

Internet Engineering Task Force  
Internet-Draft  
Intended status: Informational  
Expires: June 2, 2013

D. King  
Old Dog Consulting  
A. Farrel  
Juniper Networks  
December 2, 2012

## **A PCE-based Architecture for Application-based Network Operations**

[draft-farrkingel-pce-abno-architecture-00.txt](#)

### **Abstract**

Services such as content distribution, distributed databases, or inter-data center connectivity place a set of new requirements on the operation of networks. They need on-demand and application-specific reservation of network connectivity, reliability, and resources (such as bandwidth). An environment that operates to meet this type of requirement is said to have Application-Based Network Operations (ABNO).

ABNO brings together several existing technologies for gathering information about the resources available in a network, for consideration of topologies and how those topologies map to underlying network resources, for requesting path computation, and for provisioning or reserving network resources. Thus, ABNO may be seen as the use of a toolbox of existing components enhanced with a few new elements. The key component within an ABNO is the Path Computation Element (PCE), which can be used for computing paths and is further extended to provide policy enforcement capabilities for ABNO.

This document describes an architecture and framework for ABNO showing how these components fit together. It provides a cookbook of existing technologies to satisfy the architecture and meet the needs of the applications.

### **Status of this Memo**

This Internet-Draft is submitted in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on April 5, 2013.



## Copyright Notice

Copyright (c) 2012 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

<a href="#">1.</a>	<a href="#">Introduction .....</a>	<a href="#">2</a>
<a href="#">1.1</a>	<a href="#">Scope .....</a>	<a href="#">4</a>
<a href="#">2.</a>	<a href="#">Application-based Network Operations (ABNO) .....</a>	<a href="#">4</a>
<a href="#">2.1</a>	<a href="#">Assumptions and Requirements .....</a>	<a href="#">4</a>
<a href="#">2.2</a>	<a href="#">Generic Architecture .....</a>	<a href="#">5</a>
<a href="#">2.2.1</a>	<a href="#">ABNO Components .....</a>	<a href="#">6</a>
<a href="#">2.2.2</a>	<a href="#">ABNO Functional Interfaces .....</a>	<a href="#">10</a>
<a href="#">3.</a>	<a href="#">ABNO Use Cases .....</a>	<a href="#">16</a>
<a href="#">3.1</a>	<a href="#">Inter-AS Connectivity .....</a>	<a href="#">16</a>
<a href="#">3.2</a>	<a href="#">Multi-Layer Networking .....</a>	<a href="#">22</a>
<a href="#">3.3</a>	<a href="#">Bandwidth Scheduling .....</a>	<a href="#">25</a>
<a href="#">3.4</a>	<a href="#">Grooming and Regrooming .....</a>	<a href="#">26</a>
<a href="#">3.5</a>	<a href="#">Global Concurrent Optimization .....</a>	<a href="#">26</a>
<a href="#">3.6</a>	<a href="#">Adaptive Network Planning .....</a>	<a href="#">26</a>
<a href="#">4.</a>	<a href="#">Security Consideration .....</a>	<a href="#">26</a>
<a href="#">5.</a>	<a href="#">IANA Considerations .....</a>	<a href="#">26</a>
<a href="#">6.</a>	<a href="#">References .....</a>	<a href="#">26</a>
<a href="#">6.1</a>	<a href="#">Informative References .....</a>	<a href="#">26</a>
<a href="#">7.</a>	<a href="#">Authors' Addresses .....</a>	<a href="#">29</a>
<a href="#">A.</a>	<a href="#">Undefined Interfaces .....</a>	<a href="#">30</a>

## [1. Introduction](#)

Networks today integrate multiple technologies allowing network infrastructure to deliver a variety of services to support the different characteristics and demands of applications. There is an increasing demand to make the network responsive to service requests issued directly from the application layer. This differs from the established model where services in the network are delivered in response to management commands driven by a human user.



These application-driven requests and the services they establish place a set of new requirements on the operation of networks. They need on-demand and application-specific reservation of network connectivity, reliability, and resources (such as bandwidth). An environment that operates to meet this type of application-aware requirement is said to have Application-Based Network Operation (ABNO).

The Path Computation Element (PCE) [[RFC4655](#)] was developed to provide path computation services for GMPLS and MPLS networks. The applicability of PCE can be extended to provide path computation and policy enforcement capabilities for ABNO platforms and services.

ABNO can provide the following types of service to applications by coordinating the components that operate and manage the network:

- Optimization of traffic flows between applications to create an overlay network for communication in use cases such as file sharing, data caching or mirroring, media streaming, or real-time communications described as Application Layer Traffic Optimization (ALTO) [[RFC5693](#)].
- Remote control of network components allowing coordinated programming of network resources through such techniques as Forwarding and Control Element Separation (ForCES) [[RFC3746](#)], OpenFlow [[ONE](#)], and the Interface to the Routing System (I2RS) [[I-D.ward-irs-framework](#)].
- Interconnection of Content Delivery Networks (CDNi) [[RFC6707](#)] through the establishment and resizing of connections between content distribution networks.
- Network resource coordination to facilitate grooming and regrooming, bandwidth scheduling, and global concurrent optimization [[RFC5557](#)].
- Virtual Private Network (VPN) planning in support of deployment of new VPN customers and to facilitate inter-data center connectivity.

This document outlines the architecture and use cases for ABNO, and shows how the ABNO architecture can be used for co-ordinating control system and application requests to compute paths, enforce policies, and manage network resources for the benefit of the applications that use the network. The examination of the use cases shows the ABNO architecture as a toolkit comprising many existing components and protocols and so this document looks like a cookbook.



## **1.1 Scope**

This document describes a toolkit. It shows how existing functional components described in a large number of separate documents can be brought together within a single architecture to provide the function necessary for ABNO.

In many cases, existing protocols are known to be good enough or almost good enough to satisfy the requirements of interfaces between the components. In these cases the protocols are called out as suitable candidates for use within an implementation of ABNO.

In other cases it is clear that further work will be required, and in those cases a pointer to on-going work that may be of use will be provided.

Thus, this document may be seen as providing an applicability statement for existing protocols, and guidance for developers of new protocols or protocol extensions.

## **2. Application Based Network Operations (ABNO)**

### **2.1 Assumptions**

The principal assumption underlying this document is that existing technologies should be used where they are adequate for the task. Furthermore, when an existing technology is almost sufficient, it is assumed to be preferable to make minor extensions rather than to invent a whole new technology.

Note that this document describes an architecture. Functional components are architectural concepts and have distinct and clear responsibilities. Pairs of functional components interact at functional interfaces that are, themselves, architectural concepts.

It is not intended that this architecture constrains implementations. For example, a stateful and active PCE could be implemented as a single a server combining the ABNO components of the PCE, the Traffic Engineering Database, and the Resource Manager (see [Section 2.2](#)). However, the separation of the ABNO functions into separate functional components with clear interfaces between them enables implementations to choose which features to include and allows different functions to be distributed across distinct processes or even processors.





## 2.2 Generic ABNO Architecture

The following diagram illustrates the ABNO architecture. The components and functional interfaces are discussed in Sections [2.2.1](#) and [2.2.2](#) respectively. The use cases described in [Section 3](#) show how different components are used selectively to provide different services.

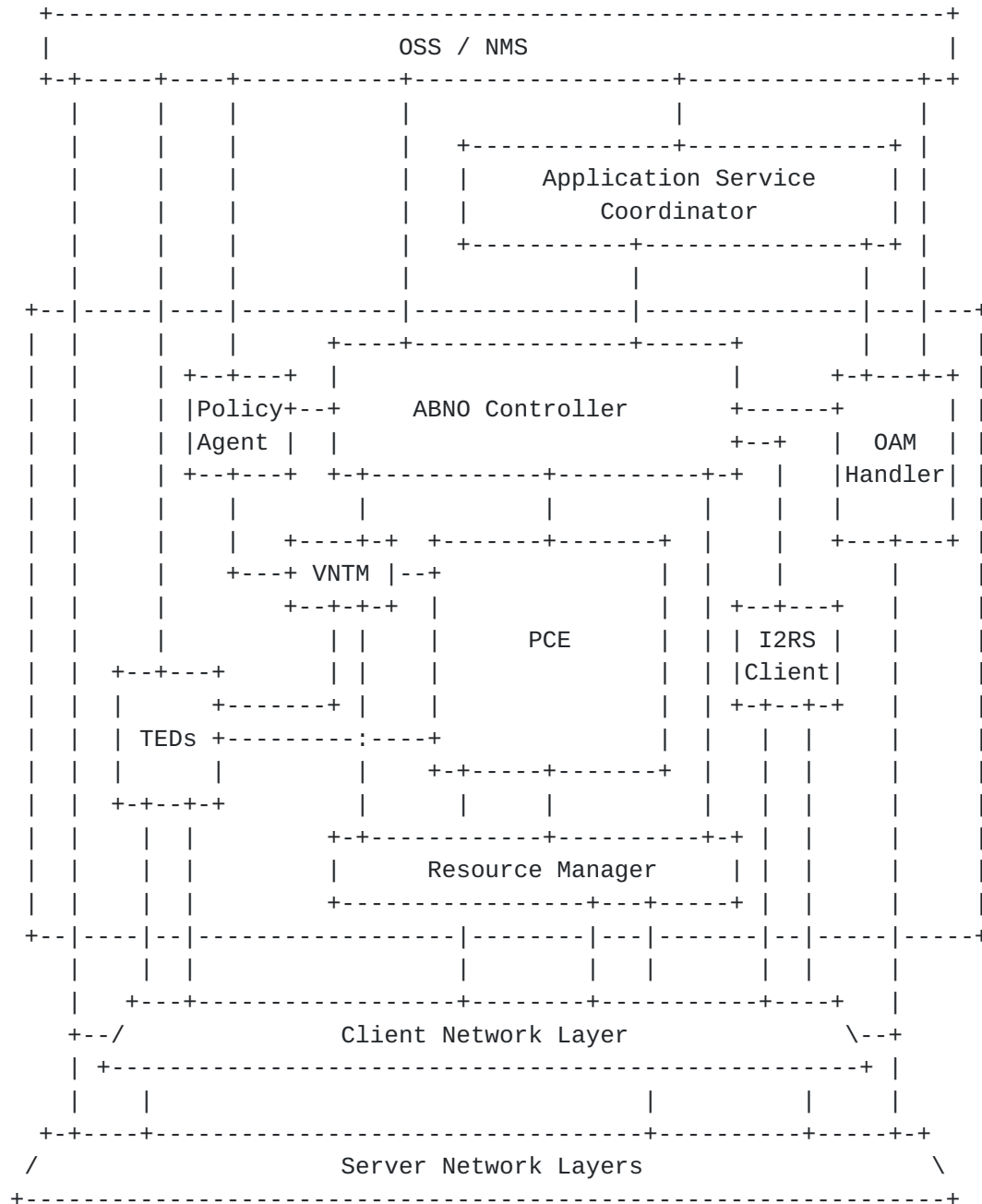


Figure 1: Generic ABNO Architecture



### **[2.2.1](#) ABNO Components**

This section describes the functional components shown as boxes in Figure 1. The interactions between those components, that is the functional interfaces, are described in [Section 2.2.2](#).

#### **[2.2.1.1](#) NMS and OSS**

A Network Management Station (NMS) or an Operations Support System (OSS) can be used to control, operate, and manage a network. Within the ABNO architecture, an NMS or OSS may issue high-level service requests to the ABNO controller. It may also establish policies for the activities of the components within the architecture.

The NMS and OSS can be consumers of network events reported through the OAM handler and can act on these reports as well as displaying them to users and raising alarms. The NMS and OSS can also access the Traffic Engineering Database (TED) to show the users the current state of the network.

Lastly, the NMS and OSS may utilize a direct programmatic or configuration interface to interact with the network elements within the network.

#### **[2.2.1.2](#) Application Service Coordinator**

In addition to the NMS and OSS, services in the ABNO architecture may be requested by or on behalf of applications. In this context the term "application" is very broad. An application may be a program that runs on a host or server and that provides services to a user, such as video conferencing application. Alternatively, an application may be a software tool with which a user makes requests of the network to set up specific services such as end-to-end connections or scheduled bandwidth reservations. Finally, an application may be a sophisticated control system that is responsible for arranging the provision of a more complex network service such as a virtual private network.

For the sake of this architecture, all of these concepts of an application are grouped together and are shown as the Application Service Coordinator since they are all in some way responsible for coordinating the activity of the network to provide services for use by applications.

The Application Service Coordinator communicates with the ABNO Controller to request operations on the network.



### **[2.2.1.3](#) ABNO Controller**

The ABNO Controller is the main gateway to the network for the NMS, OSS, and the Application Service Coordinator for the provision of advanced network coordination and functions. The ABNO Controller governs the behavior of the network in response to changing network conditions and in accordance with application network requirements and policies.

The use cases in [Section 3](#) provide a clearer picture of how the ABNO Controller interacts with the other components in the ABNO architecture.

### **[2.2.1.4](#) Policy Agent**

Policy plays a very important role in the control and management of the network. It is therefore significant in influencing how the key components of the ANBO architecture operate.

Figure 1 shows the Policy Agent as a component that is configured by the NMS/OSS with the policies that it applies. The Policy Agent is possible for propagating those policies into the other components of the system.

Simplicity in the figure necessitates leaving out many of the policy interactions that will take place. Although the Policy Agent is only shown interacting with the ABNO Controller and the Virtual Network Topology Manager (VNTM), it will also interact with the Path Computation Element (PCE), the Interface to the Routing System (I2RS) Client, and the network elements themselves.

### **[2.2.1.5](#) Interface to the Routing System (I2RS) Client**

The Interface to the Routing System (I2RS) is described in [\[I-D.ward-irs-framework\]](#). The interface provides a programmatic way to access (for read and write) the the routing state and policy information on routers in the network.

The I2RS Client is introduced in [\[I-D.atlas-irs-problem-statement\]](#). Its purpose is to manage information requests across a number of routers (each of which runs an I2RS Server) and coordinate setting or gathering state to/from those routers.

### **[2.2.1.6](#) OAM Handler**

Operations, Administration, and Maintenance (OAM) plays a critical role in understanding how a network is operating, detecting faults, and taking the necessary action to react to problems in the network.



Within the ABNO architecture, the OAM Handler is responsible for receiving notifications (often called alerts) from the network about potential problems, for correlating them, and for triggering other components of the system to take action to preserve or recover the services that were established by the ABNO Controller. The OAM Handler also reports network problems and, in particular, service-affecting problems to the NMS, OSS, and Application Service Coordinator.

Additionally, the OAM Handler interacts with the devices in the network to initiate OAM actions within the data plane such as monitoring and testing.

#### **2.2.1.7 Path Computation Element (PCE)**

The Path Computation Element (PCE) is introduced in [[RFC4655](#)]. It is a functional component that services requests to compute paths across a network graph. In particular, it can generate traffic engineered routes for MPLS-TE and GMPLS Label Switched Paths (LSPs). The PCE may receive these requests from the ABNO Controller, from the Virtual Network Topology Manager, or from network elements themselves.

The PCE operates on a view of the network topology stored in the Traffic Engineering Database (TED). A more sophisticated computation may be provided by a Stateful PCE that enhances the TED with information about the LSPs that are provisioned and operational within the network as described in [[RFC4655](#)] and [[I-D.ietf-pce-stateful-pce](#)].

Additional function in an Active PCE allows a functional component that includes a Stateful PCE to make provisioning requests to set up new services or to modify in-place services as described in [[I-D.crabbe-pce-pce-initiated-lsp](#)]. This function may directly access the network elements, or may be channelled through the Resource Manager.

Coordination between multiple PCEs operating on different TEDs can prove useful for performing path computation in multi-domain (for example, inter-AS) or multi-layer networks.

Since the PCE is a key component of the ABNO architecture, a better view of its role can be gained by examining the use cases described in [Section 3](#).





#### **2.2.1.8 Traffic Engineering Database (TED)**

The Traffic Engineering Database (TED) is data store of topology information about a network that may be enhanced with capability data (such as metrics or bandwidth capacity) and active status information (such as up/down status or residual unreserved bandwidth).

The TED may be built from information supplied by the network or from data (such as inventory details) sourced through the NMS/OSS.

The principal use of the TED in the ABNO architecture is to provide the raw data on which the Path Computation Element operates. But the TED may also be inspected by users at the NMS/OSS to view the current status of the network, and may provide information to application services such as Application Layer Traffic Optimization (ALTO) [[RFC5693](#)].

#### **2.2.1.9 Virtual Network Topology Manager (VNTM)**

A Virtual Network Topology (VNT) is defined in [[RFC5212](#)] as a set of one or more LSPs in one or more lower-layer networks that provides information for efficient path handling in an upper-layer network. For instance, a set of LSPs in a wavelength division multiplexed (WDM) network can provide connectivity as virtual links in a higher-layer packet switched network.

The VNT enhances the physical/dedicated links that are available in the upper-layer network and is configured by setting up or tearing down the lower-layer LSPs and by advertising the changes into the higher-layer network. The VNT can be adapted to traffic demands so that capacity in the higher-layer network can be created or released as needed. Releasing unwanted VNT resources makes them available in the lower-layer network for other uses.

The creation of virtual topology for inclusion in a network is not a simple task. Decisions must be made about which nodes in the upper-layer it is best to connect, in which lower-layer network to provision LSPs to provide the connectivity, and how to route the LSPs in the lower-layer network. Furthermore, some specific actions have to be taken to cause the lower-layer LSPs to be provisioned and the connectivity in the upper-layer network to be advertised.

All of these actions and decisions are heavily influenced by policy, so the Virtual Network Topology Manager (VNTM) [[RFC5623](#)] component that coordinates them takes input from the Policy Agent. The VNTM is also closely associated with the PCE for the upper-layer network and each of the PCEs for the lower-layer networks.



#### **[2.2.1.10](#) Resource Manager**

The Resource Manager is responsible for making or channelling requests for the establishment of LSPs. This may be instructions to the control plane running in the networks, or may involve the programming of individual network devices. In the latter case, the Resource Manager may act as an OpenFlow Controller [[ONE](#)].

See [Section 2.2.2.6](#) for more details of the interactions between the Resource Manager and the network.

#### **[2.2.1.11](#) Client and Server Network Layers**

The client and server networks are shown in Figure 1 as illustrative examples of the fact that the ABNO architecture may be used to coordinate services across multiple networks where lower-layer networks provide connectivity in upper-layer networks.

[Section 3.2](#) describes a use case for multi-layer networking.

### **[2.2.2](#) Functional Interfaces**

This section describes the interfaces between functional components that might be externalized in an implementation allowing the components to be distributed across platforms. Where existing protocols might provide all or most of the necessary capabilities they are noted.

#### **[2.2.2.1](#) Configuration and Programmatic Interfaces**

The network devices may be configured or programmed direct from the NMS/OSS. Many protocols already exist to perform these functions including:

- SNMP [[RFC3412](#)]
- Netconf [[RFC6241](#)]
- ForCES [[RFC5810](#)]
- OpenFlow [[ONE](#)].

From the ABNO perspective, network configuration is a pass-through function. It can be seen represented on the left hand side of Figure 1.



#### **[2.2.2.2](#) TED Construction from the Networks**

As described in [Section 2.2.1.8](#), the Traffic Engineering Database (TED) provides details of the capabilities of the network for use by the ABNO system and the PCE in particular.

The TED can be constructed by participating in the IGP-TE protocols run by the networks (for example, OSPF-TE [[RFC3630](#)] and ISIS-TE [[RFC5305](#)]). Alternatively, the TED may be fed using link-state distribution extensions to BGP [[I-D.ietf-idr-ls-distribution](#)].

The ABNO system may maintain a single TED unified across multiple networks, or may retain a separate TEDs for each network.

Additionally, an ALTO Server [[RFC5693](#)] may provide an abstracted topology from a network to build an application-level TED that can be used by a PCE to compute paths between servers and application-layer entities for the provision of application services.

#### **[2.2.2.3](#) TED Enhancement**

The TED may be enhanced by inventory information supplied from the NMS/OSS. This may supplement the data collected as described in [Section 2.2.2.2](#) with information that is not normally distributed within the network such as node types and capabilities, or the characteristics of optical links.

No protocol is currently identified for this interface, but the Interface to the Routing System (I2RS) protocol [[I-D.ward-irs-framework](#)] may be a suitable candidate because it is designed to distribute bulk routing state information in a well-defined encoding language. Another candidate protocol may be Netconf [[RFC6241](#)] passing data encoded using YANG [[RFC6020](#)].

#### **[2.2.2.4](#) TED Presentation**

The TED may be presented north-bound from the ABNO system for use by an NMS/OSS or by the Application Service Coordinator. This allows users and applications to get a view of the network topology and the status of the network resources. It also allows planning and provisioning of application services.

There are several protocols available for exporting the TED north-bound:

- The ALTO protocol [[I-D.ietf-alto-protocol](#)] is designed to distribute the abstracted topology used by an ALTO Server and may prove useful for exporting the TED.



- The same protocol used to export topology information from the network can be used to export the topology from the TED. [[I-D.ietf-idr-ls-distribution](#)].
- The Interface to the Routing System (I2RS) [[I-D.ward-irs-framework](#)] will require a protocol that is capable of handling bulk routing information exchanges that would be suitable for exporting the TED.

#### **[2.2.2.5](#) Network Making Path Computation Requests**

As originally specified in the PCE architecture [[RFC4655](#)], network elements can make path computation requests to a PCE using the PCE protocol (PCEP) [[RFC5440](#)]. This facilitates the network setting up LSPs in response to simple connectivity requests, and it allows the network to re-optimize or repair LSPs.

#### **[2.2.2.6](#) Resource Manager Control of Networks**

As described in [Section 2.2.1.10](#), the Resource Manager makes or channels requests to provision resources in the network. These operations can take place at two levels: there can be requests to program/configure specific resources in the data or forwarding planes; and there can be requests to trigger a set of actions to be programmed with the assistance of a control plane.

A number of protocols already exist to provision network resources as follows:

- Program/configure specific network resources
  - ForCES [[RFC5810](#)] defines a protocol for separation of the control element (the Resource Manager) from the forwarding elements in each node in the network.
  - The Generic Switch Management Protocol (GSMP) [[RFC3292](#)] is an asymmetric protocol that allows one or more external switch controllers (such as the Resource Manager) to establish and maintain the state of a label switch such as an MPLS switch.
  - OpenFlow [[ONE](#)] is is a communications protocol that gives an OpenFlow Controller (such as the Resource Manager) access to the forwarding plane of a network switch or router in the network.
  - Historically, other configuration-based mechanisms have been used to set up the forwarding/switching state at individual nodes within networks. Such mechanisms have ranged from non-standard command line interfaces (CLIs) to various standards-based options such as TL1 [[TL1](#)] and SNMP [[RFC3412](#)]. These mechanisms are not





designed for rapid operation of a network and are not easily programmatic. They are not proposed for use by the Resource Controller as part of the ABNO architecture.

- Netconf [[RFC6241](#)] provides a more active configuration protocol that may be suitable for bulk programming of network resources. Its use in this way is dependent on suitable YANG modules being defined for the necessary options. Early work in the IETF's Netmod working group is focused on a higher level of routing function more comparable with the function discussed in [Section 2.2.2.8](#) [I-D.[draft-ietf-netmod-routing-cfg](#)].
- Trigger actions through the control plane
  - LSPs can be requested using a management system interface to the head end of the LSP using tools such as CLIs, TL1 [[TL1](#)] or SNMP [[RFC3412](#)]. Configuration at this granularity is not as time-critical as when individual network resources are programmed because the main task of programming end-to-end connectivity is devolved to the control plane. Nevertheless, these mechanisms remain unsuitable for programmatic control of the network and are not proposed for use by the Resource Controller as part of the ABNO architecture.
  - As noted above, Netconf [[RFC6241](#)] provides a more active configuration protocol. This may be particularly suitable for requesting the establishment of LSPs. Work would be needed to complete a suitable YANG module.
  - The PCE protocol (PCEP) [[RFC5440](#)] has been proposed as a suitable protocol for requesting the establishment of LSPs [[I-D.crabbe-pce-pce-initiated-lsp](#)]. This works well because the protocol elements necessary are exactly the same as used to respond to a path computation request.

The functional element that issues PCEP requests to establish LSPs is known as an "Active PCE", however it should be noted that the ABNO functional components responsible for requesting LSPs are more likely to be the Resource Manager, the Virtual Network Topology Manager, and the ABNO Controller itself.

Note that the I2RS does not provide a mechanism for control of network resources at this level as it is designed to provide control of routing state in routers, not forwarding state in the data plane.



#### **[2.2.2.7](#) Auditing the Network**

Once resources have been provisioned or connections established in the network, it is important that the ABNO system can determine the state of the network. This function falls into four categories:

- Updates to the TED are gathered as described in [Section 2.2.2.2](#).
- OAM can be commissioned and the results inspected by the OAM Handler as described in [Section 2.2.2.13](#).
- Explicit notification of the successful establishment and the subsequent state of LSP can be provided through extensions to PCEP as described in [[I-D.ietf-pce-stateful-pce](#)] and [[I-D.crabbe-pce-pce-initiated-lsp](#)].
- ABNO components can may make enquiries and inspect network state through I2RS or using Netconf.

#### **[2.2.2.8](#) Controlling The Routing System**

As discussed in [Section 2.2.1.5](#), the Interface to the Routing System (I2RS) provides a programmatic way to access (for read and write) the routing state and policy information on routers in the network. The I2RS Client issues requests to routers in the network to establish or retrieve routing state. Those requests utilize the I2RS protocol which has yet to be selected/designed by the IETF.

#### **[2.2.2.9](#) ABNO Controller Interface to PCE**

The ABNO controller needs to be able to consult the PCE to determine what services can be provisioned in the network. There is no reason why this interface cannot be based on the standard PCE protocol as defined in [[RFC5440](#)].

#### **[2.2.2.10](#) VNTM Interface to and from PCE**

There are two interactions between the Virtual Network Topology Manager and the PCE.

The first interaction is used when VNTM wants to determine what LSPs can be set up in a network: in this case it uses the standard PCEP interface [[RFC5440](#)] to make path computation requests.

The second interaction arises when a PCE determines that it cannot compute a requested path or notices that (according to some configured policy) a network is short of resources (for example, the capacity on some key link is close to exhausted). In this case, the



PCE may notify the VNTM which may (again according to policy) act to construct more virtual topology. This second interface is not currently specified although it may be that the I2RS protocol provides suitable features.

#### **[2.2.2.11](#) ABNO Control Interfaces**

The north-bound interface from the ABNO controller is used by the NMS, OSS, and Application Service Coordinator to request services in the network in support of applications. The interface will also need to be able to report the asynchronous completion of service requests and convey changes in the status of services.

This interface will also need strong capabilities for security, authentication, and policy.

This interface is not currently specified. It needs to be a transactional interface that supports the specification of abstract services with adequate flexibility to facilitate easy extension and yet be concise and easily parsable.

It is possible that the I2RS protocol (see [Section 2.2.2.8](#)) will support the necessary features.

#### **[2.2.2.12](#) Policy Interfaces**

As described in [Section 2.2.1.4](#) and throughout this document, policy forms a critical component of the ABNO architecture. The role of policy will include enforcing the following rules and requirements:

- Adding resources on demand should be gated by the authorized capability.
- Client microflows should not trigger server-layer setup or allocation.
- Accounting capabilities should be supported.
- Security mechanisms for authorization of requests and capabilities are required.

Various policy-capable architectures have been defined including a framework for using policy with a PCE-enabled system [[RFC5394](#)]. However, the take-up of the IETF's Common Open Policy Service protocol (COPS) [[RFC2748](#)] has been poor.

New work will be needed to define all of the policy interfaces within the ABNO architecture. There is some discussion that the I2RS



protocol may support the configuration and manipulation of policies.

#### **[2.2.2.13](#) OAM and Reporting**

The OAM Handler must interact with the networks to perform several actions:

- Enabling OAM function within the network.
- Performing proactive OAM operations in the network.
- Receiving notifications of network events.

Any of the configuration and programmatic interfaces described in [Section 2.2.2.1](#) may serve this purpose, although neither Netconf nor OpenFlow currently supports asynchronous notifications. Additionally Syslog [[RFC5424](#)] is a protocol for reporting events from the network, and IPFIX [[RFC5101](#)] is designed to allow network statistics to be aggregated and reported.

The OAM Handler also correlates events reported from the network and reports them onward to the ABNO Controller (which can apply the information to the recovery of services that it has provisioned) and to the NMS, OSS, and Application Service Coordinator. The reporting mechanism used here can be essentially the same as used when events are reported from the network and no new protocol is needed.

### **[3.](#) ABNO Use Case**

This section provides a number of examples of how the ABNO architecture can be applied to provide application and NMS/OSS driven network operations.

#### **[3.1](#) Inter-AS Connectivity**

The following use case describes how the ABNO framework can be used set up an end-to-end service across multiple Autonomous Systems (ASes). Consider the simple network topology shown in Figure 2. The three ASes (ASa, ASb, and ASc) are connected as ASBRs a1, a2, b1 through b4, c1 and c2. A source node (s) located in ASa is to be connected to a destination node (d) located in ASc. The optimal path for the LSP from s to d must be computed, and then the network must be triggered to set up the LSP.





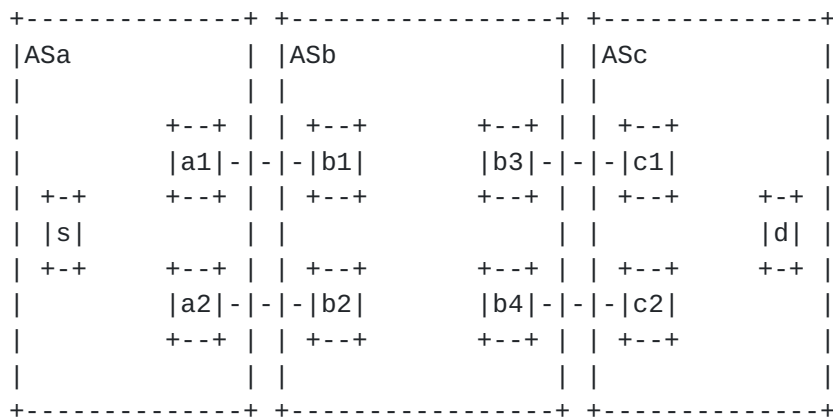
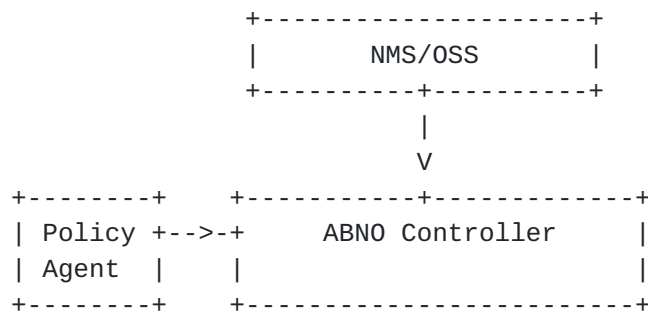


Figure 2: Inter-AS Domain Topology with H-PCE (Parent PCE)

In the ABNO architecture, the following steps are performed to deliver the service.

## 1. Request Management

As shown in Figure 3, the NMS/OSS issues a request to the ABNO Controller for a path between  $s$  and  $d$ . The ABNO Controller verifies that the NMS/OSS has sufficient rights to make the service request.



### Figure 3: ABNO Request Management

## 2. Service Path Computation with Hierarchical PCE

The ABNO Controller needs to determine an end-to-end path for the LSP. Since the ASes will want to maintain a degree of confidentiality about their internal resources and topology, they will not share a TED and each will have its own PCE. In such a situation, the Hierarchical PCE (H-PCE) architecture described in [\[RFC6805\]](#) is applicable.

As shown in Figure 4, the ABNO Controller sends a request to the



parent PCE for an end-to-end path. As described in [RFC6805], the parent PCE consults its TED that shows the connectivity between ASes. This helps it understand that the end-to-end path must cross each of ASa, ASb, and ASc, so it sends individual path computation requests to each of PCE a, b, and c to determine the best options for crossing the ASes.

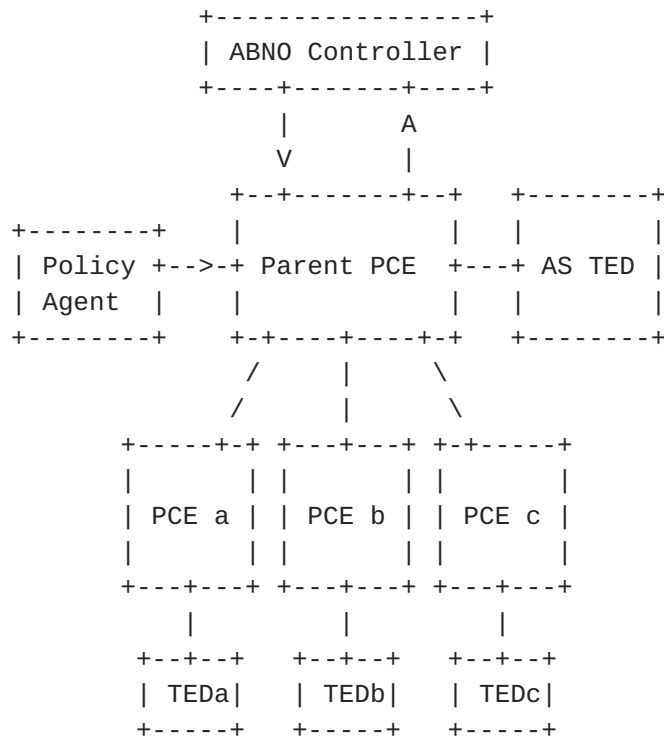


Figure 4: Path Computation Request with Hierarchical PCE

Each child PCE applies policy to the requests it receives to determine whether the request is to be allowed and to select the type of network resources that can be used in the computation result. For confidentiality reasons, each child PCE may supply its computation responses using a path key [RFC5520] to hide the details of the path segment it has computed.

The parent PCE collates the responses from the children and applies its own policy to stitch them together into the best end-to-end path which it returns as a response to the ABNO Controller.

### 3. Provisioning the End-to-End LSP

There are several options for how the end-to-end LSP gets provisioned in the ABNO architecture. Some of these are described below.



### 3a. Provisioning from the ABNO Controller With a Control Plane

Figure 5 shows how the ABNO controller makes a request through the Resource Manager to establish the end-to-end LSP. As described in [Section 2.2.2.6](#) these interactions can use the Netconf protocol [[RFC6241](#)] or the extensions to PCEP described in [[I-D.crabbe-pce-pce-initiated-lsp](#)]. In either case, the provisioning request is sent to the head end Label Switching Router (LSR) and it signals in the control plane (using a protocol such as RSVP-TE [[RFC3209](#)]) so cause the LSP to be established.

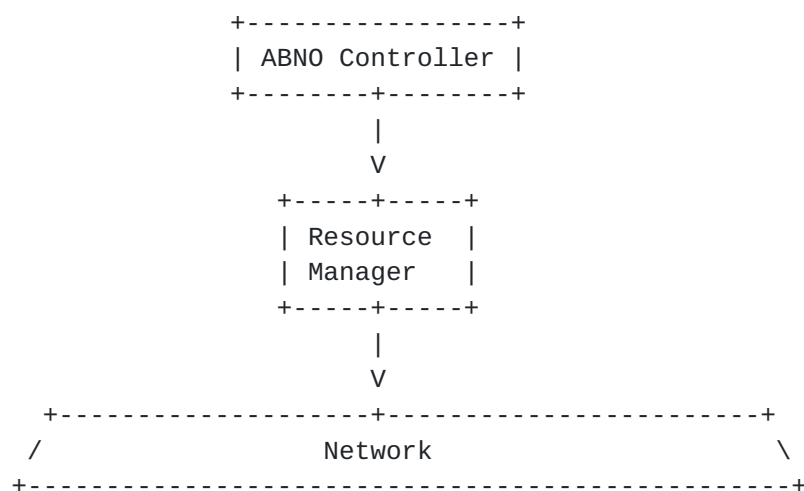


Figure 5: Provisioning the End-to-End LSP

### 3b. Provisioning through Programming Network Resources

Another option is that the LSP is provisioned hop by hop from the Resource Manager using ForCES [[RFC5810](#)] or OpenFlow [[ONF](#)] as described in [Section 2.2.2.6](#). In this case, the picture is the same as shown in Figure 5. The interaction between the ABNO Controller and the Resource Manager will be PCEP or Netconf as described in option 3a., and the Resource Manager will have the responsibility to fan out the requests to the individual network elements.

### 3c. Provisioning with an Active PCE

The active PCE is described in [Section 2.2.1.7](#) based on the concepts expressed in [[I-D.crabbe-pce-pce-initiated-lsp](#)]. In this approach, the process described in 3a is modified such that the PCE issues a PCEP command to the network direct without a response being first returned to the ABNO Controller.



This situation is shown in Figure 6, and could be modified so that the Resource Manager still programs the individual network elements as described in 3b.

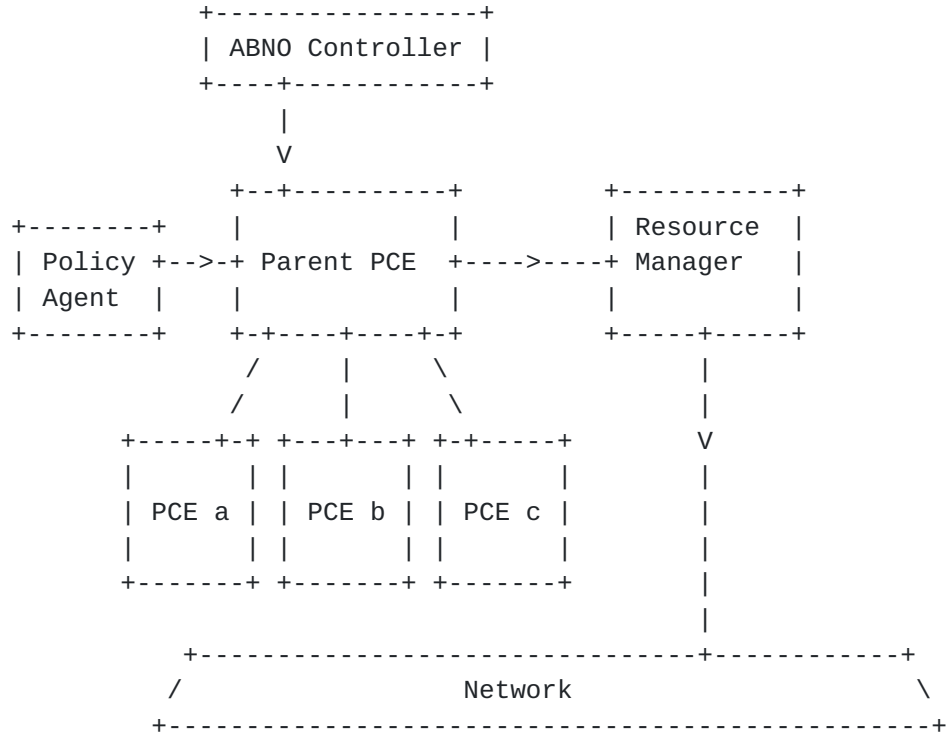


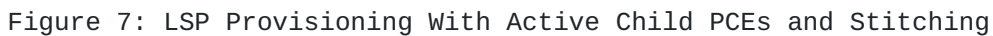
Figure 6: LSP Provisioning with an Active PCE

### 3d. Provisioning with Active Child PCEs and Segment Stitching

A mixture of the approaches described in 3b and 3c can result in a combination of mechanisms to program the network to provide the end-to-end LSP. Figure 7 shows how each child PCE can be an active PCE responsible for setting up an edge-to-edge LSP segment across one of the ASes. The ABNO Controller then uses the Resource Manager to program the inter-AS connections using ForCES or OpenFlow and the LSP segments are stitched together following the ideas described in [\[RFC5150\]](#).







The ABNO Controller will need to ascertain that the end-to-end LSP has been set up as requested. In the case of a control plane being used to establish the LSP, the head end LSR may send a notification (perhaps using PCEP) to report successful setup, but to be sure that the LSP is up, the ABNO Controller will request the OAM Handler to perform Continuity Check OAM in the Data Plane and report back that the LSP is ready to carry traffic.



## 5. Notification of Service Fulfillment

Finally, when the ABNO Controller is satisfied that the requested service is ready to carry traffic, it will notify the NMS/OSS.

## 3.2 Multi-Layer Networking

Networks typically comprise of multiple layers. These layers represent separations of administrative regions, technology, and may also represent a distinction between client and server networking roles.

It is preferable to coordinate network resource control and utilization (i.e., consideration and control of multiple layers), rather than controlling and optimizing resources at each layer independently. This facilitates network efficiency and network automation, and may be defined as inter-layer traffic engineering.

The PCE architecture supports inter-layer traffic engineering [RFC5623] and, in combination with the ABNO architecture, provides a suite of capabilities for network resource coordination across multiple layers.

The following use case demonstrates ABNO used to coordinate allocation of server-layer network resources to create virtual topology in a client-layer network in order to satisfy a request for end-to-end client-layer connectivity. Consider the simple multi-layer network in Figure 8. There are six packet-layer routers (P1 through P6) and three optical-layer lambda switches (L1 through L3). There is connectivity in the packet layer between routers P1, P2, and P3, and also between routers P4, P5, and P6, but there is no packet-layer connectivity between these two islands of routers perhaps because of a network failure or perhaps because all existing bandwidth between the islands has already been used up. However, there is connectivity in the optical layer between switches L1, L2, and L3, and the optical network is connected out to routers P3 and P4 (they have optical line cards). In this example, a packet-layer connection (an MPLS LSP) is desired between P1 and P6.

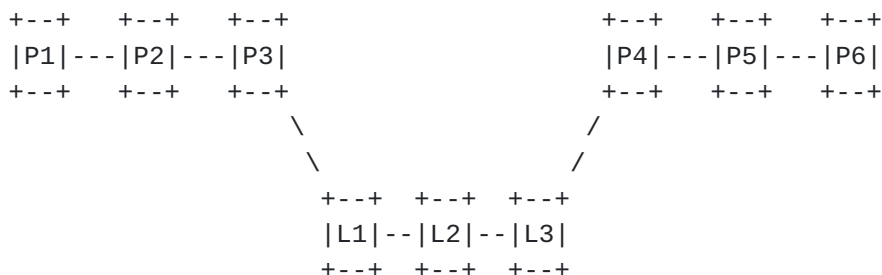


Figure 8: A Multi-Layer Network



In the ABNO architecture, the following steps are performed to deliver the service.

## 1. Request Management

As shown in Figure 9, the Application Service Coordinator issues a request for connectivity from P1 to P6 in the packet-layer network. That is, the Application Service Coordinator requests an MPLS LSP with a specific bandwidth to carry traffic for its application. The ABNO Controller verifies that the Application Service Coordinator has sufficient rights to make the service request.

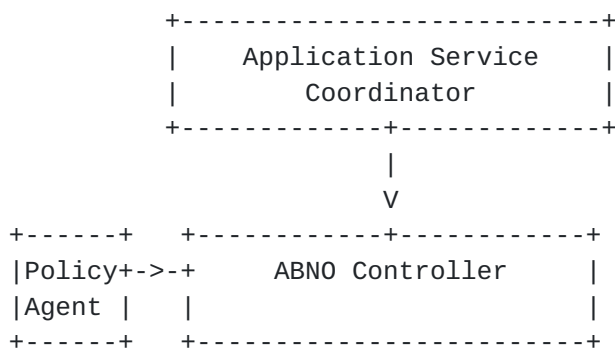


Figure 9: Application Service Coordinator Request Management

## 2. Service Path Computation in the Packet Layer

The ABNO Controller sends a path computation request to the packet layer PCE to compute a suitable path for the requested LSP as shown in Figure 10. The PCE uses the appropriate policy for the request and consults the TED for the packet layer. It determines that no path is immediately available.

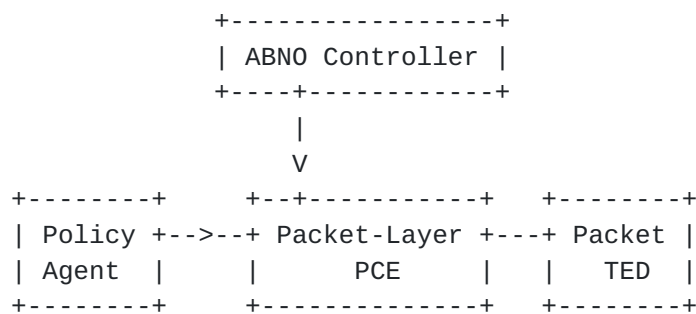


Figure 10: Path Computation Request



### 3. Invocation of VNTM and Path Computation in the Optical Layer

After the path computation failure in step 2, instead of notifying ABNO controller of the failure, the PCE invokes the VNTM to see whether it can create the necessary link in the virtual network topology to bridge the gap.

As shown in Figure 11, the packet-layer PCE reports the connectivity problem to the VNTM, and the VNTM consults policy to determine what it is allowed to do in this case. Assuming that the policy allows it, VNTM asks the optical-layer PCE to see whether it can find a path across the optical network that could be provisioned to provide a virtual link for the packet layer. In addressing this request, the optical-layer PCE consults a TED for the optical-layer network.

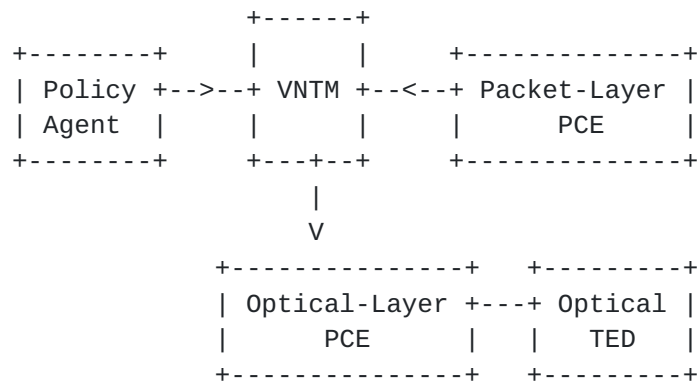


Figure 11: Invocation of VNTM and Optical Layer Path Computation

### 5. Provisioning in the Optical Layer

Once a path has been found across the optical-layer network it needs to be provisioned. The options follow those in step 3 of [Section 3.1](#). That is, provisioning can be initiated by the optical-layer PCE or by its user, the VNTM. The command can be sent to the head end of the optical LSP (P3) so that the control plane (for example, GMPLS [[RFC3473](#)]) can be used to provision the LSP. Alternatively, the network resources can be provisioned direct using any of the mechanisms described in [Section 2.2.2.6](#).

### 6. Creation of Virtual Topology in the Packet Layer

Once the LSP has been set up in the optical-layer it can be made available in the packet layer as a virtual link. If the GMPLS signaling used the mechanisms described in [[RFC6107](#)] this process can be automated within the control plane, otherwise it may





require a specific instruction to the head end router of the optical LSP (for example, through the Interface to the Routing System).

Once the virtual link is created as shown in Figure 12, it is advertised in the IGP for the packet-layer network and the link will appear in the TED for the packet-layer network.

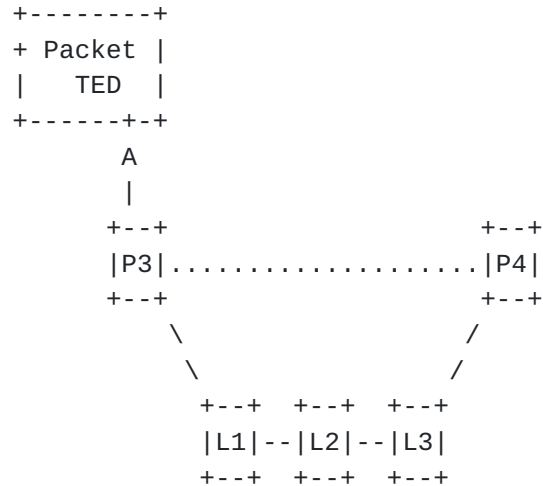


Figure 12: Advertisement of a New Virtual Link

## 7. Path Computation Completion and Provisioning in the Packet Layer

Now there are sufficient resources in the packet-layer network. The PCE for the packet-layer can complete its work and the MPLS LSP can be provisioned as described in [Section 3.1](#).

## 9. Verification and Notification of Service Fulfillment

As discussed in [Section 3.1](#), the ABNO controller will need to verify that the end-to-end LSP has been correctly established before reporting service fulfillment to the the Application Service Coordinator.

Furthermore, it is highly likely that service verification will be necessary before the optical-layer LSP can be put into service as a virtual link. Thus, the VNTM will need to coordinate with the OAM Handler to ensure that the LSP is ready for use.

## 3.3 Bandwidth Scheduling

This section to be completed in a future revision of this document.



### **[3.4 Grooming and Regrooming](#)**

This section to be completed in a future revision of this document.

This use case will cover the following scenarios:

- Nested LSPs
- Packet Classification (IP flows into LSPs at edge routers)
- Bucket Stuffing
- IP Flows into ECMP Hash Bucket

### **[3.5 Global Concurrent Optimization](#)**

This section to be completed in a future revision of this document.

### **[3.6 Adaptive Network Planning](#)**

The ABNO architecture provides the capability for reactive network control of resources based on classification, profiling and prediction based on current demands and resource utilization. ABNO would then manipulate server-layer transport network resources, including OTN and Flexi-grid to meet current and projected demands.

This section to be completed in a future revision of this document.

## **[4. Security Consideration](#)**

To be discussed.

## **[5. IANA Considerations](#)**

This document makes no requests for IANA action.

## **[6. References](#)**

### **[6.1. Informative References](#)**

[I-D.atlas-irs-problem-statement]

Atlas, A., Nadeau, T., and Ward, D., "Interface to the Routing System Problem Statement", [draft-atlas-irs-problem-statement](#), work in progress.

[I-D.crabbe-pce-pce-initiated-lsp]

Crabbe, E., Minei, I., Sivabalan, S., and Varga, R., "PCEP Extensions for PCE-initiated LSP Setup in a Stateful PCE Model", [draft-crabbe-pce-pce-initiated-lsp](#), work in progress.



[I-D.ietf-alto-protocol]

Alimi, R., Penno, R., and Yang, Y., "ALTO Protocol", [draft-ietf-alto-protocol](#), work in progress.

[I-D.ietf-idr-ls-distribution]

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

[I-D.[draft-ietf-netmod-routing-cfg](#)]

Lhotka, L., "A YANG Data Model for Routing Management", [draft-ietf-netmod-routing-cfg](#), work in progress.

[I-D.ietf-pce-stateful-pce]

Crabbe, E., Medved, J., Minei, I., and R. Varga, "PCEP Extensions for Stateful PCE", [draft-ietf-pce-stateful-pce](#), work in progress.

[I-D.ward-irs-framework]

Atlas, A., Nadeau, T. and Ward, D., "Interface to the Routing System Framework", [draft-ward-irs-framework](#), work in progress.

[ONF] Open Networking Foundation, "OpenFlow Switch Specification Version 1.1.0 Implemented (Wire Protocol 0x02)", February 2011.

[RFC2748] Durham, D., Ed., Boyle, J., Cohen, R., Herzog, S., Rajan, R., and A. Sastry, "The COPS (Common Open Policy Service) Protocol", [RFC 2748](#), January 2000.

[RFC3209] D. Awduche et al., "RSVP-TE: Extensions to RSVP for LSP Tunnels", [RFC 3209](#), December 2001.

[RFC3292] Doria, A., Hellstrand, F., Sundell, K., and Worster, T., "General Switch Management Protocol (GSMP) V3", [RFC 3292](#), June 2002.

[RFC3412] Case, J., Harrington, D., Preshun, R., and Wijnen, B., "Message Processing and Dispatching for the Simple Network Management Protocol (SNMP)", [RFC 3412](#), December 2002.

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



- [RFC3746] Yang, L., Dantu, R., Anderson, T., and Gopal, R., "Forwarding and Control Element Separation (ForCES) Framework", [RFC 3746](#), April 2004.
- [RFC3473] L. Berger et al., "Generalized Multi-Protocol Label Switching (GMPLS) Signaling Resource ReserVation Protocol-Traffic Engineering (RSVP-TE) Extensions", [RFC 3473](#), January 2003.
- [RFC4655] Farrel, A., Vasseur, J.-P., and Ash, J., "A Path Computation Element (PCE)-Based Architecture", [RFC 4655](#), August 2006.
- [RFC5101] B. Claise, "Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of IP Traffic Flow Information", [RFC 5101](#), January 2008.
- [RFC5150] Ayyangar, A., Kompella, K., Vasseur, JP. and Farrel, A., "Label Switched Path Stitching with Generalized Multiprotocol Label Switching Traffic Engineering (GMPLS TE)", [RFC 5150](#), February 2008.
- [RFC5212] Shiomoto, K., Papadimitriou, D., Le Roux, JL., Vigoureux, M., and Brungard, D., "Requirements for GMPLS-Based Multi-Region and Multi-Layer Networks (MRN/MLN)", [RFC 5212](#), July 2008.
- [RFC5305] Li, T. and H. Smit, "IS-IS Extensions for Traffic Engineering", [RFC 5305](#), October 2008.
- [RFC5394] Bryskin, I., Papadimitriou, D., Berger, L. and Ash, J., "Policy-Enabled Path Computation Framework", [RFC 5394](#), December 2008.
- [RFC5424] R. Gerhards, "The Syslog Protocol", [RFC 5424](#), March 2009.
- [RFC5440] Vasseur, JP. and Le Roux, JL., "Path Computation Element (PCE) Communication Protocol (PCEP)", [RFC 5440](#), March 2009.
- [RFC5520] Bradford, R., Vasseur, JP., and Farrel, A., "Preserving Topology Confidentiality in Inter-Domain Path Computation Using a Path-Key-Based Mechanism", RC 5520, April 2009.
- [RFC5557] Lee, Y., Le Roux, JL., King, D., and Oki, E., "Path Computation Element Communication Protocol (PCEP) Requirements and Protocol Extensions in Support of Global Concurrent Optimization", [RFC 5557](#), July 2009.





- [RFC5623] Oki, E., Takeda, T., Le Roux, J.L., and Farrel, A., "Framework for PCE-Based Inter-Layer MPLS and GMPLS Traffic Engineering", [RFC 5623](#), September 2009.
- [RFC5693] Seedorf, J., and Burger, E., "Application-Layer Traffic Optimization (ALTO) Problem Statement", [RFC 5693](#), October 2009.
- [RFC5810] A. Doria, et al., "Forwarding and Control Element Separation (ForCES) Protocol Specification", [RFC 5810](#), March 2010.
- [RFC6020] Bjorklund, M., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", [RFC 6020](#), October 2010.
- [RFC6107] Shiomoto, K. and A. Farrel, "Procedures for Dynamically Signaled Hierarchical Label Switched Paths", [RFC 6107](#), February 2011.
- [RFC6241] Enns, R., Bjorklund, M., Schoenwaelder, J., and Bierman, A., "Network Configuration Protocol (NETCONF)", [RFC 6241](#), June 2011.
- [RFC6707] Niven-Jenkins, B., Le Faucheur, F., and Bitar, N., "Content Distribution Network Interconnection (CDNI) Problem Statement", [RFC 6707](#), September 2012.
- [RFC6805] King, D. and Farrel, A., "The Application of the Path Computation Element Architecture to the Determination of a Sequence of Domains in MPLS and GMPLS", [RFC 6805](#), November 2012.
- [TL1] Telcorida, "Operations Application Messages - Language For Operations Application", GR-831, November 1996.

## **7. Authors' Addresses**

Daniel King  
Old Dog Consulting

Email: [daniel@olddog.co.uk](mailto:daniel@olddog.co.uk)

Adrian Farrel  
Juniper Networks

Email: [adrian@olddog.co.uk](mailto:adrian@olddog.co.uk)



## **Appendix A. Undefined Interfaces**

This Appendix provides a brief list of interfaces that are not yet defined at the time of writing. Interfaces where there is a choice of existing protocols are not listed.

To be completed in future release of this document.