Network Working Group Quintin Zhao
Internet-Draft Huawei Technology
Category: Standards Track Daniel King (Editor)
Expires: August 20, 2008 Aria Networks
Tomonori Takeda
NTT
Fabien Verhaeghe
Marben Products

February 17, 2008

Extensions to the Path Computation Element Communication Protocol (PCEP) for Point-to-Multipoint Traffic Engineering Label Switched Paths

draft-zhao-pce-pcep-extension-p2mp-00.txt

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. 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 [PCEP-P2MP].

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 [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 Sections 2.1.1 to 2.1.13 of [PCEP-P2MP].

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.

This document uses the terminology defined in [RFC4655], [RFC4875], and [PCEP].

In some places in this draft, multiple options are presented for consideration. After each of these options has been fully evaluated by the PCE working group, the ones which satisfy the corresponding requirements and are most practical from the point view of implementation will be chosen. [EDITORS NOTE: We will remove this paragraph before publication as an RFC.]

2. Requirements

This section summarizes the PCEP requirements specific to Point to Multi point as described in [PCEP-P2MP].

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.

3. Protocol Procedures and Extensions

The following two sections describe the procedures adopted by a PCE handling a request from a PCC for P2MP path computation, and define how those requests and their responses are encoded.

3.1 PCEP Message Header Extension for PCEP Large Message Support

As specified in Common Header Section of [PCEP], the message length is encoded in a 16 bit field, and each object in the message also has its length encoded in a 16 bit field as specified in the Object Format Section of [PCEP].

P2MP path computation may require the transfer of larger amounts of data between PCC and PCE than is required for P2P path computation. This may mean that in some circumstances a single PCEP message is unable to contain all of the necessary data. This possibility is exacerbated by the potential use of IPv6 addresses. In this case, we need to use multiple messages to represent the P2MP paths and there is a need to correlate this sequence of fragmented messages into a single computation request or response.

In this draft, we propose to extend the PCEP message header by using one of the Flags bits as the F (fragmentation) bit.

We will extend the PCEP message header as below:

```
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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Ver | Flags |F|  Message Type |  Message Length               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 1: Extended PCEP Message Common Header

In this extended format, one of the Flags bits is used as the F bit:

F: 0 This means that the message is the last of a sequence of correlated messages.
F: 1 This means that the message is one of a sequence of messages, but not the last in the sequence.

In the case of F set to zero, it will be the last of a sequence of correlated messages if the previous message has the F bit set to 1 or there is no previous message at all and it is single non-fragmented message.

In the case of F set to 1, then the receiver of the message needs to wait until it receives the message which has the F bit set to 0 and treat all the previous messages which have F bit set 1 and this last message which has the F bit set to 0 as a single complete message. The duration to wait is specified by the SyncTimerdefined in [PCEP] for the handling of the SVEC Object, it is recommended for the PCE to implement a local timer, activated upon the receipt of the first PCReq message.

This F bit can be used implicitly to signal the receiver of the message if the P2MP tree data is sent by using a single message or it is sent by using a sequence of messages. It is assumed that in-order delivery of the fragments is assumed from the use of TCP.

3.2 Open Message Extension for P2MP Capability Advertisement

Based on the Capabilities Exchange requirement described in [PCEP-P2MP], 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 as described in the following section.

3.2.1 Capability TLV

The capability TLV allows the PCE to advertise its path computation capabilities. Inside the open object, we suggest to add the P2MP capability TLV in the optional field.

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.

3.3 RP Object Definitions for P2MP Capability

3.3.1 Extend the Existing P2P RP Object

In this option, the PCE path computation request message adds an explicit parameter to allow a receiving PCE to identify that the request is for a P2MP path. The M bit is added in the flag bits field of the RP object to signal the receiver of the message that the request is for P2P or it is for P2MP.

When the M bit is set to 1, we also propose to include a TLV for the P2MP Tree ID. [EDITORS NOTE: The authors wish to highlight the need for a P2MP Tree ID in case the PCE needs to store the P2MP tree data for PCCs to reference it instead of passing the whole P2MP tree again. This function requires further discussion before text is included for the next draft.]

The extended format of the RP object body, to include the M 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    Reserved   |                 Flags             |M|O|B|R| Pri |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          Request-ID-number                     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           TYPE (Tree ID)        |           LENGTH = 4         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                              Tree ID                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ |
|                                                                |
//                      Optional TLV(s)                        //
|                                                                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 2: RP Object Body Format

The following flags are added in this draft:

M (P2MP bit - 1 bit):

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

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

3.4 P2MP END-POINT Object Extensions

There are two options to represent the end points for a P2MP path. One is that we extend the existing END-POINT object and the second option is that we define a new type of end-point object of P2MP path.

The PCE path computation request message is expanded in a way such that it allows a single request message to list multiple destinations.

Further discussion and working group feedback is required for this section.

[EDITORS NOTE: We will remove multiple options as the draft progresses and working group feedback is gained.]

3.4.1 Extending the existing END-POINT object

With the existing END-POINTS object, the same source with multiple destinations need use multiple END-POINTS object in the request message. It works fine except that it is not efficient in the case that one source have many destinations, since we need repeat the same source in each END-POINTS object.

We propose to extend the END-POINTS object such that it has one single source address and it can have one or more than one destination address. With this extension, the request message size for the multiple destinations can be reduced almost 50% for a P2MP path where a single source address has many destinations.

The format of the END-POINTS object body for IPv4 (Object-Type is as same as defined in [PCEP]) 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     Source IPv4 address                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                  Destination IPv4 address                     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                            ...                                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                            ...                                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                            ...                                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                  Destination IPv4 address                     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 3: Extended END-POINTS Object Body Format for IPv4

The format of the END-POINTS object body for IPv6 (the Object-Type is as same as defined in [PCEP]) 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                Source IPv6 address (16 bytes)                 |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|              Destination IPv6 address (16 bytes)              |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                           ...                                 |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                           ...                                 |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                           ...                                 |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|              Destination IPv6 address (16 bytes)              |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 4: Extended 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.

3.4.2 Defining a new END-POINT object type for P2MP

The format for the END-POINT object is the same as what we have described in the previous section, except that it has a new object type which is different from the END-POINT object type defined for the P2P. The new type will be assigned by IANA.

3.5 P2MP LSPs Efficient Presentation

For supporting the optimization of P2MP TE LSPs as specified in section 2.1.10 of [PCEP-P2MP], we need to pass an existing P2MP LSP from the PCC to PCE. In this case, we need a new object for efficiently passing the existing P2MP LSP from PCE to PCC.

There are two options provided here for passing an existing P2MP LSP. In option 1, we use a separate instance of the ERO which represents each individual S2L.

In option 2, we treat the P2MP LSP as a normal tree structure which is represented by tree nodes. Each tree node is a LSR which can be root node, a branch node or a leaf node.

3.5.1 P2MP LSP Presentation (using S2L)

In this option, the P2MP LSP uses Explicit Route Object which is specified in the Explicit Route Object section in [PCEP] to present each S2L in the P2MP.

3.5.2 P2MP LSP Presentation (using BRANCH object)

In this option, the P2MP LSP is presented using the BRANCH object we define here.

The format of the BRANCH object body for IPv4 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       Current IPv4 address                    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       Parent IPv4 address                     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       Child IPv4 address                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                            ...                                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       Child IPv4 address                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 5: The New BRANCH Object Body Format for IPv4

The format of the END-POINTS object for IPv6 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                   Current IPv6 address (16 bytes)             |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                   Parent IPv6 address (16 bytes)              |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                   Child IPv6 address (16 bytes)               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                           ...                                 |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                   Child IPv6 address (16 bytes)               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 6: The New BRANCH Object Body Format for IPv6

BRANCH Object-Class is to be assigned by IANA.

BRANCH Object-Type is to be assigned by IANA.

The BRANCH object is used in a PCReq message to specify a branch for a P2MP LSP. For the case that the IP address for the parent field is not set to 0, it means that this is the branch which starts from the source node of the P2MP LSP to a leaf node of the P2MP LSP. For the case that the IP address for the parent field is set to 0, it means that this is the branch which starts from a none-source node of the P2MP LSP to a leaf node of the P2MP LSP.

The BRANCH object body has a variable length of multiple of 8 bytes required and the parent is optional (if it is the root node, it is set to 0) and there should be 1 child node or more children nodes. The advantage of this option is that in the case of one parent with many children nodes, it can be efficiently represented.

The disadvantage is that in the existing RSVP-TE extension for P2MP, the P2MP LSP is presented using the S2L structure already, which requires the conversion if we use a different coding scheme.

The efficiency of this option can be seen through the following example of P2MP LSP:

```
             A
             |
             |
             B
         / / / \ \ \
         | | | | | |
         C D E F G H
```

Using the option 1 specified in section 3.5.1, we have the following 6 objects to represent the P2MP LSP:

S2L sub-LSP-C: ERO = {A, B, C}, <S2L_SUB_LSP> object-C
S2L sub-LSP-D: SERO= {B, D}, <S2L_SUB_LSP> object-D
S2L sub-LSP-E: SERO= {B, E}, <S2L_SUB_LSP> object-E
S2L sub-LSP-F: SERO= {B, F}, <S2L_SUB_LSP> object-F
S2L sub-LSP-G: SERO= {B, G}, <S2L_SUB_LSP> object-G
S2L sub-LSP-H: SERO= {B, H}, <S2L_SUB_LSP> object-H

Using the option 2 specified in the section 3.5.2, we have the following 2 branch objects to represents the P2MP LSP:

Branch Object: Current(A), Parent(Null), Child(B)
Branch Object: Current(B), Parent(A), Child(C, D, E, F, G, H)

From the above example, we can see that using the option 1, the B node is repeated 6 times in the request message and using the option2, the B node is only repeated 2 times in the request message.

3.6. 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                  Destination IPv4 address                     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                            ...                                ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                  Destination IPv4 address                     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 7: The New UNREACH-DESTINATION Object Body Format 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|            Destination IPv6 address (16 bytes)                |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                           ...                                 ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|              Destination IPv6 address (16 bytes)              |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 8: The New UNREACH_DESTINATION Object Body Format for IPv6

3.7 P2MP PCEP Error Object

To indicate errors associated with the P2MP path optimization 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 of 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 requesrt 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 [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.

3.8 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
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |C|        Flags                |          Reserved             |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                                                               |
 //                      Optional TLV(s)                        //
 |                                                               |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 9: Flags (16 bits). The C flag is defined in [PCEP].

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.

4. 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

5. Manageability Considerations

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

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

5.1 Control of Function and Policy

In addition to configuration parameters listed in [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)

5.2 Information and Data Models

As described in [PCEP-P2MP], MIB objects MUST be supported for PCEP extensions specified in this document.

5.3 Liveness Detection and Monitoring

There are no additional considerations beyond those expressed in [PCEP], since [PCEP-P2MP] does not address any additional requirements.

5.4 Verifying Correct Operation

There are no additional considerations beyond those expressed in [PCEP], since [PCEP-P2MP] does not address any additional requirements.

5.5 Requirements on Other Protocols and Functional Components

As described in [PCEP-P2MP], 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.

5.6 Impact on Network Operation

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

6. Security Considerations

As described in [PCEP-P2MP], 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.

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

7. References

7.1 Normative References

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

[PCEP] Vasseur, JP., Ed. and JL. Le Roux, Ed., "Path Computation Element (PCE) communication Protocol (PCEP) - Version 8", draft-ietf-pce-pcep, work in progress.

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

[PCEP-P2MP] S. Yasukawa, A. Farrel," PCC-PCE Communication Requirements for Point to Multipoint Multiprotocol Label Switching Traffic Engineering (MPLS-TE)"

7.2 Informative References

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

8. Acknowledgements

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.

9. Authors' Addresses

Daniel King
Aria Networks Ltd.
44-45 Market Place
Chippenham, SN153HU UK
Email: daniel.king@aria-networks.com

Tomonori Takeda
NTT Network Service Systems Laboratories, NTT Corporation
3-9-11, Midori-Cho
Musashino-Shi, Tokyo 180-8585 Japan
Email : takeda.tomonori@lab.ntt.co.jp

Fabien Verhaeghe
Marben Products
176 avenue Jean Jaures
92800 Puteaux
France
Email: fabien.verhaeghe@marben-products.com

Quintin Zhao
Huawei Technology
125 Nagog Technology Park
Acton, MA 01719
USA
Email: qzhao@huawei.com

10. Intellectual Property Statement