Network Working Group
Internet-Draft
Intended Status: Informational
Expires: 28 November 2012
D. King (Ed.)
Old Dog Consulting
A. Farrel (Ed.)
Old Dog Consulting
28 June 2012

The Application of the Path Computation Element Architecture to the Determination of a Sequence of Domains in MPLS and GMPLS

draft-ietf-pce-hierarchy-fwk-04.txt

Abstract

Computing optimum routes for Label Switched Paths (LSPs) across multiple domains in MPLS Traffic Engineering (MPLS-TE) and GMPLS networks presents a problem because no single point of path computation is aware of all of the links and resources in each domain. A solution may be achieved using the Path Computation Element (PCE) architecture.

Where the sequence of domains is known a priori, various techniques can be employed to derive an optimum path. If the domains are simply-connected, or if the preferred points of interconnection are also known, the Per-Domain Path Computation technique can be used. Where there are multiple connections between domains and there is no preference for the choice of points of interconnection, the Backward Recursive Path Computation Procedure (BRPC) can be used to derive an optimal path.

This document examines techniques to establish the optimum path when the sequence of domains is not known in advance. The document shows how the PCE architecture can be extended to allow the optimum sequence of domains to be selected, and the optimum end-to-end path to be derived through the use of a hierarchical relationship between domains.

Contents

1. Introduction

The capability to compute the routes of end-to-end inter-domain MPLS Traffic Engineering (TE) and GMPLS Label Switched Paths (LSPs) is expressed as requirements in [RFC4105] and [RFC4216]. This capability may be realized by a Path Computation Element (PCE). The PCE architecture is defined in [RFC4655]. The methods for establishing and controlling inter-domain MPLS-TE and GMPLS LSPs are documented in [RFC4726].

In this context, a domain can be defined as a separate administrative, geographic, or switching environment within the network. A domain may be further defined as a zone of routing or computational ability. Under these definitions a domain might be categorized as an Antonymous System (AS) or an Interior Gateway Protocol (IGP) area [RFC4726] and [RFC4655]. Domains are connected through ingress and egress boundary nodes (BNs). A more detailed definition is given in Section 1.2.

In a multi-domain environment, the determination of an end-to-end traffic engineered path is a problem because no single point of path computation is aware of all of the links and resources in each domain. PCEs can be used to compute end-to-end paths using a per-domain path computation technique [RFC5152]. Alternatively, the backward recursive path computation (BRPC) mechanism [RFC5441] allows multiple PCEs to collaborate in order to select an optimal end-to-end path that crosses multiple domains. Both mechanisms assume that the sequence of domains to be crossed between ingress and egress is known in advance.

This document examines techniques to establish the optimum path when the sequence of domains is not known in advance. It shows how the PCE architecture can be extended to allow the optimum sequence of domains to be selected, and the optimum end-to-end path to be derived.

The model described in this document introduces a hierarchical relationship between domains. It is applicable to environments with small groups of domains where visibility from the ingress Label Switching Router (LSR) is limited. Applying the hierarchical PCE model to large groups of domains such as the Internet, is not considered feasible or desirable, and is out of scope for this document.

This document does not specify any protocol extensions or enhancements. That work is left for future protocol specification documents. However, some assumptions are made about which protocols will be used to provide specific functions, and guidelines to future protocol developers are made in the form of requirements statements.

1.1 Problem Statement

Using a PCE to compute a path between nodes within a single domain is relatively straightforward. Computing an end-to-end path when the source and destination nodes are located in different domains requires co-operation between multiple PCEs, each responsible for its own domain.

Techniques for inter-domain path computation described so far

([RFC5152] and [RFC5441]) assume that the sequence of domains to be crossed from source to destination is well known. No explanation is given (for example, in [RFC4655]) of how this sequence is generated or what criteria may be used for the selection of paths between domains. In small clusters of domains, such as simple cooperation between adjacent ISPs, this selection process is not complex. In more advanced deployments (such as optical networks constructed from multiple sub-domains, or in multi-AS environments) the choice of domains in the end-to-end domain sequence can be critical to the determination of an optimum end-to-end path.

1.2 Definition of a Domain

A domain is defined in [RFC4726] as any collection of network elements within a common sphere of address management or path computational responsibility. Examples of such domains include IGP areas and Autonomous Systems. Wholly or partially overlapping domains are not within the scope of this document.

In the context of GMPLS, a particularly important example of a domain is the Automatically Switched Optical Network (ASON) subnetwork [G-8080]. In this case, a domain might be an ASON Routing Area [G-7715]. Furthermore, computation of an end-to-end path requires the selection of nodes and links within a routing area where some nodes may, in fact, be subnetworks. A PCE may perform the path computation function of an ASON Routing Controller as described in [G-7715-2]. See Section 5.2 for a further discussion of the applicability to the ASON architecture.

This document assumes that the selection of a sequence of domains for an end-to-end path is in some sense a hierarchical path computation problem. That is, where one mechanism is used to determine a path across a domain, a separate mechanism (or at least a separate set of paradigms) is used to determine the sequence of domains. The responsibility for the selection of domain interconnection can belong to either or both of the mechanisms.

1.3 Assumptions and Requirements

Networks are often constructed from multiple domains. These domains are often interconnected via multiple interconnect points. Its assumed that the sequence of domains for an end-to-end path is not always well known; that is, an application requesting end-to-end connectivity has no preference for, or no ability to specify, the sequence of domains to be crossed by the path.

The traffic engineering properties of a domain cannot be seen from outside the domain. Traffic engineering aggregation or abstraction,

hides information and can lead to failed path setup or the selection of suboptimal end-to-end paths [RFC4726]. The aggregation process may also have significant scaling issues for networks with many possible routes and multiple TE metrics. Flooding TE information breaks confidentiality and does not scale in the routing protocol. See Section 6 for a discussion of the concept of inter-domain traffic engineering information exchange known as BGP-TE.

The primary goal of this document is to define how to derive optimal end-to-end, multi-domain paths when the sequence of domains is not known in advance. The solution needs to be scalable and to maintain internal domain topology confidentiality while providing the optimal end-to-end path. It cannot rely on the exchange of TE information between domains, and for the confidentiality, scaling, and aggregation reasons just cited, it cannot utilize a computation element that has universal knowledge of TE properties and topology of all domains.

The sub-sections that follow set out the primary objectives and requirements to be satisfied by a PCE solution to multi-domain path computation.

1.3.1 Metric Objectives

The definition of optimality is dependent on policy, and is based on a single objective or a group objectives. An objective is expressed as an objective function [RFC5541] and may be specified on a path computation request. The following objective functions are identified in this document. They define how the path metrics and TE link qualities are manipulated during inter-domain path computation. The list is not proscriptive and may be expanded in other documents.

o Minimize the cost of the path [RFC5541]
o Select a path using links with the minimal load [RFC5541]
o Select a path that leaves the maximum residual bandwidth [RFC5541]
o Minimize aggregate bandwidth consumption [RFC5541]
o Minimize the Load of the most loaded Link [RFC5541]
o Minimize the Cumulative Cost of a set of paths [RFC5541]
o Minimize or cap the number of domains crossed
o Disallow domain re-entry

See Section 5.1 for further discussion of objective functions.

1.3.2 Diversity

1.3.2.1 Physical Diversity

Within a Carrier's Carrier environment MPLS services may share common underlying equipment and resources, including optical fiber and

nodes. An MPLS service request may require a path for traffic that is physically disjointed from another service. Thus, if a physical link or node fails on one of the disjoint paths, not all traffic is lost.

1.3.2.2 Domain Diversity

A pair of paths are domain-diverse if they do not transit any of the same domains. A pair of paths that share a common ingress and egress are domain-diverse if they only share the same domains at the ingress and egress (the ingress and egress domains). Domain diversity may be maximized for a pair of paths by selecting paths that have the smallest number of shared domains. (Note that this is not the same as finding paths with the greatest number of distinct domains!)

Path computation should facilitate the selection of paths that share ingress and egress domains, but do not share any transit domains. This provides a way to reduce the risk of shared failure along any path, and automatically helps to ensure path diversity for most of the route of a pair of LSPs.

Thus, domain path selection should provide the capability to include or exclude specific domains and specific boundary nodes.

1.3.3 Existing Traffic Engineering Constraints

Any solution should take advantage of typical traffic engineering constraints (hop count, bandwidth, lambda continuity, path cost, etc.) to meet the service demands expressed in the path computation request [RFC4655].

1.3.4 Commercial Constraints

The solution should provide the capability to include commercially relevant constraints such as policy, SLAs, security, peering preferences, and monetary costs.

Additionally it may be necessary for the service provider to request that specific domains are included or excluded based on commercial relationships, security implications, and reliability.

1.3.5 Domain Confidentiality

A key requirement is the ability to maintain domain confidentiality when computing inter-domain end-to-end paths. It should be possible for local policy to require that a PCE not disclose to any other PCE the intra-domain paths it computes or the internal topology of the domain it serves. This requirement should have no impact in the optimality or quality of the end-to-end path that is derived.

1.3.6 Limiting Information Aggregation

In order to reduce processing overhead and to not sacrifice computational detail, there should be no requirement to aggregate or abstract traffic engineering link information.

1.3.7 Domain Interconnection Discovery

To support domain mesh topologies, the solution should allow the discovery and selection of domain inter-connections. Pre-configuration of preferred domain interconnections should also be supported for network operators that have bilateral agreement, and preference for the choice of points of interconnection.

1.4 Terminology

This document uses PCE terminology defined in [RFC4655], [RFC4726], and [RFC5440]. Additional terms are defined below.

Domain Path: The sequence of domains for a path.

Ingress Domain: The domain that includes the ingress LSR of a path.

Transit Domain: A domain that has an upstream and downstream neighbor domain for a specific path.

Egress Domain: The domain that includes the egress LSR of a path.

Boundary Nodes: Each Domain has entry LSRs and exit LSRs that could be Area Border Routers (ABRs) or Autonomous System Border Routers (ASBRs) depending on the type of domain. They are defined here more generically as Boundary Nodes (BNs).

Entry BN of domain(n): a BN connecting domain(n-1) to domain(n) on a path.

Exit BN of domain(n): a BN connecting domain(n) to domain(n+1) on a path.

Parent Domain: A domain higher up in a domain hierarchy such that it contains other domains (child domains) and potentially other links and nodes.

Child Domain: A domain lower in a domain hierarchy such that it has a parent domain.

Parent PCE: A PCE responsible for selecting a path across a parent domain and any number of child domains by coordinating with child PCEs and examining a topology map that shows domain interconnectivity.

Child PCE: A PCE responsible for computing the path across one or more specific (child) domains. A child PCE maintains a relationship with at least one parent PCE.

OF: Objective Function: A set of one or more optimization criteria used for the computation of a single path (e.g., path cost minimization), or the synchronized computation of a set of paths (e.g., aggregate bandwidth consumption minimization). See [RFC4655] and [RFC5541].

2. Examination of Existing PCE Mechanisms

This section provides a brief overview of two existing PCE cooperation mechanisms called the per-domain path computation method, and the backward recursive path computation method. It describes the applicability of these methods to the multi-domain problem.

2.1 Per-Domain Path Computation

The per-domain path computation method for establishing inter-domain TE-LSPs [RFC5152] defines a technique whereby the path is computed during the signalling process on a per-domain basis. The entry BN of each domain is responsible for performing the path computation for the section of the LSP that crosses the domain, or for requesting that a PCE for that domain computes that piece of the path.

During per-domain path computation, each computation results in the best path across the domain to provide connectivity to the next domain in the domain sequence (usually indicated in signalling by an identifier of the next domain or the identity of the next entry BN).

Per-domain path computation may lead to sub-optimal end-to-end paths because the most optimal path in one domain may lead to the choice of an entry BN for the next domain that results in a very poor path across that next domain.

In the case that the domain path (in particular, the sequence of boundary nodes) is not known, the path computing entity must select an exit BN based on some determination of how to reach the destination that is outside the domain for which the path computing entity has computational responsibility. [RFC5152] suggest that this might be achieved using the IP shortest path as advertise by BGP. Note, however, that the existence of an IP forwarding path does not guarantee the presence of sufficient bandwidth, let alone an optimal TE path. Furthermore, in many GMPLS systems inter-domain IP routing will not be present. Thus, per-domain path computation may require a significant number of crankback routing attempts to establish even a sub-optimal path.

Note also that the path computing entities in each domain may have different computation capabilities, may run different path computation algorithms, and may apply different sets of constraints and optimization criteria, etc.

This can result in the end-to-end path being inconsistent and sub-optimal.

Per-domain path computation can suit simply-connected domains where the preferred points of interconnection are known.

2.2 Backward Recursive Path Computation

The Backward Recursive Path Computation (BRPC) [RFC5441] procedure involves cooperation and communication between PCEs in order to compute an optimal end-to-end path across multiple domains. The sequence of domains to be traversed can either be determined before or during the path computation. In the case where the sequence of domains is known, the ingress Path Computation Client (PCC) sends a path computation request to a PCE responsible for the ingress domain. This request is forwarded between PCEs, domain-by-domain, to a PCE responsible for the egress domain. The PCE in the egress domain creates a set of optimal paths from all of the domain entry BNs to the egress LSR. This set is represented as a tree of potential paths called a Virtual Shortest Path Tree (VSPT), and the PCE passes it back to the previous PCE on the domain path. As the VSPT is passed back toward the ingress domain, each PCE computes the optimal paths from its entry BNs to its exit BNs that connect to the rest of the tree. It adds these paths to the VSPT and passes the VSPT on until the PCE for the ingress domain is reached and computes paths from the ingress LSR to connect to the rest of the tree. The ingress PCE then selects the optimal end-to-end path from the tree, and returns the path to the initiating PCC.

BRPC may suit environments where multiple connections exist between domains and there is no preference for the choice of points of interconnection. It is best suited to scenarios where the domain path is known in advance, but can also be used when the domain path is not known.

2.2.1. Applicability of BRPC when the Domain Path is Not Known

As described above, BRPC can be used to determine an optimal inter-domain path when the domain sequence is known. Even when the sequence of domains is not known BRPC could be used as follows.

o The PCC sends a request to a PCE for the ingress domain (the ingress PCE).

- o The ingress PCE sends the path computation request direct to a PCE responsible for the domain containing the destination node (the egress PCE).

- o The egress PCE computes an egress VSPT and passes it to a PCE responsible for each of the adjacent (potentially upstream) domains.

- o Each PCE in turn constructs a VSPT and passes it on to all of its neighboring PCEs.

- o When the ingress PCE has received a VSPT from each of its neighboring domains it is able to select the optimum path.

Clearly this mechanism (which could be called path computation flooding) has significant scaling issues. It could be improved by the application of policy and filtering, but such mechanisms are not simple and would still leave scaling concerns.

3. Hierarchical PCE

In the hierarchical PCE architecture, a parent PCE maintains a domain topology map that contains the child domains (seen as vertices in the topology) and their interconnections (links in the topology). The parent PCE has no information about the content of the child domains; that is, the parent PCE does not know about the resource availability within the child domains, nor about the availability of connectivity across each domain because such knowledge would violate the confidentiality requirement and would either require flooding of full information to the parent (scaling issue) or would necessitate some form of aggregation. The parent PCE is aware of the TE capabilities of the interconnections between child domains as these interconnections are links in its own topology map.

Note that in the case that the domains are IGP areas, there is no link between the domains (the ABRs have a presence in both neighboring areas). The parent domain may choose to represent this in its TED as a virtual link that is unconstrained and has zero cost, but this is entirely an implementation issue.

Each child domain has at least one PCE capable of computing paths across the domain. These PCEs are known as child PCEs and have a relationship with the parent PCE. Each child PCE also knows the identity of the domains that neighbor its own domain. A child PCE only knows the topology of the domain that it serves and does not know the topology of other child domains. Child PCEs are also not aware of the general domain mesh connectivity (i.e., the domain

topology map) beyond the connectivity to the immediate neighbor domains of the domain it serves.

The parent PCE builds the domain topology map either from configuration or from information received from each child PCE. This tells it how the domains are interconnected including the TE properties of the domain interconnections. But the parent PCE does not know the contents of the child domains. Discovery of the domain topology and domain interconnections is discussed further in Section 4.3.

When a multi-domain path is needed, the ingress PCE sends a request to the parent PCE (using the path computation element protocol, PCEP [RFC5440]). The parent PCE selects a set of candidate domain paths based on the domain topology and the state of the inter-domain links. It then sends computation requests to the child PCEs responsible for each of the domains on the candidate domain paths. These requests may be sequential or parallel depending on implementation details.

Each child PCE computes a set of candidate path segments across its domain and sends the results to the parent PCE. The parent PCE uses this information to select path segments and concatenate them to derive the optimal end-to-end inter-domain path. The end-to-end path is then sent to the child PCE which received the initial path request and this child PCE passes the path on to the PCC that issued the original request.

Specific deployment and implementation scenarios are out of scope of this document. However the hierarchical PCE architecture described does support the function of parent PCE and child PCE being implemented as a common PCE.

4. Hierarchical PCE Procedures

4.1 Objective Functions and Policy

Deriving the optimal end-to-end domain path sequence is dependent on the policy applied during domain path computation. An Objective Function (OF) [RFC5541], or set of OFs, may be applied to define the policy being applied to the domain path computation.

The OF specifies the desired outcome of the computation. It does not describe the algorithm to use. When computing end-to-end inter-domain paths, required OFs may include (see Section 1.3.1):

- o Minimum cost path
- o Minimum load path
- o Maximum residual bandwidth path

o Minimize aggregate bandwidth consumption

o Minimize or cap the number of transit domains
o Disallow domain re-entry

The objective function may be requested by the PCC, the ingress domain PCE (according to local policy), or applied by the parent PCE according to inter-domain policy.

More than one OF (or a composite OF) may be chosen to apply to a single computation provided they are not contradictory. Composite OFs may include weightings and preferences for the fulfillment of pieces of the desired outcome.

4.2 Maintaining Domain Confidentiality

Information about the content of child domains is not shared for scaling and confidentiality reasons. This means that a parent PCE is aware of the domain topology and the nature of the connections between domains, but is not aware of the content of the domains. Similarly, a child PCE cannot know the internal topology of another child domain. Child PCEs also do not know the general domain mesh connectivity, this information is only known by the parent PCE.

As described in the earlier sections of this document, PCEs can exchange path information in order to construct an end-to-end inter-domain path. Each per-domain path fragment reveals information about the topology and resource availability within a domain. Some management domains or ASes will not want to share this information outside of the domain (even with a trusted parent PCE). In order to conceal the information, a PCE may replace a path segment with a path-key [RFC5520]. This mechanism effectively hides the content of a segment of a path.

4.3 PCE Discovery

It is a simple matter for each child PCE to be configured with the address of its parent PCE. Typically, there will only be one or two parents of any child.

The parent PCE also needs to be aware of the child PCEs for all child domains that it can see. This information is most likely to be configured (as part of the administrative definition of each domain).

Discovery of the relationships between parent PCEs and child PCEs does not form part of the hierarchical PCE architecture. Mechanisms that rely on advertising or querying PCE locations across domain or provider boundaries are undesirable for security, scaling, commercial, and confidentiality reasons.

The parent PCE also needs to know the inter-domain connectivity. This information could be configured with suitable policy and commercial rules, or could be learned from the child PCEs as described in Section 4.4.

In order for the parent PCE to learn about domain interconnection the child PCE will report the identity of its neighbor domains. The IGP in each neighbor domain can advertise its inter-domain TE link capabilities [RFC5316], [RFC5392]. This information can be collected by the child PCEs and forwarded to the parent PCE, or the parent PCE could participate in the IGP in the child domains.

4.4 Parent Domain Traffic Engineering Database

The parent PCE maintains a domain topology map of the child domains and their interconnectivity. Where inter-domain connectivity is provided by TE links the capabilities of those links may also be known to the parent PCE. The parent PCE maintains a traffic engineering database (TED) for the parent domain in the same way that any PCE does.

The parent domain may just be the collection of child domains and their interconnectivity, may include details of the inter-domain TE links, and may contain nodes and links in its own right.

The mechanism for building the parent TED is likely to rely heavily on administrative configuration and commercial issues because the network was probably partitioned into domains specifically to address these issues.

In practice, certain information may be passed from the child domains to the parent PCE to help build the parent TED. In theory, the parent PCE could listen to the routing protocols in the child domains, but this would violate the confidentiality and scaling issues that may be responsible for the partition of the network into domains. So it is much more likely that a suitable solution will involve specific communication from an entity in the child domain (such as the child PCE) to convey the necessary information. As already mentioned, the "necessary information" relates to how the child domains are inter-connected. The topology and available resources within the child domain do not need to be communicated to the parent PCE: doing so would violate the PCE architecture. Mechanisms for reporting this information are described in the examples in Section 4.6 in abstract terms as "a child PCE reports its neighbor domain connectivity to its parent PCE"; the specifics of a solution are out of scope of this document, but the requirements are indicated in Section 4.8. See Section 6 for a brief discussion of BGP-TE.

In models such as ASON (see Section 5.2), it is possible to consider a separate instance of an IGP running within the parent domain where the participating protocol speakers are the nodes directly present in that domain and the PCEs (Routing Controllers) responsible for each of the child domains.

4.5 Determination of Destination Domain

The PCC asking for an inter-domain path computation is aware of the identity of the destination node by definition. If it knows the egress domain it can supply this information as part of the path computation request. However, if it does not know the egress domain this information must be known by the child PCE or known/determined by the parent PCE.

In some specialist topologies the parent PCE could determine the destination domain based on the destination address, for example from configuration. However, this is not appropriate for many multi-domain addressing scenarios. In IP-based multi-domain networks the parent PCE may be able to determine the destination domain by participating in inter-domain routing. Finally, the parent PCE could issue specific requests to the child PCEs to discover if they contain the destination node, but this has scaling implications.

For the purposes of this document, the precise mechanism of the discovery of the destination domain is left out of scope. Suffice to say that for each multi-domain path computation some mechanism will be required to determine the location of the destination.

4.6 Hierarchical PCE Examples

The following example describes the generic hierarchical domain topology. Figure 1 demonstrates four interconnected domains within a fifth, parent domain. Each domain contains a single PCE:

- o Domain 1 is the ingress domain and child PCE 1 is able to compute paths within the domain. Its neighbors are Domain 2 and Domain 4. The domain also contains the source LSR (S) and three egress boundary nodes (BN11, BN12, and BN13).

- o Domain 2 is served by child PCE 2. Its neighbors are Domain 1 and Domain 3. The domain also contains four boundary nodes (BN21, BN22, BN23, and BN24).

- o Domain 3 is the egress domain and is served by child PCE 3. Its neighbors are Domain 2 and Domain 4. The domain also contains the destination LSR (D) and three ingress boundary nodes (BN31, BN32, and BN33).

- o Domain 4 is served by child PCE 4. Its neighbors are Domain 2 and

Domain 3. The domain also contains two boundary nodes (BN41 and BN42).

All of these domains are contained within Domain 5 which is served by the parent PCE (PCE 5).

```
 -----------------------------------------------------------------
|   Domain 5                                                      |
|                              -----                              |
|                             |PCE 5|                             |
|                              -----                              |
|                                                                 |
|    ----------------     ----------------     ----------------   |
|   | Domain 1       |   | Domain 2       |   | Domain 3       |  |
|   |                |   |                |   |                |  |
|   |        -----   |   |        -----   |   |        -----   |  |
|   |       |PCE 1|  |   |       |PCE 2|  |   |       |PCE 3|  |  |
|   |        -----   |   |        -----   |   |        -----   |  |
|   |                |   |                |   |                |  |
|   |            ----|   |----        ----|   |----            |  |
|   |           |BN11+---+BN21|      |BN23+---+BN31|           |  |
|   |   -        ----|   |----        ----|   |----        -   |  |
|   |  |S|           |   |                |   |           |D|  |  |
|   |   -        ----|   |----        ----|   |----        -   |  |
|   |           |BN12+---+BN22|      |BN24+---+BN32|           |  |
|   |            ----|   |----        ----|   |----            |  |
|   |                |   |                |   |                |  |
|   |         ----   |   |                |   |   ----         |  |
|   |        |BN13|  |   |                |   |  |BN33|        |  |
|    -----------+----     ----------------     ----+-----------   |
|                \                                /               |
|                 \       ----------------       /                |
|                  \     |                |     /                 |
|                   \    |----        ----|    /                  |
|                    ----+BN41|      |BN42+----                   |
|                        |----        ----|                       |
|                        |                |                       |
|                        |        -----   |                       |
|                        |       |PCE 4|  |                       |
|                        |        -----   |                       |
|                        |                |                       |
|                        | Domain 4       |                       |
|                         ----------------                        |
|                                                                 |
 -----------------------------------------------------------------
```

Figure 1 : Sample Hierarchical Domain Topology

Figure 2, shows the view of the domain topology as seen by the parent PCE (PCE 5). This view is an abstracted topology; PCE 5 is aware of domain connectivity, but not of the internal topology within each domain.

```
 ----------------------------
| Domain 5                   |
|            ----            |
|           |PCE5|           |
|            ----            |
|                            |
|   ----     ----     ----   |
|  |    |---|    |---|    |  |
|  | D1 |   | D2 |   | D3 |  |
|  |    |---|    |---|    |  |
|   ----     ----     ----   |
|     \      ----      /     |
|      \    |    |    /      |
|       ----| D4 |----       |
|           |    |           |
|            ----            |
|                            |
 ----------------------------
```

Figure 2 : Abstract Domain Topology as Seen by the Parent PCE

4.6.1 Hierarchical PCE Initial Information Exchange

Based on the Figure 1 topology, the following is an illustration of the initial hierarchical PCE information exchange.

1. Child PCE 1, the PCE responsible for Domain 1, is configured with the location of its parent PCE (PCE5).

2. Child PCE 1 establishes contact with its parent PCE. The parent applies policy to ensure that communication with PCE 1 is allowed.

3. Child PCE 1 listens to the IGP in its domain and learns its inter-domain connectivity. That is, it learns about the links BN11-BN21, BN12-BN22, and BN13-BN41.

4. Child PCE 1 reports its neighbor domain connectivity to its parent PCE.

5. Child PCE 1 reports any change in the resource availability on its inter-domain links to its parent PCE.

Each child PCE performs steps 1 through 5 so that the parent PCE can create a domain topology view as shown in Figure 2.

4.6.2 Hierarchical PCE End-to-End Path Computation Procedure

The procedure below is an example of a source PCC requesting an end-to-end path in a multi-domain environment. The topology is

represented in Figure 1. It is assumed that the each child PCE has connected to its parent PCE and exchanged the initial information required for the parent PCE to create its domain topology view as described in Section 5.6.1.

1. The source PCC (the ingress LSR in our example), sends a request to the PCE responsible for its domain (PCE 1) for a path to the destination LSR (D).

2. PCE 1 determines the destination is not in domain 1.

3. PCE 1 sends a computation request to its parent PCE (PCE 5).

4. The parent PCE determines that the destination is in Domain 3. (See Section 5.5).

5. PCE 5 determines the likely domain paths according to the domain interconnectivity and TE capabilities between the domains. For example, assuming that the link BN12-BN22 is not suitable for the requested path, three domain paths are determined:

   S-BN11-BN21-D2-BN23-BN31-D
   S-BN11-BN21-D2-BN24-BN32-D
   S-BN13-BN41-D4-BN42-BN33-D

6. PCE 5 sends edge-to-edge path computation requests to PCE 2 which is responsible for Domain 2 (i.e., BN21-to-BN23 and BN21-to-BN24), and to PCE 4 for Domain 4 (i.e., BN41-to-BN42).

7. PCE 5 sends source-to-edge path computation requests to PCE 1 which is responsible for Domain 1 (i.e., S-to-BN11 and S-to-BN13).

8. PCE 5 sends edge-to-egress path computation requests to PCE3 which is responsible for Domain 3 (i.e., BN31-to-D, BN32-to-D, and BN33-to-D).

9. PCE 5 correlates all the computation responses from each child PCE, adds in the information about the inter-domain links, and applies any requested and locally configured policies.

10. PCE 5 then selects the optimal end-to-end multi-domain path that meets the policies and objective functions, and supplies the resulting path to PCE 1.

11. PCE 1 forwards the path to the PCC (the ingress LSR).

Note that there is no requirement for steps 6, 7, and 8 to be carried out in parallel or in series. Indeed, they could be overlapped with step 5. This is an implementation issue.

4.7 Hierarchical PCE Error Handling

In the event that a child PCE in a domain cannot find a suitable path to the egress. The child PCE should return the relevant error to notify the parent PCE. Depending on the error response the parent PCE can elect to:

- o Cancel the request and send the relevant response back to the initial child PCE that requested an end-to-end path;
- o Relax some of the constraints associated with the initial path request;
- o Select another candidate domain and send the path request to the child PCE responsible for the domain.

If the parent PCE does not receive a response from a child PCE within an allotted time period. The parent PCE can elect to:

- o Cancel the request and send the relevant response back to the initial child PCE that requested an end-to-end path;
- o Send the path request to another child PCE in the same domain, if a secondary child PCE exists;
- o Select another candidate domain and send the path request to the child PCE responsible for that domain.

The parent PCE may also want to prune any unresponsive child PCE domain paths from the candidate set.

4.8 Requirements for Hierarchical PCEP Protocol Extensions

This section lists the high-level requirements for extensions to the PCEP to support the hierarchical PCE model. It is provided to offer guidance to PCEP protocol developers in designing a solution suitable for use in a hierarchical PCE framework.

4.8.1 PCEP Request Qualifiers

PCEP request (PCReq) messages are used by a PCC or a PCE to make a computation request or enquiry to a PCE. The requests are qualified so that the PCE knows what type of action is required.

Support of the hierarchical PCE architecture will introduce two new qualifications as follows:

- o It must be possible for a child PCE to indicate that the response it receives from the parent PCE should consist of a domain sequence only (i.e., not a fully-specified end-to-end path). This allows the child PCE to initiate per-domain or backward recursive path computation.

- A parent PCE may need to be able to ask a child PCE whether a particular node address (the destination of an end-to-end path) is present in the domain that the child PCE serves.

In PCEP, such request qualifications are carried as bit-flags in the RP object within the PCReq message.

4.8.2 Indication of Hierarchical PCE Capability

Although parent/child PCE relationships are likely configured, it will assist network operations if the parent PCE is able to indicate to the child that it really is capable of acting as a parent PCE. This will help to trap misconfigurations.

In PCEP, such capabilities are carried in the Open Object within the Open message.

4.8.3 Intention to Utilize Parent PCE Capabilities

A PCE that is capable of acting as a parent PCE might not be configured or willing to act as the parent for a specific child PCE. This fact could be determined when the child sends a PCReq that requires parental activity (such as querying other child PCEs), and could result in a negative response in a PCEP Error (PCErr) message.

However, the expense of a poorly targeted PCReq can be avoided if the child PCE indicates that it might wish to use the parent-capable as a parent (for example, on the Open message), and if the parent-capable determines at that time whether it is willing to act as a parent to this child.

4.8.4 Communication of Domain Connectivity Information

Section 4.4 describes how the parent PCE needs a parent TED and indicates that the information might be supplied from the child PCEs in each domain. This requires a mechanism whereby information about inter-domain links can be supplied by a child PCE to a parent PCE, for example on a PCEP Notify (PCNtf) message.

The information that would be exchanged includes:

- o Identifier of advertising child PCE
- o Identifier of PCE's domain
- o Identifier of the link
- o TE properties of the link (metrics, bandwidth)
- o Other properties of the link (technology-specific)
- o Identifier of link end-points
- o Identifier of adjacent domain

It may be desirable for this information to be periodically updated, for example, when available bandwidth changes. In this case, the parent PCE might be given the ability to configure thresholds in the child PCE to prevent flapping of information.

4.8.5 Domain Identifiers

Domain identifiers are already present in PCEP to allow a PCE to indicate which domains it serves, and to allow the representation of domains as abstract nodes in paths. The wider use of domains in the context of this work on hierarchical PCE will require that domains can be identified in more places within objects in PCEP messages. This should pose no problems.

However, more attention may need to be applied to the precision of domain identifier definitions to ensure that it is always possible to unambiguously identify a domain from its identifier. This work will be necessary in configuration, and also in protocol specifications (for example, an OSPF area identifier is sufficient within an Autonomous System, but becomes ambiguous in a path that crosses multiple Autonomous Systems).

5. Hierarchical PCE Applicability

As per [RFC4655], PCE can inherently support inter-domain path computation for any definition of a domain as set out in Section 1.2 of this document.

Hierarchical PCE can be applied to inter-domain environments, including Antonymous Systems and IGP areas. The hierarchical PCE procedures make no distinction between, Antonymous Systems and IGP area applications, although it should be noted that the TED maintained by a parent PCE must be able to support the concept of child domains connected by inter-domain links or directly connected at boundary nodes (see Section 3).

This section sets out the applicability of hierarchical PCE to three environments:

o MPLS traffic engineering across multiple Autonomous Systems
o MPLS traffic engineering across multiple IGP areas
o GMPLS traffic engineering in the ASON architecture

5.1 Antonymous Systems and Areas

Networks are comprised of domains. A domain can be considered to be a collection of network elements within an AS or area that has a common sphere of address management or path computational responsibility.

As networks increase in size and complexity it may be required to introduce scaling methods to reduce the amount information flooded within the network and make the network more manageable. An IGP

hierarchy is designed to improve IGP scalability by dividing the IGP domain into areas and limiting the flooding scope of topology information to within area boundaries. This restricts a router's visibility to information about links and other routers within the single area. If a router needs to compute a route to destination located in another area, a method is required to compute a path across the area boundary.

When an LSR within an AS or area needs to compute a path across an area or AS boundary it must also use an inter-AS computation technique. Hierarchical PCE is equally applicable to computing inter-area and inter-AS MPLS and GMPLS paths across domain boundaries.

5.2 ASON Architecture

The International Telecommunications Union (ITU) defines the ASON architecture in [G-8080]. [G-7715] defines the routing architecture for ASON and introduces a hierarchical architecture. In this architecture, the Routing Areas (RAs) have a hierarchical relationship between different routing levels, which means a parent (or higher level) RA can contain multiple child RAs. The interconnectivity of the lower RAs is visible to the higher level RA. Note that the RA hierarchy can be recursive.

In the ASON framework, a path computation request is termed a Route Query. This query is executed before signaling is used to establish an LSP termed a Switched Connection (SC) or a Soft Permanent Connection (SPC). [G-7715-2] defines the requirements and architecture for the functions performed by Routing Controllers (RC) during the operation of remote route queries - an RC is synonymous with a PCE. For an end-to-end connection, the route may be computed by a single RC or multiple RCs in a collaborative manner (i.e., RC federations). In the case of RC federations, [G-7715-2] describes three styles during remote route query operation:

o Step-by-step remote path computation
o Hierarchical remote path computation
o A combination of the above.

In a hierarchical ASON routing environment, a child RC may communicate with its parent RC (at the next higher level of the ASON routing hierarchy) to request the computation of an end-to-end path across several RAs. It does this using a route query message (known as the abstract message RI_QUERY). The corresponding parent RC may communicate with other child RCs that belong to other child RAs at the next lower hierarchical level. Thus, a parent RC can act as either a Route Query Requester or Route Query Responder.

It can be seen that the hierarchical PCE architecture fits the hierarchical ASON routing architecture well. It can be used to provide paths across subnetworks, and to determine end-to-end paths in networks constructed from multiple subnetworks or RAs.

When hierarchical PCE is applied to implement hierarchical remote path computation in [G-7715-2], it is very important for operators to understand the different terminology and implicit consistency between hierarchical PCE and [G-7715-2].

5.2.1 Implicit Consistency Between Hierarchical PCE and G.7715.2

This section highlights the correspondence between features of the hierarchical PCE architecture and the ASON routing architecture.

(1) RC (Routing Controller) and PCE (Path Computation Element)

[G-8080] describes the Routing Controller component as an abstract entity, which is responsible for responding to requests for path (route) information and topology information. It can be implemented as a single entity, or as a distributed set of entities that make up a cooperative federation.

[RFC4655] describes PCE (Path Computation Element) is 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.

Therefore, in the ASON architecture, a PCE can be regarded as a realizations of the RC.

(2) Route Query Requester/Route Query Responder and PCC/PCE

[G-7715-2] describes the Route Query Requester as a Connection Controller or Routing Controller that sends a route query message to a Routing Controller requesting one or more paths that satisfy a set of routing constraints. The Route Query Responder is a Routing Controller that performs path computation upon receipt of a route query message from a Route Query Requester, sending a response back at the end of the path computation.

In the context of ASON, a Signaling Controller initiates and processes signaling messages and is closely coupled to a Signaling Protocol Speaker. A Routing Controller makes routing decisions and is usually coupled to configuration entities and/or a Routing Protocol Speaker.

It can be seen that a PCC corresponds to a Route Query Requester, and a PCE corresponds to a Route Query Responder. A PCE/RC can

also act as a Route Query Requester sending requests to another Route Query Responder.

The PCEP path computation request (PCReq) and path computation reply (PCRep) messages between PCC and PCE correspond to the RI_QUERY and RI_UPDATE messages in [G-7715-2].

(3) Routing Area Hierarchy and Hierarchical Domain

The ASON routing hierarchy model is shown in Figure 6 of [G-7715] through an example that illustrates routing area levels. If the hierarchical remote path computation mechanism of [G-7715-2] is applied in this scenario, each routing area should have at least one RC for route query function and there is a parent RC for the child RCs in each routing area.

According to [G-8080], the parent RC has visibility of the structure of the lower level, so it knows the interconnectivity of the RAs in the lower level. Each child RC can compute edge-to-edge paths across its own child RA.

Thus, an RA corresponds to a domain in the PCE architecture, and the hierarchical relationship between RAs corresponds to the hierarchical relationship between domains in the hierarchical PCE architecture. Furthermore, a parent PCE in a parent domain can be regarded as parent RC in a higher routing level, and a child PCE in a child domain can be regarded as child RC in a lower routing level.

5.2.2 Benefits of Hierarchical PCEs in ASON

RCs in an ASON environment can use the hierarchical PCE model to fully match the ASON hierarchical routing model, so the hierarchical PCE mechanisms can be applied to fully satisfy the architecture and requirements of [G-7715-2] without any changes. If the hierarchical PCE mechanism is applied in ASON, it can be used to determine end-to-end optimized paths across sub-networks and RAs before initiating signaling to create the connection. It can also improve the efficiency of connection setup to avoid crankback.

6. A Note on BGP-TE

The concept of exchange of TE information between Autonomous Systems (ASes) is discussed in [BGP-TE]. The information exchanged in this way could be the full TE information from the AS, an aggregation of that information, or a representation of the potential connectivity across the AS. Furthermore, that information could be updated frequently (for example, for every new LSP that is set up across the

AS) or only at threshold-crossing events.

There are a number of discussion points associated with the use of [BGP-TE] concerning the volume of information, the rate of churn of information, the confidentiality of information, the accuracy of aggregated or potential-connectivity information, and the processing required to generate aggregated information. The PCE architecture and the architecture enabled by [BGP-TE] make different assumptions about the operational objectives of the networks, and this document does not attempt to make one of the approaches "right" and the other "wrong". Instead, this work assumes that a decision has been made to utilize the PCE architecture.

6.1 Use of BGP for TED Synchronization

Indeed, [BGP-TE] may have some uses within the PCE model. For example, [BGP-TE] could be used as a "northbound" TE advertisement such that a PCE does not need to listen to an IGP in its domain, but has its TED populated by messages received (for example) from a Route Reflector. Furthermore, the inter-domain connectivity and connectivity capabilities that is required information for a parent PCE could be obtained as a filtered subset of the information available in [BGP-TE]. This scenario is discussed further in [PCE-AREA-AS].

7. Management Considerations

General PCE management considerations are discussed in [RFC4655]. In the case of the hierarchical PCE architecture, there are additional management considerations.

The administrative entity responsible for the management of the parent PCEs must be determined. In the case of multi-domains (e.g., IGP areas or multiple ASes) within a single service provider network, the management responsibility for the parent PCE would most likely be handled by the service provider. In the case of multiple ASes within different service provider networks, it may be necessary for a third-party to manage the parent PCEs according to commercial and policy agreements from each of the participating service providers.

7.1 Control of Function and Policy

7.1.1 Child PCE

Support of the hierarchical procedure will be controlled by the management organization responsible for each child PCE. A child PCE must be configured with the address of its parent PCE in order for it to interact with its parent PCE. The child PCE must also be authorized to peer with the parent PCE.

7.1.2 Parent PCE

The parent PCE must only accept path computation requests from authorized child PCEs. If a parent PCE receives requests from an unauthorized child PCE, the request should be dropped.

This means that a parent PCE must be configured with the identities and security credentials of all of its child PCEs, or there must be some form of shared secret that allows an unknown child PCE to be authorized by the parent PCE.

7.1.3 Policy Control

It may be necessary to maintain a policy module on the parent PCE [RFC5394]. This would allow the parent PCE to apply commercially relevant constraints such as SLAs, security, peering preferences, and monetary costs.

It may also be necessary for the parent PCE to limit end-to-end path selection by including or excluding specific domains based on commercial relationships, security implications, and reliability.

7.2 Information and Data Models

A PCEP MIB module is defined in [PCEP-MIB] that describes managed objects for modeling of PCEP communication. An additional PCEP MIB will be required to report parent PCE and child PCE information, including:

o Parent PCE configuration and status,

o Child PCE configuration and information,

o Notifications to indicate session changes between parent PCEs and child PCEs.

o Notification of parent PCE TED updates and changes.

7.3 Liveness Detection and Monitoring

The hierarchical procedure requires interaction with multiple PCEs. Once a child PCE requests an end-to-end path, a sequence of events occurs that requires interaction between the parent PCE and each child PCE. If a child PCE is not operational, and an alternate transit domain is not available, then a failure must be reported.

7.4 Verifying Correct Operation

Verifying the correct operation of a parent PCE can be performed by

monitoring a set of parameters. The parent PCE implementation should provide the following parameters monitored by the parent PCE:

- o Number of child PCE requests.
- o Number of successful hierarchical PCE procedures completions on a per-PCE-peer basis.
- o Number of hierarchical PCE procedure completion failures on a per-PCE-peer basis.
- o Number of hierarchical PCE procedure requests from unauthorized child PCEs.

7.5. Impact on Network Operation

The hierarchical PCE procedure is a multiple-PCE path computation scheme. Subsequent requests to and from the child and parent PCEs do not differ from other path computation requests and should not have any significant impact on network operations.

8. Security Considerations

The hierarchical PCE procedure relies on PCEP and inherits the security requirements defined [RFC5440]. As noted in Section 7, there is a security relationship between child and parent PCEs. This relationship, like any PCEP relationship assumes pre-configuration of identities, authority, and keys, or can operate through any key distribution mechanism outside the scope of PCEP. As PCEP operates over TCP, it may make use of any TCP security mechanism.

The hierarchical PCE architecture makes use of PCE policy [RFC5394] and the security aspects of the PCE communication protocol documented in [RFC5440]. It is expected that the parent PCE will require all child PCEs to use full security when communicating with the parent and that security will be maintained by not supporting the discovery by a parent of child PCEs.

PCE operation also relies on information used to build the TED. Attacks on a PCE system may be achieved by falsifying or impeding this flow of information. The child PCE TEDs are constructed as described in [RFC4655] and are unchanged in this document: if the PCE listens to the IGP for this information, then normal IGP security measures may be applied, and it should be noted that an IGP routing system is generally assumed to be a trusted domain such that router subversion is not a risk. The parent PCE TED is constructed as described in this document and may involve:

- multiple parent-child relationships using PCEP (as already described)

- the parent PCE listening to child domain IGPs (with the same security features as a child PCE listening to its IGP)

- an external mechanism (such as [BGP-TE]) which will need to be authorized and secured.

Any multi-domain operation necessarily involves the exchange of information across domain boundaries. This is bound to represent a significant security and confidentiality risk especially when the child domains are controlled by different commercial concerns. PCEP allows individual PCEs to maintain confidentiality of their domain path information using Path Keys [RFC5520], and the hierarchical PCE architecture is specifically designed to enable as much isolation of domain topology and capabilities information as is possible.

Further considerations of the security issues related to inter-AS path computation see [RFC5376].

9. IANA Considerations

This document makes no requests for IANA action.

10. Acknowledgements

The authors would like to thank David Amzallag, Oscar Gonzalez de Dios, Franz Rambach, Ramon Casellas, Olivier Dugeon, Filippo Cugini, Dhruv Dhody and Julien Meuric for their comments and suggestions.

11. References

11.1 Normative References

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

[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.

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

[RFC5440] Ayyangar, A., Farrel, A., Oki, E., Atlas, A., Dolganow, A., Ikejiri, Y., Kumaki, K., Vasseur, J., and J. Roux, "Path Computation Element (PCE) Communication Protocol (PCEP)", RFC 5440, March 2009.

[RFC5441] Vasseur, J.P., Ed., "A Backward Recursive PCE-based Computation (BRPC) procedure to compute shortest inter-domain Traffic Engineering Label Switched Paths", RFC 5441, April 2009.

[RFC5520] Brandford, R., Vasseur J.P., and Farrel A., "Preserving Topology Confidentiality in Inter-Domain Path Computation Using a Key-Based Mechanism RFC5520, April 2009.

11.2. Informative References

[RFC4105] Le Roux, JL., Vasseur, J., Boyle, J., "Requirements for Inter-Area MPLS Traffic Engineering", RFC 4105, June 2005.

[RFC4216] Zhang, R., and Vasseur, J., "MPLS Inter-Autonomous System (AS) Traffic Engineering (TE) Requirements", RFC 4216, November 2005.

[RFC4726] Farrel, A., Vasseur, J., Ayyangar, A., "A Framework for Inter-Domain Multiprotocol Label Switching Traffic Engineering", RFC 4726, November 2006.

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

[RFC5316] Chen, M., Zhang, R., Duan, X., "ISIS Extensions in Support of Inter-Autonomous System (AS) MPLS and GMPLS Traffic Engineering", RFC 5316, December 2008.

[RFC5376] Bitar, N., et al., "Inter-AS Requirements for the Path Computation Element Communication Protocol (PCECP)", RFC 5376, November 2008.

[RFC5392] Chen, M., Zhang, R., Duan, X., "OSPF Extensions in Support of Inter-Autonomous System (AS) MPLS and GMPLS Traffic Engineering", RFC 5392, January 2009.

[RFC5541] Le Roux, J., Vasseur, J., Lee, Y., "Encoding of Objective Functions in the Path Computation Element Communication Protocol (PCEP)", RFC5541, December 2008.

[G-8080] ITU-T Recommendation G.8080/Y.1304, Architecture for the automatically switched optical network (ASON).

[G-7715] ITU-T Recommendation G.7715 (2002), Architecture and Requirements for the Automatically Switched Optical Network (ASON).

[G-7715-2] ITU-T Recommendation G.7715.2 (2007), ASON routing architecture and requirements for remote route query.

[BGP-TE] Gredler, H., Medved, J, Farrel, A. Previdi, S., "North-Bound Distribution of Link-State and TE Information using BGP", draft-gredler-idr-ls-distribution, work in progress.

[PCE-AREA-AS] King, D., Meuric, J., Dugeon, O., Zhao, Q., Gonzalez de Dios, O., "Applicability of the Path Computation Element to Inter-Area and Inter-AS MPLS and GMPLS Traffic Engineering", draft-ietf-pce-inter-area-as-applicability, work in progress.

[PCEP-MIB] Stephan, E., Koushik, K., Zhao, Q., King, D., "PCE Communication Protocol (PCEP) Management Information Base", work in progress.

12. Authors' Addresses

Daniel King
Old Dog Consulting
UK

Email: daniel@olddog.co.uk

Adrian Farrel
Old Dog Consulting
UK

Email: adrian@olddog.co.uk

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

Email: qzhao@huawei.com

Fatai Zhang
Huawei Technologies
F3-5-B R&D Center, Huawei Base
Bantian, Longgang District
Shenzhen 518129 P.R.China

Email: zhangfatai@huawei.com