Internet Engineering Task Force (IETF)
Request for Comments: 7399
Category: Informational
ISSN: 2070-1721
October 2014

A. Farrel
Juniper Networks
D. King
Old Dog Consulting

Unanswered Questions in the Path Computation Element Architecture

Abstract

The Path Computation Element (PCE) architecture is set out in RFC 4655. The architecture is extended for multi-layer networking with the introduction of the Virtual Network Topology Manager (VNTM) in RFC 5623 and generalized to Hierarchical PCE (H-PCE) in RFC 6805.

These three architectural views of PCE deliberately leave some key questions unanswered, especially with respect to the interactions between architectural components. This document draws out those questions and discusses them in an architectural context with reference to other architectural components, existing protocols, and recent IETF efforts.

This document does not update the architecture documents and does not define how protocols or components must be used. It does, however, suggest how the architectural components might be combined to provide advanced PCE function.

Status of This Memo

Table of Contents

1. Introduction

Over the years since the architecture for the Path Computation Element (PCE) was documented in [RFC4655], many new people have become involved in the work of the PCE working group and wish to use or understand the PCE architecture. These people often missed out on early discussions within the working group and are unfamiliar with questions that were raised during the development of the documentation.

Furthermore, the base architecture has been extended to handle other situations and requirements: the architecture was extended for multi-layer networking with the introduction of the Virtual Network Topology Manager (VNTM) [RFC5623] and was generalized to include Hierarchical PCE (H-PCE) [RFC6805].

These three architectural views of PCE deliberately leave some key questions unanswered, especially with respect to the interactions between architectural components. This document draws out those questions and discusses them in an architectural context with reference to other architectural components, existing protocols, and recent IETF efforts.

This document does not update the architecture documents and does not define how protocols or components must be used. It does, however, suggest how the architectural components might be combined to provide advanced PCE function.

1.1. Terminology

Readers are assumed to be thoroughly familiar with terminology defined in [RFC4655], [RFC4726], [RFC5440], [RFC5623], and [RFC6805]. More information about terms related to stateful PCE can be found in [STATEFUL-PCE].

Throughout this document, the term "area" is used to refer equally to an OSPF area and an IS-IS level. It is assumed that the reader is able to map the small differences between these two use cases.

2. What Is Topology Information?

[RFC4655] specifies that a PCE performs path computations based on a view of the available network resources and network topology. This information is collected into a Traffic Engineering Database (TED).

However, [RFC4655] does not provide a detailed description of what information is present in the TED. It simply says that the TED "contains the topology and resource information of the domain." The precise information that needs to be held in a TED depends on the type of network and nature of the computation that has to be performed. As a basic minimum, the TED must contain the nodes and links that form the domain, and it must identify the connectivity in the domain.

For most traffic-engineering needs (for example, MPLS Traffic Engineering - MPLS-TE), the TED would additionally contain a basic metric for each link and knowledge of the available (unallocated) resources on each link.

More advanced use cases might require that the TED contain additional data that represents qualitative information such as:

- link delay
- link jitter
- node throughput capabilities
- optical impairments
- switching capabilities
- limited node cross-connect capabilities

Additionally, an important information element for computing paths, especially for protected services, is the Shared Risk Group (SRG). This is an indication of resources in the TED that have a common risk of failure. That is, they have a shared risk of failure from a single event.

In short, the TED needs to contain as much information as is needed to satisfy the path computation requests subject to the objective functions (OFs). This, in itself, may not be a trivial issue in some network technologies. For example, in some optical networks, the path computation for a new Label Switched Path (LSP) may need to consider the impact that turning up a new laser would have on the optical signals already being carried by fibers. It may be possible to abstract this information as parameters of the optical links and nodes in the TED, but it may be easier to capture this information through a database of existing LSPs (see Sections 14 and 15).

3. How Is Topology Information Gathered?

Clearly, the information in the TED discussed in Section 2 needs to be gathered and maintained somehow. [RFC4655] simply says "The TED may be fed by Interior Gateway Protocol (IGP) extensions or potentially by other means." In this context, "fed" means built and maintained.

Thus, one way that the PCE may construct its TED is by participating in the IGP running in the network. In an MPLS-TE network, this would depend on OSPF TE [RFC3630] and IS-IS TE [RFC5305]. In a GMPLS network, it would utilize the GMPLS extensions to OSPF and IS-IS, [RFC4203] and [RFC5307].

However, participating in an IGP, even as a passive receiver of IGP information, can place a significant load on the PCE. The IGP can be quite "chatty" when there are frequent updates to the use of the network, meaning that the PCE must dedicate significant processing to parsing protocol messages and updating the TED. Furthermore, to be truly useful, a PCE implementation would need to support OSPF and IS-IS.

An alternative feed from the network to the PCE's TED is offered by BGP-LS [LS-DISTRIB]. This approach offers the alternative of leveraging an in-network BGP speaker (such as an Autonomous System Border Router or a Route Reflector) that already has to participate in the IGP and that is specifically designed to apply filters to IGP advertisements. In this usage, the BGP speaker filters and aggregates topology information according to configured policy before advertising it "north-bound" to the PCE to update the TED. The PCE implementation has to support just a simplified subset of BGP rather than two full IGPs.

But BGP might not be convenient in all networks (for example, where BGP is not run, such as in an optical network or a BGP-free core). Furthermore, not all relevant information is made available through standard TE extensions to the IGPs. In these cases, the TED must be built or supplemented from other sources such as the Network Management System (NMS), inventory management systems, and directly configured data.

It has also been proposed that the PCE Communication Protocol (PCEP) [RFC5440] could be extended to serve as an information collection protocol to supply information from network devices to a PCE. The logic is that the network devices may already speak PCEP; so, the protocol could easily be used to report details about the resources and state in the network, including the LSP state discussed in Sections 14 and 15.

Note that a PCE that is responsible for more than one domain must, of course, collect TE information from each domain to build its TED or TEDs.

4. How Do I Find My PCE?

A Path Computation Client (PCC) needs to know the identity/location of a PCE in order to be able to make computation requests. This is because PCEP is a transaction-based protocol carried over TCP, and the architectural decision made in Section 6.4 of RFC 4655 required targeted PCC-PCE communications.

As described in [RFC4655], a PCC could be configured with the knowledge of the IP address of its PCE. This is a relatively lightweight option considering all of the other configuration that a router may require, but it is open to configuration errors, and does not meet the need for minimal-configuration operation. Furthermore, configuration communication with multiple PCEs could become onerous, while handling changes in PCE identities and coping with failure events would be an issue for a configured system.

[RFC4655] offers the possibility for PCEs to advertise themselves in the IGP, and this requirement is developed in [RFC4674] and made possible in OSPF and IS-IS through [RFC5088] and [RFC5089]. In general, these mechanisms should be sufficient for PCCs in a network where an IGP is used and where the PCE participates in the IGP.

Note, however, that not all PCEs will participate in the IGP (see Section 3). In these cases, assuming configuration is not appropriate as a discovery mechanism, some other server announcement/discovery function may be needed, such as DNS [RFC4848] as used for discovery of the Local Location Information Server (LIS) [RFC5986] and in the Application-Layer Traffic Optimization (ALTO) discovery function [ALTO-SERVER-DISC].

5. How Do I Select between PCEs?

When more than one PCE is discovered or configured, a PCC will need to select which PCE to use. It may make this decision on any arbitrary algorithm (for example, first-listed, or round robin), but it may also be the case that different PCEs have different capabilities and path computation scope; in which case, the PCC will want to select the PCE most likely to be able to satisfy any one request. The first requirement, of course, is that the PCE can compute paths for the relevant domain.

PCE advertisement in OSPF or IS-IS per [RFC5088] and [RFC5089] allows a PCE to announce its capabilities as required in [RFC4657]. A PCC can select between PCEs based on the capabilities that they have announced. However, these capabilities are expressed as flags in the PCE advertisement so only the core capabilities are presented, and there is not scope for including detailed information (such as support for specific objective functions) in the advertisement.

Additional and more complex PCE capabilities, including the capability to perform point-to-multipoint (P2MP) path computations [RFC6006], may be announced by the PCE as optional PCEP type-length-value (TLV) Type Indicators in the Open message described in [RFC5440]. This mechanism is not limited to just a set of flags, and detailed capability information may be presented in sub-TLVs.

Note that this exchange of PCE capabilities is in the form of an announcement, not a negotiation. That is, a PCC that wants specific function from a PCE must examine the advertised capabilities and select which PCE to use for a specific request. There is no scope for a PCC to request a PCE to support features or functions that it does not offer or announce.

A PCC may also vary which PCE it uses according to congestion information reported by the PCEs using the Notification Object and Notification Type [RFC5440]. In a heavily overloaded PCE system, note that reports from one PCE that it is overloaded may simply result in all PCCs switching to another PCE, which will, itself, immediately become overloaded. Thus, PCCs should exercise a certain amount of discretion and queueing theory before selecting a PCE purely based on reported load.

Note that a PCC could send all requests to all PCEs that it knows about. It can then select between the results, perhaps choosing the first result it receives, but this approach is very likely to overload all the PCEs in the network considering that one of the reasons for multiple PCEs is to share the load.

6. How Do Redundant PCEs Synchronize TEDs?

A network may have more than one PCE, as discussed in the previous sections. These PCEs may provide redundancy for load-sharing, resilience, or partitioning of computation features.

In order to achieve some consistency between the results of different PCEs, it is desirable that they operate on the same TE information.

The TED reflects the actual state of the network and is not a resource reservation or booking scheme. Therefore, a PCE-based system does not prevent competition for network resources during the provisioning phase, although a process of "sticky resources" that are temporarily reduced in the TED after a computation may be applied purely as a local implementation feature.

One option for ensuring that multiple PCEs use the same TE information is simply to have the PCEs driven from the same TED. This could be achieved in implementations by utilizing a shared database, but it is unlikely to be efficient.

More likely is that each PCE is responsible for building its own TED independently, using the techniques described in Section 3. If the PCEs participate in the IGP, it is likely that they will attach at different points in the network; so, there may be minor and temporary inconsistencies between their TEDs caused by IGP convergence issues. If the PCEs gather TE information via BGP-LS [LS-DISTRIB] from different sources, the same inconsistencies may arise. However, if the PCEs attach to the same BGP speaker, it may be possible to achieve consistency between TEDs modulo the BGP-LS process itself.

A final option is to provide an explicit synchronization process between the TED of a "master" PCE and the TEDs of other PCEs. Such a process could be achieved using BGP-LS or a database synchronization protocol (which would allow check-pointing and sequential updates). This approach is fraught with issues around selection of the master PCE and handling failures. It is, in fact, a mirrored database scenario: a problem that is well known and the subject of plenty of work.

Noting that the provisioning protocols such as RSVP-TE [RFC3209] already handle contention for resources, that the differences between TEDs are likely to be relatively small with moderate arrival rates for new services, and that contention in all but the most busy networks is relatively unlikely, there may be no value in any attempt to synchronize TEDs between PCEs.

However, see Section 16 for a discussion of synchronizing other state between redundant PCEs.

7. Where Is the Destination?

Path computation provides an end-to-end path between a source and a destination. If the destination lies in the source domain, then its location will be known to the PCE and there are no issues to be solved. However, in a multi-domain system a path must be found to a remote domain that contains the destination, and that can only be achieved by knowledge of the location of the destination or at least knowing the next domain in the path toward the domain that contains the destination.

The simplest solution here is achieved when a PCE has visibility into multiple domains. Such may be the case in a multi-area network where the PCE is aware of the contents of all of the IGP areas. This approach is only likely to be appropriate where the number of nodes is manageable, and it is unlikely to extend over administrative boundaries.

The per-domain path computation method for establishing inter-domain traffic engineering LSPs [RFC5152] simply requires a PCE to compute a path to the next domain toward the destination. As the LSP setup (through signaling) progresses domain by domain, the Label Switching Router (LSR) at the entry point to each domain requests its local PCE to compute the next segment of the path, that is from that LSR to the next domain in the sequence toward the destination. Thus, it is not necessary for any PCE (except the last) to know in which domain the destination exists. But, in this approach, each PCE must somehow determine the next domain toward the destination, and it is not obvious how this is achieved.

[RFC5152] suggests that, in an IP/MPLS network, it is good enough to leverage the IP reachability information distributed by BGP and assume that TE reachability can follow the same Autonomous System (AS) path. This approach might not guarantee the optimal TE path and, of course, might result in no path being found in degenerate cases. Furthermore, in many network technologies (such as optical networks operated by GMPLS) there may be limited or no end-to-end IP connectivity.

The Backward Recursive PCE-based Computation (BRPC) procedure [RFC5441] is able to achieve a more optimal end-to-end path than the per-domain method, but depends on the knowledge of both the domain in which the destination is located and the sequence of domains toward the destination. This information is described in [RFC5441] as being known a priori. Clearly, however, information is not always known a priori, and it may be hard for the PCE that serves the source PCC to discover the necessary details. While there are several approaches to solving the question of establishing the domain sequence (for example, BRPC trial and error or H-PCE [RFC6805]), none of them addresses the issue of determining where the destination lies.

One argument that is often made is that an end-to-end connection expressed as an LSP is a feature of a service agreement between source and destination. If that is the case, it is argued, it stands to reason that the location of the destination must be known to the source node in the same way that the source has determined the IP address of the destination. Presumably, this would be through a commercial process or an administrative protocol.

[RFC4974] introduced the concept of Calls and Connections for LSPs. A Call does not provide the actual connectivity for transmitting user traffic, but builds a relationship that will allow subsequent Connections to be made. A Call might be considered an agreement to support an end-to-end LSP that is made between the endpoint nodes. Call messages are sent and routed as normal IP messages, so the sender does not need to know the location of the destination.

Furthermore, Call requests are responded, and the Call Response can carry information (such as the identity of the domain containing the destination) for use by Call initiator. Thus, the use of GMPLS Calls might provide a mechanism to discover destination’s location.

8. Who Runs or Owns a Parent PCE?

A parent PCE [RFC6805] is responsible for selecting inter-domain path by coordinating with child PCEs and maintaining a domain topology map.

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. Note that the H-PCE architecture does not require disclosure of internals of a child domain to the parent PCE. Thus, there is ample scope for a parent PCE to be run by one of the connected service providers or by a third party without compromising commercial issues. In fact, each service provider could run its own parent PCE while allowing its child PCEs to be contacted by outsider parent PCEs according to configured policy and security.

9. How Do I Find My Parent PCE?

[RFC6805] specifies that a child PCE must be configured with the address of its parent PCE in order for it to interact with its parent PCE. There is no scope for parent PCEs to advertise their presence; however, there is potential for directory systems (such as DNS [RFC4848] as used in the ALTO discovery function [ALTO-SERVER-DISC]) to be used as described in Section 4.

According to [RFC6805], note that the child PCE must also be authorized to peer with the parent PCE. This is discussed from the viewpoint of the parent PCE in Section 10. The child PCE may need to participate in a key distribution protocol in order to properly authenticate its identity to the parent PCE.

10. How Do I Find My Child PCEs?

Within the hierarchical PCE framework [RFC6805], the parent PCE must only accept path computation requests from authorized child PCEs. If a parent PCE receives a request from an unauthorized child PCE, the request should be dropped.

This requires a parent PCE to 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.

11. How Is the Parent PCE Domain Topology Built?

The parent PCE maintains a domain topology map of the child domains and their interconnectivity. This map does not include any visibility into the child domains. 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 TED for the parent domain in the same way that any PCE does. The nodes in the parent domain will be abstractions of the child domains (connected by real or virtual TE links), but the parent domain may also include real nodes and links.

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. However, note that in some configurations (for example, collections of small optical domains) a separate instance of a routing protocol (probably an IGP) may be run within the parent domain to advertise the domain interconnectivity. Additionally, since inter-domain TE links can be advertised by the IGPs operating in the child domains, this information could be exported to the parent PCE either by the child PCEs or using a north-bound export mechanism such as BGP-LS [LS-DISTRIB].

12. Does H-PCE Solve the Internet?

The model described in [RFC6805] introduced a hierarchical relationship between domains. It is applicable to environments with small groups of domains where visibility from the ingress LSRs is limited. Applying the hierarchical PCE model to large groups of domains such as the Internet is not considered feasible or desirable.

This does open up a harder question: how many domains can be handled by an H-PCE system? In other words: what is a small group of domains? The answer is not clear and might be "I know it when I see it." At the moment, a rough guide might be around 20 domains as a maximum.

An associated question would be: how many hierarchy levels can be handled by H-PCE? Architecturally, the answer is that there is no limit, but it is hard to construct practical examples where more than two or possibly three levels are needed.

13. What are Sticky Resources?

When a PCE computes a path, it has a reasonable idea that an LSP will be set up and that resources will be allocated within the network. If the arrival rate of computation requests is faster than the LSP setup rate combined with the IGP convergence time, it is quite possible that the PCE will perform its next computation before the TED has been updated to reflect the setup of the previous LSP. This can result in LSP setup failures if there is contention for resources. The likelihood of this problem is particularly high during recovery from network failures when a large number of LSPs might need new paths.

A PCE may choose to make a provisional assignment of the resources that would be needed for an LSP and to reduce the available resources in its TED so that the problem is mitigated. Such resources are informally known as "sticky resources".

Note that using sticky resources introduces a number of other problems that can make managing the TED difficult. For example:

- When the TED is updated as a result of new information from the IGP, how does the PCE know whether the reduction in available resources is due to the successful setup of the LSP for which it is holding sticky resources or due to some other network event (such as the setup of another LSP)? This problem may be particularly evident if there are multiple PCEs that do not synchronize their sticky resources or if not all LSPs utilize PCE computation.

- When LSP setup fails, how are the sticky resources released? Since the PCE doesn’t know about the failure of the LSP setup, it needs some other mechanism to release them.

- What happens if a path computation was made only to investigate the potential for an LSP but not to actually set one up?

- What if the path used by the LSP does not match that provided by the PCE (for example, because the control plane routes around some problem)?

Some of these issues can be mitigated by using a Stateful PCE (see Section 14) or by timers.

14. What Is a Stateful PCE for?

A Stateless PCE can perform path computations that take into account the existence of other LSPs if the paths of those LSPs are supplied on the computation request. This function can be particularly useful when arranging protection paths so that a working and protection LSP do not share any links or nodes. It can also be used when a group of LSPs are to be reoptimized at the same time in the process known as Global Concurrent Optimization (GCO) [RFC5557].

However, this mechanism can be quite a burden on the protocol messages, especially when large numbers of LSP paths need to be reported.

A Stateful PCE [STATEFUL-PCE] maintains a database of LSPs (the LSP-DB) that are active in the network, i.e., have been provisioned such that they use network resources although they might or might not be carrying traffic. This database allows a PCC to refer to an LSP using only its identifier -- all other details can be retrieved by the PCE from the LSP-DB.

A Stateful PCE can use the LSP-DB for many other functions, such as balancing the distribution of LSPs in the network. Furthermore, the PCE can correlate LSPs with network resource availability placing new LSPs more cleverly.

A Stateful PCE that is also an Active PCE (see Section 17) can respond to changes in network resource availability and predicted demands to reroute LSPs that it knows about.

Section 20 offers a brief comparison of the different modes of PCE with reference to stateful and stateless PCE.

15. How Is the LSP-DB Built?

The LSP-DB contains information about the LSPs that are active in the network, as mentioned in Section 14. This state information can be constructed by the PCE from information it receives from a number of sources including from provisioning tools and from the network, but no matter how the information is gleaned, a Stateful PCE needs to synchronize its LSP-DB with the state in the network. Just as described in Section 13, the PCE cannot rely on knowledge about previous computations it has made, but it must find out the actual LSPs in the network.

A simple solution is for all ingress LSRs to report all LSPs to the PCE as they are set up, modified, or torn down. Since PCEP already has the facility to fully describe LSP routes and resources in the protocol messages, this is not a difficult problem, and the LSP State Report (PCRpt) message has been defined for this purpose [STATEFUL-PCE].

The situation can be more complex, however, if there are ingress LSRs that do not support PCEP, support PCEP but not the PCRpt, or that are unaware of the requirement to report LSPs to the PCE. This might happen if the LSRs are able to compute paths themselves or if they receive LSP setup instructions with pre-computed paths from an NMS.

An alternative approach is to note that any LSR on the path of an LSP can probably see the whole path (through the Record Route object in RSVP-TE signaling [RFC3209]) and knows the bandwidth reserved for the LSP. Thus, any LSR could report the LSP to the PCE, noting that it will not hurt (beyond additional message processing and potential overload of the PCE or the network) for the LSP to be reported multiple times because it is clearly identified. In fact, this would also provide a cross-check mechanism.

Nevertheless, it is possible that some LSPs will traverse only LSRs that are not aware of the PCE’s need to learn LSP state and build an LSP-DB. In these cases, the stateful PCE must either only have limited knowledge of the LSPs in the network or must learn about LSPs through some other mechanism (such as reading the MPLS and GMPLS MIB modules [RFC3812] [RFC4802]).

Ultimately, there may be no substitute for all LSRs being aware of Stateful PCEs and able to respond to requests for reports on all LSPs that they know about. This will allow a Stateful PCE to build its LSP-DB from scratch (which it may need to do at start of day) and to verify its LSP-DB against the network (which may be important if the PCE has suffered some form of outage).

16. How Do Redundant Stateful PCEs Synchronize State?

It is important that two PCEs operating in a network have similar views of the available resources. That is, they should have the same or substantially similar TEDs. This is easy to achieve either by building the TEDs from the network in the same way or by one PCE synchronizing its TED to the other PCE using a TED export protocol such as BGP-LS [LS-DISTRIB] or the Network Configuration Protocol (NETCONF) [RFC6241] (see Section 6).

Synchronizing the LSP-DB can be a more complicated issue. As described in Section 15, building the LSP-DB can be an involved process, so it would be best to not have multiple PCEs each trying to build an LSP-DB from the network. However, it is still important that where multiple PCEs operate in the network (either as distributed PCEs or with one acting as a backup for the other), their LSP-DBs are kept synchronized.

Thus, there is likely to be a need for a protocol mechanism for one PCE to update its LSP-DB with that of another PCE. This is no different from any other database-synchronization problem and could use existing mechanisms or a new protocol. Note, however, that in the case of distributed PCEs that are also Active PCEs (see Section 17), each PCE will be creating entries in its own LSP-DB; so, the synchronization of databases must be incremental and bidirectional, not just simply a database dump.

It may be helpful to clarify the word "redundant" in the context of this question. One interpretation is that a redundant PCE exists solely as a backup such that it only performs a function in the network in the event of a failure of the primary PCE. This seems like a waste of expensive resources, and it would make more sense for the redundant PCE to take its share of computation load all the time. However, that scenario of two (or more) active PCEs creates exactly the state synchronization issue described above.

Various deployment options have been suggested where one PCE serves a set of PCCs as the primary computation server, and only addresses requests from other PCCs in the event of the failure of some other PCE; however, this mode of operation still raises questions about the need for synchronized state even in non-failure scenarios if the LSPs that will be computed by the different PCEs may traverse the same network resources.

17. What Is an Active PCE? What Is a Passive PCE?

A Passive PCE is one that only responds to path computation requests. It takes no autonomous actions. A Passive PCE may be stateless or stateful.

An Active PCE is one that issues provisioning "recommendations" to the network. These recommendations may be new routes for existing LSPs or routes for new LSPs (that is, an Active PCE may recommend the instantiation of new LSPs). An Active PCE may be stateless or stateful, but in order for it to reroute existing LSPs effectively, it is likely to hold state for at least those LSPs that it will reroute.

Many people consider that the PCE, itself, cannot be Active. That is, they hold that the PCE's function is purely to compute paths. In that worldview, the "Active PCE" is actually the combination of a normal, passive PCE and an additional architectural component responsible for issuing commands or recommendations to the network.

In some configurations, the VNTM discussed in Sections 21 and 22 provides this additional component.

Section 20 offers a brief comparison of the different modes of PCE with reference to passive and active PCE.

18. What is LSP Delegation?

LSP delegation [STATEFUL-PCE] is the process where a PCC (usually an ingress LSR) passes responsibility for triggering updates to the attributes of an LSP (such as bandwidth or path) to the PCE. In this case, the PCE would need to be both Stateful and Active.

LSP delegation allows an LSP to be set up under the control of the ingress LSR potentially using the services of a PCE. Once the LSP has been set up, the LSR (a PCC) tells the PCE about the LSP by providing details of the path and resources used. It delegates responsibility for the LSP to the PCE so that the PCE can make adjustments to the LSP as dictated by changes to the TED and the policies in force at the PCE. The PCE makes the adjustments by sending a new path to the LSR with the instruction/recommendation that the LSP be re-signaled.

There may be some debate over whether the PCE "owns" the LSP after delegation. That is, if the PCE supplies a new path, is the ingress LSR required to act or can it take the information "under advisement"? It may be too soon to answer this question definitively; however, there is certainly an expectation that the LSR will act on the advice it receives. A comparison may be drawn with a visit to the doctor: the doctor has an expectation that the patient will take the medicine, but the patient has free will.

It is important, however, to distinguish between an LSP established within the network and subsequently delegated to a PCE and an LSP that was established as the result of an Active PCE's recommendation for LSP instantiation.

Section 20 offers a brief comparison of the different modes of PCE with reference to LSP delegation.

19. Is an Active PCE with LSP Delegation Just a Fancy NMS?

In many ways the answer here is "yes". But the PCE architecture forms part of a new way of looking at network operation and management. In this new view, the network operation is more dynamic and under the control of software applications without direct intervention from operators. This is not to say that the operator has no say in how their network runs, but it does mean that the operator sets policies (see Section 24) and that new components (such as an Active PCE) are responsible for acting on those policies to dynamically control the network.

There is a subtle distinction between an NMS and an Active PCE with LSP delegation. An NMS is in control of the LSPs in the network and can command that they are set up, modified, or torn down. An Active PCE can only make suggestions about LSPs that have been delegated to the PCE by a PCC, or make recommendations for the instantiation of new LSPs.

For more details, see the discussion of an architecture for Application-Based Network Operation (ABNO) in [NET-OPS]

20. Comparison of Stateless and Stateful PCE

Table 1 shows a comparison of stateless and stateful PCEs to show how they how might be instantiated as passive or active PCEs with or without control of LSPs. The terms used relate to the concepts introduced in the previous sections. The entries in the table refer to the notes that follow.

	Stateless	Stateful
Passive	1	2
Active delegated LSPs	3	4
Active suggest new LSPs	5	6
Active instantiate LSPs	7	7

Notes:

1. Passive is the normal mode for a stateless PCE.
2. A passive mode stateful PCE may have value for more complex environments and for computing protected services.
3. Delegation of LSPs to a stateless PCE is relatively pointless, but could add value at moment of delegation.
4. This is the normal mode for a stateful PCE.
5. There is only marginal potential for a stateless PCE to recommend new LSPs because without a view of existing LSPs, the PCE cannot determine when new ones might be needed.
6. This mode has potential for recommending the instantiation of new LSPs.
7. These modes are out of scope for PCE as currently described. That is, the PCE can recommend instantiation, but cannot actually instantiate the LSPs.

Table 1 : Comparing Stateless and Stateful PCE

21. How Does a PCE Work with a Virtual Network Topology?

A Virtual Network Topology (VNT) is described in [RFC4397] as a set of Hierarchical LSPs that is created (or could be created) in a particular network layer to provide network flexibility (data links) in other layers. Thus, the TE topology of a network can be constructed from TE links that are simply data links, from TE links that are supported by LSPs in another layer of the network, or from TE links that could be supported by LSPs ("potential LSPs") that would be set up on demand in another network layer. This third type of TE link is known as a Virtual TE Link in [RFC5212].

[RFC5212] also gives a more detailed explanation of a VNT, and it should be noted that the network topology in a packet network could be supported by LSPs in a number of different lower-layer networks. For example, the TE links in the packet network could be achieved by connections (LSPs) in underlying Synchronous Optical Network or Synchronous Digital Hierarchy (SONET/SDH) and photonic networks. Furthermore, because of the hierarchical nature of MPLS, the TE links in a packet network may be achieved by setting up packet LSPs in the same packet network.

A PCE obviously works with the TED that contains information about the TE links in the network. Those links may be already established or may be virtual TE links. In a simple TED, there is no distinction between the types of TE link; however, there may be advantages to selecting TE links that are based on real data links over those based on dynamic LSPs in lower layers because the data links may be more stable. Conversely, the TE links based on dynamic LSPs may be able to be repaired dynamically giving better resilience. Similarly, a PCE may prefer to select a TE link that is supported by a data link or existing LSP in preference to using a virtual TE link because the latter may need to be set up (taking time) and the setup could potentially fail. Thus, a PCE might want to employ additional metrics or indicators to help it view the TED and select the right path for LSPs.

If a PCE uses a virtual TE link, then some action will be needed to establish the LSP that supports that link. Some models (such as that in [RFC5212]) trigger the setup of the lower-layer LSPs on-demand during the signaling of the upper-layer LSP (i.e., when the upper layer comes to use the virtual TE link, the upper-layer signaling is paused and the lower-layer LSP is established). Another view, described in [RFC5623], is that when the PCE computes a path that will use a virtual TE link, it should trigger the setup of the lower-layer LSP to properly create the TE link so that the path it returns will be sure to be viable. This latter mode of operation can be extended to allow the PCE to spot the need for additional TE links and to trigger LSPs in lower layers in order to create those links.

Of course, such "interference" in a lower-layer network by a PCE responsible for a higher-layer network depends heavily on policy. In order to make a clean architectural separation and to facilitate proper policy control, [RFC5623] introduces the Virtual Network Topology Manager (VNTM) as a functional element that manages and controls the VNT. [RFC5623] notes that the PCE and VNT Manager are distinct functional elements that may or may not be collocated. indeed, it should be noted that there will be a PCE for the upper layer, and a PCE for each lower layer, and a VNTM responsible for coordinating between the PCEs and for triggering LSP setup in the lower layers. Therefore, the combination of all of the PCEs and the VNTM produces functionally similar to an Active, multi-layer PCE.

See [TE-INFO] for additional discussion of the construction of networks using virtual and potential links.

22. How Does PCE Communicate with VNTM

The VNTM described in Section 21 and [RFC5623] has several interfaces (see also [NET-OPS]).

- In order to make decisions on whether to create new TE links, the VNTM needs to learn from the upper-layer PCE about resource shortages and the need for additional TE links. It can then make policy-based decisions to determine whether to create new TE links and how to support them through existing or new LSPs.

- The VNTM will need to coordinate with the PCEs in the lower layers, but this is simply a normal use of PCEP.

- The VNTM will need to issue provisioning requests/commands (via the Provisioning Manager described in [NET-OPS]) to the lower-layer networks to cause LSPs to be set up to act as TE links in the higher layer network. A number of potential protocols exist for this function as described in [NET-OPS], but it should be noted that it makes a lot of sense for this interface to be the same as that used by an Active PCE when providing paths to the network.

23. How Does Service Scheduling and Calendering Work?

LSP scheduling or calendaring is a process where LSPs are planned ahead of time, and they are only set up when needed. The challenge here is to ensure that the resources needed by an LSP and that were available when the LSP’s path was computed are still available when the LSP needs to be set up. This needs to be achieved using a mechanism that allows those resources to be used in the meantime.

Previous discussion of this topic has suggested that LSPs should be pre-signaled so that each LSR along the path could make a "temporal reservation" of resources. But this approach can become very complicated requiring each network node to store multi-dimensional state.

Conversely, a centralized database of resources and LSPs (such as the database maintained by a Stateful PCE) can be enhanced with a time-based booking system. If the PCE is also Active, then when the time comes for the LSP to be set up (or later, when it is to be torn down), the PCE can issue recommendations to the network.

In a busy network (and why would one bother with a scheduling service in a network that is not busy?), it should be noted that the computation algorithm can be quite complex. It may also be necessary to reposition existing or planned LSPs as new bookings arrive.

Furthermore, the booking database that contains both the scheduled LSPs and their impact on the network resources can become quite large. A very important factor in the size of the active database (depending on implementation) may be the timeslices that are available in the calendering process.

24. Where Does Policy Fit In?

Policy is critical to the operation of a network. In a PCE context, it provides control and management of how a PCE selects network resources for use by different PCEs.

[RFC5394] introduced the concept of PCE-based policy-enabled path computation. It is based on the Policy Core Information Model (PCIM) [RFC3060] as extended by [RFC3460], and provides a framework for supporting path computation policy.

Policy enters into all aspects of the use of a PCE starting from the very decision to use a PCE to off-load computation function from the LSRs.

- Each PCC must select which computations will be delegated to a PCE.

- Each PCC must select which PCEs it will use.

- Each PCE must determine which PCCs are allowed to use its services and for what computations.

- The PCE must determine how to collect the information in its TED, who to trust for that information, and how to refresh/update the information.

- Each PCE must determine which objective functions and which algorithms to apply.

- Inter-domain (and particularly H-PCE) computations will need to be sensitive to commercial and reliability information about domains and their interactions.

- Stateful PCEs must determine what state to hold, when to refresh it, and which network elements to trust for the supply of the state information.

- An Active PCE must have a policy relationship with its LSRs to determine which LSPs can be modified or triggered, and what LSP delegation is supported.

- Multi-layer interactions (especially those using virtual or dynamic TE links) must provide policy control to stop server layer LSPs (which are fat and expensive by definition) from being set up on a whim to address micro-flows or speculative computations in higher layers.

- A PCE may supply, along with a computed path, policy information that should be signaled during LSP setup for use by the LSRs along the path.

It may be seen, therefore, that a PCE is substantially a policy engine that computes paths. It should also be noted that the work of the PCE can be substantially controlled by configured policy in a way that will reduce the options available to the PCC, but also significantly reduce the need for the use of optional parameters in the PCEP messages.

25. Does PCE Play a Role in SDN?

Software-Defined Networking (SDN) is the latest shiny thing in networking. It refers to a separation between the control elements and the forwarding components so that software running in a centralized system called a controller, can act to program the devices in the network to behave in specific ways.

A required element in an SDN architecture is a component that plans how the network resources will be used and how the devices will be programmed. It is possible to view this component as performing specific computations to place flows within the network given knowledge of the availability of network resources, how other forwarding devices are programmed, and the way that other flows are routed. This, it may be concluded, is the same function that a PCE might offer in a network operated using a dynamic control plane. Thus, a PCE could form part of the infrastructure for an SDN.

A view of how PCE integrates into a wider network control system including SDN is presented in [NET-OPS].

26. Security Considerations

The use of a PCE-based architecture and subsequent impact on network security must, itself, be considered in the context of existing routing and signaling protocols and techniques. The nature of multi-domain network scenarios and establishment of relationships between PCCs and PCEs may increase the vulnerability of the network to security attacks. However, this informational document does not define any new protocol elements or mechanism. As such, it does not introduce any new security issues and security is deemed to be a "previously answered question" even if the answers previously supplied are not perfect. Previous PCE RFCs have given some attention to security concerns in the use of PCE (RFC 4655), PCE discovery (RFC 4674, RFC 5088, and RFC 5089), and PCEP (RFC 4657 and RFC 5440).

It is worth noting that PCEP operates over TCP. An analysis of the security issues for routing protocols that use TCP (including PCEP) is provided in [RFC6952], while [PCE-PCEPS] discusses an experimental approach to provide secure transport for PCEP.

A number of the questions raised and answered in this document should be given consideration in the light of security requirements. Some of these are called out explicitly (Sections 8 and 10), but attention should also be paid to security in all aspects of the use of PCE. For example:

- Topology and other information about the network needs to be kept private and protected from modification or forgery. That means that access to the TED, LSP-DB, etc., needs to be secured and that mechanisms used to gather topology and other information (Sections 2, 11, 14, and 15) need to include security.

- PCE discovery (Sections 4, 5, 9, and 10) needs to protect against impersonation or misconfiguration so that PCCs know that they are getting correct paths and so that PCEs know that they are only serving legitimate computation requests.

- Synchronization of information and state between PCEs (Sections 6 and 16) is subject to the same security requirements in that the information exchanged is sensitive and needs to be protected against interception and modification.

- PCE computes paths for components that may provision the network. Those component are responsible for the security of the provisioning mechanisms, however, if PCE operates as a provisioning protocol (Sections 17, 18, 19, and 25).

- A PCE may also need to interface with other network components (Sections 19, 21, 22, and 25). Those communications, if external to an implementation, also need to be secure.

27. References

27.1. Normative References

[RFC4655] Farrel, A., Vasseur, J.-P., and J. Ash, "A Path Computation Element (PCE)-Based Architecture", RFC 4655, August 2006, <http://www.rfc-editor.org/info/rfc4655>.

[RFC5440] Vasseur, JP., Ed., and JL. Le Roux, Ed., "Path Computation Element (PCE) Communication Protocol (PCEP)", RFC 5440, March 2009, <http://www.rfc-editor.org/info/rfc5440>.

[RFC5623] Oki, E., Takeda, T., Le Roux, JL., and A. Farrel, "Framework for PCE-Based Inter-Layer MPLS and GMPLS Traffic Engineering", RFC 5623, September 2009, <http://www.rfc-editor.org/info/rfc5623>.

[RFC6805] King, D., Ed., and A. Farrel, Ed., "The Application of the Path Computation Element Architecture to the Determination of a Sequence of Domains in MPLS and GMPLS", RFC 6805, November 2012, <http://www.rfc-editor.org/info/rfc6805>.

27.2. Informative References

[ALTO-SERVER-DISC] Kiesel, S., Stiemerling, M., Schwan, N., Scharf, M., and H. Song, "ALTO Server Discovery", Work in Progress, draft-ietf-alto-server-discovery-10, September 2013.

[LS-DISTRIB] Gredler, H., Medved, J., Previdi, S., Farrel, A., and S. Ray, "North-Bound Distribution of Link-State and TE Information using BGP", Work in Progress, draft-ietf-idr-ls-distribution-06, September 2014.

[NET-OPS] King, D., and A. Farrel, "A PCE-based Architecture for Application-based Network Operations", Work in Progress, draft-farrkingel-pce-abno-architecture-13, October 2014.

[PCE-PCEPS] Lopez, D., Gonzalez de Dios, O., Wu, Q., and D. Dhody, "Secure Transport for PCEP", Work in Progress, draft-ietf-pce-pceps-02, October 2014.

[RFC3060] Moore, B., Ellesson, E., Strassner, J., and A. Westerinen, "Policy Core Information Model -- Version 1 Specification", RFC 3060, February 2001, <http://www.rfc-editor.org/info/rfc3060>.

[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, <http://www.rfc-editor.org/info/rfc3209>.

[RFC3460] Moore, B., Ed., "Policy Core Information Model (PCIM) Extensions", RFC 3460, January 2003 <http://www.rfc-editor.org/info/rfc3460>.

[RFC3630] Katz, D., Kompella, K., and D. Yeung, "Traffic Engineering (TE) Extensions to OSPF Version 2", RFC 3630, September 2003, <http://www.rfc-editor.org/info/rfc3630>.

[RFC3812] Srinivasan, C., Viswanathan, A., and T. Nadeau, "Multiprotocol Label Switching (MPLS) Traffic Engineering (TE) Management Information Base (MIB)", RFC 3812, June 2004, <http://www.rfc-editor.org/info/rfc3812>.

[RFC4203] Kompella, K., Ed., and Y. Rekhter, Ed., "OSPF Extensions in Support of Generalized Multi-Protocol Label Switching (GMPLS)", RFC 4203, October 2005, <http://www.rfc-editor.org/info/rfc4203>.

[RFC4397] Bryskin, I. and A. Farrel, "A Lexicography for the Interpretation of Generalized Multiprotocol Label Switching (GMPLS) Terminology within the Context of the ITU-T's Automatically Switched Optical Network (ASON) Architecture", RFC 4397, February 2006, <http://www.rfc-editor.org/info/rfc4397>.

[RFC4657] Ash, J., Ed., and J. Le Roux, Ed., "Path Computation Element (PCE) Communication Protocol Generic Requirements", RFC 4657, September 2006, <http://www.rfc-editor.org/info/rfc4657>.

[RFC4674] Le Roux, J., Ed., "Requirements for Path Computation Element (PCE) Discovery", RFC 4674, October 2006, <http://www.rfc-editor.org/info/rfc4674>.

[RFC4726] Farrel, A., Vasseur, J.-P., and A. Ayyangar, "A Framework for Inter-Domain Multiprotocol Label Switching Traffic Engineering", RFC 4726, November 2006, <http://www.rfc-editor.org/info/rfc4726>.

[RFC4802] Nadeau, T., Ed., and A. Farrel, Ed., "Generalized Multiprotocol Label Switching (GMPLS) Traffic Engineering Management Information Base", RFC 4802, February 2007, <http://www.rfc-editor.org/info/rfc4802>.

[RFC4848] Daigle, L., "Domain-Based Application Service Location Using URIs and the Dynamic Delegation Discovery Service (DDDS)", RFC 4848, April 2007, <http://www.rfc-editor.org/info/rfc4848>.

[RFC4974] Papadimitriou, D. and A. Farrel, "Generalized MPLS (GMPLS) RSVP-TE Signaling Extensions in Support of Calls", RFC 4974, August 2007, <http://www.rfc-editor.org/info/rfc4974>.

[RFC5088] Le Roux, JL., Ed., Vasseur, JP., Ed., Ikejiri, Y., and R. Zhang, "OSPF Protocol Extensions for Path Computation Element (PCE) Discovery", RFC 5088, January 2008, <http://www.rfc-editor.org/info/rfc5088>.

[RFC5089] Le Roux, JL., Ed., Vasseur, JP., Ed., Ikejiri, Y., and R. Zhang, "IS-IS Protocol Extensions for Path Computation Element (PCE) Discovery", RFC 5089, January 2008, <http://www.rfc-editor.org/info/rfc5089>.

[RFC5152] Vasseur, JP., Ed., Ayyangar, A., Ed., and R. Zhang, "A Per-Domain Path Computation Method for Establishing Inter-Domain Traffic Engineering (TE) Label Switched Paths (LSPs)", RFC 5152, February 2008, <http://www.rfc-editor.org/info/rfc5152>.

[RFC5212] Shiomoto, K., Papadimitriou, D., Le Roux, JL., Vigoureux, M., and D. Brungard, "Requirements for GMPLS-Based Multi-Region and Multi-Layer Networks (MRN/MLN)", RFC 5212, July 2008, <http://www.rfc-editor.org/info/rfc5212>.

[RFC5305] Li, T. and H. Smit, "IS-IS Extensions for Traffic Engineering", RFC 5305, October 2008, <http://www.rfc-editor.org/info/rfc5305>.

[RFC5307] Kompella, K., Ed., and Y. Rekhter, Ed., "IS-IS Extensions in Support of Generalized Multi-Protocol Label Switching (GMPLS)", RFC 5307, October 2008, <http://www.rfc-editor.org/info/rfc5307>.

[RFC5394] Bryskin, I., Papadimitriou, D., Berger, L., and J. Ash, "Policy-Enabled Path Computation Framework", RFC 5394, December 2008, <http://www.rfc-editor.org/info/rfc5394>.

[RFC5441] Vasseur, JP., Ed., Zhang, R., Bitar, N., and JL. Le Roux, "A Backward-Recursive PCE-Based Computation (BRPC) Procedure to Compute Shortest Constrained Inter-Domain Traffic Engineering Label Switched Paths", RFC 5441, April 2009, <http://www.rfc-editor.org/info/rfc5441>.

[RFC5557] Lee, Y., Le Roux, JL., King, D., and E. Oki, "Path Computation Element Communication Protocol (PCEP) Requirements and Protocol Extensions in Support of Global Concurrent Optimization", RFC 5557, July 2009, <http://www.rfc-editor.org/info/rfc5557>.

[RFC5986] Thomson, M. and J. Winterbottom, "Discovering the Local Location Information Server (LIS)", RFC 5986, September 2010, <http://www.rfc-editor.org/info/rfc5986>.

[RFC6006] Zhao, Q., Ed., King, D., Ed., Verhaeghe, F., Takeda, T., Ali, Z., and J. Meuric, "Extensions to the Path Computation Element Communication Protocol (PCEP) for Point-to-Multipoint Traffic Engineering Label Switched Paths", RFC 6006, September 2010, <http://www.rfc-editor.org/info/rfc6006>.

[RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, June 2011, <http://www.rfc-editor.org/info/rfc6241>.

[RFC6952] Jethanandani, M., Patel, K., and L. Zheng, "Analysis of BGP, LDP, PCEP, and MSDP Issues According to the Keying and Authentication for Routing Protocols (KARP) Design Guide", RFC 6952, May 2013, <http://www.rfc-editor.org/info/rfc6952>.

[STATEFUL-PCE] Crabbe, E., Minei, I., Medved, J., and R. Varga, "PCEP Extensions for Stateful PCE", Work in Progress, draft-ietf-pce-stateful-pce-10, October 2014.

[TE-INFO] Farrel, A., Ed., Drake, J., Bitar, N., Swallow, G., Ceccarelli, D, and X. Zhang, "Problem Statement and Architecture for Information Exchange Between Interconnected Traffic Engineered Networks", Work in Progress, draft-farrel-interconnected-te-info-exchange-07, September 2014.

Acknowledgements

Thanks for constructive comments go to Fatai Zhang, Oscar Gonzalez de Dios, Xian Zhang, Cyril Margaria, Denis Ovsienko, Ina Minei, Dhruv Dhody, and Qin Wu.

This work was supported in part by the FP-7 IDEALIST project under grant agreement number 317999.

This work received funding from the European Union’s Seventh Framework Programme for research, technological development and demonstration through the PACE project under grant agreement no. 619712.

Authors’ Addresses

Adrian Farrel
Juniper Networks
EMail: adrian@olddog.co.uk

Daniel King
Old Dog Consulting
EMail: daniel@olddog.co.uk