Pce Working Group
Internet-Draft
Intended status: Standards Track
Expires: April 30, 2012

Q. Zhao
D. Dhody
Huawei Technology
Z. Ali
T. Saad
S. Sivabalan
Cisco Systems, Inc.
D. King
Old Dog Consulting
R. Casellas
CTTC - Centre Tecnologic de
Telecomunicacions de Catalunya

October 28, 2011

PCE-based Computation Procedure To Compute Shortest Constrained P2MP Inter-domain Traffic Engineering Label Switched Paths
draft-ietf-pce-pcep-inter-domain-p2mp-procedures-01

Abstract

The ability to compute paths for constrained point-to-multipoint (P2MP) Traffic Engineering Label Switched Paths (TE LSPs) across multiple domains has been identified as a key requirement for the deployment of P2MP services in MPLS and GMPLS networks. The Path Computation Element (PCE) has been recognized as an appropriate technology for the determination of inter-domain paths of P2MP TE LSPs.

This document describes the procedures and extensions to the PCE communication Protocol (PCEP) to handle requests and responses for the computation of inter-domain paths for P2MP TE LSPs.

Status of This Memo

This Internet-Draft will expire on April 30, 2012.

Copyright Notice

Table of Contents

1. Introduction

Multicast services are increasingly in demand for high-capacity applications such as multicast Virtual Private Networks (VPNs), IP-television (IPTV) which may be on-demand or streamed, and content-rich media distribution (for example, software distribution, financial streaming, or data-sharing). The ability to compute constrained Traffic Engineering Label Switched Paths (TE LSPs) for point-to-multipoint (P2MP) LSPs in Multiprotocol Label Switching (MPLS) and Generalized MPLS (GMPLS) networks across multiple domains is becoming important. A domain can be defined as a collection of network elements within a common sphere of address management or path computational responsibility such as an IGP area or an Autonomous Systems.

The applicability of the Path Computation Element (PCE) [RFC4655] for the computation of such paths is discussed in [RFC5671], and the requirements placed on the PCE communications Protocol (PCEP) for this are given in [RFC5862].

This document describes how multiple PCE techniques can be combined to address the requirements. These mechanisms include the use of the per-domain path computation technique specified in [RFC5152], extensions to the backward recursive path computation (BRPC) technique specified in [RFC5441] for P2MP LSP path computation in an inter-domain environment, and a new procedure for core-tree based path computation defined in this document. These three mechanisms are suitable for different environments (topologies, administrative domains, policies, service requirements, etc.) and can also be effectively combined.

1.1. Requirements Language

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]

2. Terminology

Terminology used in this document is consistent with the related MPLS/GMPLS and PCE documents [RFC4461], [RFC4655], [RFC4875], [RFC5376], [RFC5440], [RFC5441], [RFC5671] and [RFC5862].

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

ASBR: Autonomous System Border Router. Router 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.

Core Tree: the core tree is a P2MP tree where the root is the ingress LSR, and the leaf nodes are the entry BNs of the leaf domains.

Destination: The lead Nodes can be in Root Domain, Transit Domain and Leaf Domain.

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-AS TE LSP: a TE LSP that crosses an AS boundary.

Inter-area TE LSP: a TE LSP that crosses an IGP area boundary.

Leaf Domain: a domain that does not have a downstream neighbor domain. Note that, with this definition, a domain with one or more leaf nodes is not necessarily a leaf domain.

Leaf Boundary Nodes: the entry boundary node in the leaf domain.

Leaf Nodes: the LSR which is the P2MP LSP's final.

LSR: Label Switching Router.

LSP: Label Switched Path.

OF: Objective Function. A set of one or more optimization criterion (criteria) used for the computation of paths either for single or for synchronized requests (e.g. path cost minimization), or the synchronized computation of a set of paths (e.g. aggregate bandwidth consumption minimization, etc.). See [RFC4655] and [RFC5441].

P2MP LSP Path Tree: A set of LSRs and TE links that comprise the path of a P2MP TE LSP from its ingress LSR to all of its egress LSRs.

Path Domain Sequence: The known sequence of domains for a path between root and leaf.

Path Domain Tree: The tree formed by the domains that the P2MP path crosses, where the source (ingress) domain is the root domain.

PCC: Path Computation Client. Any client application requesting a path computation to be performed by the 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.

P2MP LSP Path Tree: A set of LSRs and TE links that comprise the path of a P2MP TE LSP from its ingress LSR to all of its egress LSRs.

Path Domain Sequence: the known sequence of domains for a path between the root node and a leaf node.

PCE Sequence: the known sequence of PCEs for calculating a path between the root node and a leaf node.

PCE Topology Tree: a list of PCE Sequences which has all the PCE Sequence for each path of the P2MP LSP path tree.

PCE(i): a PCE that performs path computations for domain(i).

Root Boundary Node: the egress LSR from the root domain on the path of the P2MP LSP.

Root Domain: the domain that includes the ingress (root) LSR.

TED: Traffic Engineering Database.

Transit/branch Domain: a domain that has an upstream and one or more downstream neighbour domain.

VSPT: Virtual Shortest Path Tree [RFC5441].

3. Problem Statement

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 P2MP TE LSPs for use in MPLS and GMPLS networks. The PCE is identified as a suitable application for the computation of paths for P2MP TE LSPs [RFC5671].

[RFC5441] specifies a procedure relying on the use of multiple PCEs to compute (P2P) inter-domain constrained shortest paths across a predetermined sequence of domains, using a backward recursive path

computation technique. The technique can be combined with the use of path keys [RFC5520] to preserve confidentiality across domains, which is sometimes required when domains are managed by different Service Providers.

The PCE communication Protocol (PCEP) [RFC5440] is extended for point-to-multipoint(P2MP) path computation requests and in [RFC6006]. However, that specification does not provide all the necessary mechanisms to request the computation of inter-domain P2MP TE LSPs.

As discussed in [RFC4461], a P2MP tree is a graphical representation of all TE links that are committed for a particular P2MP LSP. In other words, a P2MP tree is a representation of the corresponding P2MP tunnel on the TE network topology. A sub-tree is a part of the P2MP tree describing how the root or an intermediate P2MP LSPs minimizes packet duplication when P2P TE sub-LSPs traverse common links. As described in [RFC5671] the computation of a P2MP tree requires three major pieces of information. The first is the path from the ingress LSR of a P2MP LSP to each of the egress LSRs, the second is the traffic engineering related parameters, and the third is the branch capability information.

Generally, an inter-domain P2MP tree (i.e., a P2MP tree with source and at least one destination residing in different domains) is particularly difficult to compute even for a distributed PCE architecture. For instance, while the BRPC recursive path computation may be well-suited for P2P paths, P2MP path computation involves multiple branching path segments from the source to the multiple destinations. As such, inter-domain P2MP path computation may result in a plurality of per-domain path options that may be difficult to coordinate efficiently and effectively between domains. That is, when one or more domains have multiple ingress and/or egress border nodes, there is currently no known technique for one domain to determine which border routers another domain will utilize for the inter-domain P2MP tree, and no way to limit the computation of the P2MP tree to those utilized border nodes.

A trivial solution to the computation of inter-domain P2MP tree would be to compute shortest inter-domain P2P paths from source to each destination and then combine them to generate an inter-domain, shortest-path-to-destination P2MP tree. This solution, however, cannot be used to trade cost to destination for overall tree cost (i.e., it cannot produce a MCT tree) and in the context of inter-domain P2MP LSPs it cannot be used to reduce the number of domain border nodes that are transited.

Computing P2P LSPs individually is not an acceptable solution for computing a P2MP tree. Even per domain path computation [RFC5152]

can be used to compute P2P multi-domain paths, but it does not guarantee to find the optimal path which crosses multiple domains. Furthermore, constructing a P2MP tree from individual source to leaf P2P LSPs does not guarantee to produce a least-cost tree. This approach may also be considered to have scaling issues during LSP setup. That is, the LSP to each leaf is signaled separately, and each border node must perform path computation for each leaf.

P2MP Minimum Cost Tree (MCT), i.e. one which guarantees the least cost resulting tree, is an NP-complete problem. Moreover, adding and/or removing a single destination to/from the tree may result in an entirely different tree. In this case, frequent MCT path computation requests may prove computationally intensive, and the resulting frequent tunnel reconfiguration may even cause network destabilization. There are several heuristic algorithms presented in the literature that approximate the result within polynomial time that are applicable within the context of a single-domain.

This document presents a solution, and procedures and extensions to PCEP to support P2MP inter-domain path computation.

4. Assumptions

It is assumed that, due to deployment and commercial limitations (e.g., inter-AS peering agreements), the sequence of domains for a path (the path domain tree) will be known in advance.

[DOMAIN-SEQ] descibes the use of domain path tree in P2MP scenarios. In the figure below, the P2MP tree spans 6 domains, with D1 being the root domain. The corresponding domain sequences which are assumed known would be: D1-D3-D6, D1-D3-D5 and D1-D2-D4.

```
                              D1
                             / \
                            D2  D3
                           /   /  \
                          D4  D5  D6
```

Figure 1: Domain Sequence Tree

The examples and scenarios used in this document are also based on the following assumptions:

- PCC is either aware of the domain sequence for each of the P2MP destination as described in [DOMAIN-SEQ] or PCE sequence (i.e. PCE that serves each domain in the path domain tree). The set of PCEs and their relationships is propagated to each PCE during the

first exchange of path computation requests; [Editors note - this assumption needs to be more explicit.]

- Each PCE knows about any leaf LSRs in the domain it serves;

- The boundary nodes to use on the LSP are pre-determined. [Editors Note - In this version of the document we do not consider multi-homed domains.]

Additional assumptions are documented in [RFC5441] and will not be repeated here.

5. Requirements

This section summarizes the requirements specific to computing inter-domain P2MP paths. In these requirements we note that the actual computation times by any PCE implementation are outside the scope of this document, but we observe that reducing the complexity of the required computations has a beneficial effect on the computation time regardless of implementation. Additionally, reducing the number of message exchanges and the amount of information exchanged will reduce the overall computation time for the entire P2MP tree. We refer to the "Complexity of the computation" as the impact on these aspects of path computation time as various parameters of the topology and the P2MP LSP are changed.

Its also important that the solution preserves confidentiality across domains, which is required when domains are managed by different Service Providers.

Other than the requirements specified in [RFC5376], a number of requirements specific to P2MP are detailed below:

1. The computed P2MP LSP should be optimal when only considering the paths among the BNs.

2. Grafting and pruning of multicast destinations in a domain should have no impact on other domains and on the paths among BNs.

3. The complexity of the computation for each sub-tree within each domain should be dependent only on the topology of the domain and it should be independent of the domain sequence.

4. The number of PCEP request and reply messages should be independent of the number of multicast destinations in each domain.

5. Specifying the domain entry and exit nodes.

6. Specifying which nodes should be used as branch nodes.

7. Reoptimization of existing sub-trees.

8. Computation of P2MP paths that need to be diverse from existing P2MP paths.

6. Objective Functions

For the computation of a single or a set of P2MP TE LSPs, a request to meet specific optimization criteria, called an Objective Function (OF) may be indicated.

The computation of one or more P2MP TE-LSPs may be subject to an OF in order to select the "best" candidate paths. A variety of objective functions have been identified as being important during the computation of inter-domain P2MP LSPs. These include:

1. The sub-tree within each domain should be optimized, which can be either the Minimum cost tree [RFC5862] or Shortest path tree [RFC5862].

2. The P2MP LSP path, formed by considering only the entry and exit nodes of the domains (the Core Tree) should be optimal.

3. It should be possible to limit the number of entry points to a domain.

4. It should be possible to force the branches for all leaves within a domain to be in that domain.

7. P2MP Path Computation Procedures

The following sections describe the Core Tree based procedures to satisfy the requirements specified in the previous section. A core tree based solution provides an optimal inter-domain P2MP TE LSP.

7.1. Core Trees

A Core Tree is defined as a tree, which satisfies the following conditions:

- The root of the core tree is the ingress LSR in the root domain;

- The leaves of the core tree are the entry nodes in the leaf domains;

Note that Path-Key Mechanism [RFC5520] MAY be used to hide internal nodes.

An optimal core-tree [based on the OF] will be computed with analyzing the nodes and links within the domains. To support confidentiality the same nodes and links can be hidden via a path-key but they must be computed and be a part of core-tree.

For example, consider the Domain Tree from the figure below, representing a domain tree of 6 domains, and part of the resulting Core Tree which satisfies the aforementioned conditions.

```
                             +----------------+
                             |                |Domain D1
                             |        R       |
                             |                |
                             |        A       |
                             |                |
                             +-B------------C-+
                              /              \
                             /                \
                            /                  \
            Domain D2      /                    \ Domain D3
            +-------------D--+             +-----E----------+
            |                |             |                |
            |  F             |             |                |
            |          G     |             |       H        |
            |                |             |                |
            |                |             |                |
            +-I--------------+             +-J------------K-+
             /                              /              \
            /                              /                \
           /                              /                  \
          /                              /                    \
         /                              /                      \
        /                              /                        \
       / Domain D4         Domain D5  /              Domain D6   \
     +-L--------------+       +------P---------+      +-----------T----+
     |                |       |                |      |                |
     |                |       |  Q             |      |   U            |
     |  M        O    |       |         S      |      |                |
     |                |       |                |      |          V     |
     |         N      |       |   R            |      |                |
     +----------------+       +----------------+      +----------------+
```

Figure 2: Domain Tree Example

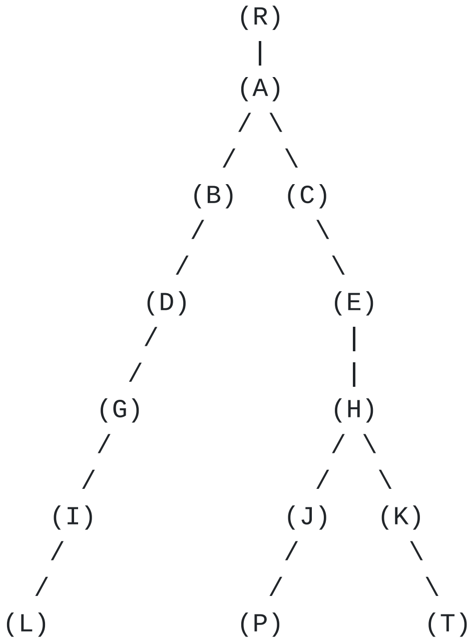

Figure 3: Core Tree

A core tree is computed such that root of the tree is R and the leaf node are the entry nodes of the destination domains (L, P and T). Path-key Mechanism can be used to hide the internal nodes and links in the final core tree.

7.2. Core Tree Computation Procedures

The algorithms to compute the optimal large core tree are outside scope of this document. The following extended BRPC based procedure can be used to compute the core tree.

BRPC Based Core Tree Path Computation Procedure:

1. Using the BRPC procedures to compute the VSPT(i) for each leaf BN(i), i=1 to n, where n is the total number of entry nodes for all the leaf domains. In each VSPT(i), there are a number of P(i) paths.

2. When the root PCE has computed all the VSPT(i), i=1 to n, take one path from each VSPT and form a set of paths, we call it a PathSet(j), j=1 to M, where M=P(1)xP(2)...xP(n);

3. For each PathSet(j), there are n S2L (Source to Leaf BN) paths and form these n paths into a Core Tree(j);

4. There will be M number of Core Trees computed from step3. Apply the OF to each of these M Core Trees and find the optimal Core Tree.

Note that the application of BRPC in the aforementioned procedure differs from the typical one since paths returned from a downstream PCE are not necessary pruned from the solution set by intermediate PCEs.

The reason for this is that if the PCE in a downstream domain does the prunning and returns the single optimal sub-path to its parent PCE, BRPC insures that the ingress PCE will get all the best optimal sub-paths for each LN (Leaf Border Nodes), but the combination of these single optimal sub-paths into a P2MP tree is not necessarily optimal even if each S2L (Source-to-Leaf) sub-path is optimal.

Without trimming, the ingress PCE will get all the possible S2L sub-paths set for LN, and eventually by looking through all the combinations, and taking one sub-path from each set to built one P2MP tree it finds the optimal tree.

Note that if the OF is SPT, VSPT is enough for computing core-tree and downstream PCE can continue to do the prunning. One way to address this would be that a transit PCE in core-tree computation can decide the numbers of paths sent upstream based on the configuration and/or OF. In case of SPT, the number of path sent will be 1.

The proposed method may present a scalability problem for the dynamic computation of the Core Tree (by iterative checking of all combinations of the solution space), specially with dense/meshed domains. Considering a domain sequence D1, D2, D3, D4, where the Leaf border node is at domain D4, PCE(4) will return 1 path. PCE(3) will return N paths, where N is E(3) x X(3), where E(k) x X(k) denotes the number of entry nodes times the number of exit nodes for that domain. PCE(2) will return M paths, where M = E(2) x X(2) x N = E(2) x X(2) x E(3) x X(3) x 1, etc. Generally speaking the number of potential paths at the ingress PCE Q = \prod E(k) x X(k).

Consequently, it is expected that the Core Path will be typically computed offline, without precluding the use of dynamic, online mechanisms such as the one presented here, in which case it SHOULD be possible to configure transit PCEs to control the number of paths sent upstream during BRPC (trading trimming for optimality at the point of trimming and downwards).

7.3. Sub Tree Computation Procedures

Once the core tree is built, the grafting of all the leaf nodes from each domain to the core tree can be achieved by a number of algorithms. One algorithm for doing this phase is that the root PCE will send the request with C bit set for the path computation to the destination(s) directly to the PCE where the destination(s) belong(s) along with the core tree computed from the phase 1.

This approach requires that the root PCE manage a potentially large number of adjacencies (either in persistent or non-persistent mode), including PCEP adjacencies to PCEs that are not within neighboring domains.

A first alternative would involve establishing PCEP adjacencies that correspond to the PCE domain tree. This would require that branch PCEs forward requests and responses from the root PCE towards the leaf PCEs and vice-versa.

Finally, another alternative would use a hierarchical PCE [H-PCE] architecture. The "hierarchically" parent would request sub tree path computations.

The algorithms to compute the optimal large sub tree are outside scope of this document. In the case that the number of destinations and the number of BNs within a domain are not big, the incremental procedure based on p2p path computation using the OSPF can be used.

7.4. PCEP Protocol Extensions

7.4.1. The Extension of RP Object

The RP Object is defined in [RFC5440] as -

```
        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                             |O|B|R| Pri |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |                        Request-ID-number                      |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |                                                               |
       //                      Optional TLV(s)                        //
       |                                                               |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 4: RP Object Body Format

The extended format of the RP object body to include the C bit is as follows:

The C 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 inter-domain P2MP Core Tree or not.

The following flag is added in this draft:

C bit (P2MP Core Tree bit - 1 bit):

0: This indicates that this is normal PCReq/PCRrep for P2MP.

1: This indicates that this is PCReq or PCRep message for inter-domain Core Tree P2MP. When the C bit is set, then the request message should have the Core Tree passed along with the destinations which and then graphed to the tree.

7.4.2. Domain or PCE Sequence

[DOMAIN-SEQ] mentions the benefit of using domain-sequence over PCE-Sequence. The domain-seq can be used in P2MP scenarios. [PER-DEST] provides a mechanims to encode Domain-Sequence (in form of IRO) per destination.

But if the administrator wants to control PCE rather than domain then PCE-SEQUENCE Object can be used.

The PCE Sequence Object is added to the existing PCE protocol. A list of this objects will represent the PCE topology tree. A list of Sequence Objects can be exchanged between PCEs during the PCE capability exchange or on the first path computation request message between PCEs. In this case, the request message format needs to be changed to include the list of PCE Sequence Objects for the PCE inter-domain P2MP calculation request.

Each PCE Sequence can be obtained from the domain sequence for a specific path. All the PCE sequences for all the paths of P2MP inter-domain form the PCE Topology Tree of the P2MP LSP.

Object Class for the PCE Sequence Object: To be assigned by IANA.

The format of the new PCE Sequence Object 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
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       | Object-Class  |   OT  |Res|P|I|   Object Length (bytes)       |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |                    IPv4 address for root PCE                  |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |                IPv4 address for the downstream PCE            |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |                IPv4 address for the downstream PCE            |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |                         !!                                    |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |   IPv4 address for the PCE corresponding to the leafDomain    |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 5: The New PCE Sequence Object Body Format for IPv4

The format of the new PCE Sequence Object for IPv6 (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
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       | Object-Class  |   OT  |Res|P|I|   Object Length (bytes)       |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |                                                               |
       |                    IPv6 address for root PCE                  |
       |                                                               |
       |                                                               |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |                                                               |
       |                IPv6 address for the downstream PCE            |
       |                                                               |
       |                                                               |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |                                                               |
       |                IPv6 address for the downstream PCE            |
       |                                                               |
       |                                                               |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |                                                               |
       |                          !!                                   |
       |                                                               |
       |                                                               |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |                                                               |
       |  IPv6 address for the PCE corresponding to the leafDomain     |
       |                                                               |
       |                                                               |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 6: The New PCE Sequence Object Body Format for IPv6

7.5. Relationship with Hierarchical PCE

The actual grafting of subtrees into the Multi-Domain tree needs to be carried out by the source node. This means that the source node needs to get the computed sub-trees from all the involved domains. This requires that the source node either has a PCEP session with all the PCEs, or PCEP messages are routed via the PCEP sessions. This may mean an excessive number of sessions or an added complexity in implementations.

Alternatively, one may use an architecture based on the concept of hierarchical PCE [H-PCE]. The parent PCE would be responsible to request Intra-domain subtrees to the PCEs, combine them and return the overall P2MP tree.

7.6. Parallelism

In order to minimize latency in path computation in multi-domain networks, intra-domain path segments and intra-domain sub-trees SHOULD be computed in parallel when possible. The proposed procedures in this draft present opportunities for parallelism:

1. The BRPC procedure for each leaf node can be launched in parallel by the ingress/root PCE if the dynamic computation of the Core Tree is enabled.

2. Intra-domain P2MP paths can also be computed in parallel by the PCEs once the entry and exit nodes within a domain are known

One of the potential issues of parallelism is that the ingress PCE would require a potentially high number of PCEP adjacencies to "remote" PCEs and that may not be desirable, but a given PCE would only receive requests for the destinations that are in its domain (+ the core nodes), without PCEs forwarding requests.

8. Protection

It is envisaged that protection may be required when deploying and using inter-domain P2MP LSPs. The procedures and mechanisms defined in this document do not prohibit the use of existing and proposed types of protection, including: end-to-end protection [RFC4875] and domain protection schemes.

Segment or facility (link and node) protection is problematic in inter-domain environment due to the limit of Fast-reroute (FRR) [RFC4875] requiring knowledge of its next-hop across domain boundaries whilst maintaining domain confidentiality. Although the FRR protection might be implemented if manually provisioned if next-hop information was known in advance.

8.1. End-to-end Protection

End-to-end protection (Node and Link Protection) principle can be applied for computing backup P2MP LSP. During computaion of Core-Tree and Sub-Tree, end-to-end protection can be taken into consideration. PCE may compute the Primary and backup P2MP LSP together or sequentially.

8.2. Domain Protection

In this protection scheme, backup P2MP Tree can be computed which excludes the transit/branch domain completely. A backup domain path tree is needed with the same source domain and destinations domains

and a new set of transit domains. The backup domain path tree can be applied to the above procedure to obtion the backup P2MP LSP with disjoint transit domains.

9. Manageability Considerations

[RFC5862] describes various manageability requirements in support of P2MP path computation when applying PCEP. This section describes how manageability requirements mentioned in [RFC5862] are supported in the context of PCEP extensions specified in this document.

Note that [RFC5440] describes various manageability considerations in PCEP, and most of manageability requirements mentioned in [PCE-P2MP] are already covered there.

9.1. Control of Function and Policy

In addition to PCE configuration parameters listed in [RFC5440], the following additional parameters might be required:

- The ability to enable or disable single domain P2MP path computations on the PCE.

- The ability to enable or disable multi-domain P2MP path computations on the PCE.

- The PCE may be configured to enable or disable the advertisement of its single domain and multi-domain P2MP path computation capability.

9.2. Information and Data Models

A number of MIB objects have been defined for general PCEP control and monitoring of P2P computations in [PCEP-MIB]. [RFC5862] specifies that MIB objects will be required to support the control and monitoring of the protocol extensions defined in this document. [PCEP-P2MP-MIB] describes managed objects for modeling of PCEP communications between a PCC and PCE, and PCE to PCE, P2MP path computation requests and responses.

In case of offline Core tree computation and configuration MAYBE stored.

9.3. Liveness Detection and Monitoring

No changes are necessary to the liveness detection and monitoring requirements as already embodied in [RFC4657].

It should be noted that multi-domain P2MP computations are likely to take longer than P2P computations, and single domain P2MP computatoins. The liveness detection and monitoring features of the PCECP SHOULD take this into account.

9.4. Verifying Correct Operation

There are no additional requirements beyond those expressed in [RFC4657] for verifying the correct operation of the PCECP. Note that verification of the correct operation of the PCE and its algorithms is out of scope for the protocol requirements, but a PCC MAY send the same request to more than one PCE and compare the results.

9.5. Requirements on Other Protocols and Functional Components

A PCE operates on a topology graph that may be built using information distributed by TE extensions to the routing protocol operating within the network. In order that the PCE can select a suitable path for the signaling protocol to use to install the P2MP LSP, the topology graph must include information about the P2MP signaling and branching capabilities of each LSR in the network.

Mechanisms for the knowledge of other domains, the discovery of corresponding PCEs and their capabilities should be provided and that this information MAY be collected by other mechanisms.

Whatever means is used to collect the information to build the topology graph, the graph MUST include the requisite information. If the TE extensions to the routing protocol are used, these SHOULD be as described in [RFC5073].

9.6. Impact on Network Operation

The use of a PCE to compute P2MP paths is not expected to have significant impact on network operations. However, it should be noted that the introduction of P2MP support to a PCE that already provides P2P path computation might change the loading of the PCE significantly, and that might have an impact on the network behavior, especially during recovery periods immediately after a network failure.

The dynamic computation of Core Trees might also have an impact on the load of the involved PCEs as well as path computation times.

9.7. Policy Control

[RFC5394] provides additional details on policy within the PCE architecture and also provides context for the support of PCE Policy. The are also applicable to Interdomain P2MP Path computation via Core Tree Mechanism.

10. Security Considerations

As described in [RFC5862], P2MP path computation requests are more CPU-intensive and also utilize more link bandwidth. In the event of an unauthorized P2MP path computation request, or a denial of service attack, the subsequent PCEP requests and processing may be disruptive to the network. Consequently, it is important that implementations conform to the relevant security requirements of [RFC5440] that specifically help to minimize or negate unauthorized P2MP path computation requests and denial of service attacks. These mechanisms include:

- Securing the PCEP session requests and responses using TCP security techniques (Section 10.2 of [RFC5440]).

- Authenticating the PCEP requests and responses to ensure the message is intact and sent from an authorized node (Section 10.3 of [RFC5440]).

- Providing policy control by explicitly defining which PCCs, via IP access-lists, are allowed to send P2MP path requests to the PCE (Section 10.6 of [RFC5440]).

PCEP operates over TCP, so it is also important to secure the PCE and PCC against TCP denial of service attacks. Section 10.7.1 of [RFC5440] outlines a number of mechanisms for minimizing the risk of TCP based denial of service attacks against PCEs and PCCs.

PCEP implementations SHOULD also consider the additional security provided by the TCP Authentication Option (TCP-AO) [RFC5925].

11. IANA Considerations

11.1. New Flag of the RP Object

A new flag of the RP object (specified in [RFC5440]) is defined in this document. IANA maintains a registry of RP object flags in the "RP Object Flag Field" sub-registry of the "Path Computation Element Protocol (PCEP) Numbers" registry.

IANA has allocated the following value:

Bit	Description	Reference
TBA	P2MP Core Tree bit	[This.I-D]

11.2. New PCEP Object

IANA is requested to assign a new object class in the registry of PCEP Objects as follows.

Object Class	Name	Object Type	Name	Reference
TBA	Pce-Seq	1	Pce Sequence [IPv4]	[This.I-D]
		2	Pce Sequence [IPv6]	[This.I-D]

12. Acknowledgements

The authors would like to thank Adrian Farrel, Dan Tappan and Olufemi Komolafe for their valuable comments on this document.

13. References

13.1. Normative References

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

[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.

13.2. Informative References

[RFC4461] Yasukawa, S., "Signaling Requirements for Point-to-Multipoint Traffic-Engineered MPLS Label Switched Paths (LSPs)", RFC 4461, April 2006.

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

RFC 4655, August 2006.

[RFC4657] Ash, J. and J. Le Roux, "Path Computation Element (PCE) Communication Protocol Generic Requirements", RFC 4657, September 2006.

[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.

[RFC5073] Vasseur, J. and J. Le Roux, "IGP Routing Protocol Extensions for Discovery of Traffic Engineering Node Capabilities", RFC 5073, December 2007.

[RFC5152] Vasseur, JP., Ayyangar, A., and R. Zhang, "A Per-Domain Path Computation Method for Establishing Inter-Domain Traffic Engineering (TE) Label Switched Paths (LSPs)", RFC 5152, February 2008.

[RFC5376] Bitar, N., Zhang, R., and K. Kumaki, "Inter-AS Requirements for the Path Computation Element Communication Protocol (PCECP)", RFC 5376, November 2008.

[RFC5394] Bryskin, I., Papadimitriou, D., Berger, L., and J. Ash, "Policy-Enabled Path Computation Framework", RFC 5394, December 2008.

[RFC5441] Vasseur, JP., Zhang, R., Bitar, N., and JL. Le Roux, "A Backward-Recursive PCE-Based Computation (BRPC) Procedure to Compute Shortest Constrained Inter-Domain Traffic Engineering Label Switched Paths", RFC 5441, April 2009.

[RFC5520] Bradford, R., Vasseur, JP., and A. Farrel, "Preserving Topology Confidentiality in Inter-Domain Path Computation Using a Path-Key-Based Mechanism", RFC 5520, April 2009.

[RFC5671] Yasukawa, S. and A. Farrel, "Applicability of the Path Computation Element (PCE) to Point-to-Multipoint (P2MP) MPLS and GMPLS Traffic Engineering (TE)", RFC 5671, October 2009.

[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.

[RFC5925] Touch, J., Mankin, A., and R. Bonica, "The TCP Authentication Option", RFC 5925, June 2010.

[H-PCE] King, D. and A. Farrel, "The Application of the Path Computation Element Architecture to the Determination of a Sequence of Domains in MPLS and GMPLS", October 2011.

[PCEP-MIB] Koushik, K., Stephan, E., Zhao, Q., and D. King, "PCE communication protocol (PCEP) Management Information Base (Work in Progress)", July 2010.

[PCEP-P2MP-MIB] Zhao, Q., Dhody, D., Palle, U., and D. King, "Management Information Base for the PCE Communications Protocol (PCEP) When Requesting Point-to-Multipoint Services (Work in Progress)", Sept 2011.

[DOMAIN-SEQ] Dhody, D., Palle, U., and R. Casellas, "Standard Representation Of Domain Sequence (Work in Progress)", Aug 2011.

[PER-DEST] Dhody, D. and U. Palle, "Supporting explicit-path per destination in Path Computation Element Communication Protocol (PCEP) - P2MP Path Request. (Work in Progress)", June 2011.

Authors' Addresses

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

EMail: quintin.zhao@huawei.com

Dhruv Dhody
Huawei Technology
Leela Palace
Bangalore, Karnataka 560008
INDIA

EMail: dhruv.dhody@huawei.com

Zafar Ali
Cisco Systems, Inc.
2000 Innovation Drive
Kanata, Ontario K2K 3E8
CANADA

EMail: zali@cisco.com

Tarek Saad
Cisco Systems, Inc.
US

EMail: tsaad@cisco.com

Siva Sivabalan
Cisco Systems, Inc.
2000 Innovation Drive
Kanata, Ontario K2K 3E8
CANADA

EMail: msiva@cisco.com

Daniel King
Old Dog Consulting
UK

EMail: daniel@olddog.co.uk

Ramon Casellas
CTTC - Centre Tecnologic de Telecomunicacions de Catalunya
Av. Carl Friedrich Gauss n7
Castelldefels, Barcelona 08860
SPAIN

EMail: ramon.casellas@cttc.e