment-routing] :

- Segment Routing (SR).
- Segment Identifier (SID).
- Maximum SID Depth (MSD).

3.1. Tree Diagrams

A graphical representation of the complete data tree is presented in Section 5. The meaning of the symbols in these diagrams is as follows and as per [I-D.ietf-netmod-rfc6087bis]:

- Brackets "[" and "]" enclose list keys.
- Curly braces "{" and "}" contain names of optional features that make the corresponding node conditional.
- Abbreviations before data node names: "rw" means configuration (read-write), and "ro" state data (read-only).

- Symbols after data node names: "?" means an optional node and "*" denotes a "list" or "leaf-list".
- Parentheses enclose choice and case nodes, and case nodes are also marked with a colon (":").
- Ellipsis ("...") stands for contents of subtrees that are not shown.

3.2. Prefixes in Data Node Names

In this document, names of data nodes and other data model objects are often used without a prefix, as long as it is clear from the context in which YANG module each name is defined. Otherwise, names are prefixed using the standard prefix associated with the corresponding YANG module, as shown in Table 1.

Prefix	YANG module	Reference
yang	ietf-yang-types	[RFC6991]
inet	ietf-inet-types	[RFC6991]

Table 1: Prefixes and corresponding YANG modules

4. Objectives

This section describes some of the design objectives for the model:

- In case of existing implementations, it needs to map the data model defined in this document to their proprietary native data model. To facilitate such mappings, the data model should be simple.
- The data model should be suitable for new implementations to use as is.
- Mapping to the PCEP MIB Module should be clear.
- The data model should allow for static configurations of peers.
- The data model should include read-only counters in order to gather statistics for sent and received PCEP messages, received messages with errors, and messages that could not be sent due to errors.

   o It should be fairly straightforward to augment the base data model for advanced PCE features.

5. The Design of PCEP Data Model

The module, "ietf-pcep", defines the basic components of a PCE speaker.

```
module: ietf-pcep
   +--rw pcep!
   |  +--rw entity
   |     +--rw addr                          inet:ip-address
   |     +--rw enabled?                      boolean
   |     +--rw role                          pcep-role
   |     +--rw description?                  string
   |     +--rw domain
   |     |  +--rw domain* [domain-type domain]
   |     |     +--rw domain-type    domain-type
   |     |     +--rw domain         domain
   |     +--rw capability
   |     |  +--rw gmpls?                 boolean {gmpls}?
   |     |  +--rw bi-dir?                boolean
   |     |  +--rw diverse?               boolean
   |     |  +--rw load-balance?          boolean
   |     |  +--rw synchronize?           boolean {svec}?
   |     |  +--rw objective-function?    boolean {obj-fn}?
   |     |  +--rw add-path-constraint?   boolean
   |     |  +--rw prioritization?        boolean
   |     |  +--rw multi-request?         boolean
   |     |  +--rw gco?                   boolean {gco}?
   |     |  +--rw p2mp?                  boolean {p2mp}?
   |     |  +--rw stateful {stateful}?
   |     |  |  +--rw enabled?         boolean
   |     |  |  +--rw active?          boolean
   |     |  |  +--rw pce-initiated?   boolean {pce-initiated}?
   |     |  +--rw sr {sr}?
   |     |     +--rw enabled?   boolean
   |     |     +--rw msd?       uint8
   |     +--rw pce-info
   |     |  +--rw scope
   |     |  |  +--rw intra-area-scope?           boolean
   |     |  |  +--rw intra-area-pref?            uint8
   |     |  |  +--rw inter-area-scope?           boolean
   |     |  |  +--rw inter-area-scope-default?   boolean
   |     |  |  +--rw inter-area-pref?            uint8
   |     |  |  +--rw inter-as-scope?             boolean
   |     |  |  +--rw inter-as-scope-default?     boolean
   |     |  |  +--rw inter-as-pref?              uint8
```

```
|     |  |  +--rw inter-layer-scope?          boolean
|     |  |  +--rw inter-layer-pref?           uint8
|     |  +--rw neigh-domains
|     |  |  +--rw domain* [domain-type domain]
|     |  |     +--rw domain-type    domain-type
|     |  |     +--rw domain         domain
|     |  +--rw (auth-type-selection)?
|     |     +--:(auth-key-chain)
|     |     |  +--rw key-chain?          key-chain:key-chain-ref
|     |     +--:(auth-key)
|     |     |  +--rw key?                string
|     |     |  +--rw crypto-algorithm
|     |     |     +--rw (algorithm)?
|     |     |        +--:(hmac-sha-1-12) {crypto-hmac-sha-1-12}?
|     |     |        |  +--rw hmac-sha1-12?             empty
|     |     |        +--:(aes-cmac-prf-128) {aes-cmac-prf-128}?
|     |     |        |  +--rw aes-cmac-prf-128?         empty
|     |     |        +--:(md5)
|     |     |        |  +--rw md5?                      empty
|     |     |        +--:(sha-1)
|     |     |        |  +--rw sha-1?                    empty
|     |     |        +--:(hmac-sha-1)
|     |     |        |  +--rw hmac-sha-1?               empty
|     |     |        +--:(hmac-sha-256)
|     |     |        |  +--rw hmac-sha-256?             empty
|     |     |        +--:(hmac-sha-384)
|     |     |        |  +--rw hmac-sha-384?             empty
|     |     |        +--:(hmac-sha-512)
|     |     |        |  +--rw hmac-sha-512?             empty
|     |     |        +--:(clear-text) {clear-text}?
|     |     |        |  +--rw clear-text?               empty
|     |     |        +--:(replay-protection-only) {replay-protection-only}?
|     |     |           +--rw replay-protection-only?   empty
|     |     +--:(auth-tls) {tls}?
|     |        +--rw tls
|     +--rw connect-timer?               uint32
|     +--rw connect-max-retry?           uint32
|     +--rw init-backoff-timer?          uint32
|     +--rw max-backoff-timer?           uint32
|     +--rw open-wait-timer?             uint32
|     +--rw keep-wait-timer?             uint32
|     +--rw keep-alive-timer?            uint32
|     +--rw dead-timer?                  uint32
|     +--rw allow-negotiation?           boolean
|     +--rw max-keep-alive-timer?        uint32
|     +--rw max-dead-timer?              uint32
|     +--rw min-keep-alive-timer?        uint32
|     +--rw min-dead-timer?              uint32
```

```
|     +--rw sync-timer?                   uint32 {svec}?
|     +--rw request-timer?                uint32
|     +--rw max-sessions?                 uint32
|     +--rw max-unknown-reqs?             uint32
|     +--rw max-unknown-msgs?             uint32
|     +--rw pcep-notification-max-rate    uint32
|     +--rw stateful-parameter {stateful}?
|     |  +--rw state-timeout?          uint32
|     |  +--rw redelegation-timeout?   uint32
|     |  +--rw rpt-non-pcep-lsp?       boolean
|     +--rw peers
|        +--rw peer* [addr]
|           +--rw addr                inet:ip-address
|           +--rw description?        string
|           +--rw domain
|           |  +--rw domain* [domain-type domain]
|           |     +--rw domain-type    domain-type
|           |     +--rw domain         domain
|           +--rw capability
|           |  +--rw gmpls?                 boolean {gmpls}?
|           |  +--rw bi-dir?                boolean
|           |  +--rw diverse?               boolean
|           |  +--rw load-balance?          boolean
|           |  +--rw synchronize?           boolean {svec}?
|           |  +--rw objective-function?    boolean {obj-fn}?
|           |  +--rw add-path-constraint?   boolean
|           |  +--rw prioritization?        boolean
|           |  +--rw multi-request?         boolean
|           |  +--rw gco?                   boolean {gco}?
|           |  +--rw p2mp?                  boolean {p2mp}?
|           |  +--rw stateful {stateful}?
|           |  |  +--rw enabled?         boolean
|           |  |  +--rw active?          boolean
|           |  |  +--rw pce-initiated?   boolean {pce-initiated}?
|           |  +--rw sr {sr}?
|           |     +--rw enabled?   boolean
|           |     +--rw msd?       uint8
|           +--rw scope
|           |  +--rw intra-area-scope?           boolean
|           |  +--rw intra-area-pref?            uint8
|           |  +--rw inter-area-scope?           boolean
|           |  +--rw inter-area-scope-default?   boolean
|           |  +--rw inter-area-pref?            uint8
|           |  +--rw inter-as-scope?             boolean
|           |  +--rw inter-as-scope-default?     boolean
|           |  +--rw inter-as-pref?              uint8
|           |  +--rw inter-layer-scope?          boolean
|           |  +--rw inter-layer-pref?           uint8
```

```
   |           +--rw neigh-domains
   |           |  +--rw domain* [domain-type domain]
   |           |     +--rw domain-type    domain-type
   |           |     +--rw domain         domain
   |           +--rw delegation-pref?    uint8 {stateful}?
   |           +--rw (auth-type-selection)?
   |              +--:(auth-key-chain)
   |              |  +--rw key-chain?          key-chain:key-chain-ref
   |              +--:(auth-key)
   |              |  +--rw key?                string
   |              |  +--rw crypto-algorithm
   |              |     +--rw (algorithm)?
   |              |        +--:(hmac-sha-1-12) {crypto-hmac-sha-1-12}?
   |              |        |  +--rw hmac-sha1-12?             empty
   |              |        +--:(aes-cmac-prf-128) {aes-cmac-prf-128}?
   |              |        |  +--rw aes-cmac-prf-128?         empty
   |              |        +--:(md5)
   |              |        |  +--rw md5?                      empty
   |              |        +--:(sha-1)
   |              |        |  +--rw sha-1?                    empty
   |              |        +--:(hmac-sha-1)
   |              |        |  +--rw hmac-sha-1?               empty
   |              |        +--:(hmac-sha-256)
   |              |        |  +--rw hmac-sha-256?             empty
   |              |        +--:(hmac-sha-384)
   |              |        |  +--rw hmac-sha-384?             empty
   |              |        +--:(hmac-sha-512)
   |              |        |  +--rw hmac-sha-512?             empty
   |              |        +--:(clear-text) {clear-text}?
   |              |        |  +--rw clear-text?               empty
   |              |        +--:(replay-protection-only) {replay-protection-only}
?
   |              |           +--rw replay-protection-only?   empty
   |              +--:(auth-tls) {tls}?
   |                 +--rw tls
   +--ro pcep-state
      +--ro entity
         +--ro addr?                  inet:ip-address
         +--ro index?                 uint32
         +--ro admin-status?          pcep-admin-status
         +--ro oper-status?           pcep-admin-status
         +--ro role?                  pcep-role
         +--ro domain
         |  +--ro domain* [domain-type domain]
         |     +--ro domain-type    domain-type
         |     +--ro domain         domain
         +--ro capability
         |  +--ro gmpls?                boolean {gmpls}?
         |  +--ro bi-dir?               boolean
```

```
|  +--ro diverse?               boolean
|  +--ro load-balance?          boolean
|  +--ro synchronize?           boolean {svec}?
|  +--ro objective-function?    boolean {obj-fn}?
|  +--ro add-path-constraint?   boolean
|  +--ro prioritization?        boolean
|  +--ro multi-request?         boolean
|  +--ro gco?                   boolean {gco}?
|  +--ro p2mp?                  boolean {p2mp}?
|  +--ro stateful {stateful}?
|  |  +--ro enabled?         boolean
|  |  +--ro active?          boolean
|  |  +--ro pce-initiated?   boolean {pce-initiated}?
|  +--ro sr {sr}?
|     +--ro enabled?   boolean
|     +--ro msd?       uint8
+--ro pce-info
|  +--ro scope
|  |  +--ro intra-area-scope?           boolean
|  |  +--ro intra-area-pref?            uint8
|  |  +--ro inter-area-scope?           boolean
|  |  +--ro inter-area-scope-default?   boolean
|  |  +--ro inter-area-pref?            uint8
|  |  +--ro inter-as-scope?             boolean
|  |  +--ro inter-as-scope-default?     boolean
|  |  +--ro inter-as-pref?              uint8
|  |  +--ro inter-layer-scope?          boolean
|  |  +--ro inter-layer-pref?           uint8
|  +--ro neigh-domains
|  |  +--ro domain* [domain-type domain]
|  |     +--ro domain-type    domain-type
|  |     +--ro domain         domain
|  +--ro (auth-type-selection)?
|     +--:(auth-key-chain)
|     |  +--ro key-chain?          key-chain:key-chain-ref
|     +--:(auth-key)
|     |  +--ro key?                string
|     |  +--ro crypto-algorithm
|     |     +--ro (algorithm)?
|     |        +--:(hmac-sha-1-12) {crypto-hmac-sha-1-12}?
|     |        |  +--ro hmac-sha1-12?             empty
|     |        +--:(aes-cmac-prf-128) {aes-cmac-prf-128}?
|     |        |  +--ro aes-cmac-prf-128?         empty
|     |        +--:(md5)
|     |        |  +--ro md5?                      empty
|     |        +--:(sha-1)
|     |        |  +--ro sha-1?                    empty
|     |        +--:(hmac-sha-1)
```

```
|     |       |  +--ro hmac-sha-1?              empty
|     |       +--:(hmac-sha-256)
|     |       |  +--ro hmac-sha-256?            empty
|     |       +--:(hmac-sha-384)
|     |       |  +--ro hmac-sha-384?            empty
|     |       +--:(hmac-sha-512)
|     |       |  +--ro hmac-sha-512?            empty
|     |       +--:(clear-text) {clear-text}?
|     |       |  +--ro clear-text?              empty
|     |       +--:(replay-protection-only) {replay-protection-only}?
|     |          +--ro replay-protection-only?  empty
|     +--:(auth-tls) {tls}?
|        +--ro tls
+--ro connect-timer?          uint32
+--ro connect-max-retry?      uint32
+--ro init-backoff-timer?     uint32
+--ro max-backoff-timer?      uint32
+--ro open-wait-timer?        uint32
+--ro keep-wait-timer?        uint32
+--ro keep-alive-timer?       uint32
+--ro dead-timer?             uint32
+--ro allow-negotiation?      boolean
+--ro max-keep-alive-timer?   uint32
+--ro max-dead-timer?         uint32
+--ro min-keep-alive-timer?   uint32
+--ro min-dead-timer?         uint32
+--ro sync-timer?             uint32 {svec}?
+--ro request-timer?          uint32
+--ro max-sessions?           uint32
+--ro max-unknown-reqs?       uint32
+--ro max-unknown-msgs?       uint32
+--ro stateful-parameter {stateful}?
|  +--ro state-timeout?         uint32
|  +--ro redelegation-timeout?  uint32
|  +--ro rpt-non-pcep-lsp?      boolean
+--ro lsp-db {stateful}?
|  +--ro association-list* [id source global-source extended-id]
|  |  +--ro type?            assoc-type
|  |  +--ro id               uint16
|  |  +--ro source           inet:ip-address
|  |  +--ro global-source    uint32
|  |  +--ro extended-id      string
|  |  +--ro lsp* [plsp-id pcc-id]
|  |     +--ro plsp-id    -> /pcep-state/entity/lsp-db/lsp/plsp-id
|  |     +--ro pcc-id     -> /pcep-state/entity/lsp-db/lsp/pcc-id
|  +--ro lsp* [plsp-id pcc-id]
|     +--ro plsp-id              uint32
|     +--ro pcc-id               inet:ip-address
```

```
         |     +--ro lsp-ref
         |     |  +--ro source?               -> /te:te/lsps-state/lsp/source
         |     |  +--ro destination?          -> /te:te/lsps-state/lsp/destinati
on
         |     |  +--ro tunnel-id?            -> /te:te/lsps-state/lsp/tunnel-id
         |     |  +--ro lsp-id?               -> /te:te/lsps-state/lsp/lsp-id
         |     |  +--ro extended-tunnel-id?   -> /te:te/lsps-state/lsp/extended-
tunnel-id
         |     |  +--ro type?                 -> /te:te/lsps-state/lsp/type
         |     +--ro admin-state?          boolean
         |     +--ro operational-state?    operational-state
         |     +--ro delegated
         |     |  +--ro enabled?   boolean
         |     |  +--ro pce?       -> /pcep-state/entity/peers/peer/addr
         |     |  +--ro srp-id?    uint32
         |     +--ro initiation {pce-initiated}?
         |     |  +--ro enabled?   boolean
         |     |  +--ro pce?       -> /pcep-state/entity/peers/peer/addr
         |     +--ro symbolic-path-name?   string
         |     +--ro last-error?           lsp-error
         |     +--ro pst?                  pst
         |     +--ro association-list* [id source global-source extended-id]
         |        +--ro id               -> /pcep-state/entity/lsp-db/associatio
n-list/id
         |        +--ro source           -> /pcep-state/entity/lsp-db/associatio
n-list/source
         |        +--ro global-source    -> /pcep-state/entity/lsp-db/associatio
n-list/global-source
         |        +--ro extended-id      -> /pcep-state/entity/lsp-db/associatio
n-list/extended-id
         +--ro peers
            +--ro peer* [addr]
               +--ro addr                    inet:ip-address
               +--ro role?                   pcep-role
               +--ro domain
               |  +--ro domain* [domain-type domain]
               |     +--ro domain-type    domain-type
               |     +--ro domain         domain
               +--ro capability
               |  +--ro gmpls?                boolean {gmpls}?
               |  +--ro bi-dir?               boolean
               |  +--ro diverse?              boolean
               |  +--ro load-balance?         boolean
               |  +--ro synchronize?          boolean {svec}?
               |  +--ro objective-function?   boolean {obj-fn}?
               |  +--ro add-path-constraint?  boolean
               |  +--ro prioritization?       boolean
               |  +--ro multi-request?        boolean
               |  +--ro gco?                  boolean {gco}?
               |  +--ro p2mp?                 boolean {p2mp}?
               |  +--ro stateful {stateful}?
               |  |  +--ro enabled?         boolean
               |  |  +--ro active?          boolean
               |  |  +--ro pce-initiated?   boolean {pce-initiated}?
```

```
               |  +--ro sr {sr}?
               |     +--ro enabled?   boolean
               |     +--ro msd?       uint8
               +--ro pce-info
               |  +--ro scope
               |  |  +--ro intra-area-scope?           boolean
               |  |  +--ro intra-area-pref?            uint8
               |  |  +--ro inter-area-scope?           boolean
               |  |  +--ro inter-area-scope-default?   boolean
               |  |  +--ro inter-area-pref?            uint8
               |  |  +--ro inter-as-scope?             boolean
               |  |  +--ro inter-as-scope-default?     boolean
               |  |  +--ro inter-as-pref?              uint8
               |  |  +--ro inter-layer-scope?          boolean
               |  |  +--ro inter-layer-pref?           uint8
               |  +--ro neigh-domains
               |     +--ro domain* [domain-type domain]
               |        +--ro domain-type    domain-type
               |        +--ro domain         domain
               +--ro delegation-pref?        uint8 {stateful}?
               +--ro (auth-type-selection)?
               |  +--:(auth-key-chain)
               |  |  +--ro key-chain?              key-chain:key-chain-ref
               |  +--:(auth-key)
               |  |  +--ro key?                    string
               |  |  +--ro crypto-algorithm
               |  |     +--ro (algorithm)?
               |  |        +--:(hmac-sha-1-12) {crypto-hmac-sha-1-12}?
               |  |        |  +--ro hmac-sha1-12?             empty
               |  |        +--:(aes-cmac-prf-128) {aes-cmac-prf-128}?
               |  |        |  +--ro aes-cmac-prf-128?         empty
               |  |        +--:(md5)
               |  |        |  +--ro md5?                     empty
               |  |        +--:(sha-1)
               |  |        |  +--ro sha-1?                   empty
               |  |        +--:(hmac-sha-1)
               |  |        |  +--ro hmac-sha-1?              empty
               |  |        +--:(hmac-sha-256)
               |  |        |  +--ro hmac-sha-256?            empty
               |  |        +--:(hmac-sha-384)
               |  |        |  +--ro hmac-sha-384?            empty
               |  |        +--:(hmac-sha-512)
               |  |        |  +--ro hmac-sha-512?            empty
               |  |        +--:(clear-text) {clear-text}?
               |  |        |  +--ro clear-text?              empty
               |  |        +--:(replay-protection-only) {replay-protection-only}
?
               |  |           +--ro replay-protection-only?   empty
               |  +--:(auth-tls) {tls}?
```

```
|     +--ro tls
+--ro discontinuity-time?     yang:timestamp
+--ro initiate-session?       boolean
+--ro session-exists?         boolean
+--ro num-sess-setup-ok?      yang:counter32
+--ro num-sess-setup-fail?    yang:counter32
+--ro session-up-time?        yang:timestamp
+--ro session-fail-time?      yang:timestamp
+--ro session-fail-up-time?   yang:timestamp
+--ro pcep-stats
|  +--ro avg-rsp-time?               uint32
|  +--ro lwm-rsp-time?               uint32
|  +--ro hwm-rsp-time?               uint32
|  +--ro num-pcreq-sent?             yang:counter32
|  +--ro num-pcreq-rcvd?             yang:counter32
|  +--ro num-pcrep-sent?             yang:counter32
|  +--ro num-pcrep-rcvd?             yang:counter32
|  +--ro num-pcerr-sent?             yang:counter32
|  +--ro num-pcerr-rcvd?             yang:counter32
|  +--ro num-pcntf-sent?             yang:counter32
|  +--ro num-pcntf-rcvd?             yang:counter32
|  +--ro num-keepalive-sent?         yang:counter32
|  +--ro num-keepalive-rcvd?         yang:counter32
|  +--ro num-unknown-rcvd?           yang:counter32
|  +--ro num-corrupt-rcvd?           yang:counter32
|  +--ro num-req-sent?               yang:counter32
|  +--ro num-req-sent-pend-rep?      yang:counter32
|  +--ro num-req-sent-ero-rcvd?      yang:counter32
|  +--ro num-req-sent-nopath-rcvd?   yang:counter32
|  +--ro num-req-sent-cancel-rcvd?   yang:counter32
|  +--ro num-req-sent-error-rcvd?    yang:counter32
|  +--ro num-req-sent-timeout?       yang:counter32
|  +--ro num-req-sent-cancel-sent?   yang:counter32
|  +--ro num-req-rcvd?               yang:counter32
|  +--ro num-req-rcvd-pend-rep?      yang:counter32
|  +--ro num-req-rcvd-ero-sent?      yang:counter32
|  +--ro num-req-rcvd-nopath-sent?   yang:counter32
|  +--ro num-req-rcvd-cancel-sent?   yang:counter32
|  +--ro num-req-rcvd-error-sent?    yang:counter32
|  +--ro num-req-rcvd-cancel-rcvd?   yang:counter32
|  +--ro num-rep-rcvd-unknown?       yang:counter32
|  +--ro num-req-rcvd-unknown?       yang:counter32
|  +--ro svec {svec}?
|  |  +--ro num-svec-sent?       yang:counter32
|  |  +--ro num-svec-req-sent?   yang:counter32
|  |  +--ro num-svec-rcvd?       yang:counter32
|  |  +--ro num-svec-req-rcvd?   yang:counter32
|  +--ro stateful {stateful}?
```

```
|  |  +--ro num-pcrpt-sent?             yang:counter32
|  |  +--ro num-pcrpt-rcvd?             yang:counter32
|  |  +--ro num-pcupd-sent?             yang:counter32
|  |  +--ro num-pcupd-rcvd?             yang:counter32
|  |  +--ro num-rpt-sent?               yang:counter32
|  |  +--ro num-rpt-rcvd?               yang:counter32
|  |  +--ro num-rpt-rcvd-error-sent?    yang:counter32
|  |  +--ro num-upd-sent?               yang:counter32
|  |  +--ro num-upd-rcvd?               yang:counter32
|  |  +--ro num-upd-rcvd-unknown?       yang:counter32
|  |  +--ro num-upd-rcvd-undelegated?   yang:counter32
|  |  +--ro num-upd-rcvd-error-sent?    yang:counter32
|  |  +--ro initiation {pce-initiated}?
|  |     +--ro num-pcinitiate-sent?            yang:counter32
|  |     +--ro num-pcinitiate-rcvd?            yang:counter32
|  |     +--ro num-initiate-sent?              yang:counter32
|  |     +--ro num-initiate-rcvd?              yang:counter32
|  |     +--ro num-initiate-rcvd-error-sent?   yang:counter32
|  +--ro num-req-sent-closed?        yang:counter32
|  +--ro num-req-rcvd-closed?        yang:counter32
+--ro sessions
   +--ro session* [initiator]
      +--ro initiator               pcep-initiator
      +--ro state-last-change?      yang:timestamp
      +--ro state?                  pcep-sess-state
      +--ro session-creation?       yang:timestamp
      +--ro connect-retry?          yang:counter32
      +--ro local-id?               uint32
      +--ro remote-id?              uint32
      +--ro keepalive-timer?        uint32
      +--ro peer-keepalive-timer?   uint32
      +--ro dead-timer?             uint32
      +--ro peer-dead-timer?        uint32
      +--ro ka-hold-time-rem?       uint32
      +--ro overloaded?             boolean
      +--ro overload-time?          uint32
      +--ro peer-overloaded?        boolean
      +--ro peer-overload-time?     uint32
      +--ro lspdb-sync?             sync-state {stateful}?
      +--ro discontinuity-time?     yang:timestamp
      +--ro pcep-stats
         +--ro avg-rsp-time?              uint32
         +--ro lwm-rsp-time?              uint32
         +--ro hwm-rsp-time?              uint32
         +--ro num-pcreq-sent?            yang:counter32
         +--ro num-pcreq-rcvd?            yang:counter32
         +--ro num-pcrep-sent?            yang:counter32
         +--ro num-pcrep-rcvd?            yang:counter32
```

```
                        +--ro num-pcerr-sent?             yang:counter32
                        +--ro num-pcerr-rcvd?             yang:counter32
                        +--ro num-pcntf-sent?             yang:counter32
                        +--ro num-pcntf-rcvd?             yang:counter32
                        +--ro num-keepalive-sent?         yang:counter32
                        +--ro num-keepalive-rcvd?         yang:counter32
                        +--ro num-unknown-rcvd?           yang:counter32
                        +--ro num-corrupt-rcvd?           yang:counter32
                        +--ro num-req-sent?               yang:counter32
                        +--ro num-req-sent-pend-rep?      yang:counter32
                        +--ro num-req-sent-ero-rcvd?      yang:counter32
                        +--ro num-req-sent-nopath-rcvd?   yang:counter32
                        +--ro num-req-sent-cancel-rcvd?   yang:counter32
                        +--ro num-req-sent-error-rcvd?    yang:counter32
                        +--ro num-req-sent-timeout?       yang:counter32
                        +--ro num-req-sent-cancel-sent?   yang:counter32
                        +--ro num-req-rcvd?               yang:counter32
                        +--ro num-req-rcvd-pend-rep?      yang:counter32
                        +--ro num-req-rcvd-ero-sent?      yang:counter32
                        +--ro num-req-rcvd-nopath-sent?   yang:counter32
                        +--ro num-req-rcvd-cancel-sent?   yang:counter32
                        +--ro num-req-rcvd-error-sent?    yang:counter32
                        +--ro num-req-rcvd-cancel-rcvd?   yang:counter32
                        +--ro num-rep-rcvd-unknown?       yang:counter32
                        +--ro num-req-rcvd-unknown?       yang:counter32
                        +--ro svec {svec}?
                        |  +--ro num-svec-sent?       yang:counter32
                        |  +--ro num-svec-req-sent?   yang:counter32
                        |  +--ro num-svec-rcvd?       yang:counter32
                        |  +--ro num-svec-req-rcvd?   yang:counter32
                        +--ro stateful {stateful}?
                           +--ro num-pcrpt-sent?             yang:counter32
                           +--ro num-pcrpt-rcvd?             yang:counter32
                           +--ro num-pcupd-sent?             yang:counter32
                           +--ro num-pcupd-rcvd?             yang:counter32
                           +--ro num-rpt-sent?               yang:counter32
                           +--ro num-rpt-rcvd?               yang:counter32
                           +--ro num-rpt-rcvd-error-sent?    yang:counter32
                           +--ro num-upd-sent?               yang:counter32
                           +--ro num-upd-rcvd?               yang:counter32
                           +--ro num-upd-rcvd-unknown?       yang:counter32
                           +--ro num-upd-rcvd-undelegated?   yang:counter32
                           +--ro num-upd-rcvd-error-sent?    yang:counter32
                           +--ro initiation {pce-initiated}?
                              +--ro num-pcinitiate-sent?           yang:counter
32
                              +--ro num-pcinitiate-rcvd?           yang:counter
32
                              +--ro num-initiate-sent?             yang:counter
32
                              +--ro num-initiate-rcvd?             yang:counter
32
```

```
                              +--ro num-initiate-rcvd-error-sent?   yang:counter
32
notifications:
   +---n pcep-session-up
   |  +--ro peer-addr?           -> /pcep-state/entity/peers/peer/addr
   |  +--ro session-initiator?   -> /pcep-state/entity/peers/peer/sessions/sessi
on/initiator
   |  +--ro state-last-change?   yang:timestamp
   |  +--ro state?               pcep-sess-state
   +---n pcep-session-down
   |  +--ro peer-addr?           -> /pcep-state/entity/peers/peer/addr
   |  +--ro session-initiator?   pcep-initiator
   |  +--ro state-last-change?   yang:timestamp
   |  +--ro state?               pcep-sess-state
   +---n pcep-session-local-overload
   |  +--ro peer-addr?           -> /pcep-state/entity/peers/peer/addr
   |  +--ro session-initiator?   -> /pcep-state/entity/peers/peer/sessions/sessi
on/initiator
   |  +--ro overloaded?          boolean
   |  +--ro overload-time?       uint32
   +---n pcep-session-local-overload-clear
   |  +--ro peer-addr?    -> /pcep-state/entity/peers/peer/addr
   |  +--ro overloaded?   boolean
   +---n pcep-session-peer-overload
   |  +--ro peer-addr?            -> /pcep-state/entity/peers/peer/addr
   |  +--ro session-initiator?    -> /pcep-state/entity/peers/peer/sessions/sess
ion/initiator
   |  +--ro peer-overloaded?      boolean
   |  +--ro peer-overload-time?   uint32
   +---n pcep-session-peer-overload-clear
      +--ro peer-addr?         -> /pcep-state/entity/peers/peer/addr
      +--ro peer-overloaded?   boolean
```

5.1. The Entity

The PCEP yang module may contain status information for the local PCEP entity.

The entity has an IP address (using ietf-inet-types [RFC6991]) and a "role" leaf (the local entity PCEP role) as mandatory.

Note that, the PCEP MIB module [RFC7420] uses an entity list and a system generated entity index as a primary index to the read only entity table. If the device implements the PCEP MIB, the "index" leaf MUST contain the value of the corresponding pcePcepEntityIndex and only one entity is assumed.

5.2. The Peer Lists

The peer list contains peer(s) that the local PCEP entity knows about. A PCEP speaker is identified by its IP address. If there is a PCEP speaker in the network that uses multiple IP addresses then it looks like multiple distinct peers to the other PCEP speakers in the network.

Since PCEP sessions can be ephemeral, the peer list tracks a peer even when no PCEP session currently exists to that peer. The statistics contained are an aggregate of the statistics for all successive sessions to that peer.

To limit the quantity of information that is stored, an implementation MAY choose to discard this information if and only if no PCEP session exists to the corresponding peer.

The data model for PCEP peer presented in this document uses a flat list of peers. Each peer in the list is identified by its IP address (addr-type, addr).

There is one list for static peer configuration ("/pcep/entity/peers"), and a separate list for the operational state of all peers (i.e. static as well as discovered)("/pcep-state/entity/peers"). The former is used to enable remote PCE configuration at PCC (or PCE) while the latter has the operational state of these peers as well as the remote PCE peer which were discovered and PCC peers that have initiated session.

5.3. The Session Lists

The session list contains PCEP session that the PCEP entity (PCE or PCC) is currently participating in. The statistics in session are semantically different from those in peer since the former applies to the current session only, whereas the latter is the aggregate for all sessions that have existed to that peer.

Although [RFC5440] forbids more than one active PCEP session between a given pair of PCEP entities at any given time, there is a window during session establishment where two sessions may exist for a given pair, one representing a session initiated by the local PCEP entity and the other representing a session initiated by the peer. If either of these sessions reaches active state first, then the other is discarded.

The data model for PCEP session presented in this document uses a flat list of sessions. Each session in the list is identified by its

initiator. This index allows two sessions to exist transiently for a given peer, as discussed above.

There is only one list for the operational state of all sessions ("/pcep-state/entity/peers/peer/sessions/session").

5.4. Notifications

This YANG model defines a list of notifications to inform client of important events detected during the protocol operation. The notifications defined cover the PCEP MIB notifications.

6. Advanced PCE Features

This document contains a specification of the base PCEP YANG module, "ietf-pcep" which provides the basic PCEP [RFC5440] data model.

This document further handles advanced PCE features like -

- Capability and Scope
- Domain information (local/neighbour)
- Path-Key
- OF
- GCO
- P2MP
- GMPLS
- Inter-Layer
- Stateful PCE
- Segement Routing
- Authentication including PCEPS (TLS)

[Editor’s Note - Some of them would be added in a future revision.]

6.1. Stateful PCE’s LSP-DB

In the operational state of PCEP which supports stateful PCE mode, the list of LSP state are maintained in LSP-DB. The key is the PLSP-ID and the PCC IP address.

The PCEP data model contains the operational state of LSPs (/pcep-state/entity/lsp-db/lsp/) with PCEP specific attributes. The generic TE attributes of the LSP are defined in [I-D.ietf-teas-yang-te]. A reference to LSP state in TE model is maintained.

7. Open Issues and Next Step

This section is added so that open issues can be tracked. This section would be removed when the document is ready for publication.

7.1. The PCE-Initiated LSP

The TE Model at [I-D.ietf-teas-yang-te] should support creationg of tunnels at the controller (PCE) and marking them as PCE-Initiated. The LSP-DB in the PCEP Yang (/pcep-state/entity/lsp-db/lsp/initiation) also marks the LSPs which are PCE-initiated.

7.2. PCEP over TLS (PCEPS)

A future version of this document would add TLS related configurations.

8. PCEP YANG Module

RFC Ed.: In this section, replace all occurrences of 'XXXX' with the actual RFC number and all occurrences of the revision date below with the date of RFC publication (and remove this note).

```
<CODE BEGINS> file "ietf-pcep@2016-07-07.yang"
module ietf-pcep {
    namespace "urn:ietf:params:xml:ns:yang:ietf-pcep";
    prefix pcep;

    import ietf-inet-types {
        prefix "inet";
    }

    import ietf-yang-types {
        prefix "yang";
    }

    import ietf-te {
        prefix "te";
    }

    import ietf-key-chain {
       prefix "key-chain";
```

```
    }


    organization
        "IETF PCE (Path Computation Element) Working Group";

    contact
        "WG Web:   <http://tools.ietf.org/wg/pce/>
         WG List:  <mailto:pce@ietf.org>
         WG Chair: JP Vasseur
                   <mailto:jpv@cisco.com>
         WG Chair: Julien Meuric
                   <mailto:julien.meuric@orange.com>
         WG Chair: Jonathan Hardwick
                   <mailto:Jonathan.Hardwick@metaswitch.com>
         Editor:   Dhruv Dhody
                   <mailto:dhruv.ietf@gmail.com>";

    description
        "The YANG module defines a generic configuration and
         operational model for PCEP common across all of the
         vendor implementations.";


    revision 2016-07-07 {
        description "Initial revision.";
        reference
            "RFC XXXX:  A YANG Data Model for Path Computation
                        Element Communications Protocol
                        (PCEP)";
    }

    /*
     * Identities
     */

    identity pcep {
        description "Identity for the PCEP protocol.";
    }

    /*
     * Typedefs
     */
    typedef pcep-role {
        type enumeration {
            enum unknown {
                value "0";
                description
```

```
                "An unknown role";
            }
            enum pcc {
                value "1";
                description
                "The role of a Path Computation Client";
            }
            enum pce {
                value "2";
                description
                "The role of Path Computation Element";
            }
            enum pcc-and-pce {
                value "3";
                description
                "The role of both Path Computation Client and
                 Path Computation Element";
            }
        }

        description
            "The role of a PCEP speaker.
             Takes one of the following values
             - unknown(0): the role is not known.
             - pcc(1): the role is of a Path Computation
               Client (PCC).
             - pce(2): the role is of a Path Computation
               Server (PCE).
             - pccAndPce(3): the role is of both a PCC and
               a PCE.";

    }

    typedef pcep-admin-status {
        type enumeration {
                enum admin-status-up {
                value "1";
                description
                "Admin Status is Up";
            }
            enum admin-status-down {
                value "2";
                description
                "Admin Status is Down";
            }
        }

        description
```

```
        "The Admin Status of the PCEP entity.
         Takes one of the following values
             - admin-status-up(1): Admin Status is Up.
             - admin-status-down(2): Admin Status is Down";
    }

    typedef pcep-oper-status {
        type enumeration {
            enum oper-status-up {
                value "1";
                description
                "The PCEP entity is active";
            }
            enum oper-status-down {
                value "2";
                description
                "The PCEP entity is inactive";
            }
            enum oper-status-going-up {
                value "3";
                description
                "The PCEP entity is activating";
            }
            enum oper-status-going-down {
                value "4";
                description
                "The PCEP entity is deactivating";
            }
            enum oper-status-failed {
                value "5";
                description
                "The PCEP entity has failed and will recover
                 when possible.";
            }
            enum oper-status-failed-perm {
                value "6";
                description
                "The PCEP entity has failed and will not recover
                 without operator intervention";
            }
        }
        description
        "The operational status of the PCEP entity.
         Takes one of the following values
             - oper-status-up(1): Active
             - oper-status-down(2): Inactive
             - oper-status-going-up(3): Activating
             - oper-status-going-down(4): Deactivating
```

```
             - oper-status-failed(5): Failed
             - oper-status-failed-perm(6): Failed Permanantly";
    }

    typedef pcep-initiator {
        type enumeration {
            enum local {
                value "1";
                description
                "The local PCEP entity initiated the session";
            }

            enum remote {
                value "2";
                description
                "The remote PCEP peer initiated the session";
            }
        }
        description
        "The initiator of the session, that is, whether the TCP
         connection was initiated by the local PCEP entity or
         the remote peer.
         Takes one of the following values
             - local(1): Initiated locally
             - remote(2): Initiated remotely";
    }

    typedef pcep-sess-state {
        type enumeration {
            enum tcp-pending {
                value "1";
                description
                    "The tcp-pending state of PCEP session.";
            }

            enum open-wait {
                value "2";
                description
                    "The open-wait state of PCEP session.";
            }

            enum keep-wait {
                value "3";
                description
                    "The keep-wait state of PCEP session.";
            }

            enum session-up {
```

```
                value "4";
                description
                    "The session-up state of PCEP session.";
            }
        }
        description
            "The current state of the session.
             The set of possible states excludes the idle state
             since entries do not exist in the idle state.
             Takes one of the following values
                - tcp-pending(1): PCEP TCP Pending state
                - open-wait(2): PCEP Open Wait state
                - keep-wait(3): PCEP Keep Wait state
                - session-up(4): PCEP Session Up state";
    }

    typedef domain-type {
        type enumeration {
            enum ospf-area {
                value "1";
                description
                    "The OSPF area.";
            }
            enum isis-area {
                value "2";
                description
                    "The IS-IS area.";
            }
            enum as {
                value "3";
                description
                    "The Autonomous System (AS).";
            }
        }
        description
            "The PCE Domain Type";
    }

    typedef domain-ospf-area {
        type union {
            type uint32;
            type yang:dotted-quad;
       }
       description
            "OSPF Area ID.";
    }

    typedef domain-isis-area {
```

```
        type string {
            pattern '[0-9A-Fa-f]{2}\.([0-9A-Fa-f]{4}\.){0,3}';
        }
        description
            "IS-IS Area ID.";
    }

    typedef domain-as {
        type uint32;
        description
            "Autonomous System number.";

    }

    typedef domain {
        type union {
            type domain-ospf-area;
            type domain-isis-area;
            type domain-as;
        }
        description
            "The Domain Information";
    }

    typedef operational-state {
        type enumeration {
            enum down {
                value "0";
                description
                    "not active.";
            }
            enum up {
                value "1";
                description
                    "signalled.";
            }
            enum active {
                value "2";
                description
                    "up and carrying traffic.";
            }
            enum going-down {
                value "3";
                description
                    "LSP is being torn down, resources are
                     being released.";
            }
            enum going-up {
```

```
                value "4";
                description
                    "LSP is being signalled.";
            }
        }
        description
            "The operational status of the LSP";
    }

    typedef lsp-error {
        type enumeration {
            enum no-error {
                value "0";
                description
                    "No error, LSP is fine.";
            }
            enum unknown {
                value "1";
                description
                    "Unknown reason.";
            }
            enum limit {
                value "2";
                description
                    "Limit reached for PCE-controlled LSPs.";
            }
            enum pending {
                value "3";
                description
                    "Too many pending LSP update requests.";
            }
            enum unacceptable {
                value "4";
                description
                    "Unacceptable parameters.";
            }
            enum internal {
                value "5";
                description
                    "Internal error.";
            }
            enum admin {
                value "6";
                description
                    "LSP administratively brought down.";
            }
            enum preempted {
                value "7";
```

```
                description
                    "LSP preempted.";
            }
            enum rsvp {
                value "8";
                description
                    "RSVP signaling error.";
            }
        }
        description
            "The LSP Error Codes.";
    }

    typedef sync-state {
        type enumeration {
            enum pending {
                value "0";
                description
                    "The state synchronization
                     has not started.";
            }
            enum ongoing {
                value "1";
                description
                    "The state synchronization
                     is ongoing.";
            }
            enum finished {
                value "2";
                description
                    "The state synchronization
                     is finished.";
            }
        }
        description
            "The LSP-DB state synchronization operational status.";
    }

        typedef pst{
                type enumeration{
                        enum rsvp-te{
                                value "0";
                                description
                                        "RSVP-TE signaling protocol";
                        }
                        enum sr{
                                value "1";
                                description
```

```
                                        "Segment Routing Traffic Engineering";
                        }
                }
                description
                        "The Path Setup Type";
        }

    typedef assoc-type{
        type enumeration{
            enum protection{
                value "1";
                description
                    "Path Protection Association Type";
            }
        }
        description
            "The PCEP Association Type";
    }

    /*
     * Features
     */

    feature svec {
        description
            "Support synchronized path computation.";
    }

    feature gmpls {
        description
            "Support GMPLS.";
    }

    feature obj-fn {
        description
            "Support OF as per RFC 5541.";
    }

    feature gco {
        description
            "Support GCO as per RFC 5557.";
    }

    feature pathkey {
        description
            "Support pathkey as per RFC 5520.";
    }
```

```
    feature p2mp {
        description
            "Support P2MP as per RFC 6006.";
    }

    feature stateful {
        description
            "Support stateful PCE.";
    }

    feature pce-initiated {
        description
            "Support PCE-Initiated LSP.";
    }

    feature tls {
        description
            "Support PCEP over TLS.";
    }

    feature sr {
        description
            "Support Segement Routing for PCE.";
    }

    /*
     * Groupings
     */



    grouping pcep-entity-info{
        description
            "This grouping defines the attributes for PCEP entity.";
        leaf connect-timer {
            type uint32 {
                range "1..65535";
            }
            units "seconds";
            default 60;
            description
                "The time in seconds that the PCEP entity will wait
                 to establish a TCP connection with a peer.  If a
                 TCP connection is not established within this time
                 then PCEP aborts the session setup attempt.";
            reference
                "RFC 5440: Path Computation Element (PCE)
                           Communication Protocol (PCEP)";
```

```
        }

        leaf connect-max-retry {
            type uint32;
            default 5;
            description
                "The maximum number of times the system tries to
                 establish a TCP connection to a peer before the
                 session with the peer transitions to the idle
                 state.";
            reference
                "RFC 5440: Path Computation Element (PCE)
                           Communication Protocol (PCEP)";
        }

        leaf init-backoff-timer {
            type uint32 {
                range "1..65535";
            }
            units "seconds";
            description
               "The initial back-off time in seconds for retrying
                a failed session setup attempt to a peer.
                The back-off time increases for each failed
                session setup attempt, until a maximum back-off
                time is reached.  The maximum back-off time is
                max-backoff-timer.";
        }

        leaf max-backoff-timer {
            type uint32;
            units "seconds";
            description
               "The maximum back-off time in seconds for retrying
                a failed session setup attempt to a peer.
                The back-off time increases for each failed session
                setup attempt, until this maximum value is reached.
                Session setup attempts then repeat periodically
                without any further increase in back-off time.";
        }

        leaf open-wait-timer {
            type uint32 {
                range "1..65535";
            }
            units "seconds";
            default 60;
            description
```

```
               "The time in seconds that the PCEP entity will wait
                to receive an Open message from a peer after the
                TCP connection has come up.
                If no Open message is received within this time then
                PCEP terminates the TCP connection and deletes the
                associated sessions.";
            reference
                "RFC 5440: Path Computation Element (PCE)
                           Communication Protocol (PCEP)";
        }

        leaf keep-wait-timer {
            type uint32 {
                range "1..65535";
            }
            units "seconds";
            default 60;
            description
                "The time in seconds that the PCEP entity will wait
                 to receive a Keepalive or PCErr message from a peer
                 during session initialization after receiving an
                 Open message.  If no Keepalive or PCErr message is
                 received within this time then PCEP terminates the
                 TCP connection and deletes the associated
                 sessions.";
            reference
                "RFC 5440: Path Computation Element (PCE)
                           Communication Protocol (PCEP)";
        }

        leaf keep-alive-timer {
            type uint32 {
                range "0..255";
            }
            units "seconds";
            default 30;
            description
                "The keep alive transmission timer that this PCEP
                 entity will propose in the initial OPEN message of
                 each session it is involved in.  This is the
                 maximum time between two consecutive messages sent
                 to a peer. Zero means that the PCEP entity prefers
                 not to send Keepalives at all.
                 Note that the actual Keepalive transmission
                 intervals, in either direction of an active PCEP
                 session, are determined by negotiation between the
                 peers as specified by RFC 5440, and so may differ
                 from this configured value.";
```

```
            reference
                "RFC 5440: Path Computation Element (PCE)
                           Communication Protocol (PCEP)";
        }

        leaf dead-timer {
            type uint32 {
                range "0..255";
            }
            units "seconds";
            must ". >= ../keep-alive-timer" {
                error-message "The dead timer must be "
                       + "larger than the keep alive timer";
                description
                    "This value MUST be greater than
                     keep-alive-timer.";

            }
            default 120;
            description
                "The dead timer that this PCEP entity will propose
                 in the initial OPEN message of each session it is
                 involved in. This is the time after which a peer
                 should declare a session down if it does not
                 receive any PCEP messages. Zero suggests that the
                 peer does not run a dead timer at all." ;
            reference
                "RFC 5440: Path Computation Element (PCE)
                           Communication Protocol (PCEP)";
        }

        leaf allow-negotiation{
            type boolean;
            description
                "Whether the PCEP entity will permit negotiation of
                 session parameters.";
        }

        leaf max-keep-alive-timer{
            type uint32 {
                range "0..255";
            }
            units "seconds";
            description
                "In PCEP session parameter negotiation in seconds,
                 the maximum value that this PCEP entity will
                 accept from a peer for the interval between
                 Keepalive transmissions. Zero means that the PCEP
```

```
                entity will allow no Keepalive transmission at
                all." ;
        }

        leaf max-dead-timer{
            type uint32 {
                range "0..255";
            }
            units "seconds";
            description
                "In PCEP session parameter negotiation in seconds,
                 the maximum value that this PCEP entity will accept
                 from a peer for the Dead timer.  Zero means that
                 the PCEP entity will allow not running a Dead
                 timer.";
        }

        leaf min-keep-alive-timer{
            type uint32 {
                range "0..255";
            }
            units "seconds";
            description
                "In PCEP session parameter negotiation in seconds,
                 the minimum value that this PCEP entity will
                 accept for the interval between Keepalive
                 transmissions. Zero means that the PCEP entity
                 insists on no Keepalive transmission at all.";
        }

        leaf min-dead-timer{
            type uint32 {
                range "0..255";
            }
            units "seconds";
            description
                 "In PCEP session parameter negotiation in
                  seconds, the minimum value that this PCEP entity
                  will accept for the Dead timer.  Zero means that
                  the PCEP entity insists on not running a Dead
                  timer.";
        }

        leaf sync-timer{
            if-feature svec;
            type uint32 {
                range "0..65535";
            }
```

```
            units "seconds";
            default 60;
            description
                "The value of SyncTimer in seconds is used in the
                 case of synchronized path computation request
                 using the SVEC object. Consider the case where a
                 PCReq message is received by a PCE that contains
                 the SVEC object referring to M synchronized path
                 computation requests.  If after the expiration of
                 the SyncTimer all the M path computation requests
                 have not been, received a protocol error is
                 triggered and the PCE MUST cancel the whole set
                 of path computation  requests.
                 The aim of the SyncTimer is to avoid the storage
                 of unused synchronized requests should one of
                 them get lost for some reasons (for example, a
                 misbehaving PCC).
                 Zero means that the PCEP entity does not use the
                 SyncTimer.";
            reference
                "RFC 5440: Path Computation Element (PCE)
                           Communication Protocol (PCEP)";
        }


        leaf request-timer{
            type uint32 {
                range "1..65535";
            }
            units "seconds";
            description
                "The maximum time that the PCEP entity will wait
                 for a response to a PCReq message.";
        }

        leaf max-sessions{
            type uint32;
            description
               "Maximum number of sessions involving this PCEP
                entity that can exist at any time.";
        }

        leaf max-unknown-reqs{
            type uint32;
            default 5;
            description
              "The maximum number of unrecognized requests and
               replies that any session on this PCEP entity is
```

```
                willing to accept per minute before terminating
                the session.
                A PCRep message contains an unrecognized reply
                if it contains an RP object whose request ID
                does not correspond to any in-progress request
                sent by this PCEP entity.
                A PCReq message contains an unrecognized request
                if it contains an RP object whose request ID is
                zero.";
            reference
                "RFC 5440: Path Computation Element (PCE)
                           Communication Protocol (PCEP)";
        }

        leaf max-unknown-msgs{
            type uint32;
            default 5;
            description
             "The maximum number of unknown messages that any
              session on this PCEP entity is willing to accept
              per minute before terminating the session.";
            reference
                "RFC 5440: Path Computation Element (PCE)
                           Communication Protocol (PCEP)";
        }

    }//pcep-entity-info

    grouping  pce-scope{
        description
            "This grouping defines PCE path computation scope
             information which maybe relevant to PCE selection.
             This information corresponds to PCE auto-discovery
             information.";
        reference
            "RFC 5088: OSPF Protocol Extensions for Path
                       Computation Element (PCE)
                       Discovery
             RFC 5089: IS-IS Protocol Extensions for Path
                       Computation Element (PCE)
                       Discovery";
        leaf intra-area-scope{
            type boolean;
            default true;
            description
                "PCE can compute intra-area paths.";
        }
        leaf intra-area-pref{
```

```
            type uint8{
                range "0..7";
            }
            description
              "The PCE's preference for intra-area TE LSP
              computation.";
        }
        leaf inter-area-scope{
            type boolean;
            default false;
            description
                "PCE can compute inter-area paths.";
        }
        leaf inter-area-scope-default{
            type boolean;
            default false;
            description
                "PCE can act as a default PCE for inter-area
                 path computation.";
        }
        leaf inter-area-pref{
            type uint8{
                range "0..7";
            }
            description
              "The PCE's preference for inter-area TE LSP
              computation.";
        }
        leaf inter-as-scope{
            type boolean;
            default false;
            description
                "PCE can compute inter-AS paths.";
        }
        leaf inter-as-scope-default{
            type boolean;
            default false;
            description
                "PCE can act as a default PCE for inter-AS
                 path computation.";
        }
        leaf inter-as-pref{
            type uint8{
                range "0..7";
            }
            description
              "The PCE's preference for inter-AS TE LSP
              computation.";
```

```
        }
        leaf inter-layer-scope{
            type boolean;
            default false;
            description
                "PCE can compute inter-layer paths.";
        }
        leaf inter-layer-pref{
            type uint8{
                range "0..7";
            }
            description
              "The PCE's preference for inter-layer TE LSP
              computation.";
        }
    }//pce-scope

    grouping  domain{
        description
            "This grouping specifies a Domain where the
             PCEP speaker has topology visibility.";
        leaf domain-type{
            type domain-type;
            description
              "The domain type.";
        }
        leaf domain{
            type domain;
            description
              "The domain Information.";
        }
    }//domain

    grouping capability{
        description
            "This grouping specifies a capability
             information of local PCEP entity. This maybe
             relevant to PCE selection as well. This
             information corresponds to PCE auto-discovery
             information.";
        reference
            "RFC 5088: OSPF Protocol Extensions for Path
                       Computation Element (PCE)
                       Discovery
             RFC 5089: IS-IS Protocol Extensions for Path
                       Computation Element (PCE)
                       Discovery";
        leaf gmpls{
```

```
            if-feature gmpls;
            type boolean;
            description
              "Path computation with GMPLS link
               constraints.";
          }
          leaf bi-dir{
            type boolean;
            description
              "Bidirectional path computation.";
          }
          leaf diverse{
            type boolean;
            description
              "Diverse path computation.";
          }
          leaf load-balance{
            type boolean;
            description
              "Load-balanced path computation.";
          }
          leaf synchronize{
            if-feature svec;
            type boolean;
            description
              "Synchronized paths computation.";
          }
          leaf objective-function{
            if-feature obj-fn;
            type boolean;
            description
              "Support for multiple objective functions.";
          }
          leaf add-path-constraint{
            type boolean;
            description
              "Support for additive path constraints (max
               hop count, etc.).";
          }
          leaf prioritization{
            type boolean;
            description
              "Support for request prioritization.";
          }
          leaf multi-request{
            type boolean;
            description
              "Support for multiple requests per message.";
```

```
        }
        leaf gco{
            if-feature gco;
            type boolean;
            description
              "Support for Global Concurrent Optimization
               (GCO).";
        }
        leaf p2mp{
            if-feature p2mp;
            type boolean;
            description
              "Support for P2MP path computation.";
        }

        container stateful{
            if-feature stateful;
            description
                "If stateful PCE feature is present";
            leaf enabled{
                type boolean;
                description
                    "Enabled or Disabled";
            }
            leaf active{
                type boolean;
                description
                  "Support for active stateful PCE.";
            }
            leaf pce-initiated{
                if-feature pce-initiated;
                type boolean;
                description
                  "Support for PCE-initiated LSP.";
            }
        }
                container sr{
                        if-feature sr;
                        description
                                "If segment routing is supported";
                        leaf enabled{
                type boolean;
                description
                    "Enabled or Disabled";
                        }
                        leaf msd{ /*should be in MPLS yang model (?)*/
                                type uint8;
                must "((../../role == 'pcc')" +
```

```
                     " or " +
                    "(../../role == 'pcc-and-pce')))"
                {
                    error-message
                        "The PCEP entity must be PCC";
                    description
                        "When PCEP entity is PCC for
                        MSD to be applicable";
                }
                                description
                                        "Maximum SID Depth";
                        }
                }
    }//capability

    grouping  info{
        description
            "This grouping specifies all information which
             maybe relevant to both PCC and PCE.
             This information corresponds to PCE auto-discovery
             information.";
        container domain{
            description
                "The local domain for the PCEP entity";
            list domain{
                key "domain-type domain";
                description
                    "The local domain.";
                uses domain{
                    description
                        "The local domain for the PCEP entity.";
                }
            }
        }
        container capability{
            description
                "The PCEP entity capability";
            uses capability{
                description
                    "The PCEP entity supported
                    capabilities.";
            }
        }
    }//info

    grouping  pce-info{
        description
            "This grouping specifies all PCE information
```

```
             which maybe relevant to the PCE selection.
             This information corresponds to PCE auto-discovery
             information.";
        container scope{
            description
                "The path computation scope";
            uses pce-scope;
        }

        container neigh-domains{
            description
                "The list of neighbour PCE-Domain
                 toward which a PCE can compute
                 paths";
            list domain{
                key "domain-type domain";

                description
                    "The neighbour domain.";
                uses domain{
                    description
                        "The PCE neighbour domain.";
                }
            }
        }
    }//pce-info

    grouping  pcep-stats{
        description
            "This grouping defines statistics for PCEP. It is used
             for both peer and current session.";
        leaf avg-rsp-time{
            type uint32;
            units "milliseconds";
            must "(/pcep-state/entity/peers/peer/role != 'pcc'" +
                 " or " +
                 "(/pcep-state/entity/peers/peer/role = 'pcc'" +
                 " and avg-rsp-time = 0))" {
                error-message
                    "Invalid average response time";
                description
                    "If role is pcc then this leaf is meaningless
                     and is set to zero.";
            }
            description
              "The average response time.
               If an average response time has not been
               calculated then this leaf has the value zero.";
```

```
        }

        leaf lwm-rsp-time{
            type uint32;
            units "milliseconds";
            must "(/pcep-state/entity/peers/peer/role != 'pcc'" +
                 " or " +
                 "(/pcep-state/entity/peers/peer/role = 'pcc'" +
                 " and lwm-rsp-time = 0))" {
                error-message
                    "Invalid smallest (low-water mark)
                     response time";
                description
                    "If role is pcc then this leaf is meaningless
                     and is set to zero.";
            }
            description
             "The smallest (low-water mark) response time seen.
              If no responses have been received then this
              leaf has the value zero.";
        }

        leaf hwm-rsp-time{
            type uint32;
            units "milliseconds";
            must "(/pcep-state/entity/peers/peer/role != 'pcc'" +
                 " or " +
                 "(/pcep-state/entity/peers/peer/role = 'pcc'" +
                 " and hwm-rsp-time = 0))" {
                error-message
                    "Invalid greatest (high-water mark)
                     response time seen";
                description
                    "If role is pcc then this field is
                     meaningless and is set to zero.";
            }
            description
             "The greatest (high-water mark) response time seen.
              If no responses have been received then this object
              has the value zero.";
        }

        leaf num-pcreq-sent{
            type yang:counter32;
            description
             "The number of PCReq messages sent.";
        }
```

```
        leaf num-pcreq-rcvd{
            type yang:counter32;
            description
              "The number of PCReq messages received.";
        }

        leaf num-pcrep-sent{
            type yang:counter32;
            description
              "The number of PCRep messages sent.";
        }

        leaf num-pcrep-rcvd{
            type yang:counter32;
            description
              "The number of PCRep messages received.";
        }

        leaf num-pcerr-sent{
            type yang:counter32;
            description
              "The number of PCErr messages sent.";
        }

        leaf num-pcerr-rcvd{
            type yang:counter32;
            description
              "The number of PCErr messages received.";
        }

        leaf num-pcntf-sent{
            type yang:counter32;
            description
              "The number of PCNtf messages sent.";
        }

        leaf num-pcntf-rcvd{
            type yang:counter32;
            description
              "The number of PCNtf messages received.";
        }

        leaf num-keepalive-sent{
            type yang:counter32;
            description
              "The number of Keepalive messages sent.";
        }
```

```
        leaf num-keepalive-rcvd{
            type yang:counter32;
            description
              "The number of Keepalive messages received.";
        }

        leaf num-unknown-rcvd{
            type yang:counter32;
            description
              "The number of unknown messages received.";
        }

        leaf num-corrupt-rcvd{
            type yang:counter32;
            description
              "The number of corrupted PCEP message received.";
        }

        leaf num-req-sent{
            type yang:counter32;
            description
              "The number of requests sent.  A request corresponds
               1:1 with an RP object in a PCReq message. This might
               be greater than num-pcreq-sent because multiple
               requests can be batched into a single PCReq
               message.";
        }

        leaf num-req-sent-pend-rep{
            type yang:counter32;
            description
              "The number of requests that have been sent for
               which a response is still pending.";
        }

        leaf num-req-sent-ero-rcvd{
            type yang:counter32;
            description
               "The number of requests that have been sent for
                which a response with an ERO object was received.
                Such responses indicate that a path was
                successfully computed by the peer.";
        }

        leaf num-req-sent-nopath-rcvd{
            type yang:counter32;
            description
              "The number of requests that have been sent for
```

```
            which a response with a NO-PATH object was
            received. Such responses indicate that the peer
            could not  find a path to satisfy the
            request.";
        }

        leaf num-req-sent-cancel-rcvd{
            type yang:counter32;
            description
              "The number of requests that were cancelled with
               a PCNtf message.
               This might be different than num-pcntf-rcvd because
               not all PCNtf messages are used to cancel requests,
               and a single PCNtf message can cancel multiple
               requests.";
        }

        leaf num-req-sent-error-rcvd{
            type yang:counter32;
            description
             "The number of requests that were rejected with a
              PCErr message.
              This might be different than num-pcerr-rcvd because
              not all PCErr messages are used to reject requests,
              and a single PCErr message can reject multiple
              requests.";
        }

        leaf num-req-sent-timeout{
            type yang:counter32;
            description
             "The number of requests that have been sent to a peer
             and have been abandoned because the peer has taken too
             long to respond to them.";
        }

        leaf num-req-sent-cancel-sent{
            type yang:counter32;
            description
             "The number of requests that were sent to the peer and
              explicitly cancelled by the local PCEP entity sending
              a PCNtf.";
        }

        leaf num-req-rcvd{
            type yang:counter32;
            description
             "The number of requests received.  A request
```

```
              corresponds 1:1 with an RP object in a PCReq
              message.
              This might be greater than num-pcreq-rcvd because
              multiple requests can be batched into a single
              PCReq message.";
        }

        leaf num-req-rcvd-pend-rep{
            type yang:counter32;
            description
             "The number of requests that have been received for
              which a response is still pending.";
        }

        leaf num-req-rcvd-ero-sent{
            type yang:counter32;
            description
             "The number of requests that have been received for
              which a response with an ERO object was sent.  Such
              responses indicate that a path was successfully
              computed by the local PCEP entity.";
        }

        leaf num-req-rcvd-nopath-sent{
            type yang:counter32;
            description
             "The number of requests that have been received for
              which a response with a NO-PATH object was sent. Such
              responses indicate that the local PCEP entity could
              not find a path to satisfy the request.";
        }

        leaf num-req-rcvd-cancel-sent{
            type yang:counter32;
            description
             "The number of requests received that were cancelled
              by the local PCEP entity sending a PCNtf message.
              This might be different than num-pcntf-sent because
              not all PCNtf messages are used to cancel requests,
              and a single PCNtf message can cancel multiple
              requests.";
        }

        leaf num-req-rcvd-error-sent{
            type yang:counter32;
            description
             "The number of requests received that were cancelled
              by the local PCEP entity sending a PCErr message.
```

```
              This might be different than num-pcerr-sent because
              not all PCErr messages are used to cancel requests,
              and a single PCErr message can cancel multiple
              requests.";
        }

        leaf  num-req-rcvd-cancel-rcvd{
            type yang:counter32;
            description
             "The number of requests that were received from the
              peer and explicitly cancelled by the peer sending
              a PCNtf.";
        }

        leaf  num-rep-rcvd-unknown{
            type yang:counter32;
            description
              "The number of responses to unknown requests
               received. A response to an unknown request is a
               response whose RP object does not contain the
               request ID of any request that is currently
               outstanding on the session.";
        }

        leaf  num-req-rcvd-unknown{
            type yang:counter32;
            description
              "The number of unknown requests that have been
               received. An unknown request is a request
               whose RP object contains a request ID of
               zero.";
        }

        container svec{
            if-feature svec;
            description
                "If synchronized path computation is supported";
            leaf num-svec-sent{
                type yang:counter32;
                description
                  "The number of SVEC objects sent in PCReq messages.
                   An SVEC object represents a set of synchronized
                   requests.";
            }

            leaf num-svec-req-sent{
                type yang:counter32;
                description
```

```
                  "The number of requests sent that appeared in one
                   or more SVEC objects.";
            }

            leaf num-svec-rcvd{
                type yang:counter32;
                description
                 "The number of SVEC objects received in PCReq
                  messages. An SVEC object represents a set of
                  synchronized requests.";
            }

            leaf num-svec-req-rcvd{
                type yang:counter32;
                description
                  "The number of requests received that appeared
                   in one or more SVEC objects.";
            }
        }
        container stateful{
            if-feature stateful;
            description
                "Stateful PCE related statistics";
            leaf num-pcrpt-sent{
                type yang:counter32;
                description
                 "The number of PCRpt messages sent.";
            }

            leaf num-pcrpt-rcvd{
                type yang:counter32;
                description
                  "The number of PCRpt messages received.";
            }

            leaf num-pcupd-sent{
                type yang:counter32;
                description
                 "The number of PCUpd messages sent.";
            }

            leaf num-pcupd-rcvd{
                type yang:counter32;
                description
                  "The number of PCUpd messages received.";
            }

            leaf num-rpt-sent{
```

```
                type yang:counter32;
                description
                  "The number of LSP Reports sent.  A LSP report
                   corresponds 1:1 with an LSP object in a PCRpt
                   message. This might be greater than
                   num-pcrpt-sent because multiple reports can
                   be batched into a single PCRpt message.";
            }

            leaf num-rpt-rcvd{
                type yang:counter32;
                description
                 "The number of LSP Reports received.  A LSP report
                  corresponds 1:1 with an LSP object in a PCRpt
                  message.
                  This might be greater than num-pcrpt-rcvd because
                  multiple reports can be batched into a single
                  PCRpt message.";
            }

            leaf  num-rpt-rcvd-error-sent{
                type yang:counter32;
                description
                  "The number of reports of LSPs received that were
                  responded by the local PCEP entity by sending a
                  PCErr message.";
            }

            leaf num-upd-sent{
                type yang:counter32;
                description
                  "The number of LSP updates sent.  A LSP update
                   corresponds 1:1 with an LSP object in a PCUpd
                   message. This might be greater than
                   num-pcupd-sent because multiple updates can
                   be batched into a single PCUpd message.";
            }

            leaf num-upd-rcvd{
                type yang:counter32;
                description
                 "The number of LSP Updates received.  A LSP update
                  corresponds 1:1 with an LSP object in a PCUpd
                  message.
                  This might be greater than num-pcupd-rcvd because
                  multiple updates can be batched into a single
                  PCUpd message.";
            }
```

```
            leaf  num-upd-rcvd-unknown{
                type yang:counter32;
                description
                  "The number of updates to unknown LSPs
                   received. An update to an unknown LSP is a
                   update whose LSP object does not contain the
                   PLSP-ID of any LSP that is currently
                   present.";
            }

            leaf  num-upd-rcvd-undelegated{
                type yang:counter32;
                description
                  "The number of updates to not delegated LSPs
                   received. An update to an undelegated LSP is a
                   update whose LSP object does not contain the
                   PLSP-ID of any LSP that is currently
                   delegated to current PCEP session.";
            }

            leaf  num-upd-rcvd-error-sent{
                type yang:counter32;
                description
                  "The number of updates to LSPs received that were
                  responded by the local PCEP entity by sending a
                  PCErr message.";
            }
            container initiation {
                if-feature pce-initiated;
                description
                    "PCE-Initiated related statistics";
                leaf num-pcinitiate-sent{
                    type yang:counter32;
                    description
                     "The number of PCInitiate messages sent.";
                }

                leaf num-pcinitiate-rcvd{
                    type yang:counter32;
                    description
                      "The number of PCInitiate messages received.";
                }

                leaf num-initiate-sent{
                    type yang:counter32;
                    description
                      "The number of LSP Initiation sent via PCE.
                       A LSP initiation corresponds 1:1 with an LSP
```

```
                        object in a PCInitiate message. This might be
                        greater than num-pcinitiate-sent because
                        multiple initiations can be batched into a
                        single PCInitiate message.";
                }

                leaf num-initiate-rcvd{
                    type yang:counter32;
                    description
                      "The number of LSP Initiation received from
                       PCE.  A LSP initiation corresponds 1:1 with
                       an LSP object in a PCInitiate message. This
                       might be greater than num-pcinitiate-rcvd
                       because multiple initiations can be batched
                       into a single PCInitiate message.";
                }

                leaf  num-initiate-rcvd-error-sent{
                    type yang:counter32;
                    description
                      "The number of initiations of LSPs received
                       that were responded by the local PCEP entity
                       by sending a PCErr message.";
                }
            }
        }
    }//pcep-stats

    grouping lsp-state{
        description
            "This grouping defines the attributes for LSP in LSP-DB.
             These are the attributes specifically from the PCEP
             perspective";
        leaf plsp-id{
            type uint32{
                range "1..1048575";
            }
            description
                "A PCEP-specific identifier for the LSP.  A PCC
                 creates a unique PLSP-ID for each LSP that is
                 constant for the lifetime of a PCEP session.
                 PLSP-ID is 20 bits with 0 and 0xFFFFF are
                 reserved";
        }
        leaf pcc-id{
            type inet:ip-address;
            description
                "The local internet address of the PCC, that
```

```
                generated the PLSP-ID.";
        }

        container lsp-ref{
            description
                "reference to ietf-te lsp state";

            leaf source {
                type leafref {
                    path "/te:te/te:lsps-state/te:lsp/te:source";
                }
                description
                  "Tunnel sender address extracted from
                  SENDER_TEMPLATE  object";
                reference "RFC3209";
            }
            leaf destination {
                type leafref {
                    path "/te:te/te:lsps-state/te:lsp/te:"
                         + "destination";
                }
                description
                    "Tunnel endpoint address extracted from
                    SESSION object";
                reference "RFC3209";
            }
            leaf tunnel-id {
                type leafref {
                    path "/te:te/te:lsps-state/te:lsp/te:tunnel-id";
                }
                description
                    "Tunnel identifier used in the SESSION
                    that remains constant over the life
                    of the tunnel.";
                reference "RFC3209";
            }
            leaf lsp-id {
                type leafref {
                    path "/te:te/te:lsps-state/te:lsp/te:lsp-id";
                }
                description
                    "Identifier used in the SENDER_TEMPLATE
                    and the FILTER_SPEC that can be changed
                    to allow a sender to share resources with
                    itself.";
                reference "RFC3209";
            }
            leaf extended-tunnel-id {
```

```
                type leafref {
                    path "/te:te/te:lsps-state/te:lsp/te:"
                    + "extended-tunnel-id";
                }
                description
                    "Extended Tunnel ID of the LSP.";
                reference "RFC3209";
            }
            leaf type {
                type leafref {
                    path "/te:te/te:lsps-state/te:lsp/te:type";
                }
                description "LSP type P2P or P2MP";
            }
        }

        leaf admin-state{
            type boolean;
            description
                "The desired operational state";
        }
        leaf operational-state{
            type operational-state;
            description
                "The operational status of the LSP";
        }
        container delegated{
            description
                "The delegation related parameters";
            leaf enabled{
                type boolean;
                description
                    "LSP is delegated or not";
            }
            leaf pce{
                type leafref {
                    path "/pcep-state/entity/peers/peer/addr";
                }
                must "((../enabled == true)" +
                     " and " +
                     "((../../role == 'pcc')" +
                     " or " +
                    "(../../role == 'pcc-and-pce')))"
                {
                    error-message
                        "The PCEP entity must be PCC
                         and the LSP be delegated";
                    description
```

```
                    "When PCEP entity is PCC for
                    delegated LSP";
                }
                description
                    "The reference to the PCE peer to
                    which LSP is delegated";
            }
            leaf srp-id{
                type uint32;
                description
                    "The last SRP-ID-number associated with this
                    LSP.";
            }
        }
        container initiation {
            if-feature pce-initiated;
            description
                "The PCE initiation related parameters";
            leaf enabled{
                type boolean;
                description
                    "LSP is PCE-initiated or not";
            }
            leaf pce{
                type leafref {
                    path "/pcep-state/entity/peers/peer/addr";
                }
                must "(../enabled == true)"
                {
                    error-message
                        "The LSP must be PCE-Initiated";
                    description
                        "When the LSP must be PCE-Initiated";
                }
                description
                    "The reference to the PCE
                    that initiated this LSP";
            }
        }
        leaf symbolic-path-name{
            type string;
            description
                "The symbolic path name associated with the LSP.";
        }
        leaf last-error{
            type lsp-error;
            description
                "The last error for the LSP.";
```

```
        }
                leaf pst{
                        type pst;
                        default "rsvp-te";
                        description
                                "The Path Setup Type";

                }

    }//lsp-state

    grouping notification-instance-hdr {
        description
            "This group describes common instance specific data
             for notifications.";

        leaf peer-addr {
            type leafref {
                path "/pcep-state/entity/peers/peer/addr";
            }
            description
                "Reference to peer address";
        }

    }// notification-instance-hdr

    grouping notification-session-hdr {
        description
        "This group describes common session instance specific
         data for notifications.";

        leaf session-initiator {
            type leafref {
            path "/pcep-state/entity/peers/peer/sessions/" +
                 "session/initiator";
            }
            description
                "Reference to pcep session initiator leaf";
        }
    }// notification-session-hdr

    grouping stateful-pce-parameter {
        description
        "This group describes stateful PCE specific
         parameters.";
        leaf state-timeout{
            type uint32;
            units "seconds";
```

```
            description
                "When a PCEP session is terminated, a PCC
                 waits for this time period before flushing
                 LSP state associated with that PCEP session
                 and reverting to operator-defined default
                 parameters or behaviours.";
        }
        leaf redelegation-timeout{
            type uint32;
            units "seconds";
            must "((../role == 'pcc')" +
                 " or " +
                 "(../role == 'pcc-and-pce'))"
            {
                error-message "The PCEP entity must be PCC";
                description
                    "When PCEP entity is PCC";
            }
            description
                "When a PCEP session is terminated, a PCC
                 waits for this time period before revoking
                 LSP delegation to a PCE and attempting to
                 redelegate LSPs associated with the
                 terminated PCEP session to an alternate
                 PCE.";
        }
        leaf rpt-non-pcep-lsp{
            type boolean;
            must "((../role == 'pcc')" +
                 " or " +
                 "(../role == 'pcc-and-pce'))"
            {
                error-message "The PCEP entity must be PCC";
                description
                    "When PCEP entity is PCC";
            }
            description
                "If set, a PCC reports LSPs that are not
                controlled by any PCE (for example, LSPs
                that are statically configured at the
                PCC). ";
        }

    }

    grouping authentication {
        description "Authentication Information";
        choice auth-type-selection {
```

```
            description
                "Options for expressing authentication setting.";
            case auth-key-chain {
                leaf key-chain {
                    type key-chain:key-chain-ref;
                    description
                        "key-chain name.";
                }
            }
            case auth-key {
                leaf key {
                    type string;
                description
                    "Key string in ASCII format.";
                }
                container crypto-algorithm {
                    uses key-chain:crypto-algorithm-types;
                        description
                            "Cryptographic algorithm associated
                             with key.";
                }
            }
            case auth-tls {
                if-feature tls;
                container tls {
                    description
                        "TLS related information - TBD";
                }
            }
        }
    }

    grouping association {
        description
            "Generic Association parameters";
        leaf type {
            type "assoc-type";
            description
                "The PCEP association type";
        }
        leaf id {
            type uint16;
            description
                "PCEP Association ID";
        }
        leaf source {
          type inet:ip-address;
          description
```

```
                "PCEP Association Source.";
        }
        leaf global-source {
          type uint32;
          description
                "PCEP Association Global
                Source.";
        }
        leaf extended-id{
            type string;
            description
                "Additional information to
                support unique identification.";
        }
    }
    grouping association-ref {
        description
            "Generic Association parameters";
        leaf id {
            type leafref {
                path "/pcep-state/entity/lsp-db/"
                + "association-list/id";
            }
            description
                "PCEP Association ID";
        }
        leaf source {
            type leafref {
                path "/pcep-state/entity/lsp-db/"
                + "association-list/source";
            }
          description
                "PCEP Association Source.";
        }
        leaf global-source {
            type leafref {
                path "/pcep-state/entity/lsp-db/"
                + "association-list/global-source";
            }
          description
                "PCEP Association Global
                Source.";
        }
        leaf extended-id{
            type leafref {
                path "/pcep-state/entity/lsp-db/"
                + "association-list/extended-id";
            }
```

```
            description
                "Additional information to
                support unique identification.";
        }
    }
    /*
     * Configuration data nodes
     */
    container pcep{

        presence
            "The PCEP is enabled";

            description
            "Parameters for list of configured PCEP entities
             on the device.";

        container entity {

            description
                "The configured PCEP entity on the device.";

            leaf addr {
                type inet:ip-address;
                mandatory true;
                description
                    "The local Internet address of this PCEP
                    entity.
                    If operating as a PCE server, the PCEP
                    entity listens on this address.
                    If operating as a PCC, the PCEP entity
                    binds outgoing TCP connections to this
                    address.
                    It is possible for the PCEP entity to
                    operate both as a PCC and a PCE Server, in
                    which case it uses this address both to
                    listen for incoming TCP connections and to
                    bind outgoing TCP connections.";
            }

            leaf enabled {
                type boolean;
                default true;
                description
                    "The administrative status of this PCEP
                     Entity.";
            }
```

```
            leaf role {
                type pcep-role;
                mandatory true;
                description
                    "The role that this entity can play.
                     Takes one of the following values.
                     - unknown(0): this PCEP Entity role is not
                       known.
                     - pcc(1): this PCEP Entity is a PCC.
                     - pce(2): this PCEP Entity is a PCE.
                     - pcc-and-pce(3): this PCEP Entity is both
                       a PCC and a PCE.";
            }

            leaf description {
                type string;
                description
                    "Description of the PCEP entity configured
                     by the user";
            }

            uses info {
                description
                    "Local PCEP entity information";
            }


            container pce-info {
                must "((../role == 'pce')" +
                     " or " +
                     "(../role == 'pcc-and-pce'))"
                {
                    error-message "The PCEP entity must be PCE";
                    description
                        "When PCEP entity is PCE";
                }
                uses pce-info {
                    description
                        "Local PCE information";
                }
                                uses authentication {
                                        description
                                                "Local PCE authentication inform
ation";
                                }


                description
                    "The Local PCE Entity PCE information";
```

```
            }

            uses pcep-entity-info {
                description
                    "The configuration related to the PCEP
                     entity.";
            }



            leaf pcep-notification-max-rate {
                type uint32;
                mandatory true;
                description
                    "This variable indicates the maximum number of
                     notifications issued per second. If events occur
                     more rapidly, the implementation may simply fail
                     to emit these notifications during that period,
                     or may queue them until an appropriate time. A
                     value of 0 means no notifications are emitted
                     and all should be discarded (that is, not
                     queued).";
            }

            container stateful-parameter{
                if-feature stateful;
                must "(../info/capability/stateful/active == true)"
                {
                    error-message
                        "The Active Stateful PCE must be enabled";
                    description
                        "When PCEP entity is active stateful
                         enabled";
                }
                uses stateful-pce-parameter;

                description
                    "The configured stateful parameters";
            }


            container peers{
                must "((../role == 'pcc')" +
                     " or " +
                     "(../role == 'pcc-and-pce'))"
                {
                    error-message
                        "The PCEP entity must be PCC";
```

```
                    description
                        "When PCEP entity is PCC, as remote
                         PCE peers are configured.";
                }
                description
                    "The list of configured peers for the
                     entity (remote PCE)";
                list peer{
                    key "addr";

                    description
                        "The peer configured for the entity.
                         (remote PCE)";

                    leaf addr {
                        type inet:ip-address;
                        description
                            "The local Internet address of this
                             PCEP peer.";
                    }

                    leaf description {
                        type string;
                        description
                            "Description of the PCEP peer
                             configured by the user";
                    }
                    uses info {
                        description
                            "PCE Peer information";
                    }
                    uses pce-info {
                        description
                            "PCE Peer information";
                    }

                    leaf delegation-pref{
                        if-feature stateful;
                        type uint8{
                            range "0..7";
                        }
                        must "(../../info/capability/stateful/active"
                             + "== true)"
                        {
                            error-message
                                "The Active Stateful PCE must be
                                 enabled";
                            description
```

```
                                "When PCEP entity is active stateful
                                 enabled";
                            }
                            description
                                "The PCE peer delegation preference.";
                        }
                        uses authentication {
                            description
                                "PCE Peer authentication";
                        }
                    }//peer
                }//peers
            }//entity
        }//pcep

        /*
         * Operational data nodes
         */

        container pcep-state{
            config false;
            description
                "The list of operational PCEP entities on the
                 device.";

            container entity{
                description
                    "The operational PCEP entity on the device.";

                leaf addr {
                    type inet:ip-address;
                    description
                        "The local Internet address of this PCEP
                        entity.
                        If operating as a PCE server, the PCEP
                        entity listens on this address.
                        If operating as a PCC, the PCEP entity
                        binds outgoing TCP connections to this
                        address.
                        It is possible for the PCEP entity to
                        operate both as a PCC and a PCE Server, in
                        which case it uses this address both to
                        listen for incoming TCP connections and to
                        bind outgoing TCP connections.";
                }

                leaf index{
                    type uint32;
```

```
            description
                "The index of the operational PECP
                 entity";
        }


        leaf admin-status {
            type pcep-admin-status;
            description
                "The administrative status of this PCEP Entity.
                 This is the desired operational status as
                 currently set by an operator or by default in
                 the implementation.  The value of enabled
                 represents the current status of an attempt
                 to reach this desired status.";
        }

        leaf oper-status {
            type pcep-admin-status;
            description
               "The operational status of the PCEP entity.
                Takes one of the following values.
                - oper-status-up(1): the PCEP entity is
                  active.
                - oper-status-down(2): the PCEP entity is
                  inactive.
                - oper-status-going-up(3): the PCEP entity is
                  activating.
                - oper-status-going-down(4): the PCEP entity is
                  deactivating.
                - oper-status-failed(5): the PCEP entity has
                  failed and will recover when possible.
                - oper-status-failed-perm(6): the PCEP entity
                  has failed and will not recover without
                  operator intervention.";
        }

        leaf role {
            type pcep-role;
            description
                "The role that this entity can play.
                 Takes one of the following values.
                 - unknown(0): this PCEP entity role is
                   not known.
                 - pcc(1): this PCEP entity is a PCC.
                 - pce(2): this PCEP entity is a PCE.
                 - pcc-and-pce(3): this PCEP entity is
                   both a PCC and a PCE.";
```

```
            }

            uses info {
                description
                    "Local PCEP entity information";
            }

            container pce-info {
                when "((../role == 'pce')" +
                     " or " +
                     "(../role == 'pcc-and-pce'))"
                {
                    description
                        "When PCEP entity is PCE";
                }
                uses pce-info {
                    description
                        "Local PCE information";
                }
                                uses authentication {
                                        description
                                                "Local PCE authentication inform
ation";
                                }
                description
                    "The Local PCE Entity PCE information";
            }

            uses pcep-entity-info{
                description
                    "The operational information related to the
                     PCEP entity.";
            }

            container stateful-parameter{
                if-feature stateful;
                must "(../info/capability/stateful/active == true)"
                {
                    error-message
                        "The Active Stateful PCE must be enabled";
                    description
                        "When PCEP entity is active stateful
                         enabled";
                }
                uses stateful-pce-parameter;

                description
                    "The operational stateful parameters";
            }
```

```
            container lsp-db{
                if-feature stateful;
                description
                    "The LSP-DB";
                list association-list {
                    key "id source global-source extended-id";
                    description
                        "List of all PCEP associations";
                    uses association {
                        description
                            "The Association attributes";
                    }
                    list lsp {
                        key "plsp-id pcc-id";
                        description
                            "List of all LSP in this association";
                        leaf plsp-id {
                            type leafref {
                                path "/pcep-state/entity/lsp-db/"
                                + "lsp/plsp-id";
                            }
                            description
                                "Reference to PLSP-ID in LSP-DB";
                        }
                        leaf pcc-id {
                            type leafref {
                                path "/pcep-state/entity/lsp-db/"
                                + "lsp/pcc-id";
                            }
                            description
                                "Reference to PCC-ID in LSP-DB";
                        }
                    }
                }
                list lsp{
                    key "plsp-id pcc-id";
                    description
                        "List of all LSPs in LSP-DB";
                    uses lsp-state{
                        description
                            "The PCEP specific attributes for
                             LSP-DB.";
                    }
                    list association-list {
                        key "id source global-source extended-id";
                        description
                            "List of all PCEP associations";
                        uses association-ref {
```

```
                        description
                            "Reference to the Association
                             attributes";
                    }
                }

            }
        }
        container peers{
            description
                    "The list of peers for the entity";

            list peer{
                key "addr";

                description
                    "The peer for the entity.";

                leaf addr {
                    type inet:ip-address;
                    description
                        "The local Internet address of this PCEP
                         peer.";
                }

                leaf role {
                    type pcep-role;
                    description
                        "The role of the PCEP Peer.
                         Takes one of the following values.
                         - unknown(0): this PCEP peer role
                           is not known.
                         - pcc(1): this PCEP peer is a PCC.
                         - pce(2): this PCEP peer is a PCE.
                         - pcc-and-pce(3): this PCEP peer
                           is both a PCC and a PCE.";
                }


                uses info {
                    description
                        "PCEP peer information";
                }


                container pce-info {
                    when "((../role == 'pce')" +
                    " or " +
```

```
            "(../role == 'pcc-and-pce'))"
            {
                description
                    "When PCEP entity is PCE";
            }
            uses pce-info {
                description
                    "PCE Peer information";
            }
        description
            "The PCE Peer information";
        }

        leaf delegation-pref{
            if-feature stateful;
            type uint8{
                range "0..7";
            }
            must "((../../role == 'pcc')" +
                 " or " +
                 "(../../role == 'pcc-and-pce'))"
            {
                error-message
                    "The PCEP entity must be PCC";
                description
                    "When PCEP entity is PCC";
            }
            must "(../../info/capability/stateful/active"
                 + " == true)"
            {
                error-message
                    "The Active Stateful PCE must be
                     enabled";
                description
                    "When PCEP entity is active stateful
                     enabled";
            }
            description
                "The PCE peer delegation preference.";
        }

        uses authentication {
            description
                "PCE Peer authentication";
        }

        leaf discontinuity-time {
            type yang:timestamp;
```

```
            description
                "The timestamp of the time when the
                 information and statistics were
                 last reset.";
        }

        leaf initiate-session {
            type boolean;
            description
                "Indicates whether the local PCEP
                 entity initiates sessions to this peer,
                 or waits for the peer to initiate a
                 session.";
        }

        leaf session-exists{
            type boolean;
            description
                "Indicates whether a session with
                 this peer currently exists.";
        }

        leaf num-sess-setup-ok{
            type yang:counter32;
            description
                "The number of PCEP sessions successfully
                 successfully established with the peer,
                 including any current session.  This
                 counter is incremented each time a
                 session with this peer is successfully
                 established.";
        }

        leaf num-sess-setup-fail{
            type yang:counter32;
            description
               "The number of PCEP sessions with the peer
                that have been attempted but failed
                before being fully established. This
                counter is incremented each time a
                session retry to this peer fails.";
        }

        leaf session-up-time{
            type yang:timestamp;
            must "(../num-sess-setup-ok != 0  or " +
                "(../num-sess-setup-ok = 0  and "  +
                "session-up-time = 0))" {
```

```
                    error-message
                        "Invalid Session Up timestamp";
                    description
                        "If num-sess-setup-ok is zero,
                         then this leaf contains zero.";
                }
            description
               "The timestamp value of the last time a
                session with this peer was successfully
                established.";
        }

        leaf session-fail-time{
            type yang:timestamp;
            must "(../num-sess-setup-fail != 0  or " +
                "(../num-sess-setup-fail = 0  and "  +
                "session-fail-time = 0))" {
                    error-message
                        "Invalid Session Fail timestamp";
                    description
                        "If num-sess-setup-fail is zero,
                         then this leaf contains zero.";
                }
            description
              "The timestamp value of the last time a
               session with this peer failed to be
               established.";
        }

        leaf session-fail-up-time{
            type yang:timestamp;
            must "(../num-sess-setup-ok != 0  or "     +
                "(../num-sess-setup-ok = 0  and "     +
                "session-fail-up-time = 0))" {
                    error-message
                        "Invalid Session Fail from
                         Up timestamp";
                    description
                        "If num-sess-setup-ok is zero,
                         then this leaf contains zero.";
                }
            description
              "The timestamp value of the last time a
               session with this peer failed from
               active.";
        }

        container pcep-stats {
```

```
                description
                    "The container for all statistics at peer
                     level.";
                uses pcep-stats{
                    description
                        "Since PCEP sessions can be
                        ephemeral, the peer statistics tracks
                        a peer even when no PCEP session
                        currently exists to that peer. The
                        statistics contained are an aggregate
                        of the statistics for all successive
                        sessions to that peer.";
                }

                leaf num-req-sent-closed{
                    type yang:counter32;
                    description
                        "The number of requests that were
                         sent to the peer and implicitly
                         cancelled when the session they were
                         sent over was closed.";
                }

                leaf  num-req-rcvd-closed{
                    type yang:counter32;
                    description
                        "The number of requests that were
                         received from the peer and
                         implicitly cancelled when the
                         session they were received over
                         was closed.";
                }
            }//pcep-stats



            container sessions {
                description
                    "This entry represents a single PCEP
                     session in which the local PCEP entity
                     participates.
                     This entry exists only if the
                     corresponding PCEP session has been
                     initialized by some event, such as
                     manual user configuration, auto-
                     discovery of a peer, or an incoming
                     TCP connection.";
```

```
                    list session {
                        key "initiator";

                        description
                            "The list of sessions, note that
                             for a time being two sessions
                             may exist for a peer";

                        leaf initiator {
                            type pcep-initiator;
                            description
                                "The initiator of the session,
                                 that is, whether the TCP
                                 connection was initiated by
                                 the local PCEP entity or the
                                 peer.
                                 There is a window during
                                 session initialization where
                                 two sessions can exist between
                                 a pair of PCEP speakers, each
                                 initiated by one of the
                                 speakers. One of these
                                 sessions is always discarded
                                 before it leaves OpenWait state.
                                 However, before it is discarded,
                                 two sessions to the given peer
                                 appear transiently in this MIB
                                 module. The sessions are
                                 distinguished by who initiated
                                 them, and so this field is the
                                 key.";
                        }

                        leaf state-last-change {
                            type yang:timestamp;
                            description
                                "The timestamp value at the
                                 time this session entered its
                                 current state as denoted by
                                 the state leaf.";
                        }

                        leaf state {
                            type pcep-sess-state;
                            description
                                "The current state of the
                                 session.
                                 The set of possible states
```

```
                excludes the idle state since
                entries do not exist in the
                idle state.";
            }

            leaf session-creation {
                type yang:timestamp;
                description
                    "The timestamp value at the
                     time this session was
                     created.";
            }

            leaf connect-retry {
                type yang:counter32;
                description
                     "The number of times that the
                      local PCEP entity has
                      attempted to establish a TCP
                      connection for this session
                      without success. The PCEP
                      entity gives up when this
                      reaches connect-max-retry.";
            }

            leaf local-id {
                type uint32 {
                  range "0..255";
                }
                description
                     "The value of the PCEP session
                      ID used by the local PCEP
                      entity in the Open message
                      for this session.
                      If state is tcp-pending then
                      this is the session ID that
                      will be used in the Open
                      message. Otherwise, this is
                      the session ID that was sent
                      in the Open message.";
            }

            leaf remote-id {
                type uint32 {
                    range "0..255";
                }
                must "((../state != 'tcp-pending'" +
                     "and " +
```

```
                    "../state != 'open-wait' )" +
                    "or " +
                    "((../state = 'tcp-pending'" +
                    " or " +
                    "../state = 'open-wait' )" +
                    "and remote-id = 0))" {
                       error-message
                           "Invalid remote-id";
                       description
                            "If state is tcp-pending
                             or open-wait then this
                             leaf is not used and
                             MUST be set to zero.";
                    }
                description
                    "The value of the PCEP session
                     ID used by the peer in its
                     Open message for this
                     session.";
            }

            leaf keepalive-timer {
                type uint32 {
                  range "0..255";
                }
                units "seconds";
                must "(../state = 'session-up'" +
                    "or " +
                    "(../state != 'session-up'" +
                    "and keepalive-timer = 0))" {
                       error-message
                           "Invalid keepalive
                            timer";
                       description
                           "This field is used if
                            and only if state is
                            session-up. Otherwise,
                            it is not used and
                            MUST be set to
                            zero.";
                    }
                description
                    "The agreed maximum interval at
                     which the local PCEP entity
                     transmits PCEP messages on this
                     PCEP session.  Zero means that
                     the local PCEP entity never
                     sends Keepalives on this
```

```
                session.";
            }

            leaf peer-keepalive-timer {
                type uint32 {
                  range "0..255";
                }
                units "seconds";
                must "(../state = 'session-up'" +
                     "or "   +
                     "(../state != 'session-up'" +
                     "and " +
                     "peer-keepalive-timer = 0))" {
                        error-message
                            "Invalid Peer keepalive
                             timer";
                        description
                            "This field is used if
                             and only if state is
                             session-up. Otherwise,
                             it is not used and MUST
                             be set to zero.";
                    }
                description
                     "The agreed maximum interval at
                      which the peer transmits PCEP
                      messages on this PCEP session.
                      Zero means that the peer never
                      sends Keepalives on this
                      session.";
            }

            leaf dead-timer {
                type uint32 {
                  range "0..255";
                }
                units "seconds";
                description
                     "The dead timer interval for
                      this PCEP session.";
            }

            leaf peer-dead-timer {
                type uint32 {
                  range "0..255";
                }
                units "seconds";
                must "((../state != 'tcp-pending'" +
```

```
                    "and " +
                    "../state != 'open-wait' )" +
                    "or " +
                    "((../state = 'tcp-pending'" +
                    " or " +
                    "../state = 'open-wait' )" +
                    "and " +
                    "peer-dead-timer = 0))" {
                       error-message
                           "Invalid Peer Dead
                            timer";
                       description
                           "If state is tcp-
                            pending or open-wait
                            then this leaf is not
                            used and MUST be set to
                            zero.";
                    }
                description
                    "The peer's dead-timer interval
                     for this PCEP session.";
            }

            leaf ka-hold-time-rem {
                type uint32 {
                  range "0..255";
                }
                units "seconds";
                must "((../state != 'tcp-pending'" +
                    "and " +
                    "../state != 'open-wait' ) " +
                    "or " +
                    "((../state = 'tcp-pending'" +
                    "or " +
                    "../state = 'open-wait' )" +
                    "and " +
                    "ka-hold-time-rem = 0))" {
                       error-message
                           "Invalid Keepalive hold
                            time remaining";
                       description
                           "If state is tcp-pending
                            or open-wait then this
                            field is not used and
                            MUST be set to zero.";
                    }
                description
                    "The keep alive hold time
```

```
                remaining for this session.";
            }

            leaf overloaded {
                type boolean;
                description
                     "If the local PCEP entity has
                      informed the peer that it is
                      currently overloaded, then this
                      is set to true.  Otherwise, it
                      is set to false.";
            }

            leaf overload-time {
                type uint32;
                units "seconds";
                must "(../overloaded = true or" +
                    "(../overloaded != true and" +
                    " overload-time = 0))" {
                      error-message
                          "Invalid overload-time";
                      description
                          "This field is only used
                           if overloaded is set to
                           true. Otherwise, it is
                           not used and MUST be set
                           to zero.";
                    }
                description
                     "The interval of time that is
                      remaining until the local PCEP
                      entity will cease to be
                      overloaded on this session.";
            }

            leaf peer-overloaded {
                type boolean;
                description
                     "If the peer has informed the
                      local PCEP entity that it is
                      currently overloaded, then this
                      is set to true. Otherwise, it
                      is set to false.";
            }

            leaf peer-overload-time {
                type uint32;
                units "seconds";
```

```
            must "(../peer-overloaded = true" +
                 " or " +
                 "(../peer-overloaded != true" +
                 " and " +
                 "peer-overload-time = 0))" {
                    error-message
                        "Invalid peer overload
                         time";
                    description
                        "This field is only used
                         if peer-overloaded is
                         set to true. Otherwise,
                         it is not used and MUST
                         be set to zero.";
                }
            description
                 "The interval of time that is
                  remaining until the peer will
                  cease to be overloaded.  If it
                  is not known how long the peer
                  will stay in overloaded state,
                  this leaf is set to zero.";
        }
        leaf lspdb-sync {
            if-feature stateful;
            type sync-state;
            description
                "The LSP-DB state synchronization
                status.";
        }
        leaf discontinuity-time {
            type yang:timestamp;
            description
                 "The timestamp value of the time
                  when the statistics were last
                  reset.";
        }

        container pcep-stats {
            description
                "The container for all statistics
                 at session level.";
            uses pcep-stats{
                description
                    "The statistics contained are
                     for the current sessions to
                     that peer. These are lost
                     when the session goes down.
```

```
                                        ";
                                    }
                                }//pcep-stats

                            } // session
                        } // sessions
                    }//peer
                }//peers
            }//entity
        }//pcep-state

        /*
         * Notifications
         */
        notification pcep-session-up {
            description
                "This notification is sent when the value of
                 '/pcep/pcep-state/peers/peer/sessions/session/state'
                 enters the 'session-up' state.";

            uses notification-instance-hdr;

            uses notification-session-hdr;

            leaf state-last-change {
                type yang:timestamp;
                description
                    "The timestamp value at the time this session entered
                    its current state as denoted by the state leaf.";
            }

            leaf state {
                type pcep-sess-state;
                description
                    "The current state of the session.
                     The set of possible states excludes the idle state
                     since entries do not exist in the idle state.";
            }
        } //notification

        notification pcep-session-down {
            description
                "This notification is sent when the value of
                 '/pcep/pcep-state/peers/peer/sessions/session/state'
                 leaves the 'session-up' state.";

            uses notification-instance-hdr;
```

```
        leaf session-initiator {
            type pcep-initiator;
            description
                "The initiator of the session.";
        }

        leaf state-last-change {
            type yang:timestamp;
            description
                "The timestamp value at the time this session entered
                its current state as denoted by the state leaf.";
        }

        leaf state {
            type pcep-sess-state;
            description
                "The current state of the session.
                 The set of possible states excludes the idle state
                 since entries do not exist in the idle state.";
        }
    } //notification

    notification pcep-session-local-overload {
        description
            "This notification is sent when the local PCEP entity
            enters overload state for a peer.";

        uses notification-instance-hdr;

        uses notification-session-hdr;

        leaf overloaded {
            type boolean;
            description
                 "If the local PCEP entity has informed the peer that
                 it is currently overloaded, then this is set to
                 true. Otherwise, it is set to false.";
        }

        leaf overload-time {
            type uint32;
            units "seconds";
            must "(../overloaded = true or "    +
                "(../overloaded != true  and "  +
                "overload-time = 0))" {
                    error-message
                        "Invalid overload-time";
                    description
```

```
                        "This field is only used if overloaded is
                         set to true. Otherwise, it is not used
                         and MUST be set to zero.";
                    }
            description
                 "The interval of time that is remaining until the
                 local PCEP entity will cease to be overloaded on
                 this session.";
        }
    } //notification

    notification pcep-session-local-overload-clear {
        description
            "This notification is sent when the local PCEP entity
            leaves overload state for a peer.";

        uses notification-instance-hdr;

        leaf overloaded {
            type boolean;
            description
                 "If the local PCEP entity has informed the peer
                 that it is currently overloaded, then this is set
                 to true.  Otherwise, it is set to false.";
        }
    } //notification

    notification pcep-session-peer-overload {
        description
            "This notification is sent when a peer enters overload
            state.";

        uses notification-instance-hdr;

        uses notification-session-hdr;

        leaf peer-overloaded {
            type boolean;
            description
                "If the peer has informed the local PCEP entity that
                it is currently overloaded, then this is set to true.
                Otherwise, it is set to false.";
        }

        leaf peer-overload-time {
            type uint32;
            units "seconds";
            must "(../peer-overloaded = true or "    +
```

```
                "(../peer-overloaded != true  and "  +
                "peer-overload-time = 0))" {
                    error-message
                        "Invalid peer-overload-time";
                    description
                        "This field is only used if
                         peer-overloaded is set to true.
                         Otherwise, it is not used and MUST
                         be set to zero.";
                }
            description
                "The interval of time that is remaining until the
                peer will cease to be overloaded.  If it is not known
                how long the peer will stay in overloaded state, this
                leaf is set to zero.";
        }
    } //notification

    notification pcep-session-peer-overload-clear {
        description
            "This notification is sent when a peer leaves overload
            state.";

        uses notification-instance-hdr;

        leaf peer-overloaded {
            type boolean;
            description
                "If the peer has informed the local PCEP entity that
                it is currently overloaded, then this is set to true.
                Otherwise, it is set to false.";
        }
    } //notification
}//module

<CODE ENDS>
```

9. Security Considerations

The YANG module defined in this memo is designed to be accessed via the NETCONF protocol [RFC6241]. The lowest NETCONF layer is the secure transport layer and the mandatory-to-implement secure transport is SSH [RFC6242]. The NETCONF access control model [RFC6536] provides the means to restrict access for particular NETCONF users to a pre-configured subset of all available NETCONF protocol operations and content.

There are a number of data nodes defined in the YANG module which are writable/creatable/deletable (i.e., config true, which is the default). These data nodes may be considered sensitive or vulnerable in some network environments. Write operations (e.g., <edit-config>) to these data nodes without proper protection can have a negative effect on network operations.

TBD: List specific Subtrees and data nodes and their sensitivity/ vulnerability.

10. Manageability Considerations

10.1. Control of Function and Policy

10.2. Information and Data Models

10.3. Liveness Detection and Monitoring

10.4. Verify Correct Operations

10.5. Requirements On Other Protocols

10.6. Impact On Network Operations

11. IANA Considerations

This document registers a URI in the "IETF XML Registry" [RFC3688]. Following the format in RFC 3688, the following registration has been made.

URI: urn:ietf:params:xml:ns:yang:ietf-pcep

Registrant Contact: The PCE WG of the IETF.

XML: N/A; the requested URI is an XML namespace.

This document registers a YANG module in the "YANG Module Names" registry [RFC6020].

```
Name:         ietf-pcep
Namespace:    urn:ietf:params:xml:ns:yang:ietf-pcep
Prefix:       pcep
Reference:    This I-D
```

12. Acknowledgements

The initial document is based on the PCEP MIB [RFC7420]. Further this document structure is based on Routing Yang Module [I-D.ietf-netmod-routing-cfg]. We would like to thank the authors of aforementioned documents.

13. References

13.1. Normative References

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

[RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, DOI 10.17487/RFC3688, January 2004, <http://www.rfc-editor.org/info/rfc3688>.

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

[RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, DOI 10.17487/RFC6020, October 2010, <http://www.rfc-editor.org/info/rfc6020>.

[RFC6991] Schoenwaelder, J., Ed., "Common YANG Data Types", RFC 6991, DOI 10.17487/RFC6991, July 2013, <http://www.rfc-editor.org/info/rfc6991>.

[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-14 (work in progress), March 2016.

[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-05 (work in progress), October 2015.

[I-D.ietf-pce-lsp-setup-type]
Sivabalan, S., Medved, J., Minei, I., Crabbe, E., Varga, R., Tantsura, J., and J. Hardwick, "Conveying path setup type in PCEP messages", draft-ietf-pce-lsp-setup-type-03 (work in progress), June 2015.

[I-D.ietf-pce-segment-routing]
Sivabalan, S., Medved, J., Filsfils, C., Crabbe, E., Lopez, V., Tantsura, J., Henderickx, W., and J. Hardwick, "PCEP Extensions for Segment Routing", draft-ietf-pce-segment-routing-07 (work in progress), March 2016.

13.2. Informative References

[RFC4655] Farrel, A., Vasseur, J., and J. Ash, "A Path Computation Element (PCE)-Based Architecture", RFC 4655, DOI 10.17487/RFC4655, August 2006, <http://www.rfc-editor.org/info/rfc4655>.

[RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <http://www.rfc-editor.org/info/rfc6241>.

[RFC6242] Wasserman, M., "Using the NETCONF Protocol over Secure Shell (SSH)", RFC 6242, DOI 10.17487/RFC6242, June 2011, <http://www.rfc-editor.org/info/rfc6242>.

[RFC6536] Bierman, A. and M. Bjorklund, "Network Configuration Protocol (NETCONF) Access Control Model", RFC 6536, DOI 10.17487/RFC6536, March 2012, <http://www.rfc-editor.org/info/rfc6536>.

[RFC7420] Koushik, A., Stephan, E., Zhao, Q., King, D., and J. Hardwick, "Path Computation Element Communication Protocol (PCEP) Management Information Base (MIB) Module", RFC 7420, DOI 10.17487/RFC7420, December 2014, <http://www.rfc-editor.org/info/rfc7420>.

[I-D.ietf-netmod-routing-cfg]
Lhotka, L. and A. Lindem, "A YANG Data Model for Routing Management", draft-ietf-netmod-routing-cfg-22 (work in progress), July 2016.

[I-D.ietf-netmod-rfc6087bis]
Bierman, A., "Guidelines for Authors and Reviewers of YANG Data Model Documents", draft-ietf-netmod-rfc6087bis-06 (work in progress), March 2016.

[I-D.ietf-teas-yang-te]
Saad, T., Gandhi, R., Liu, X., Beeram, V., Shah, H., Chen, X., Jones, R., and B. Wen, "A YANG Data Model for Traffic Engineering Tunnels and Interfaces", draft-ietf-teas-yang-te-03 (work in progress), March 2016.

Appendix A. Contributor Addresses

Rohit Pobbathi
Huawei Technologies
Divyashree Techno Park, Whitefield
Bangalore, Karnataka 560066
India

EMail: rohit.pobbathi@huawei.com

Vinod KumarS
Huawei Technologies
Divyashree Techno Park, Whitefield
Bangalore, Karnataka 560066
India

EMail: vinods.kumar@huawei.com

Zafar Ali
Cisco Systems
Canada

EMail: zali@cisco.com

Xufeng Liu
Ericsson
1595 Spring Hill Road, Suite 500
Vienna, VA 22182
USA

EMail: xufeng.liu@ericsson.com

Young Lee
Huawei Technologies
5340 Legacy Drive, Building 3
Plano, TX 75023, USA

Phone: (469) 277-5838
EMail: leeyoung@huawei.com

Udayasree Palle
Huawei Technologies
Divyashree Techno Park, Whitefield
Bangalore, Karnataka 560066
India

EMail: udayasree.palle@huawei.com

Xian Zhang
Huawei Technologies
Bantian, Longgang District
Shenzhen 518129
P.R.China

EMail: zhang.xian@huawei.com

Avantika
Huawei Technologies
Divyashree Techno Park, Whitefield
Bangalore, Karnataka 560066
India

EMail: avantika.sushilkumar@huawei.com

Authors' Addresses

Dhruv Dhody (editor)
Huawei Technologies
Divyashree Techno Park, Whitefield
Bangalore, Karnataka 560066
India

EMail: dhruv.ietf@gmail.com

Jonathan Hardwick
Metaswitch
100 Church Street
Enfield EN2 6BQ
UK

EMail: jonathan.hardwick@metaswitch.com

Vishnu Pavan Beeram
Juniper Networks
USA

EMail: vbeeram@juniper.net

Jeff Tantsura
USA

EMail: jefftant@gmail.com

PCE Working Group
Internet-Draft
Intended status: Experimental
Expires: September 29, 2017

Q. Wu
D. Dhody
Huawei
D. King
Lancaster University
D. Lopez
Telefonica I+D
J. Tantsura
March 28, 2017

Path Computation Element (PCE) Discovery using Domain Name System(DNS)
draft-wu-pce-dns-pce-discovery-10

Abstract

Discovery of the Path Computation Element (PCE) within an IGP area or routing domain is possible using OSPF and IS-IS IGP discovery. However, it has been established that in certain deployment scenarios PCEs may not wish, or be able to participate within the IGP process. In those scenarios, it is beneficial for the Path Computation Client (PCC) (or other PCE) to discover PCEs via an alternative mechanism to using an IGP discovery.

This document specifies the requirements, use cases, procedures and extensions to support PCE discovery along with certain relevant information type and capability discovery via DNS.

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 September 29, 2017.

Table of Contents

1. Introduction 3
1.1. Terminology 3
1.2. Requirements 3
2. Conventions used in this document 4
3. Motivation 4
3.1. Outside the Routing Domain 4
3.2. Discovery Mechanisms 5
3.2.1. Query-Response versus Advertisement 5
3.3. PCE Virtualization 6
3.4. Additional Capabilities 6
3.4.1. Handling Changes in PCE Identities 6
3.4.2. Secure Inter-domain Discovery 6
3.4.3. Load Sharing of Path Computation Requests 6
4. Extended Naming Authority Pointer (NAPTR)Service Field Format 7
4.1. IETF Standards Track PCE Applications 8
5. Backwards Compatibility 8
6. Discovering a Path Computation Element 9
6.1. Determining the PCE Service and transport protocol . . . 10
6.2. Determining the IP Address of the PCE 10
6.2.1. Examples 12
6.3. Determining the PCE domains and Neighbor PCE domains . . 13
7. IANA Considerations 14
7.1. IETF PCE Application Service Tags 14
7.2. PCE Application Protocol Tags 14
8. Security Considerations 14
9. Acknowledgements 15
10. References 15
10.1. Normative References 15
10.2. Informative References 16
Authors' Addresses 18

1. Introduction

The Path Computation Element Communication Protocol (PCEP) is a transaction-based protocol carried over TCP [RFC4655]. In order to be able to direct path computation requests to the Path Computation Element (PCE), a Path Computation Client (PCC) (or other PCE) needs to know the location and capability of a PCE.

In a network where an IGP is used and where the PCE participates in the IGP, discovery mechanisms exist for PCC (or PCE) to learn the identity and capability of each PCE. [RFC5088] defines a PCE Discovery (PCED) TLV carried in an OSPF Router LSA. Similarly, [RFC5089] defines the PCED sub-TLV for use in PCE Discovery using IS-IS. Scope of the advertisement is limited to IGP area/level or Autonomous System (AS).

However in certain scenarios not all PCEs will participate in the same IGP instance, section 3 (Motivation) outlines a number of use cases. In these cases, current PCE Discovery mechanisms are therefore not appropriate and another PCE discovery function would be required. (sec 4 of [PCE-QUESTION]).

This document describes PCE discovery via DNS. The mechanism with which DNS comes to know about the PCE and its capability is out of scope of this document.

1.1. Terminology

The following terminology is used in this document.

PCE-Domain: As per [RFC4655], any collection of network elements within a common sphere of address management or path computational responsibility. Examples of domains include Interior Gateway Protocol (IGP) areas and Autonomous Systems (ASs).

Domain-Name: An identification string that defines a realm of administrative autonomy, authority, or control on the Internet. Any name registered in the DNS is a domain name. DNS Domain names are used in various networking contexts and application-specific naming and addressing purposes. In general, a domain name represents an Internet Protocol (IP) resource. Examples of DNS domain name is "www.example.com" or "example.com" [RFC1035].

1.2. Requirements

As described in [RFC4674], the PCE Discovery information should at least be composed of:

- The PCE location: an IPv4 and/or IPv6 address that is used to reach the PCE. It is RECOMMENDED to use an address that is always reachable if there is any connectivity to the PCE;
- The PCE path computation scope (i.e., inter-area, inter-AS, or inter-layer);
- The set of one or more PCE-Domain(s) into which the PCE has visibility and for which the PCE can compute paths;
- The set of zero, one, or more neighbor PCE-Domain(s) toward which the PCE can compute paths;
- The set of communication and path computation-specific capabilities.

These PCE discovery information allows PCCs to select appropriate PCEs.

This document specifies the procedures and extension to facilitate DNS-based PCE information discovery for specific use cases, and to complement existing IGP discovery mechanism.

2. 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 RFC2119 [RFC2119].

3. Motivation

This section discusses in more detail the motivation and use cases for an alternative DNS-based PCE discovery mechanism.

3.1. Outside the Routing Domain

When the PCE is a router participating in the IGP, or even a server participating passively in the IGP, with all PCEP speakers in the same routing domain, a simple and efficient way to announce PCEs consists of using IGP flooding.

It has been identified that the existing PCE discovery mechanisms do not work very well in following scenarios:

Inter-AS: Per domain path computation mechanism [RFC5152] or Backward recursive path computation (BRPC) [RFC5441] MAY be used by cooperating PCEs to compute inter-domain path. In which case these cooperating PCEs should be known to other PCEs. In case of

inter-AS where the PCEs do not participate in a common IGP, the existing IGP discovery mechanism cannot be used to discover inter-AS PCE.

Hierarchy of PCE: The H-PCE [RFC6805] architecture does not require disclosure of internals of a child domain to the parent PCE. It may be necessary for a third party to manage the parent PCEs according to commercial and policy agreements from each of the participating service providers [PCE-QUESTION]. [RFC6805] specifies that a child PCE must be configured with the address of its parent PCE in order for it to interact with its parent PCE. However handling changes in parent PCE identities and coping with failure events would be an issue for a configured system. There is no scope for parent PCEs to advertise their presence to child PCEs when they are not a part of the same routing domain.

BGP-LS: [BGP-LS] describes a mechanism by which links state and traffic engineering information can be collected from networks and shared with external components using the BGP routing protocol. An external PCE MAY use this mechanism to populate its TED and not take part in the same IGP routing domain.

NMS/OSS: PCE MAY gain the knowledge of Topology information from some management system (e.g.,NMS/OSS) and not take part in the same routing domain. Also note that in some case PCC may not be a router and instead be a management system like NMS and may not be able to discover PCE via IGP discovery.

3.2. Discovery Mechanisms

3.2.1. Query-Response versus Advertisement

Advertisement based PCE discovery using IGP methods [RFC5088] and [RFC5089] floods the PCE information to an area, a subset of areas or to a full routing domain. By the very nature of flooding and advertisements it generates unwanted traffic and may lead to unnecessary advertisement, especially when PCE information needs frequent changes.

DNS is a query-response based mechanism, a client (a PCC) can use DNS to discover a PCE only when it needs to compute a path and does not require any other node in the network to be involved.

In case of Intermittent PCEP session, where PCEP sessions are systematically open and closed for each PCEP request, a DNS-based query-response mechanism is more suitable. One may also utilize DNS-based load-balancing and recovery functions.

3.3. PCE Virtualization

Server virtualization has gain importance since it provides better reliability and high availability in the event of hardware failure. It allows for higher utilization of physical resources while improving administration by having a single management interface for all virtual servers.

When one PCE instance is virtually hosted on a server and initiated as a PCE instance, another PCE instance may be created on the same server or a different server to provide better load balancing and reliability. In such a case, where there are a large number of PCCs that need to know these PCE instances' location, manual configuration on PCCs for PCC and PCE relationship is not trivial or desirable.

3.4. Additional Capabilities

3.4.1. Handling Changes in PCE Identities

In the case of H-PCE ,when a dynamic Address is assigned to the parent PCE, any existing configuration entry on child PCE becomes invalid and the parent PCE becomes unreachable. In order to handle changes in parent PCE identities, the DNS update can be used to provide IP reachability to the parent PCE with new assigned Address. The DNS update can be performed by either parent PCE or OSS/NMS that is aware of PCE Identities changes.

3.4.2. Secure Inter-domain Discovery

Applications make use of DNS lookups on FQDN to find a node(e.g., PCEP endpoint). When a PCE performs DNS lookup or dynamic DNS update with the DNS server, the PCE MUST have a security association of some type with the DNS server. The security association SHOULD be established either using DNSSEC [RFC4033] or TSIG/ TKEY[RFC2845][RFC2930]. DNS lookup for PCE Discovery can be applied either within an administration domain or spanning across administration domains. A security association is REQUIRED even if the DNS server is in the same administrative domain as the PCE.

3.4.3. Load Sharing of Path Computation Requests

Multiple PCEs can be present in a single network domain for redundancy. DNS supports inherent load balancing where multiple PCEs (with different IP addresses) are known in DNS for a single PCE server name and are hidden from the PCC.

In an IGP advertisement based PCE discovery, one learns of all the PCEs and it is the job of the PCC to do load-balancing.

A DNS-based load-balancing mechanism works well in case of Intermittent PCEP sessions and request are load-balanced among PCEs similar to HTTP request without any complexity at the client.

4. Extended Naming Authority Pointer (NAPTR)Service Field Format

The NAPTR service field format defined by the S-NAPTR DDDS application in [RFC3958] follows this Augmented Backus-Naur Form (ABNF) [RFC5234]:

```
service-parms = [ [app-service] *(":" app-protocol)]
app-service   = experimental-service  / iana-registered-service
app-protocol  = experimental-protocol / iana-registered-protocol
experimental-service      = "x-" 1*30ALPHANUMSYM
experimental-protocol     = "x-" 1*30ALPHANUMSYM
iana-registered-service   = ALPHA *31ALPHANUMSYM
iana-registered-protocol  = ALPHA *31ALPHANUMSYM
ALPHA         =  %x41-5A / %x61-7A   ; A-Z / a-z
DIGIT         =  %x30-39 ; 0-9
SYM           =  %x2B / %x2D / %x2E  ; "+" / "-" / "."
ALPHANUMSYM   =  ALPHA / DIGIT / SYM
; The app-service and app-protocol tags are limited to 32
; characters and must start with an alphabetic character.
; The service-parms are considered case-insensitive.
```

This specification refines the "iana-registered-service" tag definition for the discovery of PCE supporting a specific PCE application or multiple PCE applications as defined below.

```
iana-registered-service =/ pce-service
pce-service             = "pce" *("+" appln-name)
appln-name                = non-ws-string
non-ws-string  = 1*(%x21-FF)
```

The appln-name element is the Application Identifier used to identify a specific PCE application. The PCE Application Name are allocated by IANA as defined in section 8.1.

This specification also refines the "iana-registered-protocol" tag definition for the discovery of PCE supporting a specific transport protocol as defined below.

```
iana-registered-protocol =/ pce-protocol
pce-protocol             = "pce." pce-transport
pce-transport            = "tcp" / "tls.tcp"
```

Similar to application protocol tags defined in the [RFC6408],the S-NAPTR application protocol tags defined by this specification MUST

NOT be parsed in any way by the querying application or Resolver. The delimiter (".") is present in the tag to improve readability and does not imply a structure or namespace of any kind. The choice of delimiter (".") for the application protocol tag follows the format of existing S-NAPTR application protocol tag registry entries, but this does not imply that it shares semantics with any other specifications that create registry entries with the same format.

The S-NAPTR application service and application protocol tags defined by this specification are unrelated to the IANA "Service Name and Transport Protocol Port Number Registry" (see [RFC6335]).

The maximum length of the NAPTR service field is 256 octets, including a one-octet length field (see Section 4.1 of [RFC3403] and Section 3.3 of [RFC1035]).

4.1. IETF Standards Track PCE Applications

A PCE Client MUST be capable of using the extended S-NAPTR application service tag for dynamic discovery of a PCE supporting Standards Track applications. Therefore, every IETF Standards Track PCE application MUST be associated with a "PCE-service" tag formatted as defined in this specification and allocated in accordance with IANA policy (see Section 8).

For example, a NAPTR service field value of:

'PCE+gco:pce.tcp'

means that the PCE in the SRV or A/AAAA record supports the Global Concurrent Optimization Application (See section 8.1)and the Transport Control Protocol (TCP) as the transport protocol (See section 8.2).

5. Backwards Compatibility

Domain Name System (DNS) administrators SHOULD also provision legacy NAPTR records [RFC3403] in order to guarantee backwards compatibility with legacy PCE that only support S-NAPTR DDDS application in [RFC3958]. If the DNS administrator provisions both extended S-NAPTR records as defined in this specification and legacy NAPTR records defined in [RFC3403], then the extended S-NAPTR records MUST have higher priority(e.g., lower order and/or preference values) than legacy NAPTR records.

6. Discovering a Path Computation Element

The extended-format NAPTR records provide a mapping from a domain to the SRV record or A/AAAA record for contacting a server supporting a specific transport protocol and PCE application. The resource record will contain an empty regular expression and a replacement value, which is the SRV record or the A/AAAA record for that particular transport protocol.

The assumption for this mechanism to work is that the DNS administrator of the queried domain has first provisioned the DNS with extended-format NAPTR entries.

When the PCC or other PCEs performs a NAPTR query for a server in a particular realm, the PCC or other PCEs has to know in advance the search path of the resolver, i.e.,in which realm to look for a PCE, and in which Application Identifier it is interested.

The search path of the resolver can either be pre-configured, or discovered using Diameter, DHCP or other means. For example, the realm could be deduced from the Network Access Identifier (NAI) in the User-Name attribute-value pair (AVP) or extracted from the Destination-Realm AVP in Diameter [RFC6733].

When pre-configuration is used, PCE domain(e.g.,AS200)may be added as "subdomains" of the first-level domain of the underlying service (e.g., AS200.example.com), which allows a NAPTR query for a server in a PCE domain associated with DNS domain-name.

When DHCP is used, it SHOULD know the domain-name of that realm and use DHCP to discover IP address of the PCE in that realm that provides path computation service along with some PCE location information useful to a PCC (or other PCE) for a PCE selection, and contact it directly. In some instances, the discovery may result in a per protocol/application list of domain-names that are then used as starting points for the subsequent S-NAPTR lookups [RFC3958]. If neither the IP address nor other PCE location information can be discovered with the above procedure, the PCC (or other PCE) MAY request a domain search list, as described in [RFC3397] and[RFC3646], and use it as input to the DDDS application.

When the PCC (or other PCE) does not find valid domain-names using the mechanisms above, it MUST stop the attempt to discover any PCE.

The following procedures result in an IP address, PCE domain, neighboring PCE domain and PCE Computation Scope where the PCC (or other PCE) can contact the PCE that hosts the service it is looking for.

6.1. Determining the PCE Service and transport protocol

The PCC (or other PCE) should know the service identifier for the Path Computation service and associated transport protocol. The service identifier for the Path Computation service is defined as "PCE+apX" as specified in section 5, The PCE supporting "PCE" service MUST support TCP as transport, as described in [RFC5440].

The services relevant for the task of transport protocol selection are those with S-NAPTR service fields with values "PCE+apX:Y", where 'PCE+apX' is the service identifier defined in the previous paragraph, and ' Y' is the letter that corresponds to a transport protocol supported by the PCE. This document also establishes an IANA registry for mappings of S-NAPTR service name to transport protocol.

These NAPTR [RFC3958] records provide a mapping from a domain to the SRV [RFC2782] record for contacting a PCE with the specific transport protocol in the S-NAPTR services field. The resource record MUST contain an empty regular expression and a replacement value, which indicates the domain name where the SRV record for that particular transport protocol can be found. As per [RFC3403], the client discards any records whose services fields are not applicable.

The PCC (or other PCE) MUST discard any service fields that identify a resolution service whose value is not valid. The S-NAPTR processing as described in [RFC3403] will result in the discovery of the most preferred PCE that is supported by the client, as well as an SRV record for the PCE.

6.2. Determining the IP Address of the PCE

If the returned NAPTR service fields contain entries formatted as "pce+apX:Y" where "X" indicates the Application Identifier and "Y" indicates the supported transport protocol(s), the target realm supports the extended format for NAPTR-based PCE discovery defined in this document.

- If "X" contains the required Application Identifier and "Y" matches a supported transport protocol, the PCEP implementation resolves the "replacement" field entry to a target host using the lookup method appropriate for the "flags" field.

- If "X" does not contain the required Application Identifier or "Y" does not match a supported transport protocol, the PCEP implementation abandons the peer discovery.

If the returned NAPTR service fields contain entries formatted as "pce+apX" where "X" indicates the Application Identifier, the target realm supports the extended format for NAPTR-based PCE discovery defined in this document.

- If "X" contains the required Application Identifier, the PCEP implementation resolves the "replacement" field entry to a target host using the lookup method appropriate for the "flags" field and attempts to connect using all supported transport protocols.

- If "X" does not contain the required Application Identifier, the PCEP implementation abandons the PCE discovery.

If the returned NAPTR service fields contain entries formatted as "pce:X" where "X" indicates the supported transport protocol(s), the target realm supports PCEP but does not support the extended format for NAPTR-based PCE discovery defined in this document.

- If "X" matches a supported transport protocol, the PCEP implementation resolves the "replacement" field entry to a target host using the lookup method appropriate for the "flags" field.

If the returned NAPTR service fields contain entries formatted as "pce", the target realm supports PCEP but does not support the extended format for NAPTR-based PCE discovery defined in this document. The PCEP implementation resolves the "replacement" field entry to a target host using the lookup method appropriate for the "flags" field and attempts to connect using TCP (in future it SHOULD attempt all supported transport Protocols) .

Note that the regexp field in the S-NAPTR example above is empty. The regexp field MUST NOT be used when discovering PCE, as its usage can be complex and error prone. Also, the discovery of the PCE does not require the flexibility provided by this field over a static target present in the TARGET field.

As the default behavior, the client is configured with the information about which transport protocol is used for a path computation service in a particular domain. The client can directly perform an SRV query for that specific transport using the service identifier of the path computation Service. For example, if the client knows that it should be using TCP for path computation service, it can perform a SRV query directly for_PCE._tcp.example.com.

Once the server providing the desired service and the transport protocol has been determined, the next step is to determine the IP address.

According to the specification of SRV RRs in [RFC2782], the TARGET field is a fully qualified domain-name (FQDN) that MUST have one or more address records; the FQDN must not be an alias, i.e., there MUST NOT be a CNAME or DNAME RR at this name. Unless the SRV DNS query already has reported a sufficient number of these address records in the Additional Data section of the DNS response (as recommended by [RFC2782]), the PCC needs to perform A and/or AAAA record lookup(s) of the domain-name, as appropriate. The result will be a list of IP addresses, each of which can be contacted using the transport protocol determined previously.

6.2.1. Examples

As an example, consider a client that wishes to find PCED service in the as100.example.com domain. The client performs a S-NAPTR query for that domain, and the following NAPTR records are returned:

```
  Order Pref Flags  Service     Regexp       Replacement
   IN NAPTR  50   50   "s"  "pce:pce.tls.tcp"    ""
     _PCE._tcp.as100.example.com
   IN NAPTR  90   50   "s"  "pce:pce.tcp"    ""
      _PCE._tcp.as100.example.com
```

This indicates that the domain does have a PCE providing Path Computation services over TCP, in that order of preference. If the client only supports TCP, TCP will be used, targeted to a host determined by an SRV lookup of _PCE._tcp.example.com. That lookup would return:

```
    ;;  Priority  Weight    Port        Target
  IN  SRV    0        1      XXXX   server1.as100.example.com
  IN  SRV    0        2      XXXX   server2.as100.example.com
```

where XXXX represents the port number at which the service is reachable.

As an alternative example, a client wishes to discover a PCE in the ex2.example.com realm that supports the GCO application over TCP. The client performs a NAPTR query for that domain, and the following NAPTR records are returned:

```
;;        order pref flags service   regexp replacement
IN NAPTR  150   50   "a"   "pce:pce.tcp"  ""
             server1.ex2.example.com
IN NAPTR  150   50   "a"   "pce:pce.tls.tcp"  ""
             server2.ex2.example.com
IN NAPTR  150   50   "a"   "pce+gco:pce.tcp"  ""
             server1.ex2.example.com
IN NAPTR  150   50   "a"   "pce+gco:pce.tls.tcp"  ""
             server2.ex2.example.com
```

This indicates that the server supports GCO(ID=1) over TCP and TLS/ TCP via hosts server1.ex2.example.com and server2.ex2.example.com, respectively.

6.3. Determining the PCE domains and Neighbor PCE domains

DNS servers MAY use DNS TXT record to give additional information about PCE service and add such TXT record to the additional information section (See section 4.1 of [RFC1035]) that are relevant to the answer and have the same authenticity as the data (Generally this will be made up of A and SRV records)in the answer section. The additional information may include path computation capability, the PCE domains and Neighbor PCE domains associated with the PCE. If discovery of PCE supporting a specific PCE capability described in section 7.2 has already been performed, capability associated with the PCE does not need to be included in the additional information.

To store new types of information, the TXT record uses a structured format in its TXT-DATA field [RFC1035]. The format consists of the attribute name followed by the value of the attribute. The name and value are separated by an equals sign (=). The general syntax may follow one defined in section 2 of [RFC1464] as follows:

<owner> <class> <ttl> TXT "<attribute name>=<attribute value>"

For example, the following TXT records contain attributes specified in this fashion:

```
ex2.example.com    IN   TXT   "pce domain = as10"
ex2.example.com    IN   TXT   "neigh domain= as5"
ex2.example.com    IN   TXT   "cap=link constraint"
```

The client MAY inspect those Additional Information section in the DNS message and be capable of handling responses from nameservers that never fill in the Additional Information part of a response.

7. IANA Considerations

7.1. IETF PCE Application Service Tags

IANA specifies to create a new registry ’ S-NAPTR application service tags’ for existing IETF PCE applications.

Tag	PCE Application
pce+gco	GCO [RFC5557]
pce+p2mp	P2MP [RFC5671]
pce+stateful	Stateful [STATEFUL-PCE]
pce+gmpls	GMPLS [RFC7025]
pce+interas	Inter-AS[RFC5376]
pce+interarea	Inter-Area [RFC4927]
pce+interlayer	Inter-layer [RFC6457]

Future IETF PCE applications MUST reserve the S-NAPTR application service tag corresponding to the allocated PCE Application ID as defined in Section 3.

7.2. PCE Application Protocol Tags

IANA has reserved the following S-NAPTR Application Protocol Tags for the PCE transport protocols in the "S-NAPTR Application Protocol Tag" registry created by [RFC3958].

Tag	Protocol
pce.tcp	TCP

Future PCE versions that introduce new transport protocols MUST reserve an appropriate S-NAPTR Application Protocol Tag in the "S-NAPTR Application Protocol Tag" registry created by [RFC3958].

8. Security Considerations

This document specifies an enhancement to the NAPTR service field format. The enhancement and modifications are based on the S-NAPTR, which is actually a simplification of the NAPTR, and therefore the same security considerations described in [RFC3958] are applicable to this document.

For most of those identified threats, the DNS Security Extensions [RFC4033] does provide protection. It is therefore recommended to consider the usage of DNSSEC [RFC4033] and the aspects of DNSSEC Operational Practices [RFC6781] when deploying Path Computation Services.

In deployments where DNSSEC usage is not feasible, measures should be taken to protect against forged DNS responses and cache poisoning as much as possible. Efforts in this direction are documented in [RFC5452].

However a malicious host doing S-NAPTR queries learns applications supported by PCEs in a certain realm faster, which might help the malicious host to scan potential targets for an attack more efficiently when some applications have known vulnerabilities.

Where inputs to the procedure described in this document are fed via DHCP, DHCP vulnerabilities can also cause issues. For instance, the inability to authenticate DHCP discovery results may lead to the Path Computation service results also being incorrect, even if the DNS process was secured.

9. Acknowledgements

The author would like to thank Claire Bi,Ning Kong, Liang Xia, Stephane Bortzmeyer,Yi Yang, Ted Lemon, Adrian Farrel and Stuart Cheshire for their review and comments that help improvement to this document.

10. References

10.1. Normative References

[RFC1035] Mockapetris, P., "DOMAIN NAMES - IMPLEMENTATION AND SPECIFICATION", RFC 1035, November 1987.

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

[RFC2782] Gulbrandsen, A., "A DNS RR for specifying the location of services (DNS SRV)", RFC 2782, February 2000.

[RFC3397] Aboba, B., "Dynamic Host Configuration Protocol (DHCP) Domain Search Option", RFC 3397, November 2002.

[RFC3403] Mealling, M., "Dynamic Delegation Discovery System (DDDS) Part Three: The Domain Name System (DNS) Database", RFC 3403, October 2002.

[RFC3646] Droms, R., "DNS Configuration options for Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 3646, December 2003.

[RFC3958] Daigle, D. and A. Newton, "Domain-Based Application Service Location Using SRV RRs and the Dynamic Delegation Discovery Service (DDDS)", RFC 3958, January 2005.

[RFC4033] Arends, R., "DNS Security Introduction and Requirements", RFC 4033, March 2005.

[RFC5440] Le Roux, JL., "Path Computation Element (PCE) Communication Protocol (PCEP)", RFC 5440, April 2007.

[RFC6733] Fajardo, V., "Diameter Base Protocol", RFC 6733, October 2012.

10.2. Informative References

[BGP-LS] Gredler, H., "North-Bound Distribution of Link-State and TE Information using BGP", ID draft-ietf-idr-ls-distribution-10, January 2015.

[RFC1464] Rosenbaum, R., "Using the Domain Name System To Store Arbitrary String Attributes", RFC 1464, May 1993.

[RFC2385] Heffernan, A., "Protection of BGP Sessions via the TCP MD5 Signature Option", RFC 2385, August 1998.

[RFC2845] Vixie, P., Gudmundsson, O., Eastlake 3rd, D., and B. Wellington, "Secret Key Transaction Authentication for DNS (TSIG)", RFC 2845, DOI 10.17487/RFC2845, May 2000, <http://www.rfc-editor.org/info/rfc2845>.

[RFC2930] Eastlake 3rd, D., "Secret Key Establishment for DNS (TKEY RR)", RFC 2930, DOI 10.17487/RFC2930, September 2000, <http://www.rfc-editor.org/info/rfc2930>.

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

[RFC4674] Droms, R., "Requirements for Path Computation Element (PCE) Discovery", RFC 4674, December 2003.

[RFC4927] Le Roux, JL., "Path Computation Element Communication Protocol (PCECP) Specific Requirements for Inter-Area MPLS and GMPLS Traffic Engineering", RFC 4927, June 2007.

[RFC5088] Le Roux, JL., "OSPF Protocol Extensions for Path Computation Element (PCE) Discovery", RFC 5088, January 2008.

[RFC5089] Le Roux, JL., "IS-IS Protocol Extensions for Path Computation Element (PCE) Discovery", RFC 5089, January 2008.

[RFC5152] Vasseur, JP., Ed., Ayyangar, A., Ed., and R. Zhang, "A Per-Domain Path Computation Method for Establishing Inter-Domain Traffic Engineering (TE) Label Switched Paths (LSPs)", RFC 5152, DOI 10.17487/RFC5152, February 2008, <http://www.rfc-editor.org/info/rfc5152>.

[RFC5234] Crocker, D., Ed. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, DOI 10.17487/RFC5234, January 2008, <http://www.rfc-editor.org/info/rfc5234>.

[RFC5376] Bitar, N., "Inter-AS Requirements for the Path Computation Element Communication Protocol (PCECP)", RFC 5376, November 2008.

[RFC5441] Vasseur, JP., Ed., Zhang, R., Bitar, N., and JL. Le Roux, "A Backward-Recursive PCE-Based Computation (BRPC) Procedure to Compute Shortest Constrained Inter-Domain Traffic Engineering Label Switched Paths", RFC 5441, DOI 10.17487/RFC5441, April 2009, <http://www.rfc-editor.org/info/rfc5441>.

[RFC5452] Hubert, A., "Measures for Making DNS More Resilient against Forged Answers", RFC 5452, January 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, DOI 10.17487/RFC5557, July 2009, <http://www.rfc-editor.org/info/rfc5557>.

[RFC5671] Yasukawa, S. and A. Farrel, Ed., "Applicability of the Path Computation Element (PCE) to Point-to-Multipoint (P2MP) MPLS and GMPLS Traffic Engineering (TE)", RFC 5671, DOI 10.17487/RFC5671, October 2009, <http://www.rfc-editor.org/info/rfc5671>.

[RFC6335] Cotton, M., Eggert, L., Touch, J., Westerlund, M., and S. Cheshire, "Internet Assigned Numbers Authority (IANA) Procedures for the Management of the Service Name and Transport Protocol Port Number Registry", BCP 165, RFC 6335, DOI 10.17487/RFC6335, August 2011, <http://www.rfc-editor.org/info/rfc6335>.

[RFC6408] Jones, M., Korhonen, J., and L. Morand, "Diameter Straightforward-Naming Authority Pointer (S-NAPTR) Usage", RFC 6408, DOI 10.17487/RFC6408, November 2011, <http://www.rfc-editor.org/info/rfc6408>.

[RFC6457] Takeda, T., "PCC-PCE Communication and PCE Discovery Requirements for Inter-Layer Traffic Engineering", RFC 6457, June 2007.

[RFC6781] Kolkman, O., Mekking, W., and R. Gieben, "DNSSEC Operational Practices, Version 2", RFC 6781, December 2012.

[RFC6805] King, D. and A. Farrel, "The Application of the Path Computation Element Architecture to the Determination of a Sequence of Domains in MPLS and GMPLS", RFC 6805, November 2012.

[RFC7025] Otani, T., "Requirements for GMPLS Applications of PCE", RFC 7025, September 2013.

[RFC7399] Farrel, A. and D. King, "Unanswered Questions in the Path Computation Element Architecture", RFC 7399, DOI 10.17487/RFC7399, October 2014, <http://www.rfc-editor.org/info/rfc7399>.

[STATEFUL-PCE] Crabbe, E., Minei, I., Medved, J., and R. Varga, "PCEP Extensions for Stateful PCE", ID draft-ietf-pce-stateful-pce-11, April 2015.

Authors' Addresses

Qin Wu
Huawei
101 Software Avenue, Yuhua District
Nanjing, Jiangsu 210012
China

Email: sunseawq@huawei.com

Dhruv Dhody
Huawei
Leela Palace
Bangalore, Karnataka 560008
INDIA

Email: dhruv.dhody@huawei.com

Daniel King
Lancaster University
UK

Email: daniel@olddog.co.uk

Diego R. Lopez
Telefonica I+D

Email: diego@tid.es

Jeff Tantsura
2330 Central Expressway
Santa Clara, CA 95050
US

Email: Jefftant.ietf@gmail.com