Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: December, 2008

Q. Zhao(Editor)
Huawei Technology
M. Chaitou(Editor)
France Telecom

July 2, 2008

Extensions to the Path Computation Element Communication Protocol (PCEP) for Point-to-Multipoint Traffic Engineering Label Switched Paths
draft-zhao-pcep-p2mp-extension-00

Status of this Memo

Abstract

Point-to-point Multiprotocol Label Switching (MPLS) and Generalized MPLS (GMPLS) Traffic Engineering Label Switched Paths (TE LSPs) may be established using signaling techniques, but their paths may first be determined. The Path Computation Element (PCE) has been identified as an appropriate technology for the determination of the paths of P2MP TE LSPs.

This document describes extensions to the PCE Communication Protocol PCEP) to handle requests and responses for the computation of paths for P2MP TE LSPs.

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 RFC 2119 [RFC2119].

Table of Contents

1. Terminology 4
2. Introduction 4
3. Requirements 5
4. Protocol Procedures and Extensions 6
4.1. P2MP LSPs Efficient Presentation 6
4.2. Indication of P2MP Path Computation Request/Reply 6
4.2.1. The Extension of RP Object 6
4.2.2. The New P2MP END-POINTS Object 7
4.3. Request Message Formats 10
4.4. Reply Message Formats 11
5. P2MP Objective Functions and Metric Types 12
5.1. New Object Functions 12
5.2. New Metric Object Types 13
6. Non-Support of P2MP Path Computation. 13
7. Non-Support by Back-Level PCE Implementations. 14
8. P2MP TE Path Re-optimization Request 14
9. Adding/pruning Leaves 15
10. Branch Nodes 15
11. Synchronization of P2MP TE Path Computation Requests 15
12. P2MP Capability Advertisement 16
12.1. Extend the TLV in the Existing PCE Discovery Protocol . . 16
12.2. Open Message Extension 16
13. Multi-Message Support 17
14. UNREACH_DESTINATION object 18
15. P2MP PCEP Error Object 19
16. PCEP NO-PATH Indicator 20
17. Manageability Considerations 20
17.1. Control of Function and Policy 21
17.2. Information and Data Models 21
17.3. Liveness Detection and Monitoring 21
17.4. Verifying Correct Operation 21
17.5. Requirements on Other Protocols and Functional
Components 21
17.6. Impact on Network Operation 21
18. Security Considerations 21
19. IANA Considerations 22
20. Acknowledgement 22
21. Intellectual Property Considerations 22
22. References 23
22.1. Normative References 23
22.2. Informative References 23
23. Authors' Addresses 24
24. Full Copyright Statement 25

1. Terminology

Terminology used in this document

TE LSP: Traffic Engineered Label Switched Path.

LSR: Label Switch Router.

OF: Objective Function: A set of one or more optimization criterion (criteria) used for the computation of a single path (e.g. path cost minimization), or the synchronized computation of a set of paths (e.g. aggregate bandwidth consumption minimization, etc.).

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.

PCEP: Path Computation Element communication Protocol.

P2MP: Point-to-MultiPoint.

P2P: Point-to-Point.

This document also uses the terminology defined in [RFC4655], [RFC4875], and [ID.pcep].

2. Introduction

The Path Computation Element (PCE) defined in [RFC4655] is an entity that is capable of computing a network path or route based on a network graph, and applying computational constraints. A Path Computation Client (PCC) may make requests to a PCE for paths to be computed.

[RFC4875] describes how to set up point-to-multipoint (P2MP) Traffic Engineering Label Switched Paths (TE LSPs) for use in Multiprotocol Label Switching (MPLS) and Generalized MPLS (GMPLS) networks.

The PCE is identified as a suitable application for the computation of paths for P2MP TE LSPs [ID.pcep-p2mp-req].

The PCE communication protocol (PCEP) is designed as a communication protocol between PCCs and PCEs for point-to-point (P2P) path computations and is defined in [ID.pcep]. However, that specification does not provide a mechanism to request path computation of P2MP TE LSPs.

This document presents extensions to PCEP to support P2MP path computation satisfying the set of requirements described in [ID.pcep-p2mp-req].

This document relies on the semantics of PCEP for requesting path computation for P2MP TE LSPs. A P2MP LSP is comprised of multiple source-to-leaf (S2L) sub-LSPs. These S2L sub-LSPs are set up between ingress and egress LSRs and are appropriately combined by the branch LSRs using computation result from PCE to result in a P2MP TE LSP. One request message from a PCC may signal one or more S2L sub-LSP path computation requests to the PCE for a single P2MP LSP with certain constraints. Hence the S2L sub-LSPs belonging to a P2MP LSP can use one path computation request message or be split across multiple path computation messages.

3. Requirements

This section summarizes the PCEP requirements specific to Point to Multi point as described in [ID.pcep-p2mp-req].

R1: Indication of P2MP Path Computation Request.

R2: Indication of P2MP Objective Functions.

R3: Non-Support of P2MP Path Computation.

R4: Non-Support by Back-Level PCE Implementations.

R5: Specification of Destinations.

R6: Indication of P2MP Paths.

R7: Multi-Message Requests and Responses.

R8: Non-Specification of Per-Destination Constraints and Parameters.

R9: Path Modification and Path Diversity.

R10: Reoptimization of P2MP TE LSPs.

R11: Addition and Removal of Destinations from Existing Paths.

R12: Specification of Applicable Branch Nodes.

R13: Capabilities Exchange.

Also there are additional requirments which might be need to be added into the requirement draft. Here we list the ones which may need to be highlighted in the requirement draft (to be discussed with the authors of the requirements draft).

R14: Sender of the request message can specify if the return result from the PCE need to be represented in the compressed format or not.

4. Protocol Procedures and Extensions

The following sections describe the protocol extensions to satisfy the requirements specified in the previous section.

4.1. P2MP LSPs Efficient Presentation

In the request message of the adding of leaves, optimization of P2MP TE LSPs as specified in [ID.pcep-p2mp-req], and in the reply message, we need to pass an existing P2MP LSP between the PCC and PCE. In these cases, we need new path objects for efficiently passing the existing P2MP LSP between PCE to PCC.

We suggest to using the ERO/SERO and RRO/SRRO to represent each individual S2L sub-LSP. The ERO/RRO are same as defined in the [ID.pcep] and SERO and SRRO are same as defined in RFC4875 for the RSVP extension of P2MP.

4.2. Indication of P2MP Path Computation Request/Reply

The existing P2P RP object is extended so that it can signal to the receiver of the request or reply message that it is for P2P or P2MP path computation. Also the END- POINT object is extended to improve the efficiency of the message exchange between PCC and PCE in the case of P2MP path computation.

4.2.1. The Extension of RP Object

The PCE path computation request/reply message adds an explicit parameter to allow a receiving PCE to identify that the request/reply is for a P2MP path and also specify if the route is represented in

the compress format or not.

The M bit is added in the flag bits field of the RP object to signal the receiver of the message that the request/reply is for P2P or it is for P2MP.

The E bit is added in the flag bits field of the RP object to signal the receiver of the message that the route is in the compress format or not.

The extended format of the RP object body to include the M bit and the E bit 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 2
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |    Reserved   | Flags           |E|M|              O|B|R| Pri |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                        Request-ID-number                      |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                                                               |
 //                      Optional TLV(s)                        //
 |                                                               |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 1: RP Object Body Format

The following flags are added in this draft:

o M (P2MP bit - 1 bit):

   0: This indicates that this is not PCReq/PCRrep for P2MP.

   1: This indicates that this is PCReq or PCRep message for P2MP.

o E (ERO-compression bit - 1 bit):

   0: This indicates that the route is not in the compressed format.

   1: This indicates that the route is in the compressed format.

4.2.2. The New P2MP END-POINTS Object

To represent the end points for a P2MP path efficiently, we define a new type of end-points object for P2MP path.

With this new END-POINTS object, the PCE path computation request message is expanded in a way such that it allows a single request message to list multiple destinations.

There are 4 types of leaves in a P2MP request:

- o New leaves to add;

- o Old leaves to remove;

- o Old leaves whose path can be modified/reoptimized;

- o Old leaves whose path must be left unchanged.

A given END-POINTS object gathers the leaves of a given type. The type of leaf in a given END-POINTS object is identified by the END-POINTS object leaf type field.

So four values are possible for the leaf type field:

1. New leaves to add;

2. Old leaves to remove;

3. Old leaves whose path can be modified/reoptimized;

4. Old leaves whose path must be left unchanged.

With this new END-POINTS object, the request message size for the multiple destinations can be reduced up to 50% for a P2MP path where a single source address has many destinations.

The format of the new END-POINTS object body for IPv4 (Object-Type 3) 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 2
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |                          Leaf type                               |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |                     Source IPv4 address                          |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |                  Destination IPv4 address                        |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |                            ...                                   |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |                            ...                                   |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |                            ...                                   |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |                  Destination IPv4 address                        |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 2: The New P2MP END-POINTS Object Body Format for IPv4

The format of the END-POINTS object body for IPv6 (Object-Type 4) 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 2
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                           Leaf type                             |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                                                                 |
    |                Source IPv6 address (16 bytes)                   |
    |                                                                 |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                                                                 |
    |              Destination IPv6 address (16 bytes)                |
    |                                                                 |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                                                                 |
    |                           ...                                   |
    |                                                                 |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                                                                 |
    |                           ...                                   |
    |                                                                 |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                                                                 |
    |                           ...                                   |
    |                                                                 |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                                                                 |
    |              Destination IPv6 address (16 bytes)                |
    |                                                                 |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 3: The New P2MP END-POINTS Object Body Format for IPv6

The END-POINTS object body has a variable length of multiple of 4 bytes for IPv4 and multiple of 16 bytes for IPv6.

4.3. Request Message Formats

Below is the message format for the request message:

```
   <PCReq Message>::= <Common Header>
                      <request>
 where:
        <request>::= <RP with P2MP flag and ERO-Compress bit>
                     <end-point-rro-pair-list>
                     [<OF>]
                     [<LSPA>]
                     [<BANDWIDTH>]
                     [<metric-list>]
                     [<IRO>]
                     [<LOAD-BALANCING>]

 where:

        <end-point-rro-pair-list>::=
                        <END-POINTS>[<RRO List>]
                        [<end-point-rro-pair-list>]

        RRO-List::=<RRO>[BANDWIDTH][<RRO List>]
        <metric-list>::=<METRIC>[<metric-list>]
```

Figure 4: The Message Format for the Request Message

4.4. Reply Message Formats

Below is the message format for the reply message:

```
<PCRep Message>::= <Common Header>
                      <response>
<response>::=<RP with P2MP flag and ERO-Cpmpress bit>
                [<end-point-path-pair-list>]
                [<NO-PATH>]
                [<attribute-list>]
```

where:

```
<end-point-path-pair-list>::=
        [<END-POINTS>]<path>[<end-point-path-pair-list>]


<path> ::= <ERO>|<SERO>[<path>]

<attribute-list>::=[<OF>]
                     [<LSPA>]
                     [<BANDWIDTH>]
                     [<metric-list>]
                     [<IRO>]
```

Figure 5: The Message Format for the Reply Message

The optional END-POINTS in the reply message is used to specify which paths are removed, changed, not changed, or added for the request. The path is only needed for the end points which are added or changed.

If the ERO-Compress bit was set to 1 in request then the path will be formed by an ERO followed by a list of SERO. Otherwise it is a list of ERO.

5. P2MP Objective Functions and Metric Types

5.1. New Object Functions

Six objective functions have been defined in [ID.pce-of] for P2P path computation.

This document defines two additional objective functions, namely SPT (Shortest Path Tree) and MCT (Minimum Cost Tree) that apply to P2MP path computation. Hence two new objective function codes have to be defined.

The description of the two new objective functions is as follows.

Objective Function Code: 7 (suggested value, to be assigned by IANA)

Name: Shortest Path Tree (SPT)

Description: Minimize the maximum source-to-leaf cost with respect to a specific metric (e.g. TE or IGP metric)

Objective Function Code: 8 (suggested value, to be assigned by IANA)

Name: Minimum Cost Tree (MCT)

Description: Minimize the total cost of the tree, that is the sum of the costs of tree links, with respect to a specific metric.

Processing these two new objective functions is subject to the rules defined in [ID.pce-of].

5.2. New Metric Object Types

There are three types defined for the <METRIC> object in [ID.pcep], namely, the IGP metric, the TE metric and the hop count metric. This document defines three other types for the <METRIC> object: the P2MP IGP metric, the P2MP TE metric, and the P2MP Hop Count metric. They encode the sum of the metrics of all links of the tree. We propose the following values for these new metric types (to be assigned by IANA):

- P2MP IGP metric: T=4

- P2MP TE metric: T=5

- P2MP hop count metric: T=6

6. Non-Support of P2MP Path Computation.

- if a PCE receives a P2MP path request and it understands the P2MP flag in RP object, but the PCE is not capable of P2MP computation, the PCE MUST send a PCErr message with a PCEP-ERROR Object and an Error-Value. The corresponding P2MP path computation request MUST be cancelled. (Error-Type and Error-Value are defined in this document).

- If the PCE does not understand the P2MP flag in the RP object, then the PCE MUST send a PCErr message with a new error type "Unknown RP flag".

7. Non-Support by Back-Level PCE Implementations.

If we accidentally send the P2MP request to the PCE which does not support the P2MP yet, we have the following solution:

Using the same RP type with P2MP flag and the new END-POINTS type, the receiver will reject the request when it can not understand the new END-POINTS object.

8. P2MP TE Path Re-optimization Request

The reoptimization request for a P2MP TE path is specified by R bit in the RP object similarly to the re-optimization request for a P2P TE path. The only difference is that the user must insert the list of RRO after each type of END-POINTS as described in the PCReq message format section.

So the PCReq message would look like this:

```
        <PCReq Message>::= <Common Header>
                           <request>
            where:
                <request>::=<RP with P2MP flag/R bits set>
                        <END-POINTS for leaf type 3><RRO list>
                        <OF>
```

Figure 6: PCReq Message Example 1 for Optimization

In this example, the RRO list is representing the P2MP LSP before the optimization and the modifiable paths are indicated in the END-POINTS object.

Optionally it is possible to specify some leaves whose path cannot be modified. The PCReq message would then look like this:

```
        <PCReq Message>::= <Common Header>
                           <request>
            where:
                <request>::=<RP with P2MP flag/R bits set>
                        <END-POINTS for leaf type 3><RRO list>
                        <END-POINTS for leaf type 4><RRO list>
                        <OF>
```

Figure 7: PCReq Message Example 2 for Optimization

9. Adding/pruning Leaves

When adding new leaves or removing old leaves to the existing P2MP tree, by supplying a list of existing leaves, one may be able to optimize the new P2MP tree. This section explains ways to add new leaves or remove old leaves to the existing P2MP tree.

To add new leaves the user must build a P2MP request with END-POINTs of leaf type 1.

To remove old leaves the user must build a P2MP request with END-POINTS of leaf type 2.

In any case it must also provides the list of old leaves and indicate if they must be reoptimized or not by including END-POINTS with leaf type 3 or 4 or both. In the future version, we may want to consider to define error values when the condition is not satisfied (i.e., when there is no END-POINTS with leaf type 3 or 4, in the presence of END-POINTS with leaf type 1 or 2).

For old leaves the user must provide the old path as list of RROs that immediately follows each END-POINTS object. In the future version, we may want to consider to define error values when the condition is not satisfied.

10. Branch Nodes

Before computing the P2MP path, a PCE must be provided means to know which nodes in the network are capable of acting as branch LSRs. A PCE can discover such capability by using the mechanisms defined in [ID.node-cap].

11. Synchronization of P2MP TE Path Computation Requests

There are cases when multiple P2MP LSPs' computation need to be synchronized. For example, one P2MP LSP is the backup of another P2MP LSP. In this case, the path diversity for these two LSPs needs to be considered during the path computation.

Synchronized computation and SynchronizedVector (SVEC) association for point-to-multipoint path requests are detailed in [ID.pce-svec-list].

Example of synchronizing two P2MP LSPs, each has two leaves for Path

Computation Request Messages is illustrated as below:

```
  <PCReq Message>::= <Common Header>
                <SVEC for sync of LSP1 and LSP2>[OF?]
                <request-list>
      where:
        <request-list>::=<request1><request2>
             <request1>::= <RP with P2MP flag>
                              <END-POINTS1 for P2MP>
                              <RRO1 list>
                              [<BANDWIDTH1>]

             <request2>::= <RP with P2MP flag>
                              <END-POINTS2 for P2MP>
                              <RRO2 list>
                              [<BANDWIDTH2>]
```

Figure 8: PCReq Message Example for Synchronization

12. P2MP Capability Advertisement

12.1. Extend the TLV in the Existing PCE Discovery Protocol

Since the RFC 5088 has specified that we can not add additional sub-TLV to the PCED TLV, we will define new bits to go in the existing 32 bits PCE Caps Flags to indicate the capability of P2MP for the PCC and PCE.

12.2. Open Message Extension

Based on the Capabilities Exchange requirement described in [ID-pcep-p2mp-req], if a PCE does not advertise its P2MP capability through discovery and the capability is not configured to the PCC, we need to use PCEP to allow a PCC to discover which PCEs with which it communicates support P2MP path computation. To satisfy this requirement, we extend the OPEN object format by including a new defined TLV for the capability of P2MP in the optional field. The new defined capability TLV allows the PCE to advertise its path computation capabilities.

The TLV type number will be assigned by IANA, the LENGTH value is 2 bytes. The value field is set to default value 0.

Note that the capability TLV is meaningful only for a PCE so it will typically appear only in one of the two Open messages during PCE session establishment. However, in case of PCE cooperation (e.g.,

inter-domain), when a PCE behaving as a PCC initiates a PCE session it SHOULD also indicate its Path Computation capability.

13. Multi-Message Support

The solution follows synchronization procedures defined in [ID.pcep].

If the P2MP request (i.e. <RP><END-POINTS>) is too large to fit into a single message it is permitted to divide it into multiple requests that would be carried in different messages. That means that a P2MP request would then contain multiple requests with RP objects that have the same request IDs.

Here is an example of such P2MP request that is divided in 2 request messages:

```
  <PCReq Message1>::= <Common Header>
                <SVEC, the Req-ID1 is repeated 2 times>
                <request>

             where:

                <request>::=< RP with Req-ID1 >
                               <END-POINTs for P2MP>
                               <RRO list>


  <PCReq Message2>::= <Common Header>
                <request>

             where:

               <request>::=< RP with Req-ID1>
                              <END-POINTs for P2MP>
                              <RRO list>
```

Figure 9: PCReq Message Example for Message Fragmentation

Note that the SVEC object contains the same request Id repeated N times where N is the total number of RP objects included in all messages. This is to be able to detect that the whole P2MP request has been received. Note that this assumes that the transmission of the messages is performed reliably and in consistent order, which is not a problem since PCEP relies on TCP.

14. UNREACH_DESTINATION object

The PCE path computation request may fail because all or a subset of the destinations are unreachable.

In such a case, the UNREACH-DESTINATION object allows the PCE to optionally specify the list of unreachable destinations.

This object can be present in PCRep messages. There can be up to one such object per RP.

UNREACH_DESTINATION Object-Class is to be assigned by IANA.

UNREACH_DESTINATION Object-Type for IPv4 is to be assigned by IANA

UNREACH_DESTINATION Object-Type for IPv6 is to be assigned by IANA.

The format of the UNREACH_DESTINATION object body for IPv4 (Object-Type=1) is as follows:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                  Destination IPv4 address                     |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 ~                            ...                                ~
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                  Destination IPv4 address                     |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 10: UNREACH_DESTINATION Object Body for IPv4

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

```
      0                   1                   2                   3
      0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |                                                                 |
      |             Destination IPv6 address (16 bytes)                |
      |                                                                 |
      |                                                                 |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      ~                          ...                                    ~
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |                                                                 |
      |               Destination IPv6 address (16 bytes)              |
      |                                                                 |
      |                                                                 |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 11: UNREACH_DESTINATION Object Body for IPv6

15. P2MP PCEP Error Object

To indicate errors associated with the P2MP path request, a new Error-Type (16) and subsequent error-values are defined as follows for inclusion in the PCEP-ERROR object:

A new Error-Type (16) and subsequent error-values are defined as follows:

Error-Type=16 and Error-Value=1: if a PCE receives a P2MP path request and the PCE is not capable to satisfy the request due to insufficient memory, the PCE MUST send a PCErr message with a PCEP ERROR object (Error-Type=16) and an Error-Value(Error-Value=1). The corresponding P2MP path computation request MUST be cancelled.

Error-Type=16; Error-Value=2: if a PCE receives a P2MP path request and the PCE is not capable of P2MP computation, the PCE MUST send a PCErr message with a PCEP-ERROR Object (Error-Type=16) and an Error-Value (Error-Value=2). The corresponding P2MP path computation request MUST be cancelled.

To indicate an error associated with policy violation, a new error value "P2MP Path computation not allowed" should be added to an existing error code for policy violation (Error-Type=5) as defined in [ID.pcep].

Error-Type=5; Error-Value=4: if a PCE receives a P2MP path computation request which is not compliant with administrative

privileges (i.e., the PCE policy does not support P2MP path computation), the PCE sends a PCErr message with a PCEP-ERROR Object (Error-Type=5) and an Error-Value (Error-Value=4). The corresponding P2MP path computation request MUST be cancelled.

16. PCEP NO-PATH Indicator

To communicate the reason(s) for not being able to find P2MP path computation, the NO-PATH object can be used in the PCRep message. The format of the NO-PATH object body is as follows:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|C|          Flags               |          Reserved             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                                |
//                       Optional TLV(s)                        //
|                                                                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 12: The Format of the NO-PATH Object Body

One new bit flags are defined in the NO-PATH-VECTOR TLV carried in the NO-PATH Object:

0x20: when set, the PCE indicates that there is a reachability problem with all or a subset of the P2MP destinations. Optionally the PCE can specify the list of destination(s) that are not reachable using the new UNREACH_DESTINATION object defined in section 3.6.

17. Manageability Considerations

[ID.pcep-p2mp-req] describes various manageability requirements in support of P2MP path computation when applying PCEP. This section describes how manageability requirements mentioned in [ID.pcep-p2mp-req] are supported in the context of PCEP extensions specified in this document.

Note that [ID.pcep] describes various manageability considerations in PCEP, and most of manageability requirements mentioned in [PCEP-P2MP P2MP] are already covered there.

17.1. Control of Function and Policy

In addition to configuration parameters listed in [ID.pcep], the following parameters MAY be required.

o P2MP path computations enabled or disabled.

o Advertisement of P2MP path computation capability enabled or disabled (discovery protocol, capability exchange).

17.2. Information and Data Models

As described in [ID.pcep-p2mp-req], MIB objects MUST be supported for PCEP extensions specified in this document.

17.3. Liveness Detection and Monitoring

There are no additional considerations beyond those expressed in [ID.pcep], since [ID.pcep-p2mp-req] does not address any additional requirements.

17.4. Verifying Correct Operation

There are no additional considerations beyond those expressed in [ID.pcep], since [ID.pcep-p2mp-req] does not address any additional requirements.

17.5. Requirements on Other Protocols and Functional Components

As described in [ID.pcep-p2mp-req], the PCE MUST obtain information about the P2MP signaling and branching capabilities of each LSR in the network.

Protocol extensions specified in this document does not provide such capability. Other mechanisms MUST be present.

17.6. Impact on Network Operation

It is expected that use of PCEP extensions specified in this document does not have significant impact on network operations.

18. Security Considerations

As described in [ID.pcep], P2MP path computation requests are more CPU-intensive and also use more link bandwidth. Therefore, it may be more vulnerable to denial of service attacks.

[ID.pcep] describes various mechanisms for denial of service attacks, and these tools MAY be advantageously used.

19. IANA Considerations

A number of IANA considerations have been highlighted in the relevent sections of this document. Further clarifications of these requests will be made in a future version of this document.

20. Acknowledgement

The authors would like to thank Adrian Farrel, Young Lee, Dan Tappan,Autumn Liu and Huaimo Chen, and Eiji Oki for their valuable comments on this draft.

21. Intellectual Property Considerations

22. References

22.1. Normative References

[ID.pcep]
Ayyangar, A., Oki, E., Atlas, A., Dolganow, A., Ikejiri, Y., Kumaki, K., Vasseur, J., and J. Roux, "Path Computation Element (PCE) Communication Protocol (PCEP)", draft-ietf-pce-pcep-12 (work in progress), March 2008.

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

[RFC4875] Aggarwal, R., Papadimitriou, D., and S. Yasukawa, "Extensions to Resource Reservation Protocol - Traffic Engineering (RSVP-TE) for Point-to-Multipoint TE Label Switched Paths (LSPs)", RFC 4875, May 2007.

[ID.node-cap] Vasseur, J.-P., Le Roux, J.L., "IGP Routing Protocol Extensions for Discovery of Traffic Engineering Node Capabilities", RFC 5073, December 2007

[ID.pcep-p2mp-req]
Yasukawa, S. and A. Farrel, "PCC-PCE Communication Requirements for Point to Multipoint Multiprotocol Label Switching Traffic Engineering (MPLS-TE)", draft-yasukawa-pce-p2mp-req-05 (work in progress), May 2008.

[ID.pce-of]
Roux, J., "Encoding of Objective Functions in Path Computation Element communication Protocol (PCEP)", draft-ietf-pce-of-02 (work in progress), February 2008.

[ID.pce-svec-list]
Nishioka, I. and D. King, "The use of SVEC (Synchronization VECtor) list for Synchronized dependent path computations", draft-nishioka-pce-svec-list-02 (work in progress), July 2008.

22.2. Informative References

[RFC4655] Farrel, A., Vasseur, J., and J. Ash, "A Path Computation Element (PCE)-Based Architecture", RFC 4655, August 2006.

23. Authors' Addresses

Quintin Zhao
Huawei Technology
125 Nagog Technology Park
Acton, MA 01719
US

Email: qzhao@huawei.com

Daniel King
Old Dog Consulting
UK

Email: daniel@olddog.co.uk

Fabien Verhaeghe
Marben Products
176 avenue Jean Jaures
92800 Puteaux,
France

Email: fabien.verhaeghe@marben-products.com

Tomonori Takeda
NTT Corporation
3-9-11, Midori-Cho
Musashino-Shi, Tokyo 180-8585
Japan

Phone:
Email: takeda.tomonori@lab.ntt.co.jp

Mohamad Chaitou
France Telecom
2, avenue Pierre-Marzin
22307 Lannion Cedex,
France

Email: Mohamad.Chaitou@orange-ftgroup.com

Jean-Louis Le Roux
France Telecom
2, avenue Pierre-Marzin
22307 Lannion Cedex,
France

Email: Jeanlouis.Leroux@orange-ftgroup.com

Zafar Ali
Cisco systems, Inc.
2000 Innovation Drive
Kanata, Ontario K2K 3E8
Canada

Email: zali@cisco.com

24. Full Copyright Statement