PCE Working Group H. Chen
Internet-Draft Futurewei
Intended status: Standards Track D. Dhody
Expires: January 8, 2020 Huawei Technologies
July 7, 2019

A Forward-Search P2MP TE LSP Inter-Domain Path Computation
draft-chen-pce-forward-search-p2mp-path-15

Abstract

This document presents a forward search procedure for computing a path for a Point-to-MultiPoint (P2MP) Traffic Engineering (TE) Label Switched Path (LSP) crossing a number of domains through 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

Copyright Notice

Table of Contents

1. Introduction

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 5671 "Applicability of PCE to P2MP MPLS and GMPLS TE" examines the applicability of PCE to path computation for P2MP TE LSPs in MPLS and GMPLS networks.

This document presents a forward search procedure to address these requirements for computing a path for a P2MP TE LSP crossing domains through using multiple Path Computation Elements (PCEs).

The procedure is called "Forward Search Shortest P2MP LSP Path Crossing Domains" or FSPC for short. The major characteristics of this procedure for computing a path for a P2MP TE LSP from a source

node to a number of destination nodes crossing multiple domains include the following three ones.

1. It guarantees that the path computed from the source node to the destination nodes is shortest.

2. It does not depend on any domain path tree or domain sequences from the source node to the destination nodes.

3. 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 terminologies 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 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 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 P2MP Path Computation

This section gives an overview of the forward search path computation procedure (FSPC) to satisfy the requirements for computing a path for a P2MP TE LSP crossing multiple domains described above and describes the procedure in details.

5.1. Overview of Procedure

Simply speaking, the idea of FSPC for computing a path for an MPLS TE P2MP LSP crossing multiple domains from a source node to a number of destination nodes includes:

Start from the source node and the source domain.

Consider the optimal path segment from the source node to every exit boundary and destination 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 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 virtual topology, which contains those special links and the inter-domain links.

Compute a shortest path in this special topology from the source node to the multiple destination nodes using CSPF.

FSPC running at any PCE just grows the result path list/tree in the same way as normal CSPF on the special virtual topology. When the result path list/tree reaches all the destination nodes, the shortest path from the source node to the destination nodes is found and a PCRep message with the path is sent to the PCE/PCC that sends the PCReq message eventually.

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. Each node C in CandidateNodeList has the following information:

- 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

- 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 a Source Node, with path cost 0, PCE responsible for the source domain, and flags with S bit set. It also contains every destination node, with path cost infinity and flags with D bit set.

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

Alternatively, the result path list or tree can be combined into the candidate node list. 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 candidate list constitute the result path tree or list.

FSPC for computing a path for an MPLS TE P2MP LSP crossing a number of domains from a source node to a number of destination nodes can be described as follows:

Initially, a PCC sends a PCE responsible for the source domain a request with CandidateNodeList and ResultPathTree initialized.

When the PCE responsible for a domain (called current domain) receives a request for computing the path for the MPLS TE P2MP LSP, it checks whether the current PCE is the PCE responsible for the node C with the minimum cost in the CandidateNodeList. If it is, then remove C from CandidateNodeList and graft it into ResultPathTree (i.e., set flag bit T of node C to one); otherwise, a PCReq message is sent to the PCE for node C.

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

If node C is a destination node (i.e., the Destination Node (D) bit in the Flags is set), then check whether the path cost to node C is infinity. If it is, then we can not find any path for the P2MP LSP, and a repply message with failure reasons is sent; otherwise, if all the destinations are on the result path tree, then the shortest path is found and a PCRep message with the path is sent to the PCE/PCC which sends the request to the current PCE.

If node C is an entry boundary node or the source node, then the optimal path segments from node C to every destination node 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 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 and the cost of the special link (i.e., the path segment) between C and N. If node N is not in the candidate node list, then node N is added into the list with the cost to node N, node C as its previous hop node and a PCE for node N. The PCE for node N is the current PCE 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 candidate node list and the cost to node N through node C 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 and the previous hop to node N in the list with node C.

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

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

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 D with the minimum cost in the candidate node list and there is a PCE session between the current domain and the next domain containing node D, the current PCE sends the PCE for node D 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 D and there is not any PCE session between the current PCE and the PCE for node D, 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 current PCE in reverse direction. From the branch point, there is a downward path to the domain containing the previous hop node of node D on the result path tree and to the domain containing node D. At this branch point, the request message is sent to the PCE for node D along the downward path.

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

When a PCE (current PCE) 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 current PCE sends the reply to the PCE that sends the request to the current PCE; otherwise, it checks whether there is a path from the current domain to the domain containing the previous hop node of node D on ResultPathTree and to the domain containing node D. 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 current PCE.

When a PCE receives a request PCReq, it checks whether the current domain contains node D. If it does, then node D is removed from CandidateNodeList and grafted to ResultPathTree (i.e., set flag bit T of node D 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 D on ResultPathTree and to the domain containing node D.

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 P2MP TE LSP Inter-Domain Path Computation (FSPC). Some of the differences are briefed below.

At first, BRPC is for computing a shortest path from a source node to a destination node crossing multiple domains. FSPC is for computing a shortest path from a source node to a number of destination nodes crossing multiple domains.

Secondly, 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. FSPC does not need any sequence of domains for computing a shortest inter-domain P2MP path.

Moreover, 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). Thus it is hard for BRPC to be extended for computing a shortest path from a source node to a number of destination nodes crossing multiple domains. FSPC calculates a shortest path in a special topology from the source node to the destination nodes using CSPF.

7. Extensions to PCEP

The extensions to PCEP for FSPC include the definition of a new flag in the RP object, a result path list/tree and a candidate node list in a request message.

7.1. RP Object Extension

The following flag is 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.

```
  o F (FSPC bit - 1 bit):
      0: This indicates that this is not PCReq/PCRep for FSPC.
      1: This indicates that this is PCReq or PCRep message 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 reply 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 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 FSPC of an MPLS TE P2MP LSP or 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.
    o F = 1 and N = 0: This indicates that this is a PCReq/PCRep
                       message for FSPC of an MPLS TE P2P 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
        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                       |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

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

```
        0                   1                   2                   3
        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)                  |
       |                                                               |
       |                                                               |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

7.3. Candidate Node List Object

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

```
       0                   1                   2                   3
       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
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |                                                               |
      //                     <candidate-node-list>                   //
      |                                                               |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The following is the definition of the candidate node list.

```
         <candidate-node-list>::= <candidate-node>
                                  [<candidate-node-list>]
         <candidate-node>::= <ERO>
                             <candidate-attribute-list>

         <candidate-attribute-list>::= [<attribute-list>]
                                       [<PCE>]
                                       [<Node-Flags>]
```

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 node may contain two new objects PCE and node flags.

7.4. 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 format of the Node Flags object body is as follows:

```
    0                   1                   2                   3
    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
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |D|S|I|E|N|          Flags      |         Reserved               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

where

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

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:

```
        <PCReq Message>::= <Common Header>
                           <request>
        <request>::= <RP> <END-POINT-RRO-PAIR-LIST>  [<OF>]  [<LSPA>]
                     [<BANDWIDTH>]  [<metric-list>]  [<RRO>[<BANDWIDTH>]]
                     [<IRO>] [<LOAD-BALANCING>]
                     [<result-path-list>]
                     [<candidate-node-list-obj>]
     where:
        <result-path-list>::=<path>[<result-path-list>]
        <path>::= <ERO><attribute-list>
        <attribute-list>::=[<LSPA>]  [<BANDWIDTH>]  [<metric-list>]
                           [<IRO>]

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

        <candidate-node-list>::= <candidate-node>
                                 [<candidate-node-list>]
        <candidate-node>::= <ERO>
                            <candidate-attribute-list>

        <candidate-attribute-list>::= [<attribute-list>]
                                      [<PCE>]
                                      [<Node-Flags>]
```

Figure 1: The Format for a Request Message

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>
           <response> ::= <RP>  [<END-POINT-PATH-PAIR-LIST>]
                          [<NO-PATH>] [<attribute-list>]
                          [<result-path-list>]
                          [<candidate-node-list-obj>]
        where:
           <candidate-node-list-obj> contains a <candidate-node-list>
```

If the path from the source to the destinations 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

A new RP Object Flag has 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
18	FSPC (F-bit)	This I-D
19	Transfer Request (T-bit)	This I-D

10. Acknowledgement

The author 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

[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>.

[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, <https://www.rfc-editor.org/info/rfc3209>.

[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>.

[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>.

11.2. 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, <https://www.rfc-editor.org/info/rfc4655>.

[RFC5862] Yasukawa, S. and A. Farrel, "Path Computation Clients (PCC) - Path Computation Element (PCE) Requirements for Point-to-Multipoint MPLS-TE", RFC 5862, DOI 10.17487/RFC5862, June 2010, <https://www.rfc-editor.org/info/rfc5862>.

Authors' Addresses

Huaimo Chen
Futurewei
Boston, MA
USA

EMail: Huaimo.chen@futurewei.com

Dhruv Dhody
Huawei Technologies
Leela Palace
Bangalore, Karnataka 560008
INDIA

EMail: dhruv.dhody@huawei.com