

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.

## 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 May 11, 2015.

Internet-Draft

PCE Path Profiles

November 2014

## Copyright Notice

Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.

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

## Table of Contents

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

[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 definition 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. Alternatively, 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 completely 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 simultaneously. 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 document 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 document 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 simultaneously. 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>
```

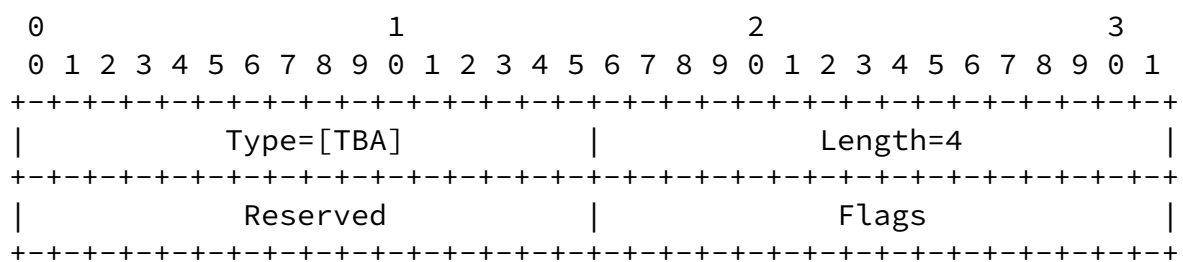
where:

<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):

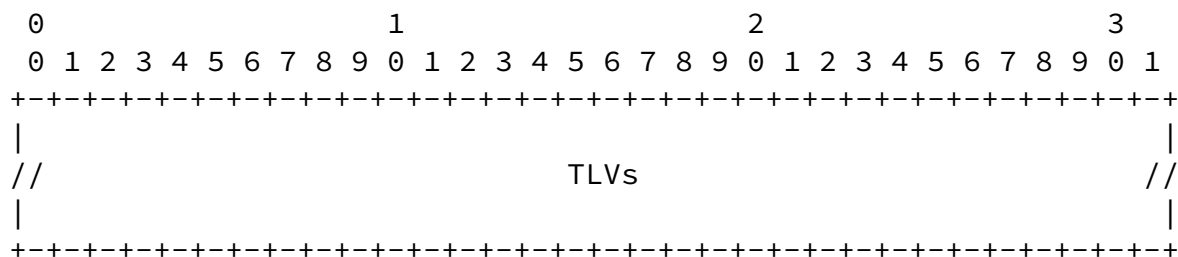
Unassigned bits are considered reserved. They MUST be set to zero on transmission and ignored on receipt. No flags are currently defined.

### 5.2. PATH-PROFILE Object

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

PATH-PROFILE Object-Class is [TBA].

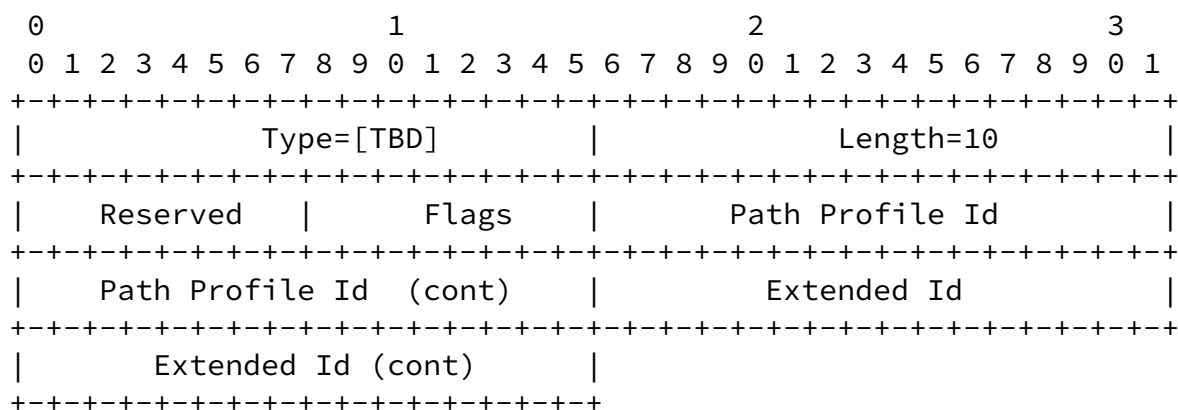
PATH-PROFILE Object-Type is 1.



## 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.

Alvarez, et al.

Expires May 11, 2015

[Page 11]

---

Internet-Draft

PCE Path Profiles

November 2014

- [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](mailto:saalvare@cisco.com)

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

Email: [msiva@cisco.com](mailto:msiva@cisco.com)

Alvarez, et al.

Expires May 11, 2015

[Page 12]

---

Internet-Draft

PCE Path Profiles

November 2014

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

Email: [zali@cisco.com](mailto:zali@cisco.com)

Luis Tomotaki  
Verizon  
400 International  
Richardson, TX 75081  
US

Email: [luis.tomotaki@verizon.com](mailto: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