Network Working Group
Request for Comments: 4655
Category: Informational

A. Farrel
Old Dog Consulting
J.-P. Vasseur
Cisco Systems, Inc.
J. Ash
AT&T

August 2006

A Path Computation Element (PCE)-Based Architecture

Status of This Memo

Copyright Notice

Abstract

Constraint-based path computation is a fundamental building block for traffic engineering systems such as Multiprotocol Label Switching (MPLS) and Generalized Multiprotocol Label Switching (GMPLS) networks. Path computation in large, multi-domain, multi-region, or multi-layer networks is complex and may require special computational components and cooperation between the different network domains.

This document specifies the architecture for a Path Computation Element (PCE)-based model to address this problem space. This document does not attempt to provide a detailed description of all the architectural components, but rather it describes a set of building blocks for the PCE architecture from which solutions may be constructed.

Table of Contents

1. Introduction

Constraint-based path computation is a fundamental building block for traffic engineering in MPLS [RFC3209] and GMPLS [RFC3473] networks. [RFC2702] describes requirements for traffic engineering in MPLS networks, while [RFC4105] and [RFC4216] describe traffic engineering requirements in inter-area and inter-AS environments, respectively.

Path computation in large, multi-domain networks is complex and may require special computational components and cooperation between the elements in different domains. This document specifies the architecture for a Path Computation Element (PCE)-based model to address this problem space.

This document describes a set of building blocks for the PCE architecture from which solutions may be constructed. For example, it discusses PCE-based implementations including composite, external, and multiple PCE path computation. Furthermore, it discusses architectural considerations including centralized computation, distributed computation, synchronization, PCE discovery and load balancing, detection of PCE liveness, communication between Path Computation Clients (PCCs) and the PCE (PCC-PCE communication) and PCE-PCE communication, Traffic Engineering Database (TED) synchronization, stateful and stateless PCEs, monitoring, policy and confidentiality, and evaluation metrics.

The model of the Internet is to distribute network functionality (e.g., routing) within the network. PCE functionality is not intended to contradict this model and can be used to match the model exactly, for example, when the PCE functionality coexists with each Label Switching Router (LSR) in the network. PCE is also able to augment functionality in the network where the Internet model cannot supply adequate solutions, for example, where traffic engineering information is not exchanged between network domains.

2. Terminology

CSPF: Constraint-based Shortest Path First.

LER: Label Edge Router.

LSDB: Link State Database.

LSP: Label Switched Path.

LSR: Label Switching Router.

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

PCE: Path Computation Element. An entity (component, application, or network node) that is capable of computing a network path or route based on a network graph and applying computational constraints (see further description in Section 3).

TED: Traffic Engineering Database, which contains the topology and resource information of the domain. The TED may be fed by Interior Gateway Protocol (IGP) extensions or potentially by other means.

TE LSP: Traffic Engineering MPLS Label Switched Path.

3. Definitions

A Path Computation Element (PCE) is an entity that is capable of computing a network path or route based on a network graph, and of applying computational constraints during the computation. The PCE entity is an application that can be located within a network node or component, on an out-of-network server, etc. For example, a PCE would be able to compute the path of a TE LSP by operating on the TED and considering bandwidth and other constraints applicable to the TE LSP service request.

A domain is any collection of network elements within a common sphere of address management or path computation responsibility. Examples of domains include IGP areas, Autonomous Systems (ASes), and multiple ASes within a Service Provider network. Domains of path computation responsibility may also exist as sub-domains of areas or ASes.

In order to fully characterize a PCE and clarify these definitions, the following important considerations must also be examined:

1) Path computation is applicable in intra-domain, inter-domain, and inter-layer contexts.

   a. Inter-domain path computation may involve the association of topology, routing, and policy information from multiple domains from which relationships may be deduced in order to help in performing path computation.

   b. Inter-layer path computation refers to the use of PCE where multiple layers are involved and when the objective is to perform path computation at one or multiple layers while taking into account topology and resource information at these layers.

Overlapping domains are not within the scope of this document. In the inter-domain case, the domains may belong to a single or to multiple Service Providers.

2) a. In "single PCE path computation", a single PCE is used to compute a given path in a domain. There may be multiple PCEs in a domain, but only one PCE per domain is involved in any single path computation.

   b. In "multiple PCE path computation", multiple PCEs are used to compute a given path in a domain.

3) a. "Centralized computation model" refers to a model whereby all paths in a domain are computed by a single, centralized PCE.

   b. Conversely, "distributed computation model" refers to the computation of paths in a domain being shared among multiple PCEs.

   Paths that span multiple domains may be computed using the distributed model with one or more PCEs responsible for each domain, or the centralized model by defining a domain that encompasses all the other domains.

   From these definitions, a centralized computation model inherently uses single PCE path computation. However, a distributed computation model could use either single PCE path computation or multiple PCE path computations. There would be no such thing as a centralized model that uses multiple PCEs.

4) The PCE may or may not be located at the head-end of the path. For example, a conventional intra-domain solution is to have path computation performed by the head-end LSR of an MPLS TE LSP; in this case, the head-end LSR contains a PCE. But solutions also exist where other nodes on the path must contribute to the path computation (for example, loose hops), making them PCEs in their own right. At the same time, the path computation may be made by some other PCE physically distinct from the computed path.

5) The path computed by the PCE may be an "explicit path" (that is, the full explicit path from start to destination, made of a list of strict hops) or a "strict/loose path" (that is, a mix of strict and loose hops comprising at least one loose hop representing the destination), where a hop may be an abstract node such as an AS.

6) A PCE-based path computation model does not mean to be exclusive and can be used in conjunction with other path computation models. For instance, the path of an inter-AS TE LSP may be computed using

a PCE-based path computation model in some ASes, whereas the set of traversed ASes may be specified by other means (not determined by a PCE). Furthermore, different path computation models may be used for different TE LSPs.

7) This document does not make any assumptions about the nature or implementation of a PCE. A PCE could be implemented on a router, an LSR, a dedicated network server, etc. Moreover, the PCE function is orthogonal to the forwarding capability of the node on which it is implemented.

4. Motivation for a PCE-Based Architecture

Several motivations for a PCE-based architecture (described in Section 5) are listed below. This list is not meant to be exhaustive and is provided for the sake of illustration.

It should be highlighted that the aim of this section is to provide some application examples for which a PCE-based path may be suitable: this also clearly states that such a model does not aim to replace existing path computation models but would apply to specific existing or future situations.

As can be seen from these examples, PCE does not replace the existing Internet model where intelligence is distributed within the network. Instead, it builds on this model and makes use of distributed centers of information or computational ability. PCE should not, therefore, necessarily be seen as a centralized, "all-seeing oracle in the sky", but as the cooperative operation of distributed functionality used to address specific challenges such as the computation of a shortest inter-domain constrained path.

4.1. CPU-Intensive Path Computation

There are many situations where the computation of a path may be highly CPU-intensive; examples of CPU-intensive path computations include the resolution of problems such as:

- Placing a set of TE LSPs within a domain so as to optimize an objective function (for example, minimization of the maximum link utilization)

- Multi-criteria path computation (for example, delay and link utilization, inclusion of switching capabilities, adaptation features, encoding types and optical constraints within a GMPLS optical network)

- Computation of minimal cost Point to Multipoint trees (Steiner trees)

In these situations, it may not be possible or desirable for some routers to perform path computation because of the constraints on their CPUs, in which case the path computations may be off-loaded to some other PCE(s) that may, themselves, be routers or may be dedicated PCE servers.

4.2. Partial Visibility

There are several scenarios where the node responsible for path computation has limited visibility of the network topology to the destination. This limitation may occur, for instance, when an ingress router attempts to establish a TE LSP to a destination that lies in a separate domain, since TE information is not exchanged across the domain boundaries. In such cases, it is possible to use loose routes to establish the TE LSP, relying on routers at the domain borders to establish the next piece of the path. However, it is not possible to guarantee that the optimal (shortest) path will be used, or even that a viable path will be discovered except, possibly, through repeated trial and error using crankback or other signaling extensions.

This problem of inter-domain path computation may most probably be addressed through distributed computation with cooperation among PCEs within each of the domains, and potentially using crankback between the domains to dynamically resolve provisioning issues. Alternatively, a central "all-seeing" PCE that has access to the complete set of topology information may be used, but in this case there are challenges of scalability (both the size of the TED and the responsiveness of a single PCE handling requests for many domains) and of preservation of confidentiality when the domains belong to different Service Providers.

Note that the issues described here can be further highlighted in the context of TE LSP reoptimization, or the establishment of multiple diverse TE LSPs for protection or load sharing.

4.3. Absence of the TED or Use of Non-TE-Enabled IGP

The traffic engineering database (TED) may be a large drain on the resources of a network node (such as an edge router or LER). Maintaining the TED may require a lot of memory and may require non-negligible CPU activity. The use of a distinct PCE may be appropriate in such circumstances, and a separate node can be used to establish and maintain the TED, and to make it available for path computation.

The IGPs run within some networks are not sufficient to build a full TED. For example, a network may run OSPF/IS-IS without the OSPF-TE/ISIS-TE extensions, or some routers in the network may not support the TE extensions. In these cases, in order to successfully compute paths through the network, the TED must be constructed or supplemented through configuration action and updated as network resources are reserved or released. Such a TED could be distributed to the routers that need to perform path computation or held centrally (on a distinct node that supports PCE) for centralized computation.

4.4. Node Outside the Routing Domain

An LER might not be part of the routing domain for administrative reasons (for example, a customer-edge (CE) router connected to the provider-edge (PE) router in the context of MPLS VPN [RFC4364] and for which it is desired to provide a CE to CE TE LSP path).

This scenario suggests a solution that does not involve doing computation on the ingress (TE LSP head-end, CE) router, and that does not rely on the configuration of static loose hops. In this case, optimal shortest paths cannot be guaranteed. A solution that a distinct PCE can help here. Note that the PCE in this case may, itself, provide a path that includes loose hops.

4.5. Network Element Lacks Control Plane or Routing Capability

It is common in legacy optical networks for the network elements not to have a control plane or routing capability. Such network elements only have a data plane and a management plane, and all cross-connections are made from the management plane. It is desirable in this case to run the path computation on the PCE, and to send the cross-connection commands to each node on the computed path. That is, the PCC would be an element of the management plane, perhaps residing in the Network Management System (NMS) or Operations Support System (OSS).

This scenario is important for Automatically Switched Optical Network (ASON)-capable networks and may also be used for interworking between GMPLS-capable and GMPLS-incapable networks.

4.6. Backup Path Computation for Bandwidth Protection

A PCE can be used to compute backup paths in the context of fast reroute protection of TE LSPs. In this model, all backup TE LSPs protecting a given facility are computed in a coordinated manner by a PCE. This allows complete bandwidth sharing between backup tunnels protecting independent elements, while avoiding any extensions to TE

LSP signaling. Both centralized and distributed computation models are applicable. In the distributed case each LSR can be a PCE to compute the paths of backup tunnels to protect against the failure of adjacent network links or nodes.

4.7. Multi-layer Networks

A server-layer network of one switching capability may support multiple networks of another (more granular) switching capability. For example, a Time-Division Multiplexing (TDM) network may provide connectivity for client-layer networks such as IP, MPLS, or Layer 2 [MLN].

The server-layer network is unlikely to provide the same connectivity paradigm as the client networks, so bandwidth granularity in the server-layer network may be much coarser than in the client-layer network. Similarly, there is likely to be a management separation between the two networks providing independent address spaces. Furthermore, where multiple client-layer networks make use of the same server-layer network, those client-layer networks may have independent policies, control parameters, address spaces, and routing preferences.

The different client- and server-layer networks may be considered distinct path computation regions within a PCE domain, so the PCE architecture is useful to allow path computation from one client-layer network region, across the server-layer network, to another client-layer network region.

In this case, the PCEs are responsible for resolving address space issues, handling differences in policy and control parameters, and coordinating resources between the networks. Note that, because of the differences in bandwidth granularity, connectivity across the server-layer network may be provided through virtual TE links or Forwarding Adjacencies: the PCE may offer a point of control responsible for the decision to provision new TE links or Forwarding Adjacencies across the server-layer network.

4.8. Path Selection Policy

A PCE may have a local policy that impacts path computation and selection in response to a path computation request. Such policy may act on information provided by the requesting PCC. The result of applying such policy includes, for example, rejection of the path computation request, or provision of a path that does not meet all of the requested constraints. Further, the policy may support

administratively configured paths, or selection among transit providers. Inclusion of policy within PCE may simplify the application of policy within the path computation/selection process.

Similarly, a PCC may apply local policy to the selection of a PCE to compute a specific path, and to the constraints that are requested.

In a PCE context, the policy may be sensitive to the type of path that is being computed. For example, a different set of policies may be applied for an intra-area or single-layer path than would be provided for an inter-area or multi-layer path.

Note that synchronization of policy between PCEs or between PCCs and PCEs may be necessary. Such issues are outside the scope of the PCE architecture, but within scope for the PCE policy framework and application which is described in a separate document.

4.9. Non-Motivations

4.9.1. The Whole Internet

PCE is not considered to be a solution that is applicable to the entire Internet. That is, the applicability of PCE is limited to a set of domains with known relationships. The scale of this limitation is similar to the peering relationships between Service Providers.

4.9.2. Guaranteed TE LSP Establishment

When two or more paths for TE LSPs are computed on the same set of TE link state information, it is possible that the resultant paths will compete for limited resources within the network. This may result in success for only the first TE LSP to be signaled, or it might even mean that no TE LSP can be established.

Batch processing of computation requests, back-off times, computation of alternate paths, and crankback can help to mitigate this sort of problem, and PCE may also improve the chances of successful TE LSP setup. However, a single, centralized PCE is not viewed as a solution that can guarantee TE LSP establishment since the potential for network failures or contention for resources still exists where the centralized TED cannot fully reflect current (i.e., real-time) network state.

5. Overview of the PCE-Based Architecture

This section gives an overview of the architecture of the PCE model. It needs to be read in conjunction with the details provided in the next section to provide a full view of the flexibility of the model.

5.1. Composite PCE Node

Figure 1 below shows the components of a typical composite PCE node (that is, a router that also implements the PCE functionality) that utilizes path computation. The routing protocol is used to exchange TE information from which the TED is constructed. Service requests to provision TE LSPs are received by the node and converted into signaling requests, but this conversion may require path computation that is requested from a PCE. The PCE operates on the TED subject to local policy in order to respond with the requested path.

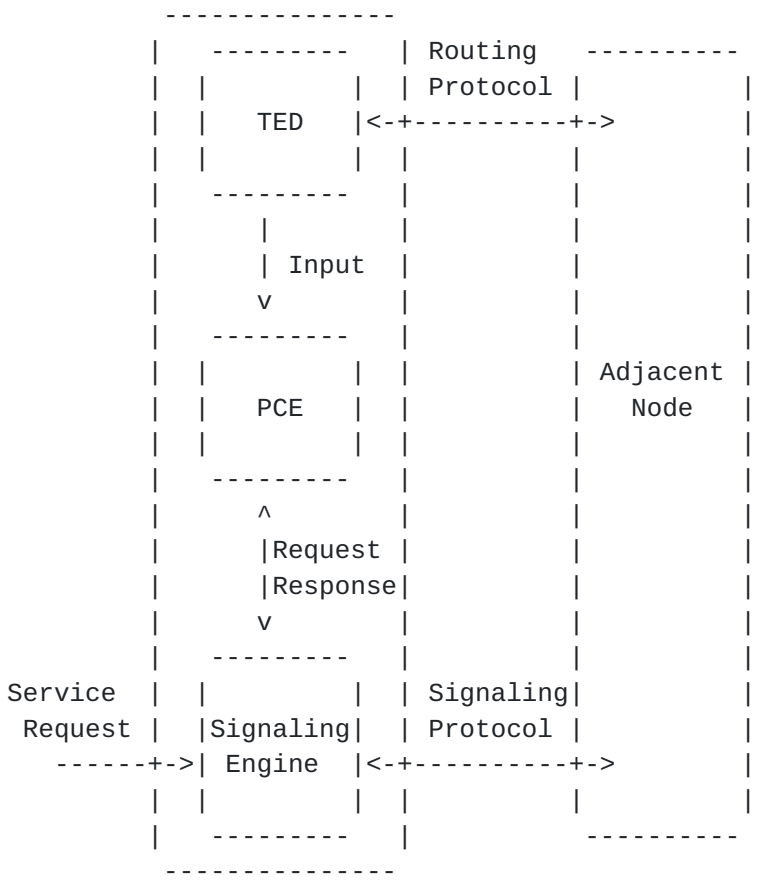

Figure 1. Composite PCE Node

Note that the routing adjacency between the composite PCE node and any other router may be performed by means of direct connectivity or any tunneling mechanism.

5.2. External PCE

Figure 2 shows a PCE that is external to the requesting network element. A service request is received by the head-end node, and before it can initiate signaling to establish the service, it makes a path computation request to the external PCE. The PCE uses the TED subject to local policy as input to the computation and returns a response.

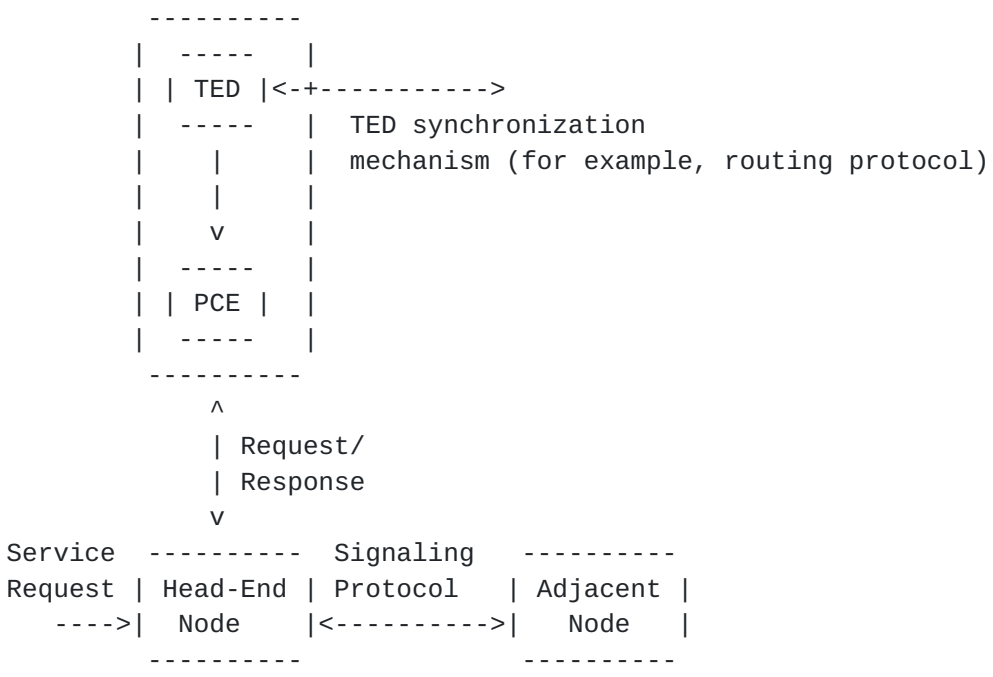

Figure 2. External PCE Node

Note that in this case, the node that supports the PCE function may also be an LSR or router performing forwarding in its own right (i.e., it may be a composite PCE node), but those functions are purely orthogonal to the operation of the function in the instance being considered here.

5.3. Multiple PCE Path Computation

Figure 3 illustrates how multiple PCE path computations may be performed along the path of a signaled service. As in the previous example, the head-end PCC makes a request to an external PCE, but the path that is returned is such that the next network element finds it necessary to perform further computation. This may be the case when the path returned is a partial path that does not reach the intended destination or when the computed path is loose. The downstream network element consults another PCE to establish the next hop(s) in the path. In this case, all policy decisions are made independently at each PCE based on information passed from the PCC.

Note that either or both PCEs in this case could be composite PCE nodes, as in Section 5.1.

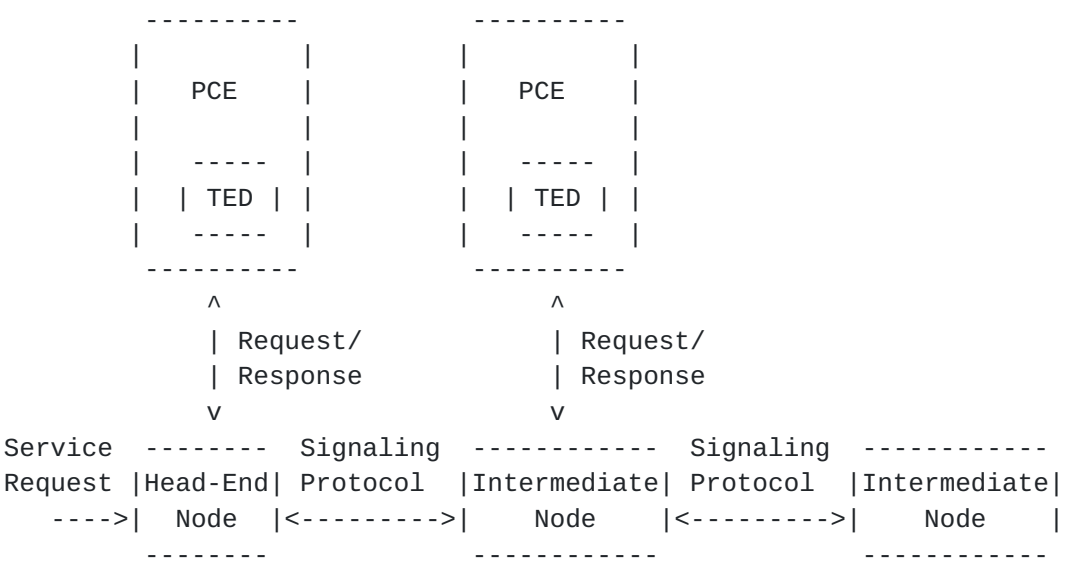

Figure 3. Multiple PCE Path Computation

5.4. Multiple PCE Path Computation with Inter-PCE Communication

The PCE in Section 5.3 was not able to supply a full path for the requested service, and as a result the adjacent node needs to make its own computation request. As illustrated in Figure 4, the same problem may be solved by introducing inter-PCE communication, and cooperation between PCEs so that the PCE consulted by the head-end network node makes a request of another PCE to help with the computation.

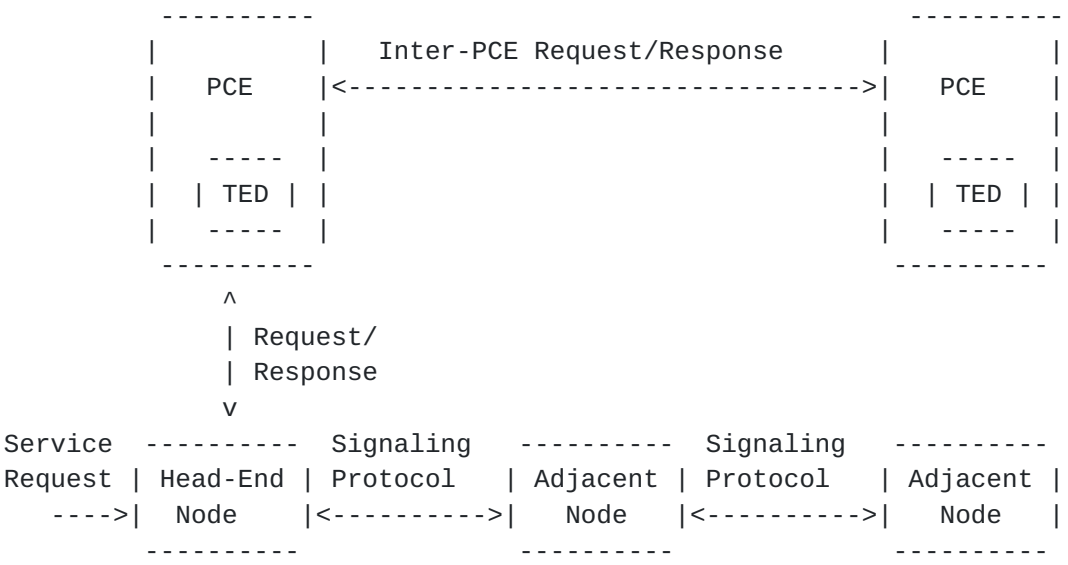

Figure 4. Multiple PCE Path Computation with Inter-PCE Communication

Multiple PCE path computation with inter-PCE communication involves coordination between distinct PCEs such that the result of the computation performed by one PCE depends on path fragment information supplied by other PCEs. This model does not provide a distributed computation algorithm, but it allows distinct PCEs to be responsible for computation of parts (segments) of the path.

PCE-PCE communication is discussed further in Section 6.6.

Note that a PCC might not see the difference between centralized computation and multiple PCE path computation with inter-PCE communication. That is, the PCC network node or component that requests the computation makes a single request and receives a full or partial path in response, but the response is actually achieved through the coordinated, cooperative efforts of more than one PCE.

In this model, all policy decisions may be made independently at each PCE based on computation information passed from the previous PCE. Alternatively, there may be explicit communication of policy information between PCEs.

5.5. Management-Based PCE Usage

It must be observed that the PCC is not necessarily an LSR. For example, in Figure 5 the NMS supplies the head-end LSR with a fully computed explicit path for the TE LSP that it is to establish through signaling. The NMS uses a management plane mechanism to send this request and encodes the data using a representation such as the TE MIB module [RFC3812].

The NMS constructs the explicit path that it supplies to the head-end LSR using information provided by the operator. It consults the PCE, which returns a path for the NMS to use.

Although Figure 5 shows the PCE as remote from the NMS, it could, of course, be collocated with the NMS.

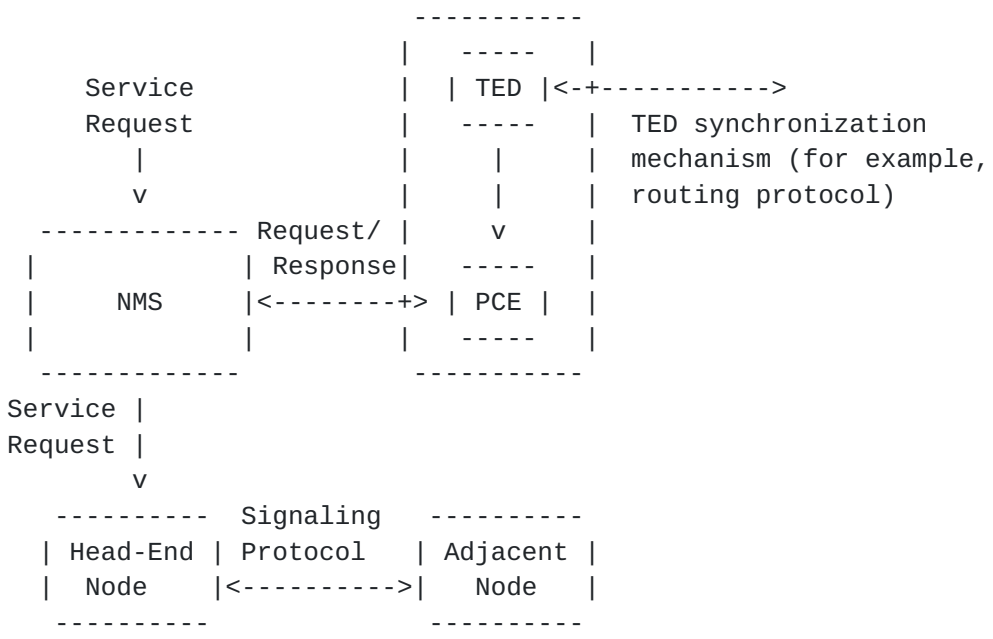

Figure 5. Management-Based PCE Usage

5.6. Areas for Standardization

The following areas require standardization within the PCE architecture.

- communication between PCCs and PCEs, and between cooperating PCEs, including the communication of policy-related information

- requirements for extending existing routing and signaling protocols in support of PCE discovery and signaling of inter-domain paths

- definition of metrics to evaluate path quality, scalability, responsiveness, robustness, and policy support of path computation models.

- MIB modules related to communication protocols, routing and signaling extensions, metrics, and PCE monitoring information

6. PCE Architectural Considerations

This section provides a list of the PCE architectural components. Specific realizations and implementation details (state machines or algorithms, etc.) of PCE-based solutions are out of the scope of this document.

Note also that PCE-based path computation does not affect in any way the use of the computed paths. For example, the use of PCE does not change the way in which Traffic Engineering LSPs are signaled, maintained, and torn down, but it strictly relates to the path computation aspects of such TE LSPs.

This section presents an architectural view of PCE. That is, it describes the components that exist and how they interact. Note that the architectural model, and in particular the functional model, may be perceived differently by different components of the PCE system. For example, the PCC will not be aware of whether a PCE consults other PCEs. The PCC view of the PCE architecture is discussed in Section 7.

6.1. Centralized Computation Model

A "centralized computation model" considers that all path computations for a given domain will be performed by a single, centralized PCE. This may be a dedicated server (for example, an external PCE node), or a designated router (for example, a composite PCE node) in the network. In this model, all PCCs in the domain would send their path computation requests to the central PCE. While

a domain in this context might be an IGP area or AS, it might also be a sub-group of network nodes that is defined by its dependence on the PCE.

This model has a single point of failure: the PCE. In order to avoid this issue, the centralized computation model may designate a backup PCE that can take over the computation responsibility in a controlled manner in the event of a failure of the primary PCE. Any policies present on the primary PCE should also be present on the backup, although the primary policies may themselves be subject to policy governing how they are implemented on the backup. Note that at any moment in time there is only one active PCE in any domain.

6.2. Distributed Computation Model

A "distributed computation model" refers to a domain or network that may include multiple PCEs, and where computation of paths is shared among the PCEs. A given path may in turn be computed by a single PCE ("single PCE path computation") or multiple PCEs ("multiple PCE path computation"). A PCC may be linked to a particular PCE or may be able to choose freely among several PCEs; the method of choice between PCEs is out of scope of this document, but see Section 6.4 for a discussion of PCE discovery that affects this choice. Implementation of policy should be consistent across the set of available PCEs.

Often, the computation of an individual path is performed entirely by a single PCE. For example, this is usually the case in MPLS TE within a single IGP area where the ingress LSR/composite PCE node is responsible for computing the path or for contacting an external PCE. Conversely, multiple PCE path computation implies that more than one PCE is involved in the computation of a single path. An example of this is where loose hop expansion is performed by transit LSRs/composite PCE nodes on an MPLS TE LSP. Another example is the use of multiple cooperating PCEs to compute the path of a single TE LSP across multiple domains.

6.3. Synchronization

Often, multiple paths need to be computed to support a single service (for example, for protection or load sharing). A PCC that determines that it requires more than one path to be computed may send a series of individual requests to the PCE. In this case of non-synchronized path computation requests, the PCE may make multiple individual path computations to generate the paths, and the PCC may send its individual requests to different PCEs.

Alternatively, the PCC may send a single request to a PCE asking for a set of paths to be computed, but specifying that non-synchronized path computation is acceptable. The PCE may compute each path in turn exactly as it would have done had the PCC made multiple requests, and the PCE may devolve some computations to other PCEs if it chooses. On the other hand, the PCE is not prohibited from performing all computations together in a synchronized manner as described below.

The PCC may also issue a single request to the PCE asking for all the paths to be computed in a synchronized manner. The PCE will then perform simultaneous computation of the set of requested paths. Such synchronized computation can often provide better results.

The involvement of more than one PCE in the computation of a series of paths is by its nature non-synchronized. However, a set of cooperating PCEs may be synchronized under the control of a single PCE. For example, a PCC may send a request to a PCE that invokes domain-specific computations by other PCEs before supplying a result to the PCC.

It is desirable to add a parameter to the PCC-PCE protocol to request that the PCE supply a set of alternate paths for use by the PCC, should the establishment of the TE LSP using the principal path fail to complete. While alternate paths may not always be successful if the first path fails, including alternate paths in a PCE response could have less overhead than having the PCC make separate requests for subsequent path computations as the need arises. This technique is used in some existing CSPF implementations.

6.4. PCE Discovery and Load Balancing

In order that a PCC can communicate efficiently with a PCE, it must know the location of the PCE. That is, it is an architectural decision made here that PCC requests be targeted to a specific PCE, and not broadcast to the network for any PCE to respond. This decision means that only the selected PCE will operate on any single request, and it saves network resources during request propagation and processing resources at the PCEs that are not required to respond.

The knowledge of the location of a PCE may be achieved through local configuration at the PCC or may rely on a protocol-based discovery mechanism that may be governed by policy.

Where more than one PCE is known to a PCC, the PCC must have sufficient information to select an appropriate PCE for its purposes, under the control of policy. Such a selection procedure allows for

load sharing between PCEs and supports PCEs with different computation capabilities including different visibility scopes. Thus, the information available to the PCC must include details of the PCE capabilities, which may be fixed or may vary dynamically in time.

The PCC may learn PCE capabilities through static configuration, or it may discover the information dynamically. Note that even when the location of the PCE is configured at the PCC, the PCC may still discover the PCE capabilities dynamically. Dynamic PCE capabilities cannot be configured and can only be discovered.

Proxy PCE advertisement whereby the existence of a PCE is advertised via a proxy PCE is a viable alternative, should the PCE be incapable of such advertisement itself. In this case, it is a requirement that the proxy adequately advertise the PCE status and capability in a timely and synchronized fashion.

In the event that multiple PCEs are available to serve a particular path computation request, the PCC must select a PCE to satisfy the request. The details of such a selection (for instance, to efficiently share the computation load across multiple PCEs or to request secondary computations after partial or failed computations) are local to the PCC, may be based on policy, and are out of the scope of this document.

PCE capabilities that may be advertised or configured could include (and are not be limited to):

- a set of constraints that it can account for (diversity, shared risk link groups (SRLGs), optical impairments, wavelength continuity, etc.)

- computational capacity (for example, the number of computations it can perform per second)

- the number of switching capability layers (and which ones)

- the number of path selection criteria (and which ones)

- whether it is a stateless PCE or it can send updates about better paths that might be available in the future

- whether it can compute P2MP trees (and which types)

- whether it can ensure resource sharing between backup tunnels

This information would help a PCC to decide which PCE to use.

Requirements for PCE advertisement will be documented separately. Note that there is no restriction within the architecture about how location and capabilities are advertised, and the two elements should be considered functionally distinct.

A PCC might also ask a PCE to perform a particular type of service without knowledge of the PCE's capabilities and receive a response that says that the PCE is unable to perform the service. The response could specify the capabilities of the PCE and might also suggest another PCE that has the requested capabilities.

6.5. Detecting PCE Liveness

The ability to detect a PCE's liveness is a mandatory piece of the overall architecture and could be achieved by several means. If some form of regular advertisement (such as through IGP extensions) is used for PCE discovery, it is expected that the PCE liveness will be determined by means of status advertisement (for example, IGP LSA/LSPs).

The inability of a PCE to service a request (perhaps due to excessive load) may be reported to the PCC through a failure message, but the failure of a PCE or the communications mechanism while processing a request cannot be reported in this way. Furthermore, in the case of excessive load, the PCE may not have sufficient resources to send a failure message. Thus, the PCC should employ other mechanisms, such as protocol timers, to determine the liveness of the PCE. This is particularly important in the case of inter-domain path computation where the PCE liveness may not be detected by means of the IGP that runs in the PCC's domain.

6.6. PCC-PCE and PCE-PCE Communication

Once the PCC has selected a PCE, and provided that the PCE is not local to the PCC, a request/response protocol is required for the PCC to communicate the path computation requests to the PCE and for the PCE to return the path computation response. Discussion of the security requirements and implications for this protocol is provided in Section 10 of this document.

The path computation request may include a significant set of requirements, including the following:

- the source and destination of the path

- the bandwidth and other Quality of Service (QoS) parameters desired

- resources, resource affinities, and shared risk link groups (SRLGs) to use/avoid

- the number of disjoint paths required and whether near-disjoint paths are acceptable

- the levels of resiliency, reliability, and robustness of the path resources

- policy-related information

The level of robustness of the path resources covers a qualitative assessment of the vulnerability of the resources that may be used. For example, one might grade resources based on empirical evidence (mean time between failures), on known risks (there is major building work going on near this conduit), or on prejudice (vendor X's software is always crashing). A PCC could request that only robust resources be used, or it could allow any resource.

In case of a positive response from the PCE, one or more paths would be returned to the requesting node. In the event of a failure to compute the desired path(s), an error is returned together with as much information as possible about the reasons for the failure(s), and potentially with advice about which constraints might be relaxed so that a positive result is more likely in a future request.

Note that the resultant path(s) may be made up of a set of strict or loose hops, or any combination of strict and loose hops. Moreover, a hop may have the form of a non-explicit abstract node.

A request/response protocol is also required for a PCE to communicate path computation requests to another PCE and for the PCE to return the path computation response. The path computation request may include a significant set of requirements including those defined above. In case of a positive response from the PCE, one or more paths would be returned to the requesting PCE. In the event of a failure to compute the desired path(s), an error is returned together with as much information as possible about the reasons for the failure, and potentially advice about which constraints might be relaxed so that a positive result is more likely. Note that the resultant path(s) may be made up of a set of strict or loose hops, or any combination of strict and loose hops. Moreover, a hop may have the form of a non-explicit abstract node.

An important feature of PCEs that are cooperating to compute a path is that they apply compatible or identical computation algorithms and coordinated policies. This may require coordination through the communication between the PCEs.

Note that when multiple PCEs cooperate to compute a path, it is important that they have a coordinated view of the meaning of constraints such as costs, resource affinities, and class of service. This is particularly significant where the PCEs are responsible for different domains. It is assumed that this is a matter of policy between domains and between PCEs.

No assumption is made in this architecture about whether the PCC-PCE and PCE-PCE communication protocols are identical.

6.7. PCE TED Synchronization

As previously described, the PCE operates on a TED. Information on network status to build the TED may be provided in the domain by various means:

1) Participation in IGP distribution of TE information. The standard method of distribution of TE information within an IGP area is through the use of extensions to the IGP [RFC3630, RFC3748]. This mechanism allows participating nodes to build a TED, and this is the standard technique, for example, within a single area MPLS or GMPLS network. A node that hosts the PCE function may collect TE information in this way by maintaining at least one routing adjacency with a router in the domain. The PCE node may be adjacent or non-adjacent (via some tunneling techniques) to the router. Such a technique provides a mechanism for ensuring that the TED is efficiently synchronized with the network state and is the normal case, for example, when the PCE is co-resident with the LSRs in an MPLS or GMPLS network.

2) Out-of-band TED synchronization. It may not be convenient or possible for a PCE to participate in the IGPs of one or more domains (for example, when there are very many domains, when IGP participation is not desired, or when some domains are not running TE-aware IGPs). In this case, some mechanism may need to be defined to allow the PCE node to retrieve the TED from each domain. Such a mechanism could be incremental (like the IGP in the previous case), or it could involve a bulk transfer of the complete TED. The latter might significantly limit the capability to ensure TED synchronization, which might result in an increase in the failure rate of computed paths, or the computation of sub-optimal paths. Consideration should also be given to the impact of the TED distribution on the network and on the network node within the domain that is asked to distribute the database. This is particularly relevant in the case of frequent network state changes.

3) Information in the TED can include information obtained from sources other than the IGP. For example, information about link usage policies can be configured by the operator. Path computation can also act on a far wider set of information that includes data about the TE LSPs provisioned within the network. This information can include TE LSP routes, reserved bandwidth, and measured traffic volume passing through the TE LSP.

   Such TE LSP information can enhance TE LSP (re)optimization to provide "full network" (re)optimization and can allow traffic fluctuations to be taken into account. Detailed TE LSP information may also facilitate reconfiguration of the Virtual Network Topology (VNT) [MLN], in which lower-layer TE LSPs, such as optical paths, provide TE links for use by the higher layer, since this reconfiguration is also a "full network" problem.

Note that synchronization techniques may apply to both intra- and inter-domain TEDs. Furthermore, the techniques can be mixed for use in different domains. The degree of synchronization between the PCE and the network is subject to implementation and/or policy. However, better synchronization generally leads to paths that are more likely to succeed.

Note also that the PCE may have access to only a partial TED: for instance, in the case of inter-domain path computation where each such domain may be managed by different entities. In such cases, each PCE may have access to a partial TED, and cooperative techniques between PCEs may be used to achieve end-to-end path computation without any requirement that any PCE handle the complete TED related to the set of traversed domains by the TE LSP in question.

6.8. Stateful versus Stateless PCEs

A PCE can be either stateful or stateless. In the former case, there is a strict synchronization between the PCE and not only the network states (in term of topology and resource information), but also the set of computed paths and reserved resources in use in the network. In other words, the PCE utilizes information from the TED as well as information about existing paths (for example, TE LSPs) in the network when processing new requests. Note that although this allows for optimal path computation and increased path computation success, stateful PCEs require reliable state synchronization mechanisms, with potentially significant control plane overhead and the maintenance of a large amount of data/states (for example, full mesh of TE LSPs).

For example, if there is only one PCE in the domain, all TE LSP computation is done by this PCE, which can then track all the existing TE LSPs and stay synchronized (each TE LSP state change must

be tracked by the PCE). However, this model could require substantial control plane resources. If there are multiple PCEs in the network, TE LSP computation and information are distributed among PCEs and so the resources required to perform the computations are also distributed. However, synchronization issues discussed in Section 6.7 also come into play.

The maintenance of a stateful database can be non-trivial. However, in a single centralized PCE environment, a stateful PCE is almost a simple matter of remembering all the TE LSPs the PCE has computed, that the TE LSPs were actually set up (if this can be known), and when they were torn down. Out-of-band TED synchronization can also be complex, with multiple PCE setup in a distributed PCE computation model, and could be prone to race conditions, scalability concerns, etc. Even if the PCE has detailed information on all paths, priorities, and layers, taking such information into account for path computation could be highly complex. PCEs might synchronize state by communicating with each other, but when TE LSPs are set up using distributed computation performed among several PCEs, the problems of synchronization and race condition avoidance become larger and more complex.

There is benefit in knowing which TE LSPs exist, and their routing, to support such applications as placing a high-priority TE LSP in a crowded network such that it preempts as few other TE LSPs as possible (also known as the "minimal perturbation" problem). Note that preempting based on the minimum number of links might not result in the smallest number of TE LSPs being disrupted. Another application concerns the construction and maintenance of a Virtual Network Topology [MLN]. It is also helpful to understand which other TE LSPs exist in the network in order to decide how to manage the forward adjacencies that exist or need to be set up. The cost-benefit of stateful PCE computation would be helpful to determine if the benefit in path computation is sufficient to offset the additional drain on the network and computational resources.

Conversely, stateless PCEs do not have to remember any computed path and each set of request(s) is processed independently of each other. For example, stateless PCEs may compute paths based on current TED information, which could be out of sync with actual network state given other recent PCE-computed paths changes. Note that a PCC may include a set of previously computed paths in its request, in order to take them into account, for instance, to avoid double bandwidth accounting or to try to minimize changes (minimum perturbation problem).

Note that the stateless PCE does operate on information about network state. The TED contains link state and bandwidth availability information as distributed by the IGPs or collected through some other means. This information could be further enhanced to provide increased granularity and more detail to cover, for example, the current bandwidth usage on certain links according to resource affinities or forwarding equivalence classes. Such information is, however, not PCE state information and so a model that uses it is still described as stateless in the PCE context.

A limited form of statefulness might be applied within an otherwise stateless PCE. The PCE may retain some context from paths it has recently computed so that it avoids suggesting the use of the same resources for other TE LSPs.

6.9. Monitoring

PCE monitoring is undoubtedly of the utmost importance in any PCE architecture. This must include the collection of variables related to the PCE status and operation. For example, it will be necessary to understand the way in which the TED is being kept synchronized, the rate of arrival of new requests and the computation times, the range of PCCs that are using the PCE, and the operation of any PCC-PCE protocol.

6.10. Confidentiality

As stated in [RFC4216], the case of inter-provider TE LSP computation requires the ability to compute a path while preserving confidentiality across multiple Service Providers cores. That is, one Service Provider must not be required to divulge any information about its resources or topology in order to support inter-provider TE LSP path computation. Thus, any PCE architecture solution must support the ability to return partial paths by means of loose hops (for example, where each loose hop would, for instance, identify a boundary LSR).

This requirement is not a security issue, but relates to Service Provider policy. Confidentiality, integrity, and authentication of PCC-PCE and PCE-PCE messages must also be ensured and are described in Section 10.

The ability to compute a path at the request of the head-end PCC, but to supply the path in segments to the domain boundary PCCs, may also be desirable.

6.11. Policy

Policy impacts multiple aspects of the PCE architecture. There are two applications of policy for consideration:

- application of policy within an architectural entity (PCC or PCE)

- application of policy to PCE-related communications

As directly applicable to TE LSPs, policy forms part of the signaling mechanism for the establishment of the TE LSPs and is not described here.

It is envisioned that policy will be largely applied as a local matter within each PCC and PCE. However, this document needs to define policy models that can be supported within the PCE architecture and by PCE-related communication.

Some example policies include:

- selection of a PCE by a PCC

- rejection of a request by the PCE based on the identity of the requesting PCC

- selection by the PCE of a path or application of additional constraints to a computation based on the PCC, the computation target, the time of day, etc.

6.11.1. PCE Policy Architecture

Two examples of the use of policy components within the PCE architecture are illustrated in Figures 6 and 7. Policy components could equally be applied to the other PCE configurations shown in Section 5. In each configuration, policy may be consulted before a response is provided by a PCE and may also be consulted by the PCC/PCE that receives the response.

A PCE may have a local policy that impacts the paths selected to satisfy a particular PCE request. A policy may be applied based on any information provided from a PCC.

In Figure 6, the policy component is shown providing input to the PCE component. This policy component may consult an external policy database, but this is outside the scope of this document.

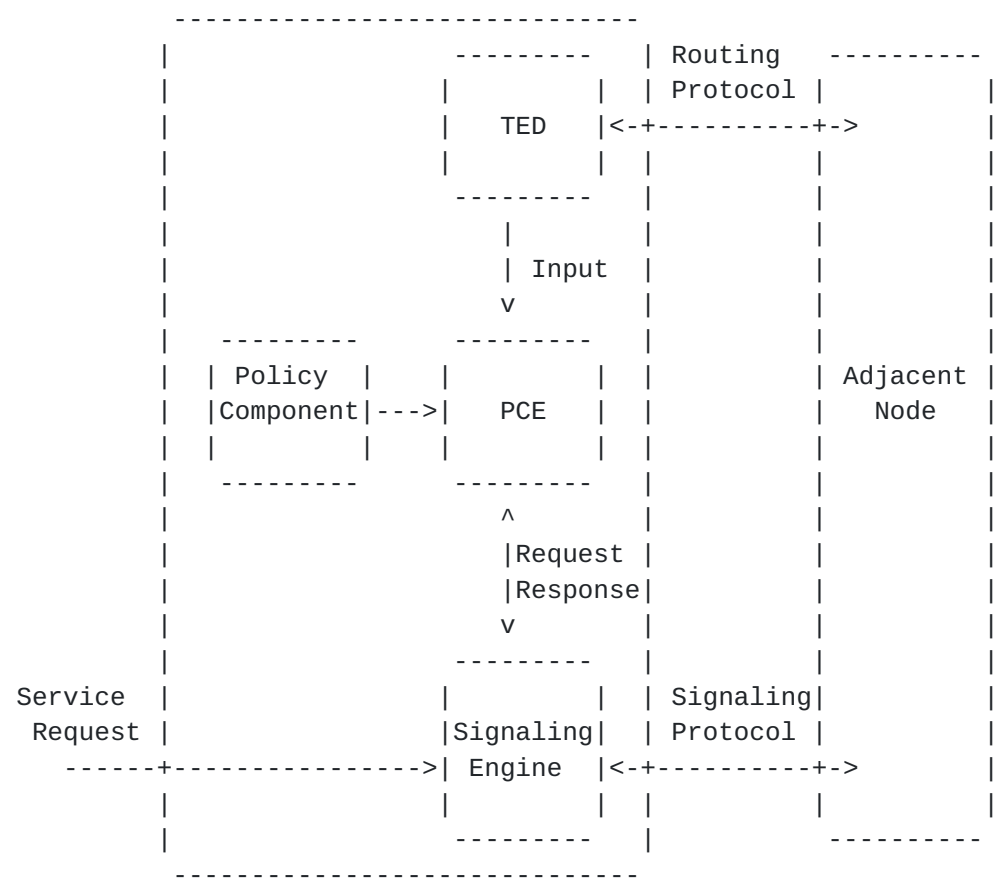

Figure 6. Policy Component in the Composite PCE Node

Note that policy information may be conveyed on the internal interfaces, and on the external protocol interfaces.

Figure 7 displays the case of a distinct PCE function through the example of the multiple PCE with inter-PCE communication example (compare with Figure 4). Each PCE takes input from local policy as part of the router computation/determination process. The local policy components may consult external policy components or databases, but that is out of the scope of this document.

Note that policy information may be conveyed on the external protocol interfaces, including the inter-PCE interface.

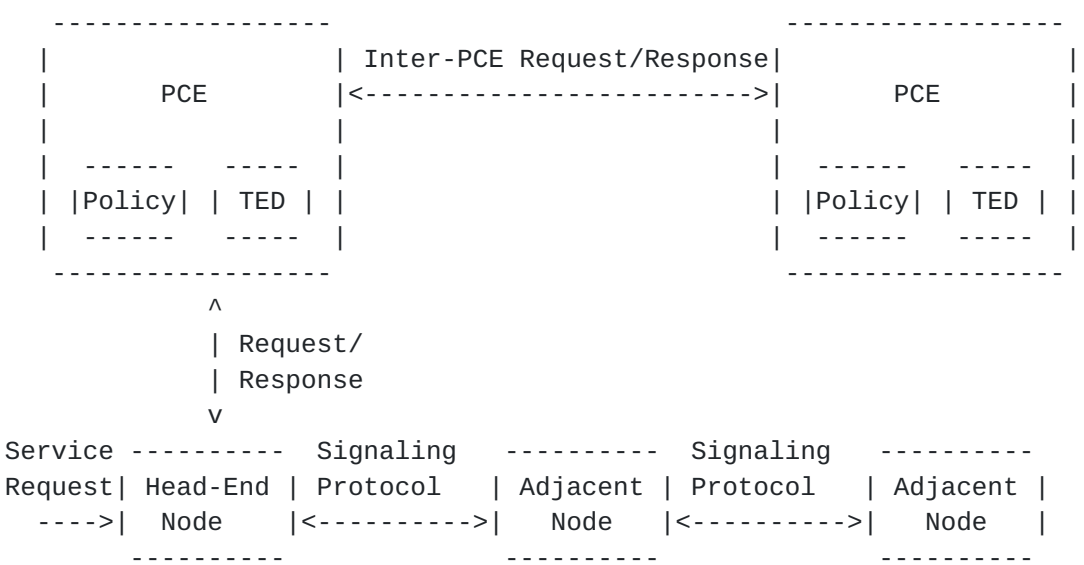

Figure 7. Policy Components in Multiple PCEs

6.11.2. Policy Realization

There are multiple options for how policy information is coordinated.

- Policy decisions may be made by PCCs before consulting PCEs. This type of decision includes selection of PCE, application of constraints, and interpretation of service requests.

- Policy decisions may be made independently at a PCE, or at each cooperating PCE. That is, the PCE(s) may make policy decisions independent of other policy decisions made at PCCs or other PCEs.

- There may also be explicit communication of policy information between PCC and PCE, or between PCEs to achieve some level of coordination of policy between entities. The type of information conveyed to support policy has important implications on what policies may be applied at each PCE, and the requirements for the exchange of policy information inform the choice or implementation of communication protocols including PCC-PCE, PCE-PCE, and discovery protocols.

6.11.3. Type of Policies

Within the context of PCE, we identify several types of policies:

o User-specific policies operate on information that is specific to the user of a service or the service itself, that is, the service for which the path is being computed, not the computation service. Examples of such information includes the contents of objects of a

signaling or provisioning message, the port ID over which the message was received, a VPN ID, a reference point type, or the identity of the user initiating the request. User-specific policies could be applied by a PCC while building a path computation request, or by a PCE while processing the request provided that sufficient information is supplied by the PCC to the PCE.

- Request-specific policies operate on information that is specific to a path computation request and is carried in the request. Examples of such information include constraints, diversities, constraint and diversity relaxation strategies, and optimization functions. Request-specific policies directly affect the path selection process because they specify which links, nodes, path segments, and/or paths are not acceptable or, on the contrary, may be desirable in the resulting paths.

- Domain-specific policies operate on the identify of the domain in which the requesting PCC exists, and upon the identities of the domains through which the resulting paths are routed. These policies have the same effect as user-specific policies, with the difference that they can be applied to a group of users rather than an individual user. One example of domain-specific policy is a restriction on what information a PCE publishes within a given domain. In such a case, PCEs in some domains may advertise just their presence, while others may advertise details regarding their capabilities, client authentication process, and computation resource availability.

6.11.4. Relationship to Signaling

When a path for an inter-domain TE LSP is being computed, it is not required to consider signaling plane policy. However, failure to do so may result in the TE LSP failing to be established, or being assigned fewer resources than intended resulting in a substandard service. Thus, where a PCE invoked by a head-end LSR has visibility into other domains, it should be capable of applying policy considerations to the computation and should be aware of the inter-domain policy agreements. Where path computation is the result of cooperation between PCEs, each of which is responsible for a particular domain, the policy issues should, where possible, be resolved at the time of computation so that the TE LSP is more likely to be signaled successfully. In this context, policy violation during inter-domain TE LSP computation may lead to path computation interruption, about which the requester should be notified along with the cause.

6.12. Unsolicited Interactions

It may be that the PCC-PCE communications (see Section 6.6) can be usefully extended beyond a simple request/response interaction. For example, the PCE and PCC could exchange capabilities using this protocol. Additionally, the protocol could be used to collect and report information in support of a stateful PCE.

Furthermore, it may be the case that a PCE is able to update a path that it computed earlier (perhaps in reaction to a change in the network or a change in policy), and in this case the PCE-PCC communication could support an "unsolicited" path computation message to supply this new path to the PCC. Note, however, that this function would require that the PCE retained a record of previous computations and had a clear trigger for performing recomputations. The PCC would also need to be able to identify the new path with the old path and determine whether it should act on the new path. Further, the PCC should be able to report the outcome of such path changes to the requesting PCE. Note that the PCE-PCC interaction is not a management interaction and the PCC is not obliged to utilize any additional path supplied by the PCE.

These functions fit easily within the architecture described here but are left for further discussion within separate requirements documents.

6.13. Relationship with Crankback

Crankback routing is a mechanism whereby a failure to establish a path or a failure of an existing path may be corrected by a new path computation and fresh signaling. Crankback routing relies on the distribution of crankback information along with the failure notification so that the new computation can be performed avoiding the failure or blockage point.

In the context of PCE, crankback information may be passed back to the head-end where the process of computation and signaling can be repeated using the failed resource as an exclusion in the computation process. But crankback may be used to attempt to correct the problem at intermediate points along the path. Such crankback recomputation nodes are most likely to be domain boundaries where the PCC had already invoked a PCE. Thus, a failure within a domain is reported to the ingress domain boundary, which will attempt to compute an alternate path across the domain. Failing this, the problem may be reported to the previous domain and communicated to the ingress boundary for that domain, which may attempt to select a more

successful path either by choosing a different entry point into the next domain, or by selecting a route through a different set of domains.

7. The View from the Path Computation Client

The view of the PCE architecture, and particularly the functional model, is subtly different from the PCC's perspective. This is partly because the PCC has limited knowledge of the way in which the PCEs cooperate to answer its requests, but depends more on the fact that the PCC is concerned with different questions.

The PCC is interested in the following:

- Selecting a PCE that is able to promptly provide a computed path that meets the supplied constraints.

- How many computation requests will the PCC have to send? Will the desired path be computed by the first PCE contacted (possibly in cooperation with other PCEs), or will the PCC have to consult other PCEs to fill in gaps in the path?

- How many other path computations will need to be issued from within the network in order to establish the TE LSP?

This last question might be considered out of scope for the head-end LSR, but an important constraint that the PCC may wish to apply is that the path should be computed in its entirety and supplied without loose hops or non-simple abstract nodes.

Thus, with its limited perspective, the PCC will see Multiple PCE Path Computation (Section 5.3) as important and will distinguish two subcases. The first is as shown in Figure 3 with subsequent computation requests made by other PCCs along the path of the TE LSP. In the second, multiple computation requests are issued by the head-end LSR. On the other hand, the PCC will not be aware of Multiple PCE Path Computation with Inter-PCE Communication (Section 5.4), which it will perceive as no different from the simple External PCE Node case (Section 5.2).

The PCC, therefore, will be acutely aware that a Centralized PCE Model (Section 6.1) might still require Multiple PCE Path Computations with the head-end or subsequent PCCs required to issue further requests to the central PCE. Conversely, the PCC may be protected from the Distributed PCE Model (Section 6.2) because the first PCE it consults uses inter-PCE communication to achieve a complete computation result so that no further computation requests are required.

These distinctions can be completely classified by determining whether the computation response includes all necessary paths, and whether those paths are fully explicit (that is, containing only strict hops between simple abstract nodes).

8. Evaluation Metrics

Evaluation metrics that may be used to evaluate the efficiency and applicability of any PCE-based solution are listed below. Note that these metrics are not being used to determine paths, but are used to evaluate potential solutions to the PCE architecture.

- Optimality: The ability to maximize network utilization and minimize cost, considering QoS objectives, multiple regions, and network layers. Note that models that require the sequential involvement of multiple PCEs (for example, the multiple PCE model described in Section 5.3) might create path loops unless careful policy is applied.

- Scalability: The implications of routing, TE LSP signaling, and PCE communication overhead, such as the number of messages and the size of messages (including LSAs, crankback information, queries, distribution mechanisms, etc.).

- Load sharing: The ability to allow multiple PCEs to spread the path computation load by allowing multiple PCEs each to take responsibility for a subset of the total path computation requests.

- Multi-path computation: The ability to compute multiple and potentially diverse paths to satisfy load-sharing of traffic and protection/restoration needs including end-to-end diversity and protection within individual domains.

- Reoptimization: The ability to perform TE LSP path reoptimization. This also includes the ability to perform inter-layer correlation when considering the reoptimization at any specific layer.

- Path computation time: The time to compute individual paths and multiple diverse paths and to satisfy bulk path computation requests. (Note that such a metric can only be applied to problems that are not NP-complete.)

- Network stability: The ability to minimize any perturbation on existing TE state resulting from the computation and establishment of new TE paths.

- Ability to maintain accurate synchronization between TED and network topology and resource states.

- Speed with which TED synchronization is achieved.

- Impact of the synchronization process on the data flows in the network.

- Ability to deal with situations where paths satisfying a required set of constraints cannot be found by the PCE.

- Policy: Application of policy to the PCC-PCE and PCE-PCE communications as well as to the computation of paths that respect inter-domain TE LSP establishment policies.

Note that other metrics may also be considered. Such metrics should be used when evaluating a particular PCE-based architecture. The potential tradeoffs of the optimization of such metrics should be evaluated (for instance, increasing the path optimality is likely to have consequences on the computation time).

9. Manageability Considerations

The PCE architecture introduces several elements that are subject to manageability. The PCE itself must be managed, as must its communications with PCCs and other PCEs. The mechanism by which PCEs and PCCs discover each other are also subject to manageability.

Many of the issues of manageability are already covered in other sections of this document.

9.1. Control of Function and Policy

It must be possible to enable and disable the PCE function at a PCE, and this will lead to the PCE accepting, rejecting, or simply not receiving requests from PCCs. Graceful shutdown of the PCE function should also be considered so that in controlled circumstances (such as software upgrade) a PCE does not just 'disappear' but warns its PCCs and gracefully handles any queued computation requests (perhaps by completing them, forwarding them to another PCE, or rejecting them).

Similarly it must be possible to control the application of policy at the PCE through configuration. This control may include the restriction of certain functions or algorithms, the configuration of access rights and priorities for PCCs, and the relationships with other PCEs both inside and outside the domain.

The policy configuration interface is yet to be determined. The interface may be purely a local matter, or it may be supported via a standardized interface (such as a MIB module).

9.2. Information and Data Models

It is expected that the operations of PCEs and PCCs will be modeled and controlled through appropriate MIB modules. The tables in the new MIB modules will need to reflect the relationships between entities and to control and report on configurable options.

Statistics gathering will form an important part of the operation of PCEs. The operator must be able to determine the historical interactions of a PCC with its PCEs, the performance that it has seen, and the success rate of its requests. Similarly, it is important for an operator to be able to inspect a PCE and determine its load and whether an individual PCC is responsible for a disproportionate amount of the load. It will also be important to be able to record and inspect statistics about the communications between the PCC and PCE, including issues such as malformed messages, unauthorized messages, and messages discarded because of congestion. In this respect, there is clearly an overlap between manageability and security.

Statistics for the PCE architecture can be made available through appropriate tables in the new MIB modules.

The new MIB modules should also be used to provide notifications when key thresholds are crossed or when important events occur. Great care must be exercised to ensure that the network is not flooded with Simple Network Management Protocol (SNMP) notifications. Thus, it might be inappropriate to issue a notification every time a PCE receives a request to compute a path. In any case, full control must be provided to allow notifications to be disabled using, for example, the mechanisms defined in the SNMP-NOTIFICATION-MIB module in [RFC3413].

9.3. Liveness Detection and Monitoring

Section 6.5 discusses the importance of a PCC being able to detect the liveness of a PCE. PCE-PCC communications techniques must enable a PCC to determine the liveness of a PCE both before it sends a request and in the period between sending a request and receiving a response.

It is less important for a PCE to know about the liveness of PCCs, and within the simple request/response model, this is only helpful

- to gain a predictive view of the likely loading of a PCE in the future, or

- to allow a PCE to abandon processing of a received request.

9.4. Verifying Correct Operation

Correct operation for the PCE architecture can be classified as determining the correct point-to-point connectivity between PCCs and PCEs, and as assessing the validity of the computed paths. The former is a security issue that may be enhanced by authentication and monitored through event logging and records as described in Section 9.1. It may also be a routing issue to ensure that PCC-PCE connectivity is possible.

Verifying computed paths is more complex. The information to perform this function can, however, be made available to the operator through MIB tables, provided that full records are kept of the constraints passed on the request, the path computed and provided on the response, and any additional information supplied by the PCE such as the constraint relaxation policies applied.

9.5. Requirements on Other Protocols and Functional Components

At the architectural stage, it is impossible to make definitive statements about the impact on other protocols and functional components since the solution's work has not been completed. However, it is possible to make some observations.

- Dependence on underlying transport protocols

  PCE-PCC communications may choose to utilize underlying protocols to provide transport mechanisms. In this case, some of the manageability considerations described in the previous sections may be devolved to those protocols.

- Re-use of existing protocols for discovery

  Without prejudicing the requirements and solutions work for PCE discovery (see Section 6.4), it is possible that use will be made of existing protocols to facilitate this function. In this case some of the manageability considerations described in the previous sections may be devolved to those protocols.

- Impact on LSRs and TE LSP signaling

  The primary example of a PCC identified in this architecture is an MPLS or a GMPLS LSR. Consideration must therefore be given to the manageability of the LSRs and the additional manageability constraints applicable to the TE LSP signaling protocols.

In addition to allowing the PCC management described in the previous sections, an LSR must be configurable to determine whether it will use a remote PCE at all, the options being to use hop-by-hop routing or to supply the PCE function itself. It is likely to be important to be able to distinguish within an LSR whether the route used for a TE LSP was supplied in a signaling message from another LSR, by an operator, or by a PCE, and, in the case where it was supplied in a signaling message, whether it was enhanced or expanded by a PCE.

- Reuse of existing policy models and mechanisms

  As policy support mechanisms can be quite extensive, it is worthwhile to explore to what extent this prior work can be leveraged and applied to PCE. This desire to leverage prior work should not be interpreted as a requirement to use any particular solution or protocol.

9.6. Impact on Network Operation

This architecture may have two impacts on the operation of a network. It increases TE LSP setup times while requests are sent to and processed by a remote PCE, and it may cause congestion within the network if a significant number of computation requests are issued in a small period of time. These issues are most severe in busy networks and after network failures, although the effect may be mitigated if the protection paths are precomputed or if the path computation load is distributed among a set of PCEs.

Issues of potential congestion during recovery from failures may be mitigated through the use of pre-established protection schemes such as fast reroute.

It is important that network congestion be managed proactively because it may be impossible to manage it reactively once the network is congested. It should be possible for an operator to rate limit the requests that a PCC sends to a PCE, and a PCE should be able to report impending congestion (according to a configured threshold) both to the operator and to its PCCs.

9.7. Other Considerations

No other management considerations have been identified.

10. Security Considerations

The impact of the use of a PCE-based architecture must be considered in the light of the impact that it has on the security of the existing routing and signaling protocols and techniques in use within the network. The impact may be less likely to be an issue in the case of intra-domain use of PCE, but an increase in inter-domain information flows and the facilitation of inter-domain path establishment may increase the vulnerability to security attacks.

Of particular relevance are the implications for confidentiality inherent in a PCE-based architecture for multi-domain networks. It is not necessarily the case that a multi-domain PCE solution will compromise security, but solutions MUST examine their effects in this area.

Applicability statements for particular combinations of signaling, routing and path computation techniques are expected to contain detailed security sections.

Note that the use of a non-local PCE (that is, one not co-resident with the PCC) does introduce additional security issues. Most notable among these are:

- interception of PCE requests or responses;

- impersonation of PCE or PCC;

- falsification of TE information, policy information, or PCE capabilities; and

- denial-of-service attacks on PCE or PCE communication mechanisms.

It is expected that PCE solutions will address these issues in detail using authentication and security techniques.

11. Acknowledgements

The authors would like to extend their warmest thanks to (in alphabetical order) Arthi Ayyangar, Zafar Ali, Lou Berger, Mohamed Boucadair, Igor Bryskin, Dean Cheng, Vivek Dubey, Kireeti Kompella, Jean-Louis Le Roux, Stephen Morris, Eiji Oki, Dimitri Papadimitriou, Richard Rabbat, Payam Torab, Takao Shimizu, and Raymond Zhang for their review and suggestions. Lou Berger provided valuable and detailed contributions to the discussion of policy in this document.

Thanks also to Pekka Savola, Russ Housley and Dave Kessens for review and constructive discussions during the final stages of publication.

12. Informative References

[RFC2702] Awduche, D., Malcolm, J., Agogbua, J., O'Dell, M., and J. McManus, "Requirements for Traffic Engineering Over MPLS", RFC 2702, September 1999.

[RFC4364] Rosen, E. and Y. Rekhter, "BGP/MPLS IP Virtual Private Networks (VPNs)", RFC 4364, February 2006.

[RFC3209] Awduche, D., Berger, L., Gan, D., Li, T., Srinivasan, V., and G. Swallow, "RSVP-TE: Extensions to RSVP for LSP Tunnels", RFC 3209, December 2001.

[RFC3630] Katz, D., Kompella, K., and D. Yeung, "Traffic Engineering (TE) Extensions to OSPF Version 2", RFC 3630, September 2003.

[RFC3413] Levi, D., Meyer, P., and B. Stewart, "Simple Network Management Protocol (SNMP) Applications", STD 62, RFC 3413, December 2002.

[RFC3473] Berger, L., "Generalized Multi-Protocol Label Switching (GMPLS) Signaling Resource ReserVation Protocol-Traffic Engineering (RSVP-TE) Extensions", RFC 3473, January 2003.

[RFC3748] Smit, H. and T. Li, "Intermediate System to Intermediate System (IS-IS) Extensions for Traffic Engineering (TE)", RFC 3784, June 2004.

[RFC3812] Srinivasan, C., Viswanathan, A., and T. Nadeau, "Multiprotocol Label Switching (MPLS) Traffic Engineering (TE) Management Information Base (MIB)", RFC 3812, June 2004.

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

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

[MLN] Shiomoto, K., Papdimitriou, D., Le Roux, J.-L., Vigoureux, M., and D. Brungard, "Requirements for GMPLS-based multi-region and multi-layer networks (MRN/MLN)", Work in Progress, June 2006.

Authors' Addresses

Adrian Farrel
Old Dog Consulting

EMail: adrian@olddog.co.uk

Jean-Philippe Vasseur
1414 Massachussetts Avenue
Boxborough, MA 01719
USA

EMail: jpv@cisco.com

Jerry Ash
AT&T
Room MT D5-2A01
200 Laurel Avenue
Middletown, NJ 07748,
USA

Phone: (732)-420-4578
Fax: (732)-368-8659
EMail: gash@att.com

Full Copyright Statement

Intellectual Property

Acknowledgement

Funding for the RFC Editor function is provided by the IETF Administrative Support Activity (IASA).