                 PCE-Based Computation Procedure to Compute
           Shortest Constrained Point-to-Multipoint (P2MP) Inter-Domain
                   Traffic Engineering Label Switched Paths

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-controlled 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 an experiment to provide procedures and
   extensions to the PCE Communication Protocol (PCEP) for the
   computation of inter-domain paths for P2MP TE LSPs.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

   Multicast services are increasingly in demand for high-capacity
   applications such as multicast VPNs, IPTV (which may be on-demand or
   streamed), and content-rich media distribution (for example, software
   distribution, financial streaming, or database replication). The
   ability to compute constrained Traffic Engineering Label Switched
   Paths (TE LSPs) for point-to-multipoint (P2MP) LSPs in MPLS and GMPLS
   networks across multiple domains is therefore required.

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

   This document details the requirements for inter-domain P2MP path
   computation and then describes the experimental procedure "core-tree"
   path computation, developed to address the requirements and
   objectives for inter-domain P2MP path computation.

   When results of implementation and deployment are available, this
   document will be updated and refined, and it will then be moved from
   Experimental status to Standards Track.

1.1. Scope

   The inter-domain P2MP path computation procedures described in this
   document are experimental. The experiment is intended to enable
   research for the usage of the PCE to support inter-domain P2MP path
   computation.

   This document is not intended to replace the intra-domain P2MP path
   computation approach defined by [RFC6006] and will not impact
   existing PCE procedures and operations.

1.2. 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 RFC 2119 [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].

   Additional terms are defined below:

   Core-Tree: a P2MP tree where the root is the ingress Label Switching
   Router (LSR) and the leaf nodes are the entry boundary nodes of the
   leaf domains.

   Entry BN of domain(n): a boundary node (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.

   H-PCE: Hierarchical PCE (as per [RFC6805]).

   Leaf Domain: a domain with one or more leaf nodes.

   Path Tree: a set of LSRs and TE links that comprise the path of a
   P2MP TE LSP from the ingress LSR to all egress LSRs (the leaf nodes).

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

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

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

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

   Sub-tree: a P2MP tree where the root is the selected entry BN of the
   leaf domain and the leaf nodes are the destinations (leaves) in that
   domain. The sub-trees are grafted to the core-tree.

   Transit/Branch Domain: a domain that has an upstream and one or more
   downstream neighbor domains.

3. Examination of Existing Mechanisms

   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-controlled 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 point-to-point (P2P) inter-domain constrained shortest
   paths across a predetermined sequence of domains, using a Backward-
   Recursive PCE-Based Computation (BRPC) 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.

   PCEP [RFC5440] was extended for point-to-multipoint (P2MP) path
   computation requests in [RFC6006].

   As discussed in [RFC4461], a P2MP tree is the ordered 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. A P2MP LSP is set up with TE constraints
   and allows efficient packet or data replication at various branching
   points in the network. As per [RFC5671], selection of branch points
   is fundamental to the determination of the paths for a P2MP TE LSP.
   Not only is this selection constrained by the network topology and
   available network resources, but it is also determined by the
   objective functions (OFs) that may be applied to path computation.

   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 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 boundary nodes
   (i.e., when the domains are multiply inter-connected), existing
   techniques may be convoluted when used to determine which boundary
   node of another domain will be utilized for the inter-domain P2MP
   tree, and there is no way to limit the computation of the P2MP tree
   to those utilized boundary nodes.

A trivial solution to the computation of the 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 Minimum Cost Tree (MCT)), and in the
context of inter-domain P2MP TE LSPs, it cannot be used to reduce the
number of domain boundary nodes that are transited. Computing P2P TE
LSPs individually does not guarantee the generation of an optimal
P2MP tree for every definition of "optimal" in every topology.

Per-domain path computation [RFC5152] may be used to compute P2MP
multi-domain paths but may encounter the issues previously described.
Furthermore, this approach may be considered to have scaling issues
during LSP setup. That is, the LSP to each leaf is signaled
separately, and each boundary node needs to perform path computation
for each leaf.

P2MP MCT, i.e., a computation that guarantees the least cost
resulting tree, typically 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.

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

4. Assumptions

   Within this document we make the following assumptions:

   o Due to deployment and commercial limitations (e.g., inter-AS
      (Autonomous System) peering agreements), the path domain tree will
      be known in advance.

   o Each PCE knows about any leaf LSRs in the domain it serves.

   Additional assumptions are documented in [RFC5441] and are not
   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 time taken by any PCE implementation is 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 TE LSP are
   changed.

   It is also important that the solution can preserve confidentiality
   across domains, which is required when domains are managed by
   different Service Providers via the Path-Key mechanism [RFC5520].

   Other than the requirements specified in [RFC5862], a number of
   requirements specific to inter-domain P2MP are detailed below:

   1. 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.

   2. The number of PCReq (Path Computation Request) and PCRep (Path
       Computation Reply) messages SHOULD be independent of the number
       of multicast destinations in each domain.

   3. It SHOULD be possible to specify the domain entry and exit nodes
       in the PCReq.

   4. Specifying which nodes are to be used as branch nodes SHOULD be
       supported in the PCReq.

   5. Reoptimization of existing sub-trees SHOULD be supported.

   6. It SHOULD be possible to compute diverse P2MP paths from existing
       P2MP paths.

6. Objective Functions and Constraints

   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 used. SPT (Shortest Path Tree) and MCT (Minimum Cost
   Tree), defined in [RFC6006], are two such OF optimization criteria
   for the sub-tree within each domain used to select the "best"
   candidate path.

   In addition to the OFs, the following constraints MAY also be
   beneficial for inter-domain P2MP path computation:

   1. The computed P2MP "core-tree" SHOULD be optimal when only
       considering the paths to the leaf domain entry BNs.

   2. Grafting and pruning of multicast destinations (sub-trees) within
       a leaf domain SHOULD ensure minimal impact on other domains and
       on the core-tree.

   3. It SHOULD be possible to choose to optimize the core-tree.

   4. It SHOULD be possible to choose to optimize the entire tree (P2MP
       LSP).

   5. It SHOULD be possible to combine the aforementioned OFs and
       constraints for P2MP path computation.

   When implementing and operating P2MP LSPs, the following need to be
   taken into consideration:

   o The complexity of computation.

   o The optimality of the tree (core-tree as well as full P2MP LSP
      tree).

   o The stability of the core-tree.

   The solution SHOULD allow these trade-offs to be made at computation
   time.

   The algorithms used to compute optimal paths using a combination of
   OFs and multiple constraints are out of the scope of this document.

7. P2MP Path Computation Procedures

7.1. General

   A P2MP path computation can be broken down into two steps: core-tree
   computation and grafting of sub-trees. Breaking the procedure into
   these specific steps has the following impact, allowing the core-
   tree-based solution to provide an optimal inter-domain P2MP TE LSP:

   o The core-tree and sub-tree are smaller in comparison to the full
      P2MP tree and are thus easier to compute.

   o An implementation MAY choose to keep the core-tree fairly static
      or computed offline (trade-off with optimality).

   o Adding/pruning of leaves requires changes to the sub-tree in the
      leaf-domain only.

   o The PCEP message size is smaller in comparison.

   The following sub-sections describe the core-tree-based mechanism,
   including procedures and PCEP extensions that satisfy the
   requirements and objectives specified in Sections 5 and 6 of this
   document.

7.2. Core-Trees

   A core-tree is defined as a tree that satisfies the following
   conditions:

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

   o The leaves of the core-tree are the entry boundary nodes in the
      leaf domains.

   To support confidentiality, these nodes and links MAY be hidden using
   the Path-Key mechanism [RFC5520], but they MUST be computed and be a
   part of the core-tree.

For example, consider the domain tree in Figure 1, 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------------W+    +------P---------+    +----------T----+
    |              |    |                |    |               |
    |              |    |   Q            |    |   U           |
    |   M      O   |    |        S       |    |               |
    |              |    |                |    |         V     |
    |        N     |    |   R            |    |               |
    +--------------+    +----------------+    +---------------+
```

Figure 1: Domain Tree Example

```
                            (R)
                             |
                            (A)
                            / \
                           /   \
                        (B)     (C)
                        /          \
                       /            \
                    (D)              (E)
                    /                 |
                   /                  |
                (G)                  (H)
                /                    / \
               /                    /   \
            (I)                  (J)     (K)
            / \                  /          \
           /   \                /            \
        (L)     (W)          (P)              (T)
```

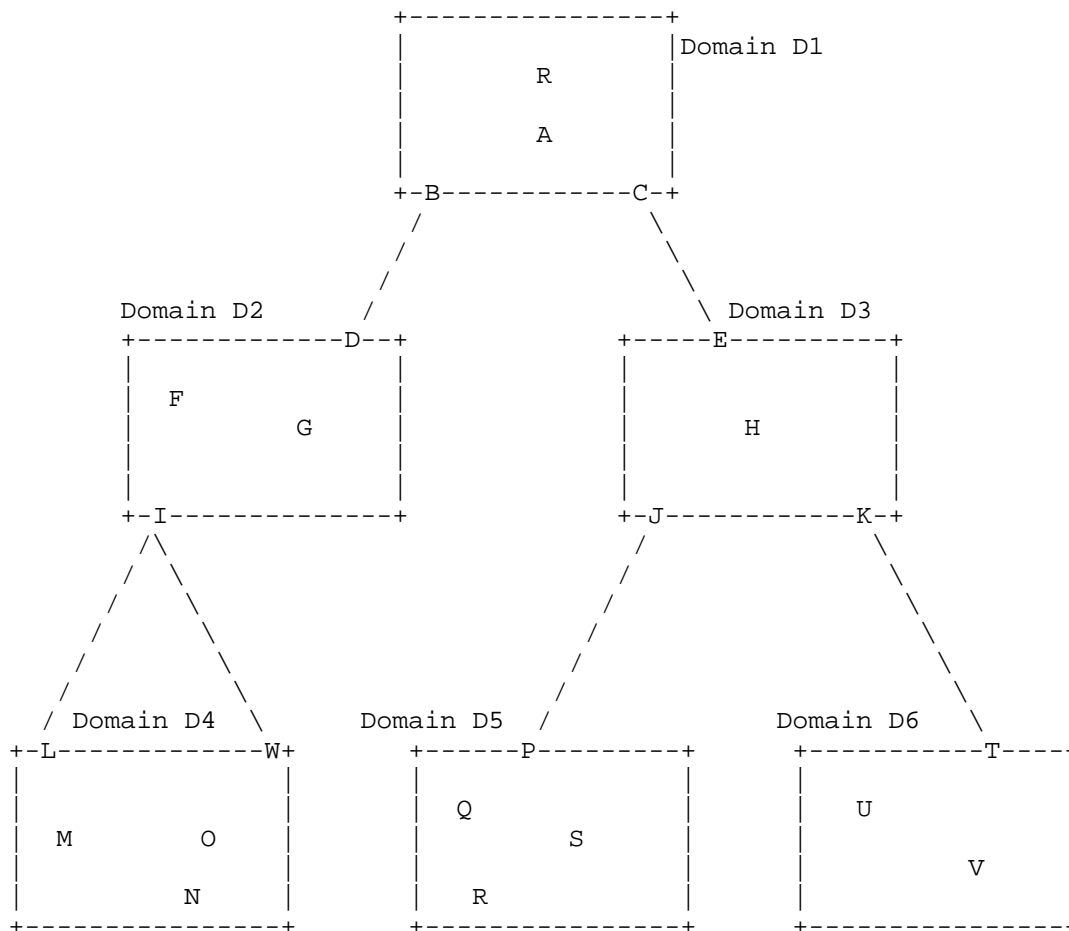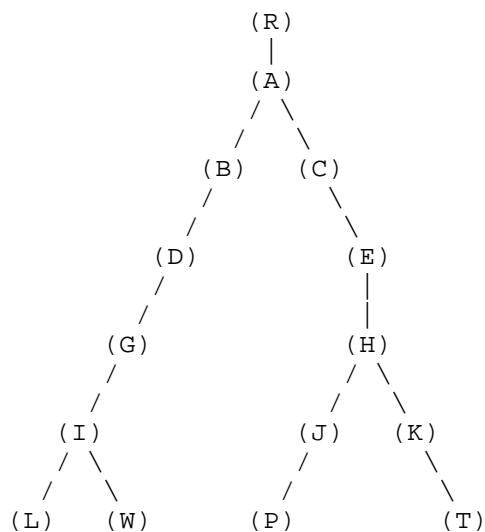   Figure 2: Core-Tree

   A core-tree is computed such that the root of the tree is R and the
   leaf nodes are the entry nodes of the destination domains (L, W, P,
   and T). The Path-Key mechanism can be used to hide the internal
   nodes and links in the final core-tree as shown below for domains D2
   and D3 (nodes G and H are hidden via Path-Keys PK1 and PK2,
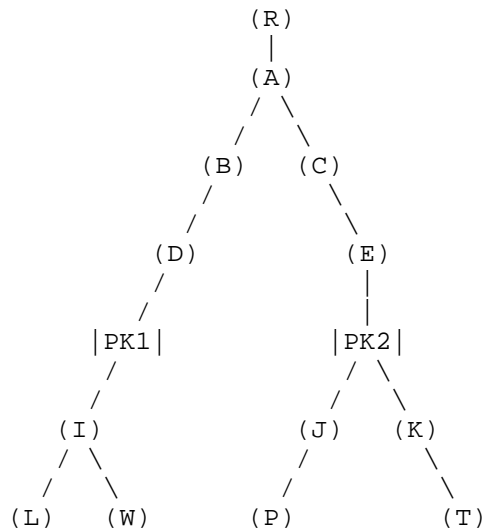   respectively).

```
                                (R)
                                 |
                                (A)
                                / \
                               /   \
                             (B)   (C)
                             /       \
                            /         \
                          (D)         (E)
                          /            |
                         /             |
                      |PK1|          |PK2|
                       /              / \
                      /              /   \
                    (I)            (J)   (K)
                    / \            /       \
                   /   \          /         \
                 (L)   (W)      (P)         (T)
```

                 Figure 3: Core-Tree with Path-Key

7.3. Optimal Core-Tree Computation Procedure

   Applying the core-tree procedure to large groups of domains, such as
   the Internet, is not considered feasible or desirable and is out of
   the scope of this document.

   The following extended BRPC-based procedure can be used to compute
   the core-tree. Note that a root PCE MAY further use its own enhanced
   optimization techniques in the future to compute the core-tree.

   A BRPC-based core-tree path computation procedure is described below:

   1. Use the BRPC procedures to compute the VSPT(i) (Virtual Shortest
       Path Tree) 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 all possible sets of paths. We
       call them 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.
       Form these n paths into a core-tree(j).

   4. There will be M number core-trees computed from step 3. An
       optimal core-tree is selected based on the OF and constraints.

Note that since the point-to-point BRPC procedure is used to compute
VSPT, the path request and response message formats defined in
[RFC5440] are used.

Also note that the application of BRPC in the aforementioned
procedure differs from the typical one since paths returned from a
downstream PCE are not necessarily pruned from the solution set
(extended VSPT) by intermediate PCEs. The reason for this is that if
the PCE in a downstream domain does the pruning and returns the
single optimal sub-path to the upstream PCE, the combination of these
single optimal sub-paths into a core-tree is not necessarily optimal
even if each S2L (Source-to-Leaf) sub-path is optimal.

Without trimming, the ingress PCE will obtain all the possible S2L
sub-paths set for the entry boundary nodes of the leaf domain. By
looking through all the combinations and taking one sub-path from
each set to build one tree, the PCE will then select the optimal
core-tree.

A PCE MAY add equal-cost paths within the domain while constructing
an extended VSPT. This will provide the ingress PCE more candidate
paths for an optimal core-tree.

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), especially with dense/meshed
domains. Considering a domain sequence D1, D2, D3, D4, where the
leaf boundary node is at domain D4, PCE(4) will return 1 path.
PCE(3) will return N paths, where N is $E(3) \times X(3)$, where $E(k) \times X(k)$
denotes the number of entry nodes times the number of exit nodes for
that domain. PCE(2) will return M paths, where $M = E(2) \times X(2) \times N =
E(2) \times X(2) \times E(3) \times X(3) \times 1$, etc. Generally speaking, the number
of potential paths at the ingress PCE $Q = \text{prod } E(k) \times X(k)$.

Consequently, it is expected that the core-tree will typically be
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).

7.4. 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 the C-bit set (as defined in Section 7.5.1
   of this document) for the path computation to the destination(s)
   directly to the PCE where the destination(s) belong(s) along with the
   core-tree computed per Section 7.2.

   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 neighbor
   domains.

   An 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.

   Note that the P2MP path request and response format is as per
   [RFC6006], where Record Route Objects (RROs) are used to carry the
   core-tree paths in the P2MP grafting request.

   The algorithms to compute the optimal large sub-tree are outside the
   scope of this document.

7.5. PCEP Protocol Extensions

7.5.1. Extension of RP Object

   This experiment will be carried out by extending the RP (Request
   Parameters) object (defined in [RFC5440]) used in PCEP requests and
   responses.

   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 whether or not the request/reply is for
   inter-domain P2MP core-tree.

   The following flag is added in this document:

      Bit Number   Name Flag
      17   Core-tree computation (C-bit)

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

      0: Indicates that this is not for an inter-domain P2MP core-tree.

      1: Indicates that this is a PCEP request or a response for the
          computation of an inter-domain core-tree or for the grafting
          of a sub-tree to an inter-domain core-tree.

7.5.2. Domain and PCE Sequence

   The procedure described in this document requires the domain tree to
   be known in advance. This information MAY be either administratively
   predetermined or dynamically discovered by some means, such as the
   Hierarchical PCE (H-PCE) framework [RFC6805], or derived through the
   IGP/BGP routing information.

   Examples of ways to encode the domain path tree are found in
   [RFC5886], which uses PCE-ID Objects, and [DOMAIN-SEQ].

7.6. Using H-PCE for Scalability

   The ingress/root PCE is responsible for the core-tree computation as
   well as grafting of sub-trees into the multi-domain tree. Therefore,
   the ingress/root PCE will receive all computed path segments from all
   the involved domains. When the ingress/root PCE chooses to have a
   PCEP session with all involved PCEs, this may cause an excessive
   number of sessions or added complexity in implementations.

   The H-PCE framework [RFC6805] may be used to establish a dedicated
   PCE with the capability (memory and CPU) and knowledge to maintain
   the necessary PCEP sessions. The parent PCE would be responsible for
   sending an intra-domain path computation request to the PCEs,
   combining them, and returning the overall P2MP tree.

7.7. Parallelism

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

   1. The BRPC procedure for each leaf boundary node can be launched in
       parallel by the ingress/root PCE for dynamic computation of the
       core-tree.

   2. The grafting of sub-trees can be triggered in parallel once the
       core-tree is computed.

   One of the potential issues of parallelism is that the ingress PCE
   would require a potentially high number of PCEP adjacencies to
   "remote" PCEs at the same time; this situation may not be desirable.

8. Protection

   It is envisaged that protection may be required when deploying and
   using inter-domain P2MP TE 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
   [RFC4872] and domain protection schemes.

   Segment or facility (link and node) protection is problematic in
   inter-domain environments due to the limit of fast reroute (FRR)
   [RFC4875] requiring knowledge of its next hop across domain
   boundaries while maintaining domain confidentiality. However, the
   FRR protection might be implemented if next-hop information was known
   in advance.

8.1. End-to-End Protection

   An end-to-end protection (for nodes and links) principle can be
   applied for computing backup P2MP TE LSPs. During computation of the
   core-tree and sub-trees, protection may also be taken into
   consideration. A PCE may compute the primary and backup P2MP TE LSP
   together or sequentially.

8.2. Domain Protection

   In this protection scheme, a backup P2MP tree can be computed that
   excludes the transit/branch domain completely. A backup domain path
   tree is needed with the same source domain and destination domains
   and a new set of transit domains. The backup path tree can be
   applied to the above procedure to obtain the backup P2MP TE 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
   the 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 the manageability requirements mentioned in
   [RFC6006] are already covered there.

9.1. Control of Function and Policy

   In addition to the PCE configuration parameters listed in [RFC5440]
   and [RFC6006], the following additional parameters might be required:

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

   o Configuration of the PCE to enable or disable the advertisement of
      its 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, communication between PCEs, and
   P2MP path computation requests and responses.

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
   computations. The liveness detection and monitoring features of the
   PCEP 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 PCEP. Note that
   verification of the correct operation of the PCE and its algorithms
   is out of the scope of 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
   TE 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 and the discovery of
   corresponding PCEs and their capabilities SHOULD be provided, and
   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.

It should be noted that pre-computing and maintaining domain trees
might introduce considerable administration effort for the operator.

9.7. Policy Control

[RFC5394] provides additional details on policy within the PCE
architecture and also provides context for the support of PCE Policy.
They are also applicable to inter-domain P2MP path computation via
the 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:

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

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

o 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].

Finally, any multi-domain operation necessarily involves the exchange
of information across domain boundaries. This may represent a
significant security and confidentiality risk, especially when the
domains are controlled by different commercial entities. PCEP allows

      individual PCEs to maintain confidentiality of their domain path
      information by using Path-Keys [RFC5520] and would allow for securing
      of domain path information when performing core-tree-based path
      computations.

11. IANA Considerations

      IANA maintains the "Path Computation Element Protocol (PCEP) Numbers"
      registry and the "RP Object Flag Field" sub-registry.

      IANA has allocated a new bit from this registry as follows:

         Bit   Description   Reference
         17   Core-tree computation (C-bit)   [RFC7334]

12. Acknowledgements

      The authors would like to thank Adrian Farrel, Dan Tappan, Olufemi
      Komolafe, Oscar Gonzalez de Dios, and Julien Meuric 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., Ed., and JL. Le Roux, Ed., "Path
                       Computation Element (PCE) Communication Protocol
                       (PCEP)", RFC 5440, March 2009.

      [RFC5441]   Vasseur, JP., Ed., 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.

      [RFC6006]   Zhao, Q., Ed., King, D., Ed., Verhaeghe, F., Takeda,
                       T., Ali, Z., and J. Meuric, "Extensions to the Path
                       Computation Element Communication Protocol (PCEP)
                       for Point-to-Multipoint Traffic Engineering Label
                       Switched Paths", RFC 6006, September 2010.

13.2. Informative References

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

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

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

   [RFC4872]   Lang, J., Ed., Rekhter, Y., Ed., and D.
                   Papadimitriou, Ed., "RSVP-TE Extensions in Support
                   of End-to-End Generalized Multi-Protocol Label
                   Switching (GMPLS) Recovery", RFC 4872, May 2007.

   [RFC4875]   Aggarwal, R., Ed., Papadimitriou, D., Ed., and S.
                   Yasukawa, Ed., "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., Ed., and J. Le Roux, Ed., "IGP Routing
                   Protocol Extensions for Discovery of Traffic
                   Engineering Node Capabilities", RFC 5073, December
                   2007.

   [RFC5152]   Vasseur, JP., Ed., Ayyangar, A., Ed., 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.

   [RFC5520]   Bradford, R., Ed., 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, Ed., "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.

   [RFC5886]   Vasseur, JP., Ed., Le Roux, JL., and Y. Ikejiri, "A
                      Set of Monitoring Tools for Path Computation Element
                      (PCE)-Based Architecture", RFC 5886, June 2010.

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

   [RFC6805]   King, D., Ed., and A. Farrel, Ed., "The Application
                      of the Path Computation Element Architecture to the
                      Determination of a Sequence of Domains in MPLS and
                      GMPLS", RFC 6805, November 2012.

   [PCEP-MIB]   Koushik, A., Stephan, E., Zhao, Q., King, D., and J.
                      Hardwick, "Path Computation Element Protocol (PCEP)
                      Management Information Base", Work in Progress,
                      July 2014.

   [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,
                      August 2012.

   [DOMAIN-SEQ]   Dhody, D., Palle, U., and R. Casellas, "Standard
                      Representation Of Domain-Sequence", Work in
                      Progress, July 2014.

14. Contributors' Addresses

   Siva Sivabalan
   Cisco Systems
   2000 Innovation Drive
   Kanata, Ontario K2K 3E8
   Canada

   EMail: msiva@cisco.com

   Tarek Saad
   Cisco Systems, Inc.
   2000 Innovation Drive
   Kanata, Ontario K2K 3E8
   Canada

   EMail: tsaad@cisco.com

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

   Daniel King
   Old Dog Consulting
   UK

   EMail: daniel@olddog.co.uk

   Zafar Ali
   Cisco Systems
   2000 Innovation Drive
   Kanata, Ontario   K2K 3E8
   Canada

   EMail: zali@cisco.com

   Ramon Casellas
   CTTC
   Av. Carl Friedrich Gauss n7
   Castelldefels, Barcelona   08860
   Spain

   EMail: ramon.casellas@cttc.es