           An Architecture for Use of PCE and the PCE Communication Protocol (PCEP)
                      in a Network with Central Control

Abstract

   The Path Computation Element (PCE) is a core component of Software-
   Defined Networking (SDN) systems. It can compute optimal paths for
   traffic across a network and can also update the paths to reflect
   changes in the network or traffic demands.

   PCE was developed to derive paths for MPLS Label Switched Paths
   (LSPs), which are supplied to the head end of the LSP using the Path
   Computation Element Communication Protocol (PCEP).

   SDN has a broader applicability than signaled MPLS traffic-engineered
   (TE) networks, and the PCE may be used to determine paths in a range
   of use cases including static LSPs, segment routing, Service Function
   Chaining (SFC), and most forms of a routed or switched network. It
   is, therefore, reasonable to consider PCEP as a control protocol for
   use in these environments to allow the PCE to be fully enabled as a
   central controller.

   This document briefly introduces the architecture for PCE as a
   central controller, examines the motivations and applicability for
   PCEP as a control protocol in this environment, and introduces the
   implications for the protocol. A PCE-based central controller can
   simplify the processing of a distributed control plane by blending it
   with elements of SDN and without necessarily completely replacing it.

   This document does not describe use cases in detail and does not
   define protocol extensions: that work is left for other documents.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

   The Path Computation Element (PCE) [RFC4655] was developed to offload
   path computation function from routers in an MPLS traffic-engineered
   network. Since then, the role and function of the PCE has grown to
   cover a number of other uses (such as GMPLS [RFC7025]) and to allow
   delegated control [RFC8231] and PCE-initiated use of network
   resources [RFC8281].

   According to [RFC7399], Software-Defined Networking (SDN) 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 traffic 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 is the function and purpose of a
PCE, and the way that a PCE integrates into a wider network control
system (including an SDN system) is presented in [RFC7491].

In early PCE implementations, where the PCE was used to derive paths
for MPLS Label Switched Paths (LSPs), paths were requested by network
elements (known as Path Computation Clients (PCCs)), and the results
of the path computations were supplied to network elements using the
Path Computation Element Communication Protocol (PCEP) [RFC5440].
This protocol was later extended to allow a PCE to send unsolicited
requests to the network for LSP establishment [RFC8281].

SDN has a far broader applicability than just signaled MPLS or GMPLS
traffic-engineered networks. The PCE component in an SDN system may
be used to determine paths in a wide range of use cases including
static LSPs, segment routing [SR-ARCH], SFC [RFC7665], and indeed any
form of routed or switched network. It is, therefore, reasonable to
consider PCEP as a general southbound control protocol (i.e., a
control protocol for communicating from the central controller to
network elements) for use in these environments to allow the PCE to
be fully enabled as a central controller.

This document introduces the architecture for PCE as a central
controller as an extension of the architecture described in [RFC4655]
and assumes the continued use of PCEP as the protocol used between
PCE and PCC. This document also examines the motivations and
applicability for PCEP as a Southbound Interface (SBI) and introduces
the implications for the protocol used in this way. A PCE-based
central controller can simplify the processing of a distributed
control plane by blending it with elements of SDN and without
necessarily completely replacing it.

This document does not describe use cases in detail and does not
define protocol extensions: that work is left for other documents.

2. Architecture

   The architecture for the use of PCE within centralized control of a
   network is based on the understanding that a PCE can determine how
   connections should be placed and how resources should be used within
   the network, and that the PCE can then cause those connections to be
   established. Figure 1 shows how this control relationship works in a
   network with an active control plane. This is a familiar view for
   those who have read and understood [RFC4655] and [RFC8281].

   In this mode of operation, the central controller is asked to create
   connectivity by a network orchestrator, a service manager, an
   Operations Support System (OSS), a Network Management Station (NMS),
   or some other application. The PCE-based controller computes paths
   with awareness of the network topology, the available resources, and
   the other services supported in the network. This information is
   held in the Traffic Engineering Database (TED) and other databases
   available to the PCE. Then the PCE sends a request using PCEP to one
   of the Network Elements (NEs), and that NE uses a control plane to
   establish the requested connections and reserve the network
   resources.

   Note that other databases (such as an LSP Database (LSP-DB)) might
   also be used, but for simplicity of illustration, just the TED is
   shown.

```
            ----------------------------------------------
           | Orchestrator / Service Manager / OSS / NMS |
            ----------------------------------------------
                      ^
                      |
                      v
               -----------
              |           |    -----
              | PCE-Based |<---| TED |
              | Controller|     -----
              |           |
               -----------
                   ^
             PCEP|
                 v
            ----         ----       ----         ----
           | NE |<--------->| NE |<--->| NE |<--->| NE |
            ----  Signaling  ----       ----         ----
                  Protocol
```
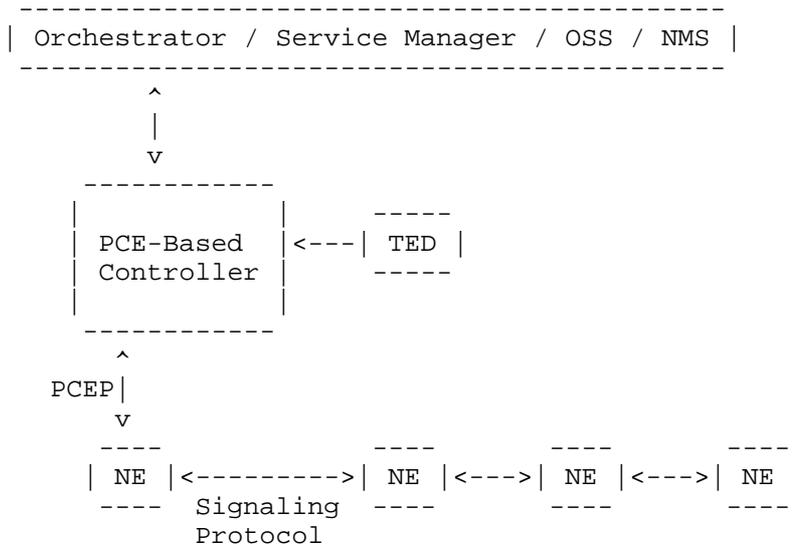
         Figure 1: Architecture for the Central Controller with
                          a Control Plane

Although the architecture shown in Figure 1 represents a form of SDN,
one objective of SDN in some environments is to remove the dependency
on a control plane. A transition architecture toward this goal is
presented in [RFC7491] and is shown in Figure 2. In this case,
services are still requested in the same way, and the PCE-based
controller still requests use of the network using PCEP. The main
difference is that the consumer of the PCEP messages is a network
controller that provisions the resources and instructs the data plane
using an SBI that provides an interface to each NE.

```
            ---------------------------------------------
           | Orchestrator / Service Manager / OSS / NMS |
            ---------------------------------------------
                                 ^
                                 |
                                 v
                      -----------
                     |           |    -----
                     | PCE-Based |<---| TED |
                     | Controller|     -----
                     |           |
                      -----------
                           ^
                           | PCEP
                           v
                      -----------
                     |  Network  |
                     | Controller|
                     /-----------\
                SBI /   ^       ^   \
                   /    |       |    \
                  /     v       v     \
            ----/     ----     ----     \----
           | NE |    | NE |    | NE |    | NE |
            ----      ----      ----      ----
```

           Figure 2: Architecture Including a Network Controller

The approach in Figure 2 delivers the SDN functionality but is overly
complicated and insufficiently flexible.

o The complication is created by the use of two controllers in a
   hierarchical organization and the resultant use of two protocols
   in a southbound direction.

o The lack of flexibility arises from the assumed or required lack
   of a control plane.

   This document describes an architecture that reduces the number of
   components and is flexible to a number of deployment models and use
   cases. In this hybrid approach (shown in Figure 3), the network
   controller is PCE enabled and can also speak PCEP as the SBI (i.e.,
   it can communicate with each node along the path using PCEP). That
   means that the controller can communicate with a conventional
   control-plane-enabled NE using PCEP and can also use the same
   protocol to program individual NEs. In this way, the PCE-based
   controller can control a wider range of networks and deliver many
   different functions as described in Section 3.
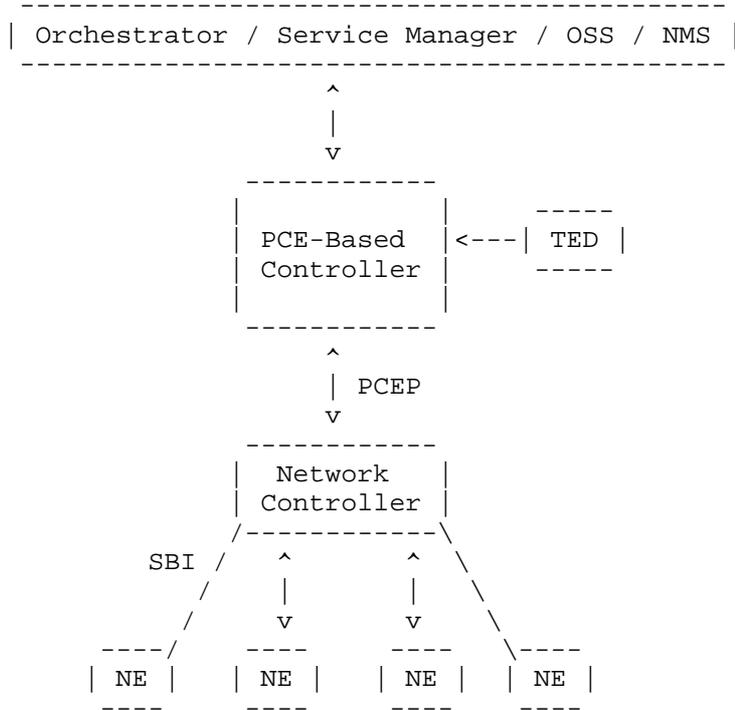
   There will be a trade-off in different application scenarios. In
   some cases, the use of a control plane will simplify deployment (for
   example, by distributing recovery actions), and in other cases, a
   control plane may add operational complexity.

   PCEP is essentially already capable of acting as an SBI and only
   small, use-case-specific modifications to the protocol are needed to
   support this architecture. The implications for the protocol are
   discussed further in Section 4.

```
              --------------------------------------------
              | Orchestrator / Service Manager / OSS / NMS |
              --------------------------------------------
                                 ^
                                 |
                                 v
                       ------------
                       |          |    -----
                       | PCE-Based |<---| TED |
                       | Controller |    -----
                       |          |
                      /------------\
                PCEP /   ^       ^  \
                    /    |       |   \
                   /     v       v    \
                  /    ----    ----     \
                 /    | NE |   | NE |      \
             ----/     ----    ----      \----
             | NE |                      | NE |
             ----                         ----
               ^      ----    ----        ^
             :......>| NE |...| NE |<....:
         Signaling Protocol ----    ----
```
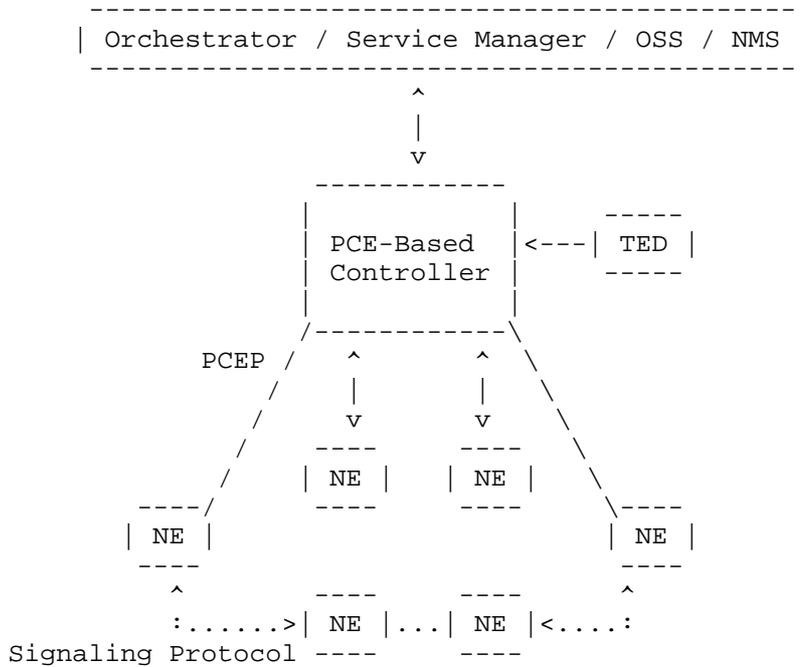
           Figure 3: Architecture for Node-by-Node Central Control

2.1. Resilience and Scaling

   Systems with central controllers are vulnerable to two problems:
   failure of the controller or overload of the controller. These
   concerns are not unique to the use of a PCE-based controller, but
   they need to be addressed in this document before the PCE-based
   controller architecture can be considered for use in all but the
   smallest networks.

   There are three architectural mechanisms that can be applied to
   address these issues. The mechanisms are described separately for
   clarity, but a deployment may use any combination of the approaches.

   For simplicity of illustration, these three approaches are shown in
   the sections that follow without a control plane. However, the
   general, hybrid approach of Figure 3 is applicable in each case.

2.1.1. Partitioned Network

   The first and simplest approach to handling controller overload or
   scalability is to use multiple controllers, each responsible for a
   part of the network. We can call the resultant areas of control
   "domains" [RFC4655].

   This approach is shown in Figure 4. It can clearly address some of
   the scaling and overload concerns since each controller now only has
   responsibility for a subset of the network elements. But this comes
   at a cost because end-to-end connections require coordination between
   the controllers. Furthermore, this technique does not remove the
   concern about a single point-of-failure even if it does reduce the
   impact on the network of the failure of a single controller.

   Note that PCEP is designed to work as a PCE-to-PCE protocol as well
   as a PCE-to-PCC protocol, so it should be possible to use it to
   coordinate between PCE-based controllers in this model.

```
                   --------------------------------------------
                   | Orchestrator / Service Manager / OSS / NMS |
                   --------------------------------------------
                            ^                        ^
                            |                        |
                            v                        v
                    ----------- Coordi-  -----------
        -----       |         | nation  |         |       -----
       | TED |--->| PCE-Based |<-------->| PCE-Based |<---| TED |
        -----       | Controller |        | Controller |    -----
                    |         |     ::  |         |
                 /----------- ::  -----------\
                /    ^       ^  ::  ^       ^    \
               /     |       |  ::  |       |     \
              |      |       |  ::  |       |      |
              v      v       v  ::  v       v      v
            ----   ----    ----  :: ----    ----    ----
           | NE | | NE |  | NE | :: | NE |  | NE |  | NE |
            ----   ----    ----  :: ----    ----    ----
                                  ::
                        Domain 1 :: Domain 2
                                  ::
```
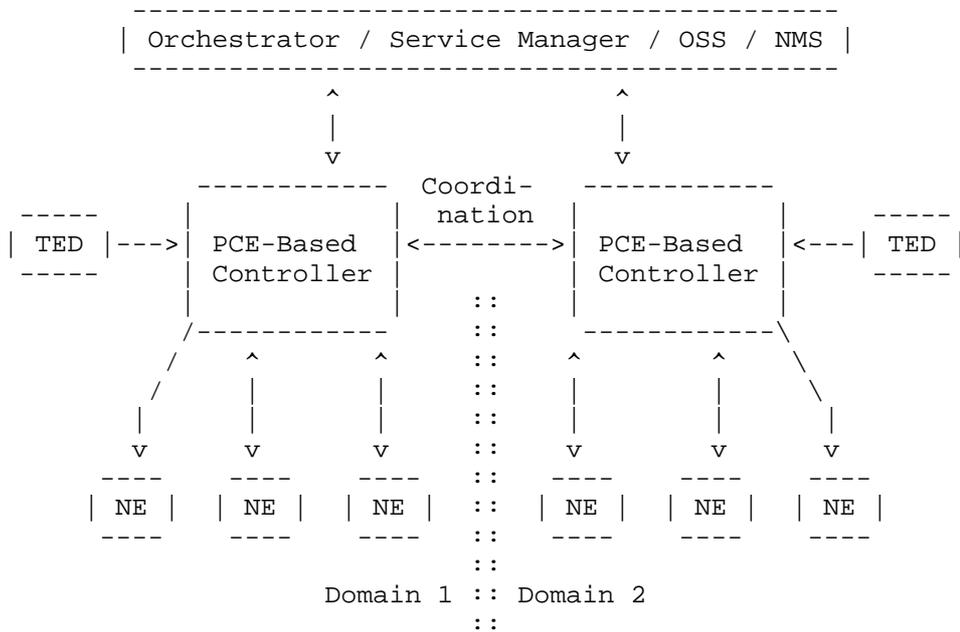
           Figure 4: Multiple Controllers on a Partitioned Network

2.1.2. Multiple Parallel Controllers

   Multiple controllers may be deployed where each controller is capable
   of controlling all of the network elements. Thus, the failure of any
   one controller will not leave the network unmanageable and, in normal
   circumstances, the load can be distributed across the controllers.

   Multiple parallel controllers may be deployed as shown in Figure 5.
   Each controller is capable of controlling all of the network
   elements; thus, the failure of any one controller will not leave the
   network unmanageable, and in normal circumstances, the load can be
   distributed across the controllers. In this model, the orchestrator
   (or any requester) must select a controller to consume its request.

```
                 ---------------------------------------------
                 | Orchestrator / Service Manager / OSS / NMS |
                 ---------------------------------------------
                       ^                            ^
                       |    _____       |
                       |   | Synchronization |      |
                       v   v                 v      v
                  -----------                 -----------
                 |           |   -----       |           |
                 | PCE-Based |<---| TED |--->| PCE-Based |
                 | Controller|   -----       | Controller|
                 |           |__ ...........  |           |
                  -----------\  \_:__      :-----------
                     ^  ^     \  \___:  \  .....:  ^   ^
                     |  | .....:\   \_:___   ..:    :
                     |  |__:___     \___:_  \_:___  :
                     | ....: |   .....: |  ..:   | :
                     | :     | :        | :      | :
                     v v     v v        v v      v v
                    ----    ----       ----     ----
                   | NE |  | NE |     | NE |   | NE |
                    ----    ----       ----     ----
```
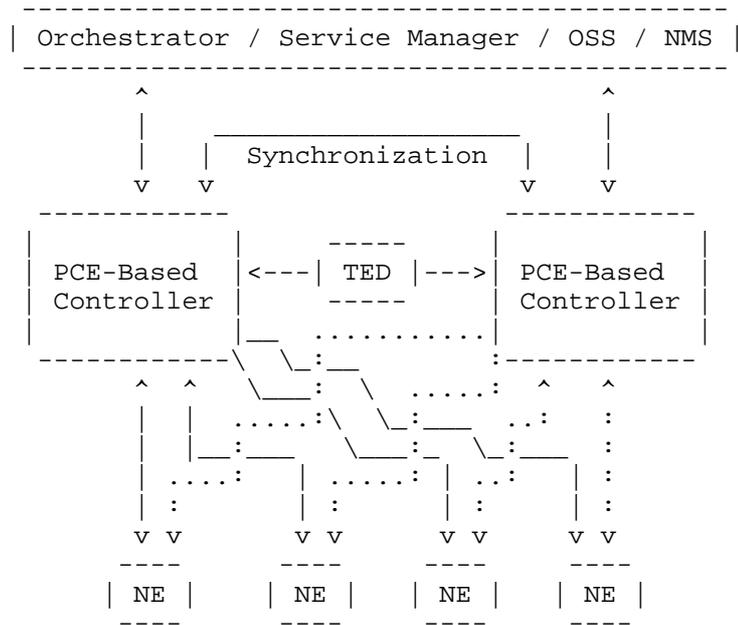
               Figure 5: Multiple Redundant Controllers

An alternate approach is to present the controllers as a "cluster"
that represents itself externally as a single controller as in
Figure 3 but that is actually comprised of multiple controllers. The
size of the cluster may be varied according to the load in the manner
of Network Functions Virtualization (NFV), and the cluster is
responsible for sharing load among the members of the cluster. This
approach is shown in Figure 6.

```
                -----------------------------------------------
              | Orchestrator / Service Manager / OSS / NMS |
                -----------------------------------------------
                                    ^
                                    |
              --------------------------+-----------------------
            | Controller _____|_____           |
            | Cluster    |             |              |           |
            |            |   _____           |           |
            |            |  | Synchronization |        |           |
            |         v  v  |                 |     v  v           |
            |        -----------    -----     -----------          |
            |       | PCE-Based  |<---| TED |--->| PCE-Based  |     |
            |       | Controller |    -----    | Controller |     |
            |       | Instance   |             | Instance   |     |
            |        -----------               -----------        |
            |            ^                          ^              |
            |            |_____|         |
            |                          |                          |
              --------------------------+-----------------------
                  _____|_____
                |         |    |     |         |
                v         v    v     v         v
               ----      ----      ----      ----
              | NE |    | NE |    | NE |    | NE |
               ----      ----      ----      ----
```
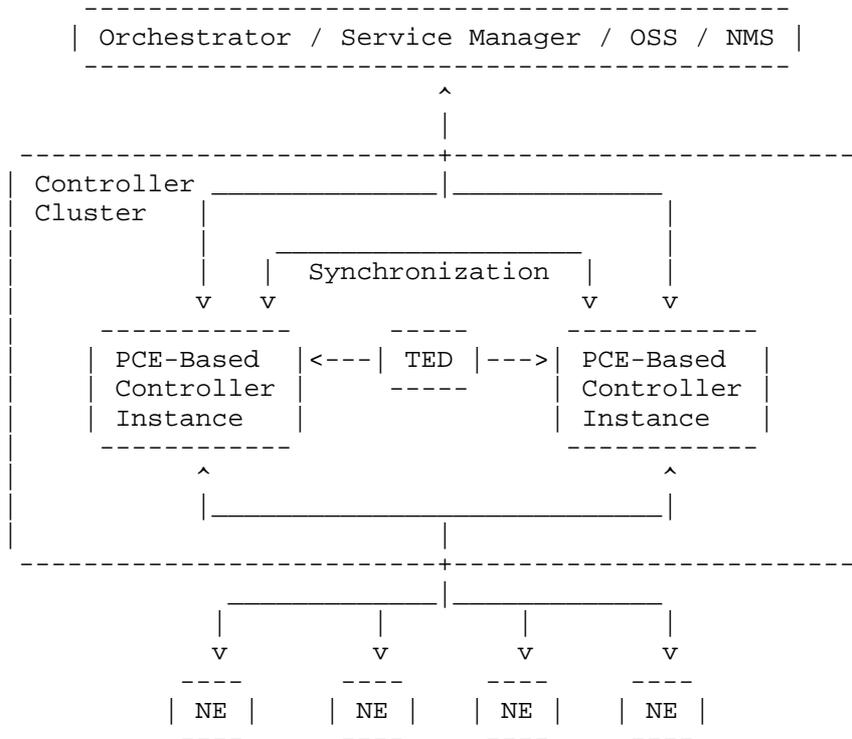
            Figure 6: Multiple Controllers Presented as a Cluster

   To achieve full redundancy and to be able to continue to provide full
   function in the event of a controller failure, the controllers must
   synchronize with each other. This is nominally a simple task if
   there are just two controllers but can actually be quite complex if
   state changes in the network are not to be lost. Furthermore, if
   there are more than two controllers, the synchronization between
   controllers can become a hard problem.

Synchronization issues are often off-loaded as "database
synchronization" problems, because distributed database packages have
already had to address these challenges, or by using a shared
database. In networking, the problem may also be addressed by
collecting the state from the network (effectively using the network
as a database) using normal routing protocols such as OSPF, IS-IS,
and BGP. It should be noted that addressing the synchronization
problem through a shared database may be hiding the issues of
congestion and of a single point of failure: while the controllers
may have been made resilient by allowing redundancy, the shared
database is still a problem, so the whole system is still vulnerable.

2.1.3. Hierarchical Controllers

Figure 7 shows an approach with hierarchical controllers. This
approach was developed for PCEs in [RFC6805] and appears in various
SDN architectures where a "parent PCE", an "orchestrator", or a
"super controller" takes responsibility for a high-level view of the
network before distributing tasks to lower-level PCEs or controllers.

On its own, this approach does little to protect against the failure
of a controller, but it can make significant improvements in loading
and scaling of the individual controllers. It also offers a good way
to support end-to-end connectivity across multiple administrative or
technology-specific domains.

Note that this model can be arbitrarily recursive with a PCE-based
controller being the child of one parent PCE-based controller while
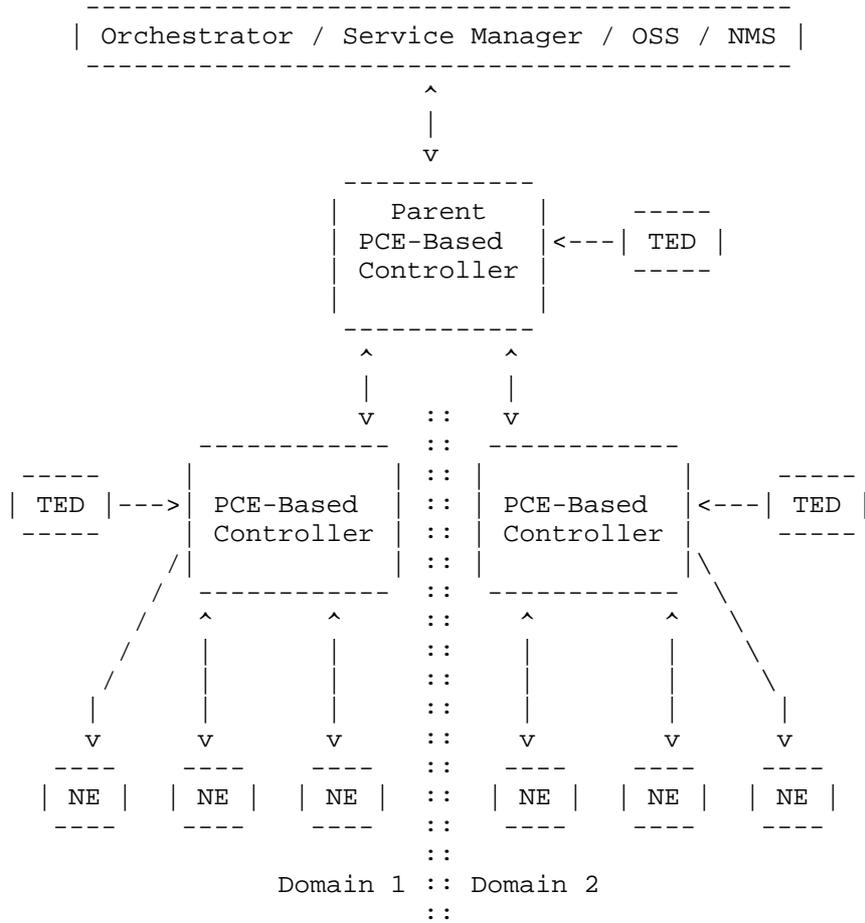acting as the parent of another set of PCE-based controllers.

```
                 -----------------------------------------
                 | Orchestrator / Service Manager / OSS / NMS |
                 -----------------------------------------
                                      ^
                                      |
                                      v
                          ------------
                          |   Parent   |    -----
                          | PCE-Based  |<---| TED |
                          | Controller |    -----
                          |            |
                          ------------
                           ^          ^
                           |          |
                           v    ::    v
                    ------------ :: ------------
            -----   |          | :: |          |    -----
            | TED |--->| PCE-Based | :: | PCE-Based |<---| TED |
            -----   | Controller | :: | Controller |    -----
                   /|          | :: |          |\
                  / ------------ :: ------------ \
                 /  ^        ^  ::  ^        ^  \
                /   |        |  ::  |        |   \
               /    |        |  ::  |        |    \
              |     |        |  ::  |        |     |
              v     v        v  ::  v        v     v
            ----  ----    ----  ::  ----    ----  ----
            | NE | | NE |  | NE |  ::  | NE |  | NE |  | NE |
            ----  ----    ----  ::  ----    ----  ----
                                  ::
                      Domain 1 :: Domain 2
                                  ::
```

                   Figure 7: Hierarchical Controllers

3. Applicability

   This section gives a very high-level introduction to the
   applicability of a PCE-based centralized controller. There is no
   attempt to explain each use case in detail, and the inclusion of a
   use case is not intended to suggest that deploying a PCE-based
   controller is a mandatory or recommended approach. The sections
   below are provided as a stimulus to the discussion of the
   applicability of a PCE-based controller, and it is expected that
   separate documents will be written to develop the use cases in which
   there is interest for implementation and deployment. As described in

   Section 4, specific enhancements to PCEP may be needed for some of
   these use cases, and it is expected that the documents that develop
   each use case will also address any extensions to PCEP.

   The rest of this section is divided into two sub-sections. The first
   approaches the question of applicability from a consideration of the
   network technology. The second looks at the high-level functions
   that can be delivered by using a PCE-based controller.

   As previously mentioned, this section is intended to just make
   suggestions. Thus, the material supplied is very brief. The
   omission of a use case is in no way meant to imply some limit on the
   applicability of PCE-based control.

3.1. Technology-Oriented Applicability

   This section provides a list of use cases based on network
   technology.

3.1.1. Applicability to Control-Plane Operated Networks

   This mode of operation is the common approach for an active, stateful
   PCE to control a traffic-engineered MPLS or GMPLS network [RFC8231].
   Note that the PCE-based controller determines what LSPs are needed
   and where to place them. PCEP is used to instruct the head end of
   each LSP, and the head end signals in the control plane to set up the
   LSP.

   In this mode of operation, the PCE may construct its TED in a number
   of ways as described in [RFC4655], including (but not limited to)
   participating in the IGP or receiving information from a network
   element via BGP-LS [RFC7752].

3.1.2. Static LSPs in MPLS

   Static LSPs are provisioned without the use of a control plane. This
   means that they are established using a management plane or "manual"
   configuration.

   Static LSPs can be provisioned as explicit label instructions at each
   hop on the end-to-end path LSP. Each router along the path must be
   told what label-forwarding instructions to program and what resources
   to reserve. The PCE-based controller keeps a view of the network and
   determines the paths of the end-to-end LSPs just as it does for the
   use case described in Section 3.1.1, but the controller uses PCEP to
   communicate with each router along the path of the end-to-end LSP.
   In this case, the PCE-based controller will take responsibility for
   managing some part of the MPLS label space for each of the routers

that it controls, and it may taker wider responsibility for
partitioning the label space for each router and allocating different
parts for different uses, communicating the ranges to the router
using PCEP.

3.1.3. MPLS Multicast

Multicast LSPs may be provisioned with a control plane or as static
LSPs. No extra considerations apply above those described in
Sections 3.1.1 and 3.1.2 except, of course, to note that the PCE must
also include the instructions about where the LSP branches, i.e.,
where packets must be copied.

3.1.4. Transport SDN

Transport SDN (T-SDN) is the application of SDN techniques to
transport networks. In this respect, a transport network is a
network built from any technology below the IP layer and designed to
carry traffic transparently in a connection-oriented way. Thus, an
MPLS traffic-engineered network is a transport network, although it
is more common to consider technologies such as Time Division
Multiplexing (TDM) and Optical Transport Networks (OTNs) to be
transport networks.

Transport networks may be operated with or without a control plane
and may have point-to-point or point-to-multipoint connections.
Thus, all of the considerations in Sections 3.1.1, 3.1.2, and 3.1.3
apply so that the normal PCEP message allows a PCE-based central
controller to provision a transport network. It is usually the case
that additional technology-specific parameters are needed to
configure the NEs or LSPs in transport networks, such as optical
characteristic. Such parameters will need to be carried in the PCEP
messages: new protocol extensions may be needed, as described, for
example, in [PCEP-WSON-RWA].

3.1.5. Segment Routing

Segment routing is described in [SR-ARCH]. It relies on a series of
forwarding instructions being placed in the header of a packet. At
each hop in the network, a router looks at the first instruction and
may: continue to forward the packet unchanged; strip the top
instruction and forward the packet; or strip the top instruction,
insert some additional instructions, and forward the packet.

The segment routing architecture supports operations that can be used
to steer packet flows in a network, thus providing a form of traffic
engineering. A PCE-based controller can be responsible for computing
the paths for packet flows in a segment routing network, configuring

the forwarding actions on the routers, and telling the edge routers
what instructions to attach to packets as they enter the network.
These last two operations can be achieved using PCEP, and the
PCE-based controller will assume responsibility for managing the
space of labels or path identifiers used to determine how packets are
forwarded.

3.1.6. Service Function Chaining

SFC is described in [RFC7665]. It is the process of directing
traffic in a network such that it passes through specific hardware
devices or virtual machines (known as service function nodes) that
can perform particular desired functions on the traffic. The set of
functions to be performed and the order in which they are to be
performed is known as a service function chain. The chain is
enhanced with the locations at which the service functions are to be
performed to derive a Service Function Path (SFP). Each packet is
marked as belonging to a specific SFP, and that marking lets each
successive service function node know which functions to perform and
to which service function node to send the packet next.

To operate an SFC network, the service function nodes must be
configured to understand the packet markings, and the edge nodes must
be told how to mark packets entering the network. Additionally, it
may be necessary to establish tunnels between service function nodes
to carry the traffic.

Planning an SFC network requires load balancing between service
function nodes and traffic engineering across the network that
connects them. These are operations that can be performed by a
PCE-based controller, and that controller can use PCEP to program the
network and install the service function chains and any required
tunnels.

3.2. High-Level Applicability

This section provides a list of the high-level functions that can be
delivered by using a PCE-based controller.

3.2.1. Traffic Engineering

According to [RFC2702], TE is concerned with performance optimization
of operational networks. In general, it encompasses the application
of technology and scientific principles to the measurement, modeling,
characterization, control of Internet traffic, and application of
such knowledge and techniques to achieve specific performance
objectives.

   From a practical point of view, this involves having an understanding
   of the topology of the network, the characteristics of the nodes and
   links in the network, and the traffic demands and flows across the
   network. It also requires that actions can be taken to ensure that
   traffic follows specific paths through the network.

   PCE was specifically developed to address TE in an MPLS network, so a
   PCE-based controller is well suited to analyze TE problems and supply
   answers that can be installed in the network using PCEP. PCEP can be
   responsible for initiating paths across the network through a control
   plane or for installing state in the network node by node such as in
   a segment-routed network (see Section 3.1.5) or by configuring IGP
   metrics.

3.2.2. Traffic Classification

   Traffic classification is an important part of traffic engineering.
   It is the process of looking at a packet to determine how it should
   be treated as it is forwarded through the network. It applies in
   many scenarios including MPLS traffic engineering (where it
   determines what traffic is forwarded onto which LSPs); segment
   routing (where it is used to select which set of forwarding
   instructions to add to a packet); and SFC (where it indicates along
   which service function path a packet should be forwarded). In
   conjunction with traffic engineering, traffic classification is an
   important enabler for load balancing.

   Traffic classification is closely linked to the computational
   elements of planning for the network functions just listed because it
   determines how traffic load is balanced and distributed through the
   network. Therefore, selecting what traffic classification should be
   performed by a router is an important part of the work done by a
   PCE-based controller.

   Instructions can be passed from the controller to the routers using
   PCEP. These instructions tell the routers how to map traffic to
   paths or connections.

3.2.3. Service Delivery

   Various network services may be offered over a network. These
   include protection services (including end-to-end protection
   [RFC4427], restoration after failure, and fast reroute [RFC4090]);
   Virtual Private Network (VPN) services (such as Layer 3 VPNs
   [RFC4364] or Ethernet VPNs [RFC7432]); or Pseudowires [RFC3985].

      Delivering services over a network in an optimal way requires
      coordination in the way that network resources are allocated to
      support the services. A PCE-based central controller can consider
      the whole network and all components of a service at once when
      planning how to deliver the service. It can then use PCEP to manage
      the network resources and to install the necessary associations
      between those resources.

4. Protocol Implications / Guidance for Solution Developers

      PCEP is a push-pull protocol that is designed to move requests and
      responses between a server (the PCE) and clients (the PCCs, i.e., the
      network elements). In particular, it has a message (the LSP Initiate
      Request (PCInitiate); see [RFC8281]) that can be sent by the PCE to
      install state or cause actions at the PCC and a response message
      (Path Computation State Report (PCRpt)) that is used to confirm the
      request.

      As such, there is an expectation that only relatively minor changes
      to PCEP are required to support the concept of a PCE-based
      controller. The only work expected to be needed is extensions to
      existing PCEP messages to carry additional or specific information
      elements for the individual use cases, which maintain backward
      compatibility and do not impact existing PCEP deployments. [RFC5440]
      already describes how legacy implementations handle unknown protocol
      extensions and how to use the PCEP Open message to indicate support
      for PCEP features. Where possible, consistent with the general
      principles of how protocols are extended, any additions to the
      protocol should be made in a generic way such that they are open to
      use in a range of applications.

      It is anticipated that new documents (such as [PCEP-CONTROLLER]) will
      be produced for each use case dependent on support and demand. Such
      documents will explain the use case and define the necessary protocol
      extensions.

      Protocol extensions could have impact on existing PCEP deployments
      and the interoperability between different implementations. It is
      anticipated that changes of the PCEP protocol or addition of
      information elements could require additional testing to ensure
      interoperability between different PCEP implementations.

      It is reasonable to expect that implementations are able to select a
      subset or profile of the protocol extensions and PCEP features that
      are relevant for the application scenario in which they will be
      deployed. Identification of these profiles should form part of the
      protocol itself so that interoperability can be easily determined and
      testing can be limited to the specific profiles.

Note that protocol mechanisms to handle synchronization of state in
parallel PCE-based controllers will also be required if parallel
controllers are used as described in Section 2.1.2. In [RFC8231],
there is a discussion of mechanisms to achieve PCE state
synchronization.

5. Security Considerations

Security considerations for a PCE-based controller are little
different from those for any other PCE system. That is, the
operation relies heavily on the use and security of PCEP, so
consideration should be given to the security features discussed in
[RFC5440] and the additional mechanisms described in [RFC8253].

It should be observed that the trust model of a network that operates
without a control plane is different from one with a control plane.
The conventional "chain of trust" used with a control plane is
replaced by individual trust relationships between the controller and
each individual NE. This model may be considerably easier to manage,
so it is more likely to be operated with a high level of security.

However, an architecture with a central controller has a central
point of failure, and this is also a security weakness since the
network can be vulnerable to denial-of-service attacks on the
controller. Similarly, the central controller provides a focus for
interception and modification of messages sent to individual NEs. In
short, while the interactions with a PCE-based controller are not
substantially different to those in any other SDN architecture, the
security implications of SDN have not been fully discussed or
described. Therefore, protocol and applicability work-around
solutions for this architecture must take proper account of these
concerns.

It is expected that each new document that is produced for a specific
use case will also include considerations of the security impacts of
the use of a PCE-based central controller on the network type and
services being managed.

6. Manageability Considerations

The architecture described in this document is a management
architecture: the PCE-based controller is a management component that
controls the network through a southbound control protocol (PCEP).

An implementation of a PCE-based controller will require access to
information about the state of the network, its nodes, and its links.
Some of this will be the TED as is normal for a PCE and can be
collected using the mechanisms already in place (such as listening to

   the IGPs, using BGP-LS [RFC7752], or northbound export of
   YANG-encoded data [YANG-TE] from the network elements to the
   controller). More information may be collected in the LSP database
   for stateful PCEs as described in [RFC7399] and [RFC8231].
   Additional information may be needed for other specific use cases and
   will need to be collected and passed to the controller. This may
   require protocol extensions for the mechanisms listed in this
   paragraph.

   The use of different PCEP options and protocol extensions may have an
   impact on interoperability, which is a management issue. As noted in
   Section 4, protocol extensions should be done in a way that makes it
   possible to identify profiles of PCEP to aid interoperability, and
   this will aid deployment and manageability.

   [RFC5440] contains a substantive Manageability Considerations section
   that examines how a PCE-based system and a PCE-enabled system may be
   managed. A MIB module for PCEP was published as [RFC7420], and a
   YANG module for PCEP has also been proposed [YANG-PCEP].

7. IANA Considerations

   This document does not require any IANA actions.

8. References

8.1. Normative References

   [RFC4655] Farrel, A., Vasseur, J., and J. Ash, "A Path Computation
              Element (PCE)-Based Architecture", RFC 4655,
              DOI 10.17487/RFC4655, August 2006,
              <https://www.rfc-editor.org/info/rfc4655>.

   [RFC5440] Vasseur, JP., Ed. and JL. Le Roux, Ed., "Path Computation
              Element (PCE) Communication Protocol (PCEP)", RFC 5440,
              DOI 10.17487/RFC5440, March 2009,
              <https://www.rfc-editor.org/info/rfc5440>.

   [RFC8281] Crabbe, E., Minei, I., Sivabalan, S., and R. Varga, "Path
              Computation Element Communication Protocol (PCEP)
              Extensions for PCE-Initiated LSP Setup in a Stateful PCE
              Model", RFC 8281, DOI 10.17487/RFC8281, December 2017,
              <https://www.rfc-editor.org/info/rfc8281>.

8.2. Informative References

   [PCECC]   Zhao, Q., Li, Z., Khasanov, B., Ke, Z., Fang, L., Zhou,
                  C., Communications, T., Rachitskiy, A., and A. Gulida,
                  "The Use Cases for Using PCE as the Central
                  Controller(PCECC) of LSPs", Work in Progress,
                  draft-zhao-teas-pcecc-use-cases-02, October 2016.

   [PCEP-CONTROLLER]
                  Zhao, Q., Li, Z., Dhody, D., Karunanithi, S., Farrel, A.,
                  and C. Zhou, "PCEP Procedures and Protocol Extensions for
                  Using PCE as a Central Controller (PCECC) of LSPs", Work
                  in Progress, draft-zhao-pce-pcep-extension-for-pce-
                  controller-06, October 2017.

   [PCEP-WSON-RWA]
                  Lee, Y. and R. Casellas, "PCEP Extension for WSON Routing
                  and Wavelength Assignment", Work in Progress,
                  draft-ietf-pce-wson-rwa-ext-07, November 2017.

   [RFC2702]   Awduche, D., Malcolm, J., Agogbua, J., O'Dell, M., and J.
                  McManus, "Requirements for Traffic Engineering Over MPLS",
                  RFC 2702, DOI 10.17487/RFC2702, September 1999,
                  <https://www.rfc-editor.org/info/rfc2702>.

   [RFC3985]   Bryant, S., Ed. and P. Pate, Ed., "Pseudo Wire Emulation
                  Edge-to-Edge (PWE3) Architecture", RFC 3985,
                  DOI 10.17487/RFC3985, March 2005,
                  <https://www.rfc-editor.org/info/rfc3985>.

   [RFC4090]   Pan, P., Ed., Swallow, G., Ed., and A. Atlas, Ed., "Fast
                  Reroute Extensions to RSVP-TE for LSP Tunnels", RFC 4090,
                  DOI 10.17487/RFC4090, May 2005,
                  <https://www.rfc-editor.org/info/rfc4090>.

   [RFC4364]   Rosen, E. and Y. Rekhter, "BGP/MPLS IP Virtual Private
                  Networks (VPNs)", RFC 4364, DOI 10.17487/RFC4364, February
                  2006, <https://www.rfc-editor.org/info/rfc4364>.

   [RFC4427]   Mannie, E., Ed. and D. Papadimitriou, Ed., "Recovery
                  (Protection and Restoration) Terminology for Generalized
                  Multi-Protocol Label Switching (GMPLS)", RFC 4427,
                  DOI 10.17487/RFC4427, March 2006,
                  <https://www.rfc-editor.org/info/rfc4427>.

   [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,
              DOI 10.17487/RFC6805, November 2012,
              <https://www.rfc-editor.org/info/rfc6805>.

   [RFC7025] Otani, T., Ogaki, K., Caviglia, D., Zhang, F., and C.
              Margaria, "Requirements for GMPLS Applications of PCE",
              RFC 7025, DOI 10.17487/RFC7025, September 2013,
              <https://www.rfc-editor.org/info/rfc7025>.

   [RFC7399] Farrel, A. and D. King, "Unanswered Questions in the Path
              Computation Element Architecture", RFC 7399,
              DOI 10.17487/RFC7399, October 2014,
              <https://www.rfc-editor.org/info/rfc7399>.

   [RFC7420] Koushik, A., Stephan, E., Zhao, Q., King, D., and J.
              Hardwick, "Path Computation Element Communication Protocol
              (PCEP) Management Information Base (MIB) Module",
              RFC 7420, DOI 10.17487/RFC7420, December 2014,
              <https://www.rfc-editor.org/info/rfc7420>.

   [RFC7432] Sajassi, A., Ed., Aggarwal, R., Bitar, N., Isaac, A.,
              Uttaro, J., Drake, J., and W. Henderickx, "BGP MPLS-Based
              Ethernet VPN", RFC 7432, DOI 10.17487/RFC7432, February
              2015, <https://www.rfc-editor.org/info/rfc7432>.

   [RFC7491] King, D. and A. Farrel, "A PCE-Based Architecture for
              Application-Based Network Operations", RFC 7491,
              DOI 10.17487/RFC7491, March 2015,
              <https://www.rfc-editor.org/info/rfc7491>.

   [RFC7665] Halpern, J., Ed. and C. Pignataro, Ed., "Service Function
              Chaining (SFC) Architecture", RFC 7665,
              DOI 10.17487/RFC7665, October 2015,
              <https://www.rfc-editor.org/info/rfc7665>.

   [RFC7752] Gredler, H., Ed., Medved, J., Previdi, S., Farrel, A., and
              S. Ray, "North-Bound Distribution of Link-State and
              Traffic Engineering (TE) Information Using BGP", RFC 7752,
              DOI 10.17487/RFC7752, March 2016,
              <https://www.rfc-editor.org/info/rfc7752>.

   [RFC8231] Crabbe, E., Minei, I., Medved, J., and R. Varga, "Path
              Computation Element Communication Protocol (PCEP)
              Extensions for Stateful PCE", RFC 8231,
              DOI 10.17487/RFC8231, September 2017,
              <https://www.rfc-editor.org/info/rfc8231>.

   [RFC8253]   Lopez, D., Gonzalez de Dios, O., Wu, Q., and D. Dhody,
               "PCEPS: Usage of TLS to Provide a Secure Transport for the
               Path Computation Element Communication Protocol (PCEP)",
               RFC 8253, DOI 10.17487/RFC8253, October 2017,
               <https://www.rfc-editor.org/info/rfc8253>.

   [SR-ARCH]   Filsfils, C., Previdi, S., Ginsberg, L., Decraene, B.,
               Litkowski, S., and R. Shakir, "Segment Routing
               Architecture", Work in Progress, draft-ietf-spring-
               segment-routing-13, October 2017.

   [YANG-PCEP]
               Dhody, D., Hardwick, J., Beeram, V., and j.
               jefftant@gmail.com, "A YANG Data Model for Path
               Computation Element Communications Protocol (PCEP)", Work
               in Progress, draft-ietf-pce-pcep-yang-05, June 2017.

   [YANG-TE]   Liu, X., Bryskin, I., Beeram, V., Saad, T., Shah, H., and
               O. Dios, "YANG Data Model for Traffic Engineering (TE)
               Topologies", Work in Progress, draft-ietf-teas-yang-te-
               topo-13, October 2017.

Contributors

    The following people contributed to discussions that led to the
    development of this document:

       Cyril Margaria
       Email: cmargaria@juniper.net

       Sudhir Cheruathur
       Email: scheruathur@juniper.net

       Dhruv Dhody
       Email: dhruv.dhody@huawei.com

       Daniel King
       Email: daniel@olddog.co.uk

       Iftekhar Hussain
       Email: IHussain@infinera.com

       Anurag Sharma
       Email: AnSharma@infinera.com

       Eric Wu
       Email: eric.wu@huawei.com

Authors' Addresses

    Adrian Farrel (editor)
    Juniper Networks

    Email: afarrel@juniper.net

    Quintin Zhao (editor)
    Huawei Technologies
    125 Nagog Technology Park
    Acton, MA 01719
    United States of America

    Email: quintin.zhao@huawei.com

    Robin Li
    Huawei Technologies
    Huawei Bld., No.156 Beiqing Road
    Beijing 100095
    China

    Email: lizhenbin@huawei.com

    Chao Zhou
    Cisco Systems

    Email: chao.zhou@cisco.com