Internet Engineering Task Force
Internet-Draft
Intended status: Standards Track
Expires: May 2, 2012

H. Chen
Huawei Technologies
O. Gonzalez de Dios
Telefonica I+D

October 30, 2011

A Forward-Search P2P TE LSP Inter-Domain Path Computation
draft-chen-pce-forward-search-p2p-path-computation-02.txt

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 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

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 2, 2012.

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.

RFC 4105 "Requirements for Inter-Area MPLS TE" lists the requirements for computing a shortest path for a TE LSP acrossing 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 acrossing multiple ASes. RFC 4655 "A PCE-Based Architecture" discusses centralized and distributed computation models for the computation of a path for a TE LSP acrossing multiple domains.

This document presents a forward search procedure to address these requirements through 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 a determined sequence of domains.

Exit BN of domain(n): a BN connecting domain(n) to domain(n+1) along a determined sequence of domains.

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 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 acrossing 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 acrossing 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 acrossing 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 details.

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 acrossing 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 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, the normal links in the destination domain and the inter-domain links.

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

The forward search path computation procedure for computing a path for an MPLS TE P2P LSP starts at the source domain, in which the source (or ingress) node of the MPLS TE LSP locates. When a PCE in the source domain receives a PCReq for the path for the MPLS TE LSP, it computes the optimal path from the source node to every exit boundary node of the domain towards the destination node.

There are two lists involved in the path computation. One list is called candidate node list, which contains the nodes with brief information about the temporary optimal paths from the source node to each of these nodes currently found. The nodes in the candidate list are ordered by the cost of the path. Initially, the candidate node list contains only source node with cost 0.

The other is called result path list or tree, which contains the final optimal paths from the source node to the boundary nodes or the nodes in the destination domain. Initially, the result path list is empty.

When a PCE responsible for a domain (called current domain) receives a PCReq for computing the path for the MPLS TE LSP, it removes the node with the minimum cost from the candidate node list and put or graft the node to the result path list or tree.

If the destination node is in the current domain, the PCE tries to compute the optimal path from the source node to the destination node and sends a PCRep with the optimal path to the PCE or PCC from which the PCReq is received.

Otherwise (i.e., if the destination is not in the domain), the PCE computes the optimal path from the source node to every exit boundary node of the current domain towards the destination node and further to the entry boundary nodes of the domain connected to the current domain, puts the new node into the candidate list in order by path cost, updates the existing node in the candidate node list with the new node with lower cost, and then sends a PCReq with the new candidate node list to the PCE that is responsible for the domain with the first node in the candidate node list.

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. The information about each node C in CandidateNodeList consists of

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

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

the PCE responsible for C, and

the flags for C. The flags include

one bit D indicating that node C is a Destination node if it is set;

one bit S indicating that C is the Source node if it is set;

one bit E indicating that C is an Exit boundary node if it is set;

one bit I indicating that C is an entry boundary node if it is set; and

one bit N indicating that C is a Node in the destination domain 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, and flags with S bit set.

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 nodes in the destination domain. Initially, ResultPathTree is empty.

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 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 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; otherwise, a PCReq message is sent to the PCE for node C.

Suppose that node C has Flags. The ResultPathTree is built from C in the following steps.

If the D (Destination Node) bit in the Flags is set, 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 current PCE.

If the N (Node in Destination domain) bit in the Flags is set, then for every node N connected to node C and not on ResultPathTree, it is merged into CandidateNodeList. The cost to node N is the sum of the cost to node C and the cost of the link between C and N. The PCE for node N is the current PCE.

If the Entry/Incoming Boundary Node (I) bit or the Source Node (S) bit is set), then path segments from node C to every exit boundary

node of the current domain that is not on the result path tree are computed through using CSPF and as special links. For every node N connected to node C through a special link (i.e., a path segment), 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., path segment) between C and N. The PCE for node N is the current PCE.

If the Exit Boundary Node (E) bit is set and there exist inter-domain links connected to it, then for every node N connected to C 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. The PCE for node N is the PCE responsible for node N.

If the CurrentPCE is the same as the PCE of the node with the minimum cost in CandidateNodeList, then the node is removed from CandidateNodeList, grafted to ResultPathTree, and the above steps are repeated; otherwise, the CurrentPCE sends the PCE a request with the source node, CandidateNodeList and ResultPathTree.

5.3. 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. 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 sesuence. 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 destionation, 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.

6. 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.

6.1. 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 Forward Search Path Computation.

- F (Forward search Path Computation bit - 1 bit):

  - 0: This indicates that this is not PCReq/PCRep for Forward Search Path Computation.

  - 1: This indicates that this is PCReq or PCRep message for Forward Search Path Computation.

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.

6.2. PCE Object

The figure below illustrates a PCE IPv4 object body (Object-Type=2), 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.

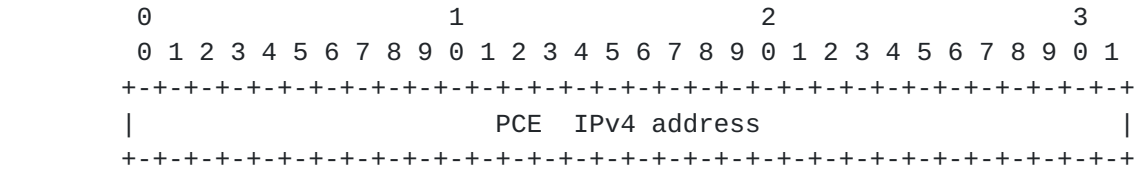

Figure 1: PCE Object Body for IPv4

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

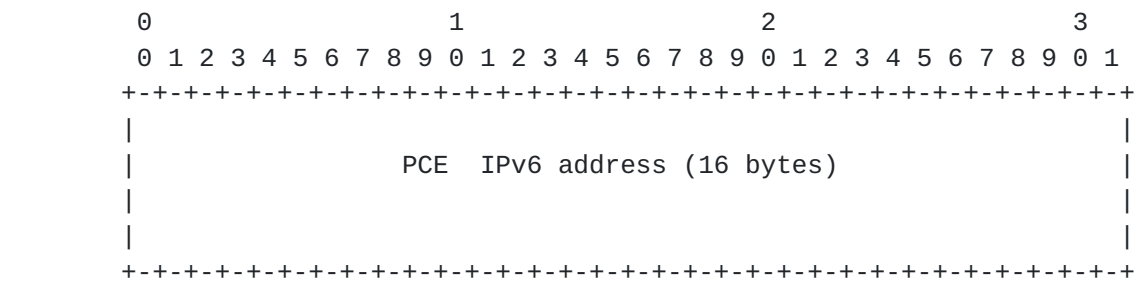

Figure 2: PCE Object Body for IPv6

6.3. 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:

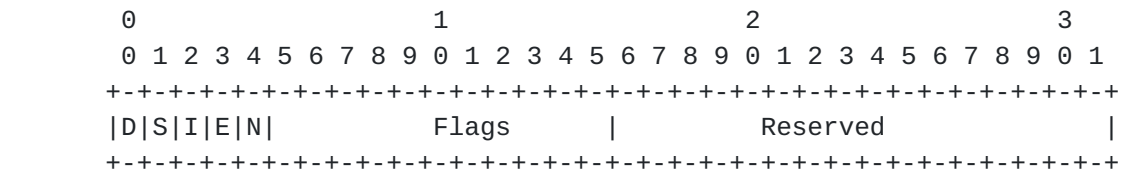

Figure 3: Node Flags Object Body

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 N = 1: The node is a node in a destination domain.

6.4. 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:

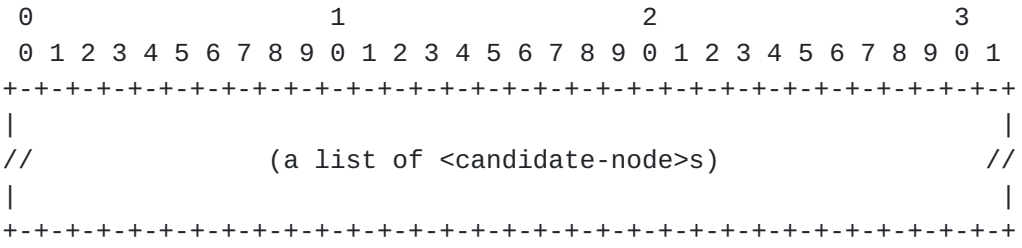

Figure 4: Candidate Node List Object

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>]
```

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.

6.5. 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>
                   [<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-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>
```

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. Security Considerations

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

8. IANA Considerations

This section specifies requests for IANA allocation.

8.1. 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	Forward Path Computation (F-bit)	This I-D

9. Acknowledgement

The authors would like to thank Julien Meuric, Daniel King, Cyril Margaria, Ramon Casellas, Olivier Dugeon and Dhruv Dhody for their valuable comments on this draft.

10. References

10.1. Normative References

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

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

[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.

[RFC5862] Yasukawa, S. and A. Farrel, "Path Computation Clients (PCC) - Path Computation Element (PCE) Requirements for Point-to-Multipoint MPLS-TE", RFC 5862, June 2010.

Authors' Addresses

Huaimo Chen
Huawei Technologies
Boston, MA
USA

Email: Huaimochen@huawei.com

Oscar Gonzalez de Dios
Telefonica I+D
Emilio Vargas 6, Madrid
Spain

Email: ogondio@tid.es