PCE Working Group
Internet-Draft
Intended status: Standards Track
Expires: May 3, 2018

H. Chen
D. Dhody
Huawei Technologies

October 30, 2017

A Forward-Search P2P TE LSP Inter-Domain Path Computation draft-chen-pce-forward-search-p2p-path-computation-14

Abstract

This document presents a forward search procedure for computing paths for Point-to-Point (P2P) Traffic Engineering (TE) Label Switched Paths (LSPs) crossing a number of domains using multiple Path Computation Elements (PCEs). In addition, extensions to the Path Computation Element Communication Protocol (PCEP) for supporting the forward search procedure are described.

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 https://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 3, 2018.

Table of Contents

1. Introduction

It would be useful to extend MPLS TE capabilities across multiple domains (i.e., IGP areas or Autonomous Systems) to support inter-domain resources optimization, to provide strict QoS guarantees between two edge routers located within distinct domains.

[RFC4105] "Requirements for Inter-Area MPLS TE" lists the requirements for computing a shortest path for a TE LSP crossing multiple IGP areas; and [RFC4216] "MPLS Inter-Autonomous System (AS) TE Requirements" describes the requirements for computing a shortest path for a TE LSP crossing multiple ASes. [RFC4655] "A PCE-Based Architecture" discusses centralized and distributed computation models for the computation of a path for a TE LSP crossing multiple domains.

This document presents a forward search procedure to address these requirements using multiple Path Computation Elements (PCEs). This procedure guarantees that the path found from the source to the destination is shortest. It does not depend on any sequence of domains from the source node to the destination node. Navigating a mesh of domains is simple and efficient.

2. Terminology

The following terminology is used in this document.

ABR: Area Border Router. Router used to connect two IGP areas (Areas in OSPF or levels in IS-IS).

ASBR: Autonomous System Border Router. Router used to connect together ASes of the same or different service providers via one or more inter-AS links.

BN: Boundary Node. A boundary node is either an ABR in the context of inter-area Traffic Engineering or an ASBR in the context of inter-AS Traffic Engineering.

Entry BN of domain(n): a BN connecting domain(n-1) to domain(n) along the path found from the source node to the BN, where domain(n-1) is the previous hop domain of domain(n).

Exit BN of domain(n): a BN connecting domain(n) to domain(n+1) along the path found from the source node to the BN, where domain(n+1) is the next hop domain of domain(n).

Inter-area TE LSP: a TE LSP that crosses an IGP area boundary.

Inter-AS TE LSP: a TE LSP that crosses an AS boundary.

LSP: Label Switched Path

LSR: Label Switching Router

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.

PCE(i): a PCE with the scope of domain(i).

TED: Traffic Engineering Database.

This document uses terminology defined in [RFC5440].

3. 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].

4. Requirements

This section summarizes the requirements specific for computing a path for a P2P Traffic Engineering (TE) LSP crossing multiple domains (areas or ASes). More requirements for Inter-Area and Inter-AS MPLS Traffic Engineering are described in [RFC4105] and [RFC4216].

A number of requirements specific for a solution to compute a path for a P2P TE LSP crossing multiple domains is listed as follows:

1. The solution SHOULD provide the capability to compute a shortest path dynamically, satisfying a set of specified constraints across multiple IGP areas.

2. The solution MUST provide the ability to reoptimize in a minimally disruptive manner (make before break) an inter-area TE LSP, should a more optimal path appear in any traversed IGP area.

3. The solution SHOULD provide mechanism(s) to compute a shortest end-to-end path for a TE LSP crossing multiple ASes and satisfying a set of specified constraints dynamically.

4. Once an inter-AS TE LSP has been established, and should there be any resource or other changes inside anyone of the ASes, the solution MUST be able to re-optimize the LSP accordingly and non-disruptively, either upon expiration of a configurable timer or upon being triggered by a network event or a manual request at the TE tunnel Head-End.

5. Forward Search Path Computation

This section gives an overview of the forward search path computation procedure (FSPC for short) to satisfy the requirements described above and describes the procedure in detail.

5.1. Overview of Procedure

Simply speaking, the idea of FSPC for computing a path for an MPLS TE P2P LSP crossing multiple domains from a source node to a destination node includes:

Start from the source node and the source domain.

Consider the optimal path segment from the source node to every exit boundary node of the source domain as a special link;

Consider the optimal path segment from an entry boundary node to every exit boundary node and the destination node of a domain as a special link; and the optimal path segment is computed as needed.

The whole topology consisting of many domains can be considered as a special topology, which contains those special links and the inter-domain links.

Compute an optimal path in this special topology from the source node to the destination node using CSPF.

5.2. Description of Procedure

Suppose that we have the following variables:

A current PCE named as CurrentPCE which is currently computing the path.

A candidate node list named as CandidateNodeList, which contains the nodes to each of which the temporary optimal path from the source node is currently found and satisfies a set of given constraints. The information about each node C in CandidateNodeList consists of:

o the cost of the path from the source node to node C,

- the hopcount of the path from the source node to node C,
- the previous hop node P and the link between P and C,
- the domain list of C (ABR or ASBR) where C has visibility to multiple domains and clearly mark the domain by which C is added to CandidateNodeList,
- the PCE responsible for C (i.e., the PCE responsible for the domain containing C. Alternatively, we may use the above mentioned domain instead of the PCE.), and
- the flags for C.

The flags include:

- bit D indicating that C is a Destination node if it is set,
- bit S indicating that C is the Source node if it is set,
- bit T indicating that C is on result path Tree if it is set.

The nodes in CandidateNodeList are ordered by path cost. Initially, CandidateNodeList contains only a Source Node, with path cost 0, PCE responsible for the source domain.

A result path list or tree named as ResultPathTree, which contains the final optimal paths from the source node to the boundary nodes or the destination node. Initially, ResultPathTree is empty.

Alternatively, the result path list or tree can be combined into the CandidateNodeList. We may set bit T to one in the NODE-FLAGS object for the candidate node when grafting it into the existing result path list or tree. Thus all the candidate nodes with bit T set to one in the CandidateNodeList constitute the result path tree or list.

FSPC for computing the path for the MPLS TE P2P LSP is described as follows:

Initially, a PCC sets ResultPathTree to empty and CandidateNodeList to contain the source node and sends PCE responsible for the source domain a PCReq with the source node, the destination node, CandidateNodeList and ResultPathTree.

When the PCE responsible for a domain (called current domain) receives a request for computing the path for the MPLS TE P2P LSP, it obtains node Cm with the minimum path cost in the CandidateNodeList. The node Cm is the first node in the CandidateNodeList, which is

sorted by path cost. It checks whether the CurrentPCE is the PCE responsible for the node Cm(always expand node Cm only if it is an entry boundary node). If it is, then remove Cm from CandidateNodeList and graft it into ResultPathTree (i.e., set flag bit T of node Cm to one); otherwise, a PCReq message is sent to the PCE for node Cm (see Section 5.3 for the case that there is not any direct session between the CurrentPCE and the PCE for node Cm).

Suppose that node Cm is in the current domain. The ResultPathTree is built from Cm in the following steps.

If node Cm is the destination node, then the optimal path from the source node to the destination node is found, and a PCRep message with the path is sent to the PCE/PCC which sends the request to the CurrentPCE.

If node Cm is an entry boundary node or the source node, then the optimal path segments from node Cm to the destination node (if it is in the current domain) and every exit boundary node of the current domain that is not on the result path tree and satisfies the given constraints are computed through using CSPF and as special links.

For every node N connected to node Cm through a special link (i.e., the optimal path segment satisfying the given constraints), it is merged into CandidateNodeList. The cost to node N is the sum of the cost to node Cm and the cost of the special link (i.e., the path segment) between Cm and N. If node N is not in the CandidateNodeList, then node N is added into the list with the cost to node N, node Cm as its previous hop node and the PCE for node N. The PCE for node N is the CurrentPCE if node N is an ASBR; otherwise (node N is an ABR, an exit boundary node of the current domain and an entry boundary node of the domain next to the current domain) the PCE for node N is the PCE for the next domain. If node N is in the CandidateNodeList and the cost to node N through node Cm is less than the cost to node N in the list, then replace the cost to node N in the list with the cost to node N through node Cm and the previous hop to node N in the list with node Cm.

If node Cm is an exit boundary node and there are inter-domain links connecting to it (i.e., node Cm is an ASBR) and satisfying the constraints, then for every node N connecting to Cm, satisfying the constraints and not on the result path tree, it is merged into the CandidateNodeList. The cost to node N is the sum of the cost to node Cm and the cost of the link between Cm and N. If node N is not in the CandidateNodeList, then node N is added into the list with the cost to node N, node Cm as its previous hop node and the PCE for node N. If node N is in the CandidateNodeList and the cost to node N through node Cm is less than the cost to node N in the list, then

replace the cost to node N in the list with the cost to node N through node Cm and the previous hop to node N in the list with node Cm.

After the CandidateNodeList is updated, there will be a new node Cm with the minimum cost in the updated CandidateNodeList. If the CurrentPCE is the same as the PCE for the new node Cm, then the node Cm is removed from the CandidateNodeList and grafted to ResultPathTree (i.e., set flag bit T of node Cm to one), and the above steps are repeated; otherwise, a request message is to be sent to the PCE for node Cm.

Note that if node Cm has visibility to multiple domains, do not remove it from the CandidateNodeList until it is expanded in all domains. Also mark in the domain list of node Cm, for which domains it is already expanded.

5.3. Processing Request and Reply Messages

In this section, we describe the processing of the request and reply messages with Forward search bit set for FSPC. Each of the request and reply messages mentioned below has its Forward search bit set even though we do not indicate this explicitly.

In the case that a reply message is a final reply, which contains the optimal path from the source to the destination, the reply message is sent toward the PCC along the path that the request message goes from the PCC to the current PCE in reverse direction.

In the case that a request message is to be sent to the PCE for node Cm with the minimum cost in the CandidateNodeList and there is a PCE session between the current domain and the next domain containing node Cm, the CurrentPCE sends the PCE for node Cm through the session a request message with the source node, the destination node, CandidateNodeList and ResultPathTree.

In the case that a request message is to be sent to the PCE for node Cm and there is not any PCE session between the CurrentPCE and the PCE for node Cm, a request message with T bit set to one in RP is sent toward a branch point on the result path tree from the current domain along the path that the request message goes from the PCC to the CurrentPCE in reverse direction. From the branch point, there is a downward path to the domain containing the previous hop node of node Cm on the result path tree and to the domain containing node Cm. At this branch point, the request message with T bit set to zero is sent to the PCE for node Cm along the downward path.

6. Comparing to BRPC

[RFC5441] describes the Backward Recursive Path Computation (BRPC) algorithm or procedure for computing an MPLS TE P2P LSP path from a source node to a destination node crossing multiple domains. Comparing to BRPC, there are a number of differences between BRPC and the Forward-Search P2P TE LSP Inter-Domain Path Computation (FSPC). Some of the differences are briefed below.

First, for BRPC to compute a shortest path from a source node to a destination node crossing multiple domains, we MUST provide a sequence of domains from the source node to the destination node to BRPC in advance. It is a big burden and very challenging for users to provide a sequence of domains for every LSP path crossing domains in general. In addition, it increases the cost of operation and maintenance of the network. FSPC does not need any sequence of domains for computing a shortest path.

Secondly, for a given sequence of domains domain(1), domain(2), ... ,domain(n), BRPC searches the shortest path from domain(n), to domain(n-1), until domain(1) along the reverse order of the given sequence of domain. It will get the shortest path within the given domain sesuence. FSPC calculates an optimal path in a special topology from the source node to the destination node. It will find the shortest path within all the domains.

Moreover, if the sequence of domains from the source node to the destination node provided to BRPC does not contain the shortest path from the source to the destination, then the path computed by BRPC is not optimal. FSPC guarantees that the path found is optimal.

7. Extensions to PCEP

This section describes the extensions to PCEP for FSPC. The extensions include the definition of a new flag in the RP object, a result path list and a candidate node list in the PCReq and PCRep message.

7.1. RP Object Extension

The following flags are added into the RP Object:

The F bit is added in the flag bits field of the RP object to tell the receiver of the message that the request/reply is for FSPC.

- F (FSPC bit - 1 bit):
  - 0: This indicates that this is not a PCReq/PCRep for FSPC.
  - 1: This indicates that this is a PCReq or PCRep for FSPC.

The T bit is added in the flag bits field of the RP object to tell the receiver of the message that the request is for transferring a request message to the domain containing the node with minimum cost in the candidate list.

- o T (Transfer request bit - 1 bit):
  - 0: This indicates that this is not a PCReq for transferring a request message.
  - 1: This indicates that this is a PCReq message for transferring a request message.

Setting Transfer request T-bit in a RP Object to one indicates that a reqest message containing the RP Object is for transferring a request message to the domain containing the node with minimum cost in the candidate list.

The IANA request is referenced in Section below (Request Parameter Bit Flags) of this document.

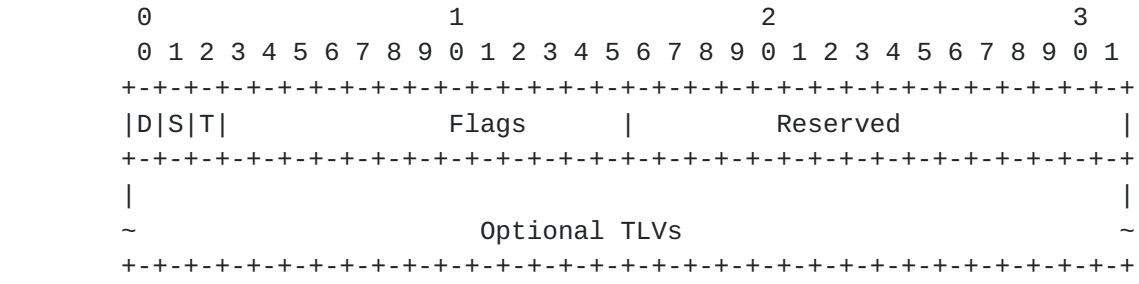
This F bit with the N bit defined in [RFC6006] can indicate whether the request/reply is for FSPC of an MPLS TE P2P LSP or an MPLS TE P2MP LSP.

- o F = 1 and N = 0: This indicates that this is a PCReq/PCRep message for FSPC of an MPLS TE P2P LSP.
- o F = 1 and N = 1: This indicates that this is a PCReq/PCRep message for FSPC of an MPLS TE P2MP LSP.

7.2. NODE-FLAGS Object

The NODE-FLAGS object is used to indicate the characteristics of the node in a Candidate Node List in a request or reply message for FSPC. The NODE-FLAGS object comprises a Reserved field, and a number of Flags. The NODE-FLAGS object may also contain a set of TLVs.

The format of the NODE-FLAGS object body is as follows:



NODE-FLAGS Object Body

where

o D = 1: The node is a destination node.

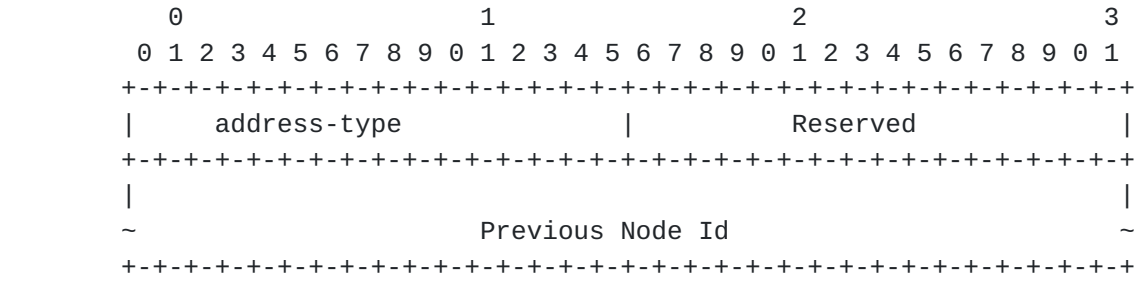
o S = 1: The node is a source node.

o T = 1: The node is on the result path tree.

Following are the TLVs defined to convey the characteristics of the candidate node.

7.2.1. PREVIOUS-NODE TLV

The PREVIOUS-NODE TLV contains the Previous Node Id of the candidate node. The PREVIOUS-NODE TLV has the following format:



PREVIOUS-NODE TLV format

The Type of PREVIOUS-NODE TLV is to be assigned by IANA.

The length is 8 if the address type is IPv4 or 20 if the address type is IPV6.

Address Type (16 bits): Indicates the address type of Previous Node Id. 1 means IPv4 address type, 2 means IPv6 address type.

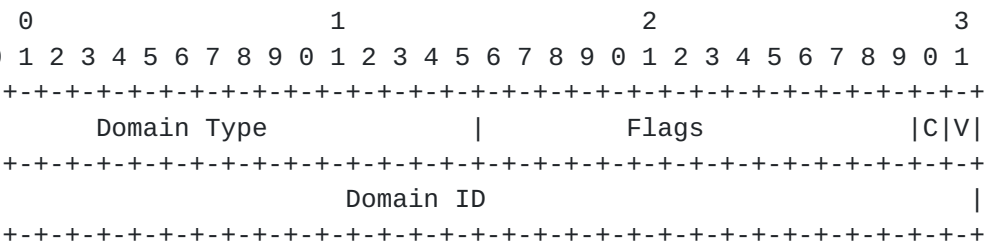
Reserved(16 bits): SHOULD be set to zero on transmission and MUST be ignored on receipt.

Previous Node Id : IP address of the node.

7.2.2. DOMAIN-ID TLV

The DOMAIN-ID TLV contains the domain Id of the candidate node (ABR or ASBR). The NODE-FLAG Object may include multiple DOMAIN-ID TLVs when the candidate node has visibility into multiple Domains.

The DOMAIN-ID TLV has the following format:



DOMAIN-ID TLV format

The Type of DOMAIN-ID TLV is to be assigned by IANA.

The length is 8.

Domain Type (8 bits): Indicates the domain type. There are two types of domain defined currently:

o Type=1: the Domain ID field carries an IGP Area ID.

o Type=2: the Domain ID field carries an AS number.

C Flag (1 bit): If the flag is set to 1, it indicates the candidate node is added into Candidate Node List by this domain.

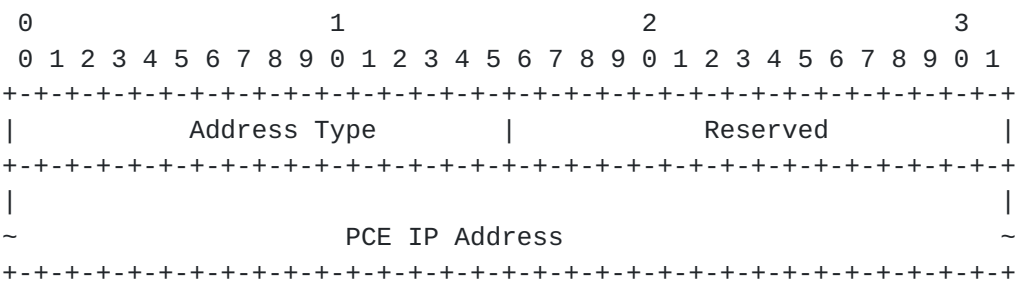
V Flag (1 bit): If the flag is set to 1, it indicates the candidate node has been expanded in this domain.

Domain ID (32 bits): With the Domain Type set to 1, this indicates the 32-bit Area ID of an IGP area where the candidate belongs. With Domain Type set to 2, this indicates an AS number of an AS where the candidate belongs. When the AS number is coded in two octets, the AS Number field MUST have its first two octets set to 0.

[Editor's note: [PCE-HIERARCHY-EXT], section 3.1.3 deals with the encoding of Domain-Id TLV in OPEN Object. Later on DOMAIN-ID TLV defined here can be incorporate with this draft]

7.2.3. PCE-ID TLV

The PCE-ID TLV is used to indicate the PCE that added this node to the CandidateList. The PCE-ID TLV has the following format:



PCE-ID TLV format

The type of PCE-ID TLV is to be assigned by IANA.

The length is 8.

Address Type (16 bits): Indicates the address type of PCE IP Address. 1 means IPv4 address type, 2 means IPv6 address type.

PCE IP Address: Indicates the reachable address of a PCE.

[Editor's note: [PCE-HIERARCHY-EXT], section 3.1.4 deals with the encoding of PCE-Id TLV in OPEN Object. Later on PCE-ID TLV defined here can be incorporate with this draft]

7.3. Candidate Node List

The Candidate Node List has the following format:

```
<candidate-node-list>::= <node>
                         [<candidate-node-list>]
where

<node>::= <ERO>  <NODE-FLAGS>
          <attribute-list>

<attribute-list>::=<metric-list>
                   [<IRO>]

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

The ERO in a candidate node contain just the path segment of the last link of the path, which is from the previous hop node of the tail end node of the path to the tail end node. With this information, we can graft the candidate node into the existing result path list or tree.

Simply speaking, a candidate node has the same or similar format of a path defined in [RFC5440], but the ERO in the candidate node just contain the tail end node of the path and its previous hop, and the candidate path may contain a new object NODE-FLAGS along with new TLVs.

7.4. Result Path List

The Result Path List has the following format:

```
<result-path-list>::= <node>
                      [<result-path-list>]
where

<node>::= <ERO>  <NODE-FLAGS>
          <attribute-list>

<attribute-list>::=<metric-list>
                   [<IRO>]

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

The usage of ERO, NODE-FLAGS objects etc, is similar to Candidate Node List. The T-bit of NODE-FLAGS Object would be set denoting that the best path to this node is already found.

7.5. Request Message Extension

Below is the message format for a request message with the extension of result-path-list and candidate-node-list:

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

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

<request>::= <RP> <END-POINTS> [<OF>] [<LSPA>] [<BANDWIDTH>]
             [<metric-list>]  [<RRO>[<BANDWIDTH>]]  [<IRO>]
             [<LOAD-BALANCING>]
             [<result-path-list>]
             [<candidate-node-list>]
   where:
             <result-path-list> and <candidate-node-list>
             are as defined above.
```

7.6. Reply Message Extension

Below is the message format for a reply message with the extension of result-path-list and candidate-node-list:

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

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

<response> ::= <RP> [<NO-PATH>] [<attribute-list>]
               [<path-list>]
               [<result-path-list>]
               [<candidate-node-list >]
where:
              <result-path-list> and <candidate-node-list>
              are as defined above.
```

If the path from the source to the destination is found, the reply message contains path-list comprising the information of the path.

8. Suggestion to improve performance

To get much better performance all the candidate nodes of current domain can be expanded before moving on to a new domain. Note in this case, after expanding the least cost candidate node, PCE can look for and expand any other candidates in this domain.

9. Manageability Considerations

9.1. Control of Function and Policy

TBD

9.2. Information and Data Models

TBD

9.3. Liveness Detection and Monitoring

TBD

9.4. Verify Correct Operations

TBD

9.5. Requirements On Other Protocols

TBD

9.6. Impact On Network Operations

TBD

10. Security Considerations

The mechanism described in this document does not raise any new security issues for the PCEP protocols.

11. IANA Considerations

This section specifies requests for IANA allocation.

11.1. Request Parameter Bit Flags

Two new RP Object Flags have been defined in this document. IANA is requested to make the following allocation from the "PCEP RP Object Flag Field" Sub-Registry:

Bit	Description		Reference
TBA	FSPC	(F-bit)	This I-D
TBA	Transfer Request	(T-bit)	This I-D

Setting FSPC F-bit in a RP Object to one indicates that a request/ reply message containing the RP Object is for FSPC.

Setting Transfer Request T-bit in a RP Object to one indicates that a request message containing the RP Object is for transferring a request message to the domain containing the node with minimum cost in the candidate list.

11.2. New PCEP Object

11.2.1. NODE-FLAGS Object

The NODE-FLAGS Object-Type and Object-Class has been defined in this document. IANA is requested to make the following allocation:

NODE-FLAGS Object-Type : TBA

NODE-FLAGS Object-Class: TBA

Flag field of the NODE-FLAG Object:

Bit	Description	Reference
0	The node is a destination node	This I-D
1	The node is a source node	This I-D
2	The node is on the result path tree	This I-D

Each bit should be tracked with the following qualities:

- Bit number (counting from bit 0 as the most significant bit)
- Name flag
- Reference

11.3. New PCEP TLV

IANA is requested to make the following allocation:

Value	Meaning	Reference
TBA	DOMAIN-ID TLV	This I-D
TBA	PCE-ID TLV	This I-D
TBA	PREVIOUS-NODE TLV	This I-D

11.3.1. DOMAIN-ID TLV

IANA is requested to make the following allocation:

Flag field of the DOMAIN-ID TLV

Bit	Name	Description	Reference
15	V-bit	Candidate Node has been expanded by the domain	This I-D
14	C-bit	Candidate Node added by the domain	This I-D

Each bit should be tracked with the following qualities:

- Bit number (counting from bit 0 as the most significant bit)
- Name flag
- Reference

12. Acknowledgement

The authors would like to thank Julien Meuric, Daniel King, Ramon Casellas, Cyril Margaria,Olivier Dugeon, Oscar Gonzalez de Dios, Udayasree Palle, Reeja Paul and Sandeep Kumar Boina for their valuable comments on this draft.

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, <https://www.rfc-editor.org/info/rfc2119>.

[RFC4655] Farrel, A., Vasseur, J., and J. Ash, "A Path Computation Element (PCE)-Based Architecture", RFC 4655, DOI 10.17487/RFC4655, August 2006, <https://www.rfc-editor.org/info/rfc4655>.

[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>.

13.2. Informative References

[RFC4105] Le Roux, J., Ed., Vasseur, J., Ed., and J. Boyle, Ed., "Requirements for Inter-Area MPLS Traffic Engineering", RFC 4105, DOI 10.17487/RFC4105, June 2005, <https://www.rfc-editor.org/info/rfc4105>.

[RFC4216] Zhang, R., Ed. and J. Vasseur, Ed., "MPLS Inter-Autonomous System (AS) Traffic Engineering (TE) Requirements", RFC 4216, DOI 10.17487/RFC4216, November 2005, <https://www.rfc-editor.org/info/rfc4216>.

[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, <https://www.rfc-editor.org/info/rfc5441>.

[RFC6006] Zhao, Q., Ed., King, D., Ed., 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, DOI 10.17487/RFC6006, September 2010, <https://www.rfc-editor.org/info/rfc6006>.

[PCE-HIERARCHY-EXT]
Zhang, F., Zhao, Q., King, O., Casellas, R., and D. King, "Extensions to Path Computation Element Communication Protocol (PCEP) for Hierarchical Path Computation Elements (PCE) (draft-zhang-pce-hierarchy-extensions-02)", August 2012.

Authors' Addresses

Huaimo Chen
Huawei Technologies
Boston, MA
USA

EMail: Huaimo.chen@huawei.com

Dhruv Dhody
Huawei Technologies
Leela Palace
Bangalore, Karnataka 560008
INDIA

EMail: dhruv.dhody@huawei.com