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 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 17, 2013.

Copyright Notice

Copyright (c) 2012 IETF Trust and the persons identified as the document authors. All rights reserved.

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

Table of Contents

1. Introduction ........................................... 3
2. Terminology ........................................... 3
3. Conventions Used in This Document ....................... 4
4. Requirements ........................................... 4
5. Forward Search Path Computation ........................ 5
   5.1. Overview of Procedure ............................... 5
   5.2. Description of Procedure ............................ 5
   5.3. Processing Request and Reply Messages .............. 8
6. Comparing to BRPC ...................................... 9
7. Extensions to PCEF ..................................... 10
   7.1. RP Object Extension ................................. 10
   7.2. PCE Object .......................................... 11
   7.3. Node Flags Object .................................... 12
   7.4. Candidate Node List Object ........................ 12
   7.5. Request Message Extension ........................... 13
   7.6. Reply Message Extension ............................. 14
8. Security Considerations ................................ 15
9. IANA Considerations .................................... 15
   9.1. Request Parameter Bit Flags ......................... 15
10. Acknowledgement ....................................... 15
11. References ............................................ 15
   11.1. Normative References ............................... 16
   11.2. Informative References ............................. 16
Authors’ Addresses ....................................... 16
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.

RFC 4105 "Requirements for Inter-Area MPLS TE" lists the requirements for computing a shortest path for a TE LSP crossing multiple IGP areas; and RFC 4216 "MPLS Inter-Autonomous System (AS) TE Requirements" describes the requirements for computing a shortest path for a TE LSP crossing multiple ASes. RFC 4655 "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

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

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

Boundary Node (BN): 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) is a PCE with the scope of domain(i).

TED: Traffic Engineering Database.

This document uses terminology defined in RFC 5440.

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 RFC 4105 and RFC 4216.

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 to satisfy the requirements described above and describes the procedure in detail.

5.1. Overview of Procedure

Simply speaking, the idea of the forward search path computation procedure 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,

o the previous hop node P and the link between P and C,

o the PCE responsible for C (i.e., the PCE responsible for the domain containing C. Alternatively, we may use the domain instead of the PCE.), and

o 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 E indicating that C is an Exit boundary node if it is set,
  * bit I indicating that C is an entry boundary node if it is set, and
  * 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 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.

The Forward Search Path Computation procedure 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 PCC 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
If node \(C_m\) 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 \(C_m\) is an entry boundary node or the source node, then the optimal path segments from node \(C_m\) 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 \(C_m\) 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 \(C_m\) and the cost of the special link (i.e., the path segment) between \(C_m\) 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 \(C_m\) 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 \(C_m\) 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 \(C_m\) and the previous hop to node \(N\) in the list with node \(C_m\).

If node \(C_m\) is an exit boundary node and there are inter-domain links connecting to it (i.e., node \(C_m\) is an ASBR) and satisfying the constraints, then for every node \(N\) connecting to \(C_m\), 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 \(C_m\) and the cost of the link between \(C_m\) 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 \(C_m\) 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 \(C_m\) 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 \(C_m\) and the previous hop to node \(N\) in the list with node \(C_m\).
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.

### 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 forward search inter-domain path computation. 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 reply message 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 is sent to the PCE for node Cm along the downward path.

Suppose that node Cm has the minimum cost in CandidateNodeList when a PCE receives a request message or a reply message containing CandidateNodeList.

When a PCE (CurrentPCE) for a domain (current domain) receives a reply message PCRep, it checks whether the reply is a final reply with the optimal path from the source to the destination. If the reply is the final reply, the CurrentPCE sends the reply to the PCE that sends the request to the CurrentPCE; otherwise, it checks...
whether there is a path from the current domain to the domain containing the previous hop node of node Cm on ResultPathTree and to the domain containing node Cm. If there is a path, the PCE sends a request PCReq to the PCE responsible for the next domain along the path; otherwise, it sends a reply PCRep to the PCE that sends the request to the CurrentPCE.

When a PCE receives a request PCReq, it checks whether the current domain contains node Cm. If it does, then node Cm is removed from CandidateNodeList and grafted to ResultPathTree (i.e., set flag bit T of node Cm to one), and the above steps in the previous sub section are repeated; otherwise, the PCE sends a request PCReq to the PCE responsible for the next domain along the path from the current domain to the domain containing the previous hop node of node Cm on ResultPathTree and to the domain containing node Cm.

6. Comparing to BRPC

RFC 5441 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. 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. The Forward-Search P2P TE LSP Inter-Domain Path Computation 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 sequence. The Forward-Search P2P TE LSP Inter-Domain Path Computation calculates an optimal path in a special topology from the source node to the destination node using CSPF. 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. The Forward-Search P2P TE LSP Inter-Domain Path Computation guarantees that the path found is optimal.

7. Extensions to PCEP

This section describes the extensions to PCEP for Forward Search Path Computation. 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 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 Forward Search Path Computation.

- F (Forward search Path Computation bit - 1 bit):
  - 0: This indicates that this is not a PCReq/PCRep for Forward Search Path Computation.
  - 1: This indicates that this is a PCReq or PCRep message for Forward Search Path Computation.

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

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

Setting Transfer request T-bit in a RP Object to one indicates that a reply 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 Forward Search Path Computation of an MPLS TE P2P LSP or an MPLS TE P2MP LSP.

- F = 1 and N = 0: This indicates that this is a PCReq/PCRep message for Forward Search Path Computation of an MPLS TE P2P LSP.

- F = 1 and N = 1: This indicates that this is a PCReq/PCRep message for Forward Search Path Computation of an MPLS TE P2MP LSP.

7.2. PCE Object

The figure below illustrates a PCE IPv4 object body (Object-Type=1), which comprises a PCE IPv4 address. The PCE IPv4 address object indicates the IPv4 address of a PCE, with which a PCE session may be established and to which a request message may be sent.

```
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       PCE  IPv4 address                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 1: PCE Object Body for IPv4

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

```
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| PCE  IPv6 address (16 bytes)                     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 2: PCE Object Body for IPv6
7.3. 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 Forward Search Inter-domain Path Computation. The Node Flags object comprises a Reserved field, and a number of Flags.

The format of the Node Flags object body is as follows:

```
+-----------------+-----------------+-----------------+
| D | S | I | E | T | Flags | Reserved |
+-----------------+-----------------+-----------------+
```

Figure 3: Node Flags Object Body

where

- D = 1: The node is a destination node.
- S = 1: The node is a source node.
- I = 1: The node is an entry boundary node.
- E = 1: The node is an exit boundary node.
- T = 1: The node is on the result path tree.

7.4. Candidate Node List Object

The candidate-node-list-obj object contains the nodes in the candidate node list. A new PCEP object class and type are requested for it. The format of the candidate-node-list-obj object body is as follows:

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

Figure 4: Candidate Node List Object

The following is the definition of candidate node list, which may contain Node Flags.
<candidate-node-list>::= <candidate-node>
[<candidate-node-list>]
<candidate-node>::= <ERO>
<candidate-attribute-list>
<candidate-attribute-list>::= [<attribute-list>]
[<PCE>]
[<Node-Flags>]
<attribute-list>::= [<LSPA>]
[<BANDWIDTH>]
[<metric-list>]
[<IRO>]

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 RFC 5440, 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 two new objects PCE and node flags.

7.5. Request Message Extension

Below is the message format for a request message with the extension of a result path list and a candidate node list:
The definition for the result path list that may be added into a request message is the same as that for the path list in a reply message that is described in RFC5440.

7.6. Reply Message Extension

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

```
<PCRep Message> ::= <Common Header>
   <response-list>
   <response> ::= <RP>
      [NO-PATH]
      [attribute-list]
      [path-list]
      [result-path-list]
      [candidate-node-list-obj]
```

where:

- `<candidate-node-list-obj>` contains a `<candidate-node-list>`

Chen & Dhody Expires January 17, 2013 [Page 14]
If the path from the source to the destination is not found yet and there are still chances to find a path (i.e., the candidate list is not empty), the reply message contains candidate-node-list-obj consisting of the information of the candidate list, which is encoded. In this case, the Transfer request T-bit in the RP Object is set to one.

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

8. Security Considerations

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

9. IANA Considerations

This section specifies requests for IANA allocation.

9.1. 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:

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>18</td>
<td>Forward Path Computation (F-bit)</td>
<td>This I-D</td>
</tr>
<tr>
<td>19</td>
<td>Transfer Request (T-bit)</td>
<td>This I-D</td>
</tr>
</tbody>
</table>

Setting Forward Path Computation F-bit in a RP Object to one indicates that a request/reply message containing the RP Object is for forward path computation.

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

10. Acknowledgement

The authors would like to thank Julien Meuric, Daniel King, Cyril Margaria, Ramon Casellas, Olivier Dugeon, and Oscar Gonzalez de Dios for their valuable comments on this draft.
11. References

11.1. Normative References


11.2. Informative References


Authors' Addresses

Huaimo Chen
Huawei Technologies
Boston, MA
USA

Email: Huaimochen@huawei.com

Dhruv Dhody
Huawei Technologies
Leela Palace, Bangalore, Karnataka 560008
INDIA

Email: dhruv.dhody@huawei.com