

Internet Engineering Task Force
Internet-Draft
Intended status: Informational
Expires: 21 April, 2014

D. King
Old Dog Consulting
A. Farrel
Juniper Networks
21 October 2013

A PCE-based Architecture for Application-based Network Operations

draft-farrkingel-pce-abno-architecture-06.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) in a variety of network applications (such as point-to-point connectivity, network virtualization, or mobile back-haul) and in a range of network technologies from packet (IP/MPLS) down to optical. An environment that operates to meet this type of requirement is said to have Application-Based Network Operations (ABNO).

ABNO brings together many 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."

Copyright Notice

Copyright (c) 2013 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

1. Introduction	4
1.1 Scope	5
2. Application-based Network Operations (ABNO)	5
2.1 Assumptions and Requirements	5
2.2 Implementation of the Architecture	6
2.3 Generic Architecture	8
2.3.1 ABNO Components	9
2.3.2 ABNO Functional Interfaces	15
3. ABNO Use Cases	22
3.1 Inter-AS Connectivity	22
3.2 Multi-Layer Networking	28
3.2.1 Data Center (DC) Interconnection across MLNs.....	31
3.3 Make-Before-Break	35
3.3.1 Make-Before-Break for Re-optimization	35
3.3.2 Make-Before-Break for Restoration	36
3.3.3 Make-Before-Break for Path Test and Selection	37
3.4 Global Concurrent Optimization	39
3.4.1 Use Case: GCO with MPLS LSPs	40
3.5 Adaptive Network Management (ANM)	42
3.5.1. ANM Trigger	43
3.5.2. Processing request and GCO computation	43
3.5.3. Automated Provisioning Process	44
3.6 Pseudowire Operations and Management	45
3.6.1 Multi-Segment Pseudowires	45
3.6.2 Path-Diverse Pseudowires	47
3.6.3 Path-Diverse Multi-Segment Pseudowires	48
3.6.4 Pseudowire Segment Protection	49
3.6.5 Applicability of ABNO to Pseudowires	49
3.7 Cross-Stratum Optimization	50
3.7.1. Data Center Network Operation	50
3.7.2. Application of the ABNO Architecture	52
3.8 Other Potential Use Cases	54
3.8.1 Grooming and Regrooming	54
3.8.2 Bandwidth Scheduling	54
3.8.3 ALTO Server	55
4. Survivability and Redundancy within the ABNO Architecture ...	55
5. Security Consideration	55
6. Manageability Considerations	55
7. IANA Considerations	55
8. Acknowledgements	55
9. References	55
9.1 Informative References	55
10. Contributors' Addresses	59
11. Authors' Addresses	60
A. Undefined Interfaces	61

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) in a variety of network applications (such as point-to-point connectivity, network virtualization, or mobile back-haul) and in a range of network technologies from packet (IP/MPLS) down to optical. 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 [ONF], and the Interface to the Routing System (I2RS) [I-D.ietf-i2rs-architecture].
- Interconnection of Content Delivery Networks (CDNi) [RFC6707] through the establishment and resizing of connections between content distribution networks. Similarly, ABNO can coordinate inter-data center connections.
- Network resource coordination to automate provisioning, 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 coordinating 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. ABNO is compatible with pre-existing Network Management Station (NMS) and Operations Support System (OSS) as well as with more recent developments programmatic networks such as Software Defined Networking (SDN).

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 is provided. Where there is no current work that can be identified by the authors, a short description of the missing interface protocol is given in the Appendix.

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 over functional interfaces that are, themselves, architectural concepts.

2.2 Implementation of the Architecture

It needs to be strongly emphasized that this document describes a functional architecture. It is not a software design. Thus, it is not intended that this architecture constrain implementations. 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.

An implementation of this architecture may make several important decisions about the functional components:

- Multiple functional components may be grouped together into one software component such that all of the functions are bundled and only the external interfaces are exposed. This may have distinct advantages for fast paths within the software, and can reduce inter-process communication overhead.

For example, an active, stateful PCE could be implemented as a single server combining the ABNO components of the PCE, the Traffic Engineering Database, the LSP Database, and the Provisioning Manager (see Section 2.3).

- The functional components could be distributed across separate processes, processors, or servers so that the interfaces are exposed as external protocols.

For example, the OAM Handler (see Section 2.3.1.6) could be presented on a dedicated server in the network that consumes all status reports from the network, aggregates them, correlates them, and then dispatches notifications to other servers that need to understand what has happened.

- There could be multiple instances of any or each of the components. That is, the function of a functional component could be partitioned across multiple software components with each responsible for handling a specific feature or a partition of the network.

For example, there may be multiple Traffic Engineering Databases (see Section 2.3.1.8) in an implementation with each holding the

topology information of a separate network domain (such as a network layer or an Autonomous System). Similarly there could be multiple PCE instances each processing on a different Traffic Engineering Database, and potentially distributed on different servers under different management control. As a final example, there could be multiple ABNO Controllers each with capability to support different classes of application or application service.

The purpose of the description of this architecture is to facilitate different implementations while offering interoperability between implementations of key components and easy interaction with the applications and with the network devices.

2.3 Generic ABNO Architecture

The following diagram illustrates the ABNO architecture. The components and functional interfaces are discussed in Sections 2.3.1 2.3.2 respectively. The use cases described in Section 3 show how different components are used selectively to provide different services.

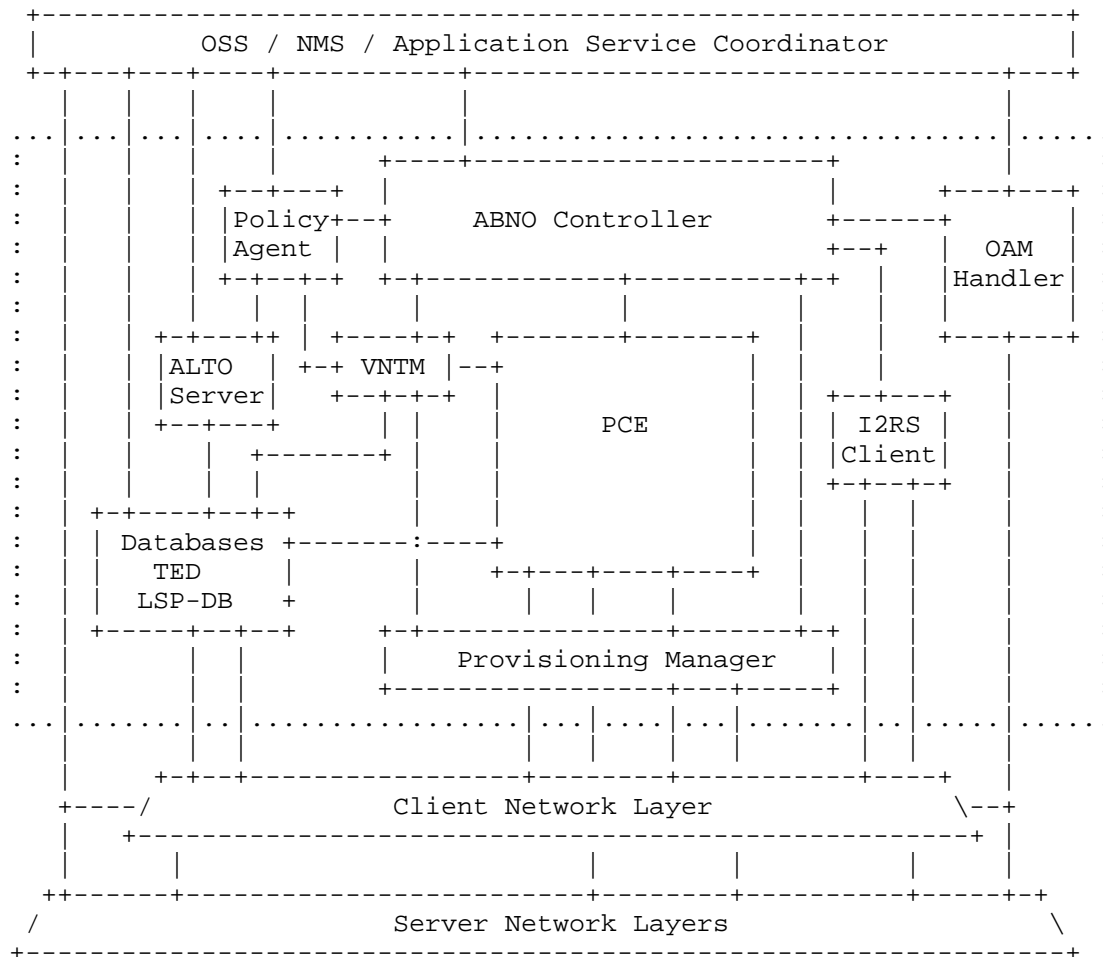


Figure 1: Generic ABNO Architecture

2.3.1 ABNO Components

This section describes the functional components shown as boxes in Figure 1. The interactions between those components, the functional interfaces, are described in Section 2.3.2.

2.3.1.1 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) and Label Switched Path Database (LSP-DB) 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.

2.3.1.2 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 a 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. In practice, the function of the Application Service Coordinator may be distributed across multiple applications or servers.

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

2.3.1.3 ABNO Controller

The ABNO Controller is the main gateway to the network for the NMS, OSS, and 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. It is the point of attachment, and invokes the right components in the right order.

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

2.3.1.4 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 ABNO 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 responsible 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, the Alto Server, and the Virtual Network Topology Manager (VNTM), it will also interact with a number of other components and the network elements themselves. For example, the Path Computation Element (PCE) will be a Policy Enforcement Point (PEP) [RFC2753] as described in [RFC5394], and the Interface to the Routing System (I2RS) Client will also be a PEP as noted in [I-D.ietf-i2rs-architecture].

2.3.1.5 Interface to the Routing System (I2RS) Client

The Interface to the Routing System (I2RS) is described in [I-D.ietf-i2rs-architecture]. The interface provides a programmatic way to access (for read and write) the routing state and policy information on routers in the network.

The I2RS Client is introduced in [I-D.ietf-i2rs-problem-statement]. Its purpose is to manage information requests across a number of routers (each of which runs an I2RS Agent) and coordinate setting or gathering state to/from those routers.

2.3.1.6 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.

2.3.1.7 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.ietf-pce-stateful-pce] and [I-D.crabbe-pce-pce-initiated-lsp]. This function may directly access the network elements, or may be channelled through the Provisioning 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.

2.3.1.8 Databases

The ABNO Architecture includes a number of databases that contain information stores for use by the system. The two main databases are the Traffic Engineering Database (TED) and the LSP Database (LSP-DB), but there may be a number of other databases to contain information about topology (ALTO Server), policy (Policy Agent), services (ABNO Controller), etc.

2.3.1.8.1 Traffic Engineering Database (TED)

The Traffic Engineering Database (TED) is a 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].

2.3.1.8.2 LSP Database

The LSP Database (LSP-DB) is a data store of information about LSPs that have been set up in the network, or that could be established. The information stored includes the paths and resource usage of the LSPs.

The LSP-DB may be built from information generated locally . For example, when LSPs are provisioned, the LSP-DB can be updated. The database can also be constructed from information gathered from the network by polling or reading the state of LSPs that have already been set up.

The main use of the LSP-DB within the ABNO architecture is to enhance the planning and optimization of LSPs. New LSPs can be established to be path-disjoint from other LSPs in order to offer protected services; LSPs can be rerouted in order to put them on more optimal paths or to make network resources available for other LSPs; LSPs can

be rapidly repaired when a network failure is reported; LSPs can be moved onto other paths in order to avoid resources that have planned maintenance outages.

2.3.1.8.3 Shared Risk Link Group (SRLG) Databases

The TED may, itself, be supplemented by SRLG information that assigns to each network resource one or more identifiers that associates the resource with other resources in the same TED that share the same risk of failure.

While this information can be highly useful, it may be supplemented by additional detailed information maintained in a separate database and indexed using the SRLG identifier from the TED. Such a database can interpret SRLG information provided by other networks (such as server networks), can provide failure probabilities associated with each SRLG, can offer prioritization when SRLG-disjoint paths can't be found, and can correlate SRLGs between different server networks or between different peer networks.

2.3.1.8.4 Other Databases

There may be other databases that are built within the ABNO system and which are referenced when operating the network. These databases might include information about, for example, traffic flows and demands, predicted or scheduled traffic demands, links and node failure and repair history, network resources such as packet labels and physical labels (i.e., MPLS and GMPLS labels), etc.

As mentioned in Section 2.3.1.8.1, the TED may be enhanced by inventory information. It is quite likely in many networks that such an inventory is held in a separate database (the Inventory Database) that includes details of manufacturer, model, installation date, etc.

2.3.1.9 ALTO Server

The ALTO server provides network information to the application layer based on abstract maps of a network region. This information provides a simplified view, but it is useful to route the traffic of the application layers. ALTO services enable Service Providers to share information about network locations and costs between them. The selection criteria to choose between two locations may depend of information such as maximum bandwidth, minimum cross-domain traffic, lower cost to the user, etc.

As the ALTO protocol exchanges information between network and application layer, ALTO server defines Provider-defined Identifiers (PIDs) as the entrance points to the network. Each node in the

application layer is known as an End Point (EP). Each EP is assigned to a PID, because PIDs are the entry point of the application in the network. Typically, multiple EPs are grouped in a single PID. As defined in [I-D.ietf-alto-protocol], a PID can denote a subnet, a set of subnets, a metropolitan area, a PoP, etc. The ALTO server provides cost information between the EPs or the PIDs in its network region. Such network region can be a single domain or multiple networks, it is just the view that the ALTO server is exposing to the application layer.

2.3.1.10 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.

[RFC5623] describes how the VNTM may instantiate connections in the server-layer in support of connectivity in the client-layer. Within the ABNO architecture, the creation of new connections may be delegated to the Provisioning Manager as discussed in Section 2.3.1.11.

All of these actions and decisions are heavily influenced by policy, so the VNTM 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.

2.3.1.11 Provisioning Manager

The Provisioning 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 Provisioning Manager may act as an OpenFlow Controller [ONF].

See Section 2.3.2.6 for more details of the interactions between the Provisioning Manager and the network.

2.3.1.12 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 set of use cases for multi-layer networking.

2.3.2 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. Appendix A notes the interfaces where more protocol specification may be needed.

2.3.2.1 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 [ONF]
- PCEP [I-D.crabbe-pce-pce-initiated-lsp].

[Editor note: need to add the correct TMF/MTOSI interface with a reference]

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

2.3.2.2 TED Construction from the Networks

As described in Section 2.3.1.8, the TED provides details of the capabilities and state 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.

2.3.2.3 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 2.3.2.2 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 protocol developed or adopted to satisfy the requirements of the Interface to the Routing System (I2RS) [I-D.ietf-i2rs-architecture] may be a suitable candidate because it is required to be able 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].

Note that, in general, any protocol and encoding that is suitable for presenting the TED as described in Section 2.3.2.4 will likely be suitable (or could be made suitable) for enabling write-access to the TED as described in this section.

2.3.2.4 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. ALTO server provides the cost between EPs or between PIDs, so the application layer can select which is the most appropriate connection for the information exchange between its application end points.
- 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.ietf-i2rs-architecture] will require a protocol that is capable of handling bulk routing information exchanges that would be suitable for exporting the TED. In this case it would make sense to have a standardized representation of the TED in a formal data modelling language such as YANG [RFC6020] so that an existing protocol could be used such as Netconf [RFC6241] or XMPP [RFC6120].

Note that export from the TED can be a full dump of the content (expressed in a suitable abstraction language) as described above, or it could be an aggregated or filtered set of data based on policies or specific requirements. Thus, the relationships shown in Figure 1 may be a little simplistic in that the ABNO Controller may also be involved in preparing and presenting the TED information over a north-bound interface.

2.3.2.5 Path Computation Requests from the Network

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.

2.3.2.6 Provisioning Manager Control of Networks

As described in Section 2.3.1.11, the Provisioning 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 Provisioning 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 Provisioning Manager) to establish and maintain the state of a label switch such as an MPLS switch.
- OpenFlow [ONF] is a communications protocol that gives an OpenFlow Controller (such as the Provisioning 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 Provisioning Manager 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 2.3.2.8 [I-D.ietf-netmod-routing-cfg].
- [Editor note: need to add the correct TMF interface with a reference]
- 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 Provisioning Manager 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 is the Provisioning Manager. Other controllers like the the VNTM and the ABNO Controller use the services of the Provisioning Manager to isolate the twin functions of computing and requesting paths from the provisioning mechanisms in place with any given network.

Note that 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.

2.3.2.7 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 2.3.2.2.
- 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].
- OAM can be commissioned and the results inspected by the OAM Handler as described in Section 2.3.2.13.
- A number of ABNO components may make inquiries and inspect network state through a variety of techniques including I2RS, Netconf, or SNMP.

2.3.2.8 Controlling The Routing System

As discussed in Section 2.3.1.5, 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.

2.3.2.9 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].

2.3.2.10 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 protocol selected or designed to satisfy I2RS will provide suitable features (see Section 2.3.2.8) or an extension could be made to the PCEP Notify message (PCNtf) [RFC5440].

2.3.2.11 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 protocol selected or designed to satisfy I2RS will provide suitable features (see Section 2.3.2.8).

2.3.2.12 Policy Interfaces

As described in Section 2.3.1.4 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.

Other policy-related function in the system might include the policy behavior of the routing and forwarding system such as:

- ECMP behavior
- Classification of packets onto LSPs.

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 and to determine which are internal interfaces and which may be external and so in need of a protocol specification. There is some discussion that the I2RS protocol may support the configuration and manipulation of policies.

2.3.2.13 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 2.3.2.1 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 [RFC7011] 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 Cases

This section provides a number of examples of how the ABNO architecture can be applied to provide application-driven 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 MPLS service across multiple Autonomous Systems (ASes). Consider the simple network topology shown in Figure 2. The three ASes (ASa, ASb, and ASc) are connected at 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.

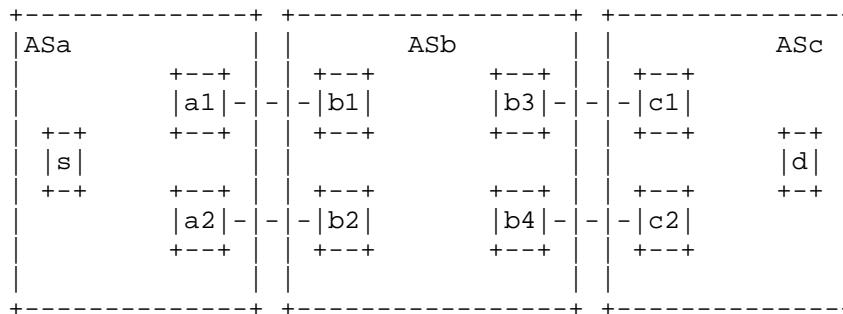


Figure 2: Inter-AS Domain Topology with H-PCE (Parent PCE)

The following steps are performed to deliver the service within the ABNO architecture.

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.

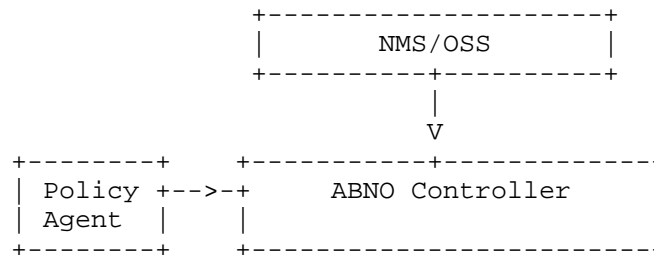


Figure 3: ABNO Request Management

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.

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.

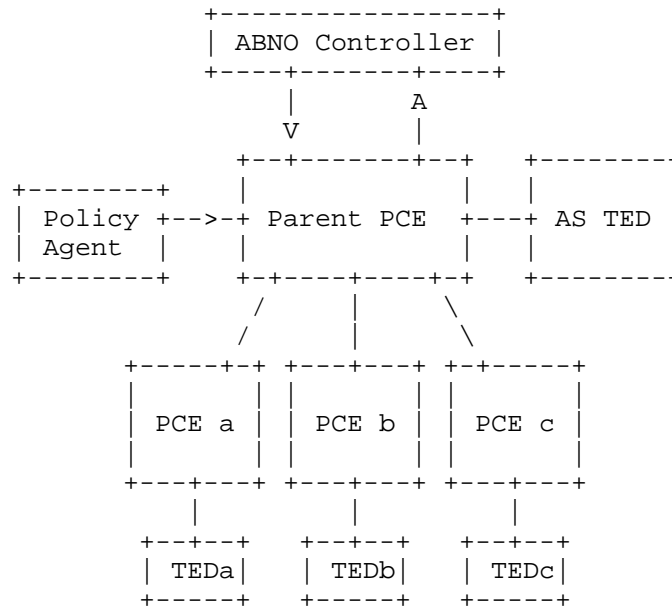


Figure 4: Path Computation Request with Hierarchical PCE

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 Provisioning Manager to establish the end-to-end LSP. As described in Section 2.3.2.6 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.

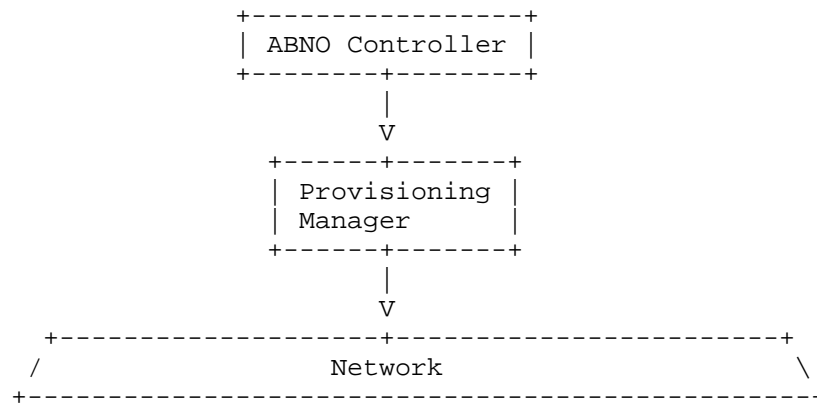


Figure 5: Provisioning the End-to-End LSP

3b. Provisioning through Programming Network Resources

Another option is that the LSP is provisioned hop by hop from the Provisioning Manager using a mechanism such as ForCES [RFC5810] or OpenFlow [ONF] as described in Section 2.3.2.6. In this case, the picture is the same as shown in Figure 5. The interaction between the ABNO Controller and the

Provisioning Manager will be PCEP or Netconf as described in option 3a., and the Provisioning 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 2.3.1.7 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 Provisioning Manager still programs the individual network elements as described in 3b.

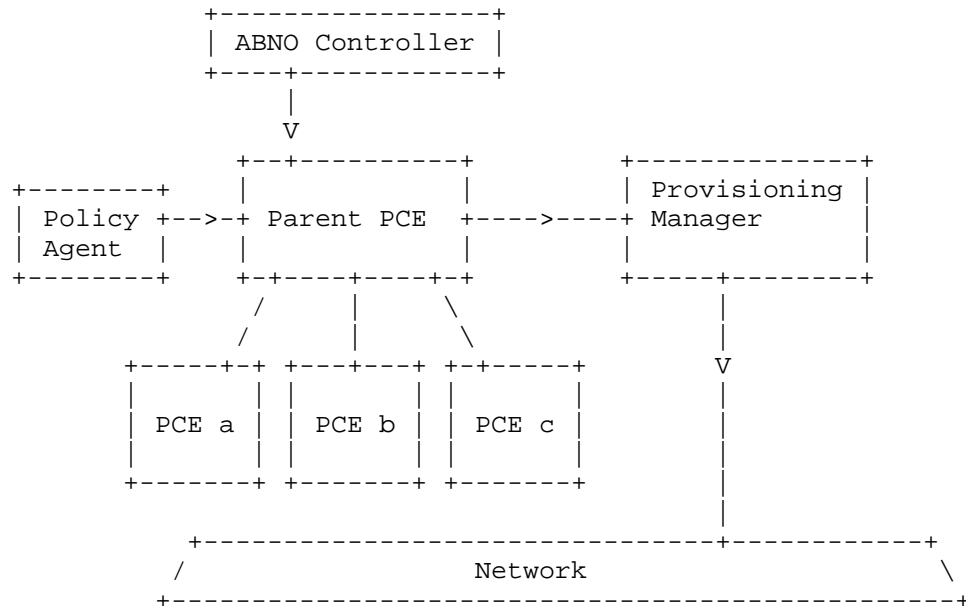


Figure 6: LSP Provisioning with an Active PCE

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 Provisioning Manager to program the inter-AS connections using ForCES or OpenFlow and the LSP segments are stitched together following the ideas described in [RFC5150].

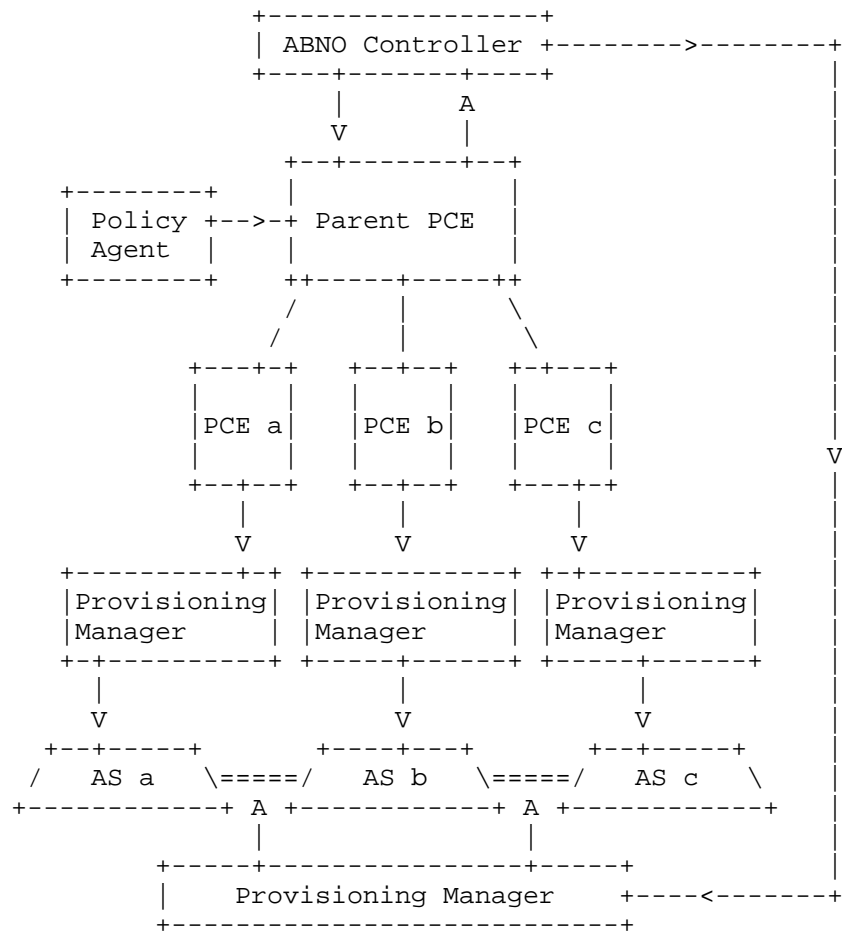


Figure 7: LSP Provisioning With Active Child PCEs and Stitching

4. Verification of Service

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. The delivery of the service may be further checked through auditing the network as described in 2.3.2.7.

3.2 Multi-Layer Networking

Networks are typically constructed using multiple layers. These layers represent separations of administrative regions or of technologies, 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.

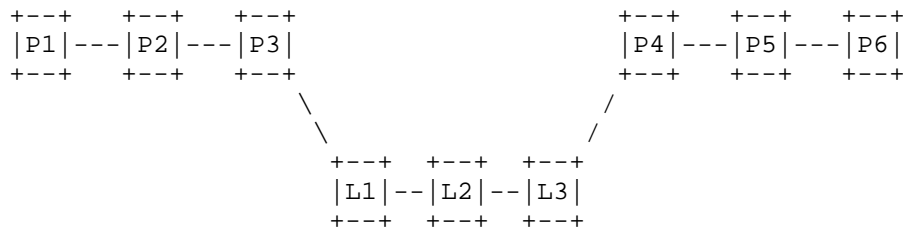


Figure 8: A Multi-Layer Network

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.

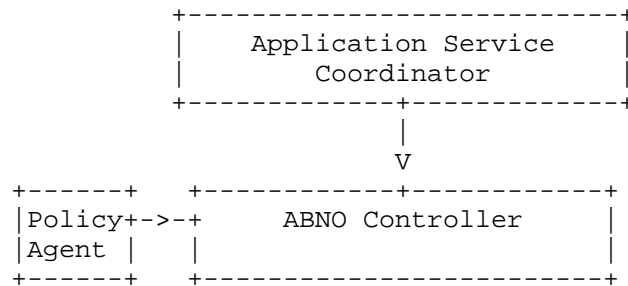


Figure 9: Application Service Coordinator Request Management

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.

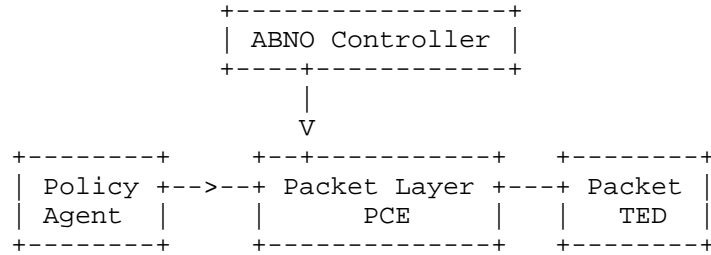


Figure 10: Path Computation Request

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.

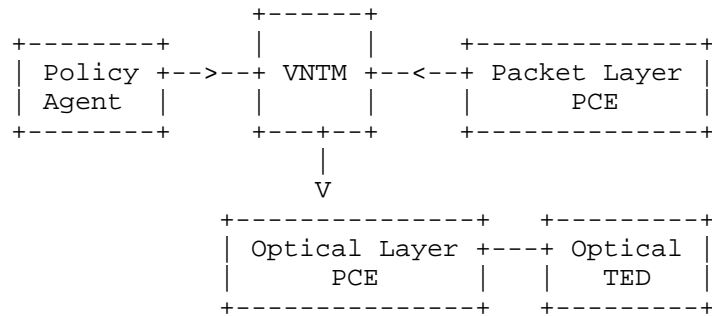


Figure 11: Invocation of VNTM and Optical Layer Path Computation

4. 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 2.3.2.6.

5. 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.

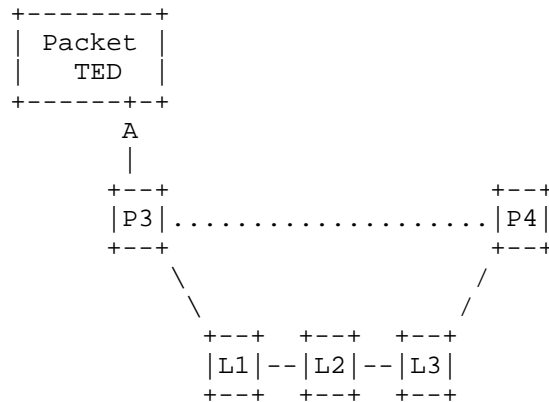


Figure 12: Advertisement of a New Virtual Link

6. 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.

7. 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 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.

3.2.1 Data Center (DC) Interconnection across MLNs

In order to support new and emerging cloud-based applications, such as real-time data backup, virtual machine migration, server clustering or load reorganization, the dynamic provisioning and allocation of IT resources and the interconnection of multiple, remote Data Centers (DC) is a growing requirement.

These operations require traffic being delivered between data centers, and, typically, the connections providing such inter-DC connectivity are provisioned using static circuits or dedicated leased lines, leading to an inefficiency in terms of resource utilization. Moreover, a basic requirement is that such a group of remote DCs can be operated logically as one.

In such environments, the data plane technology is operator and provider dependent. Their customers may rent LSC, PSC or TDM services, and the application and usage of the ABNO architecture and Controller enables the required dynamic end to end network service provisioning, regardless of underlying service and transport layers.

Consequently, the interconnection of remote DCs may involve the operation, control and management of heterogeneous environments, namely, each DC site and the metro-core network segment used to interconnect them, with regard to not only the underlying data plane technology, but also the control plane. For example, each DC site or domain could be controlled locally in a centralized way (e.g. via OpenFlow [ONF]), whereas the metro-core transport infrastructure is controlled by GMPLS. Although OpenFlow is specially adapted to single domain intra-data center networks (packet level control, lots of routing exceptions), a standardized GMPLS based architecture would enable dynamic optical resources allocation and restoration in multi-domain (e.g., multi-vendor) core networks interconnecting distributed data centers.

The application of an ABNO architecture and related procedures would involve the following aspects:

1. Request Application Service Coordinator or NMS

The ABNO Controller receives a request from the Application Service Coordinator or from the NMS, in order to create a new end-to-end connection between two end points. the actual addressing of these end points is discussed in the next section. The ABNO Controller asks the PCE for a path between these two endpoints, after considering any applicable policy as defined by the Policy Agent (see Figure 1).

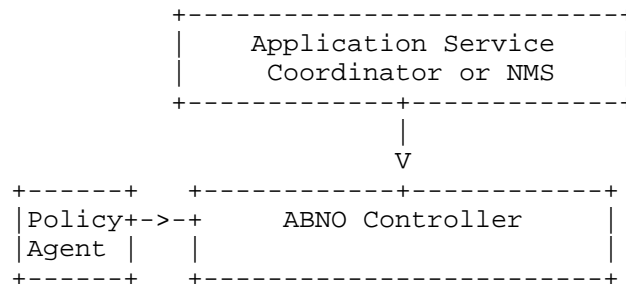


Figure 14: Application Service Coordinator Request Management

2. Cross-Stratum Addressing Mapping

In order to compute an end to end path, the PCE needs to have a unified view of the overall topology, which means that it has to consider and identify the actual endpoints with regard to the client network addresses. The ABNO Controller and/or the PCE may need to translate or map addresses from different address spaces. Depending on how the topological information is disseminated and gathered, there are two possible scenarios:

- a. The Application Layer knows Client Network Layer. Entities belonging to the application layer may have an interface with the TED or with an ALTO server, allowing the mapping of the high level endpoints to network addresses. The Application Layer may have an interface with TEDs or with ALTO server, it may know which are the client network layer addresses, where DCs are connected. This address correlation can be done via manual configuration or any other mechanism which is out of the scope of this draft. In this scenario, request from NMS or Application layer contains addresses in the client layer network. Therefore, when ABNO request to PCE for a path between these two end points, PCE can compute the path and continue the work-flow talking with the provisioning manager.
- b. Application Layer knows Server Network Layer. In this case, when ABNO asks PCE for a path, there is no route between two end

points. Similarly to use case in section 3.1, PCE asks to the VNTM to create a new connection between two addresses in the Server Network Layer. As VNT can access to TED information and this mapping have to exist, VNT can determine which are Client Layer addresses and continue with provisioning process.

3. Provisioning Process

Once the path has been obtained, the provisioning manager receives a high level provisioning request to provision the service. Since, in the considered use case, the network elements are not necessarily configured using the same protocol, the end to end path is split into segments, and the ABNO Controller coordinates or orchestrates the establishment by adapting and/or translating the abstract provisioning request to concrete segment requests, by means of a VNTM or PCE, which issue the corresponding commands or instructions. The provisioning may involve configuring the data plane elements directly or delegating the establishment of the underlying connection to a dedicated control plane instance, responsible for that segment.

The provisioning manager needs to know which technology is used for the actual provisioning at each segment, by ether manual configuration or discovery. Once the technology is selected, this configuration process can follow the steps.

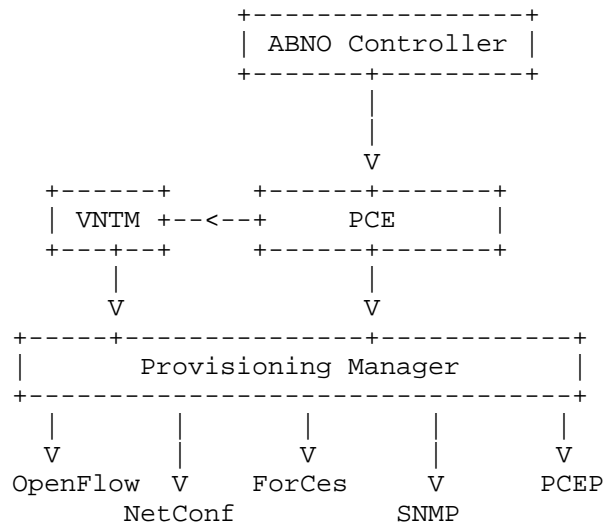


Figure 15: Provisioning Process

4. Verification and Notification of Service Fulfillment

Once the end-to-end connectivity service has been provisioned, and after the verification of the correct operation of the service, the ABNO Controller needs to notify the Application Service Coordinator or NMS.

3.3 Make-Before-Break

A number of different services depend on the establishment of a new LSP so that traffic supported by an existing LSP can be switched without disruption. This section describes those use cases, presents a generic model for make-before-break within the ABNO architecture, and shows how each use case can be supported by using elements of the generic model.

3.3.1 Make-Before-Break for Re-optimization

Make-before-break is a mechanism supported in RSVP-TE signaling where a new LSP is set up before the LSP it replaces is torn down [RFC3209]. This process has several benefits in situations such as re-optimization of in-service LSPs.

The process is simple, and the example shown in Figure 16 utilizes a stateful PCE [I-D.ietf-pce-stateful-pce] to monitor the network and take re-optimization actions when necessary. In this process a service request is made to the ABNO Controller by a requester such as the OSS. The service request indicates that the LSP should be re-optimized under specific conditions according to policy. This allows the ABNO Controller to manage the sequence and prioritization of re-optimizing multiple LSPs using elements of Global Concurrent Optimization (GCO) as described in Section 3.4, and applying policies across the network so that, for instance, LSPs for delay-sensitive services are re-optimized first.

The ABNO Controller commissions the PCE to compute and set up the initial path.

Over time, the PCE monitors the changes in the network as reflected in the TED, and according to the configured policy may compute and set up a replacement path, using make-before-break within the network.

Once the new path has been set up and the Network reports that it is in use correctly, PCE tears down the old path and may report the re-optimization event to the ABNO Controller.

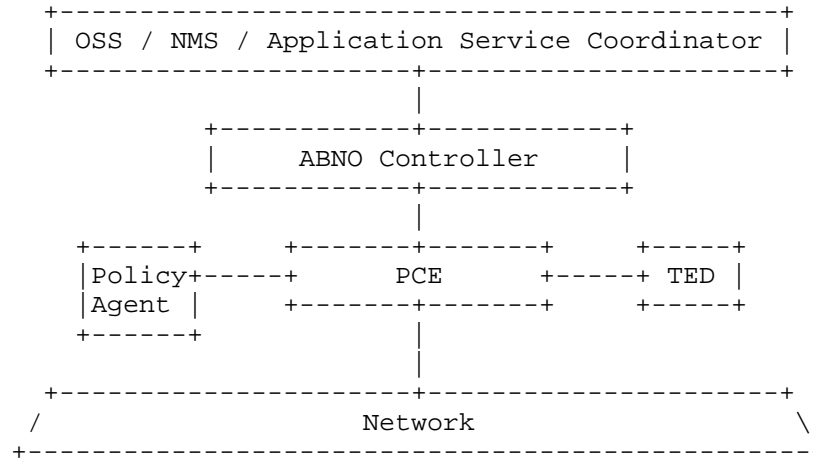


Figure 16: The Make-Before-Break Process

3.3.2 Make-Before-Break for Restoration

Make-before-break may also be used to repair a failed LSP where there is a desire to retain resources along some of the path, and where there is the potential for other LSPs to "steal" the resources if the failed LSP is torn down first. Unlike the example in Section 3.3.1, this case is service-interrupting, but that arises from the break in service introduced by the network failure. Obviously, in the case of a point-to-multipoint LSP, the failure might only affect part of the tree and the disruption will only be to a subset of the destination leafs so that a make-before-break restoration approach will not cause disruption to the leafs that were not affected by the original failure.

Figure 17 shows the components that interact for this use case. A service request is made to the ABNO Controller by a requester such as the OSS. The service request indicates that the LSP may be restored after failure and should attempt to reuse as much of the original path as possible.

The ABNO Controller commissions the PCE to compute and set up the initial path. The ABNO Controller also requests the OAM Handler to initiate OAM on the LSP and to monitor the results.

At some point the network reports a fault to the OAM Handler which notifies the ABNO Controller.

The ABNO Controller commissions the PCE to compute a new path, re-using as much of the original path as possible, and PCE sets up the new LSP.

Once the new path has been set up and the Network reports that it is in use correctly, the ABNO Controller instructs the PCE to tear down the old path.

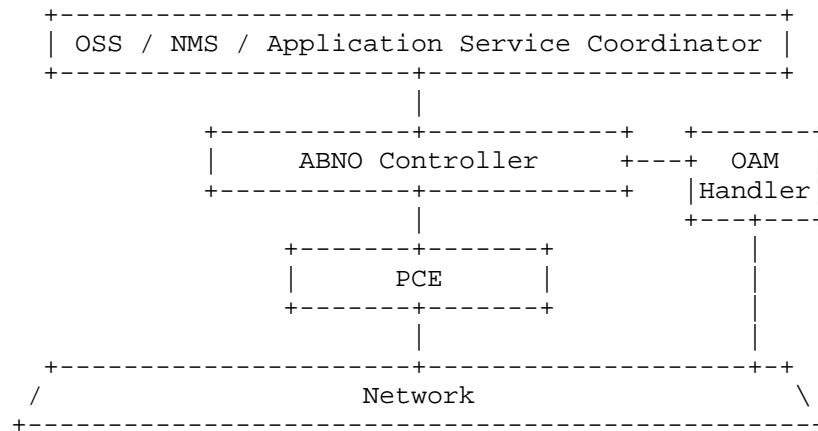


Figure 17: The Make-Before-Break Restoration Process

3.3.3 Make-Before-Break for Path Test and Selection

In a more complicated use case, an LSP may be monitored for a number of attributes such as delay and jitter. When the LSP falls below a threshold, the traffic may be moved to another LSP that offers the desired (or at least a better) quality of service. To achieve this, it is necessary to establish the new LSP and test it, and because the traffic must not be interrupted, make-before-break must be used.

Moreover, it may be the case that no new LSP can provide the desired attributes, and that a number of LSPs need to be tested so that the best can be selected. Furthermore, even when the original LSP is set up, it could be desirable to test a number of LSPs before deciding which should be used to carry the traffic.

Figure 18 shows the components that interact for this use case. Because multiple LSPs might exist at once, a distinct action is needed to coordinate which one carries the traffic, and this is the job of the I2RS Client acting under the control of the ABNO

Controller.

The OAM Handler is responsible for initiating tests on the LSPs and for reporting the results back to the ABNO Controller. The OAM Handler can also check end-to-end connectivity test results across a multi-domain network even when each domain runs a different technology. For example, an end-to-end might be achieved by stitching together an MPLS segment, an Ethernet/VLAN segment, and an IP etc.

Otherwise, the process is similar to that for re-optimization discussed in Section 3.3.1.

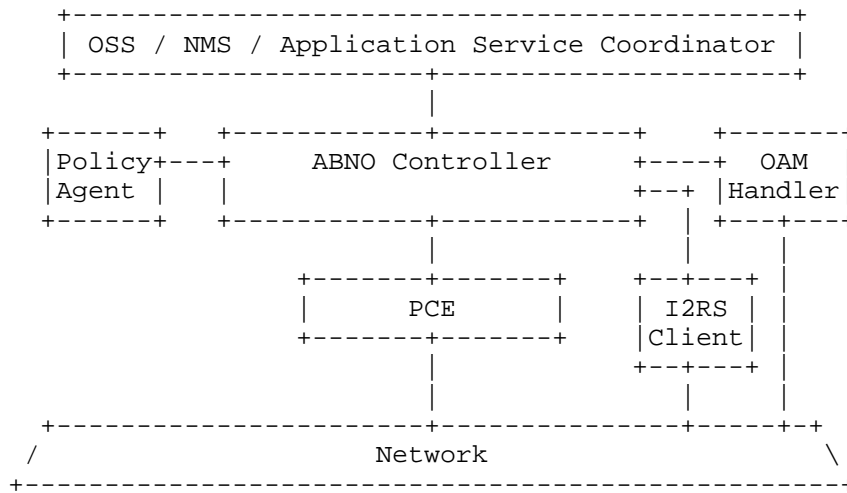


Figure 18: The Make-Before-Break Path Test and Selection Process

The pseudo-code that follows gives an indication of the interactions between ABNO components.

OSS requests quality-assured service

:Label1

DoWhile not enough LSPs (ABNO Controller)

 Instruct PCE to compute and provision the LSP (ABNO Controller)

 Create the LSP (PCE)

EndDo

:Label2

```

DoFor each LSP (ABNO Controller)
  Test LSP (OAM Handler)
  Report results to ABNO Controller (OAM Handler)
EndDo

Evaluate results of all tests (ABNO Controller)
Select preferred LSP and instruct I2RS client (ABNO Controller)
Put traffic on preferred LSP (I2RS Client)

DoWhile too many LSPs (ABNO Controller)
  Instruct PCE to tear down unwanted LSP (ABNO Controller)
  Tear down unwanted LSP (PCE)
EndDo

DoUntil trigger (OAM controller, ABNO Controller, Policy Agent)
  keep sending traffic (Network)
  Test LSP (OAM Handler)
Endif
EndDo

If there is already a suitable LSP (ABNO Controller)
  GoTo Label2
Else
  GoTo Label1
EndIf

```

3.4 Global Concurrent Optimization

Global Concurrent Optimization (GCO) is defined in [RFC5557] and represents a key technology for maximizing network efficiency by computing a set of traffic engineered paths concurrently. A GCO path computation request will simultaneously consider the entire topology of the network, and the complete set of new LSPs together with their respective constraints. Similarly, GCO may be applied to recompute the paths of a set of existing LSPs.

GCO may be requested in a number of scenarios. These include:

- o Routing of new services where the PCE should consider other services or network topology.
- o A reoptimization of existing services due to fragmented network resources or sub-optimized placement of sequentially computed services.
- o Recovery of connectivity for bulk services in the event of a catastrophic network failure.

A service provider may also want to compute and deploy new bulk services based on a predicted traffic matrix. The GCO functionality and capability to perform concurrent computation provides a significant network optimization advantage, thus utilizing network resources optimally and avoiding blocking.

The following use case shows how the ABNO architecture and components are used to achieve concurrent optimization across a set of services.

3.4.1 Use Case: GCO with MPLS LSPs

When considering the GCO path computation problem, we can split the GCO objective functions into three optimization categories, these are:

- o Minimize aggregate Bandwidth Consumption (MBC).
- o Minimize the load of the Most Loaded Link (MLL).
- o Minimize Cumulative Cost of a set of paths (MCC).

This use case assumes the GCO request will be offline and be initiated from an NMS/OSS, that is it may take significant time to compute the service, and the paths reported in the response may want to be verified by the user before being provisioned within the network.

1. Request Management

The NMS/OSS issues a request for new service connectivity for bulk services. The ABNO Controller verifies that the NMS/OSS has sufficient rights to make the service request and apply a GCO attribute with a request to Minimize aggregate Bandwidth Consumption (MBC).

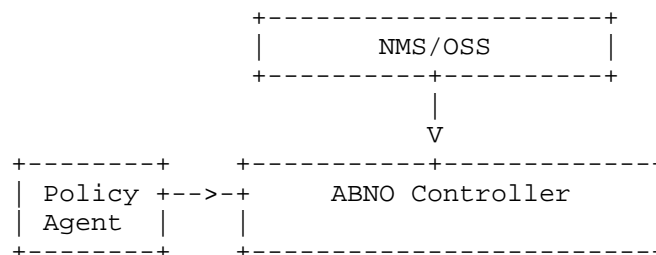


Figure 19: NMS Request to ABNO Controller

- 1a. Each service request has a source, destination and bandwidth request. These service requests are sent to the ABNO Controller and categorized as a GCO. The PCE uses the appropriate policy for the request and consults the TED for the packet layer.

2. Service Path Computation in the Packet Layer

To compute a set of services for the GCO application, PCEP supports synchronization vector (SVEC) lists for synchronized dependent path computations as defined in [RFC5440] and described in [RFC6007].

- 2a. The ABNO Controller sends the bulk service request to the GCO-capable packet layer PCE using PCEP messaging. The PCE uses the appropriate policy for the request and consults the TED for the packet layer.

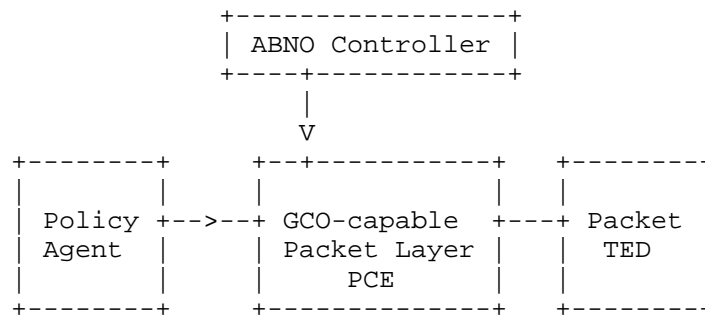


Figure 20: Path Computation Request from GCO-capable PCE

- 2b. Upon receipt of the bulk (GCO) service requests, the PCE applies the MBC objective function and computes the services concurrently.
- 2c. Once the requested GCO service path computation completes, the PCE sends the resulting paths back to the ABNO Controller as a PCEP response. The response includes a fully computed explicit path for each service (TE LSP).

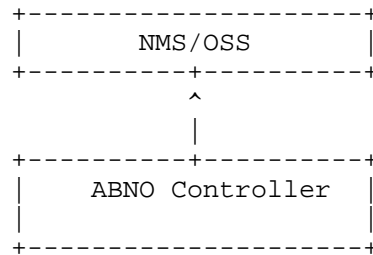


Figure 21: ABNO Sends Solution to the NMS/OSS

3. The concurrently computed solution received from the PCE is sent back to the NMS/OSS by the ABNO Controller. The NMS/OSS user can then check the candidate paths and either provision the new services, or save the solution for deployment in the future.

3.5 Adaptive Network Management (ANM)

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. Server-layer transport network resources, such as Optical Transport Network (OTN) time-slicing [G.709], or the fine granularity grid of wavelengths with variable spectral bandwidth (flexi-grid) [G.694.1], can be manipulated to meet current and projected demands in a model called Elastic Optical Networks (EON).

EON provides spectrum-efficient and scalable transport by introducing flexible granular grooming in the optical frequency domain. This is achieved using arbitrary contiguous concatenation of optical spectrum that allows creation of customized bandwidth. This bandwidth is defined in slots of 12,5GHz.

Adaptive Network Management (ANM) with EON allows appropriately-sized optical bandwidth to be allocated to an end-to-end optical path. In flexi-grid, the allocation is performed according to the traffic volume or following user requests, and can be achieved in a highly spectrum-efficient and scalable manner. Similarly, OTN provides an adaptive and elastic provisioning of bandwidth on top of wavelength switched optical networks (WSON).

To efficiently use optical resources, a system is required which can monitor network resources, and decide the optimal network configuration based on the status, bandwidth availability and user service. We call this ANM.

3.5.1. ANM Trigger

There are different reasons to trigger an adaptive network management process, these include:

- o Measurement: traffic measurements can be used in order to cause spectrum allocations that fit the traffic needs as efficiently as possible. This function may be influenced by measuring the IP router traffic flows, by examining traffic engineering or link state databases, by usage thresholds for critical links in the network, or by requests from external entities. Nowadays, network operators have active monitoring probes in the network, which store their results in the OSS. The OSS or OAM Handler components activate this measurement-based trigger, so the ABNO Controller would not be directly involved in this case.
- o Human: operators may request ABNO to run an adaptive network planning process via a NMS.
- o Periodic: adaptive network planning process can be run periodically to find an optimum configuration.

An ABNO Controller would receive a request from OSS or NMS to run an adaptive network manager process.

3.5.2. Processing request and GCO computation

Based on the human or periodic trigger requests described in the previous Section, the OSS or NMS will send a request to the ABNO Controller to perform EON-based GCO. The ABNO Controller will select a set of services to be reoptimized and choose an objective function that will deliver the best use of network resources. In making these choices, the ABNO Controller is guided by network-wide policy on the use of resources, the definition of optimization, and the level of perturbation to existing services that is tolerable.

Much as in Section 3.5, this request for GCO is passed to the PCE. The PCE could then consider the end-to-end paths and every channel's optimal spectrum assignment in order to satisfy traffic demands and optimize the optical spectrum consumption within the network.

The PCE will operate on the TED, but is likely to also be stateful so that it knows which LSPs correspond to which waveband allocations on which links in the network. Once PCE arrives at an answer, it returns a set of potential paths to the ABNO Controller which passes them on to the NMS or OSS to supervise/select the subsequent path set-up/modification process.

This exchange is shown in Figure 22. Note that the figure does not show the interactions used by the OSS/NMS for establishing or modifying LSPs in the network.

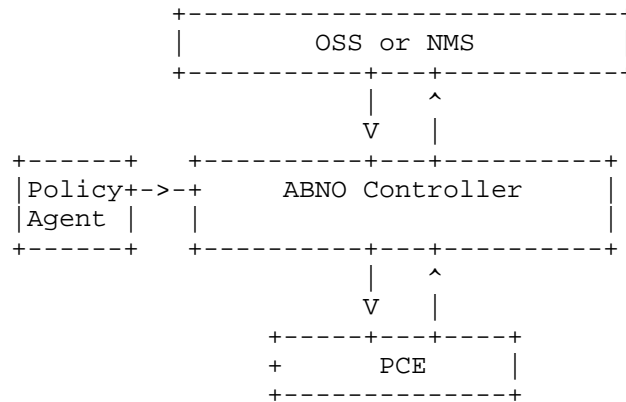


Figure 22: Adaptive Network Management with human intervention

3.5.3. Automated Provisioning Process

Although most of network operations are supervised by the operator, there are some actions, which may not require supervision, like a simple modification of a modulation format in a Bit rate variable transponder (BVT) (to increase the optical spectrum efficiency or reduce energy consumption). In this processes, where human intervention is not required, PCE sends provisioning manager new configuration to configure the network elements.

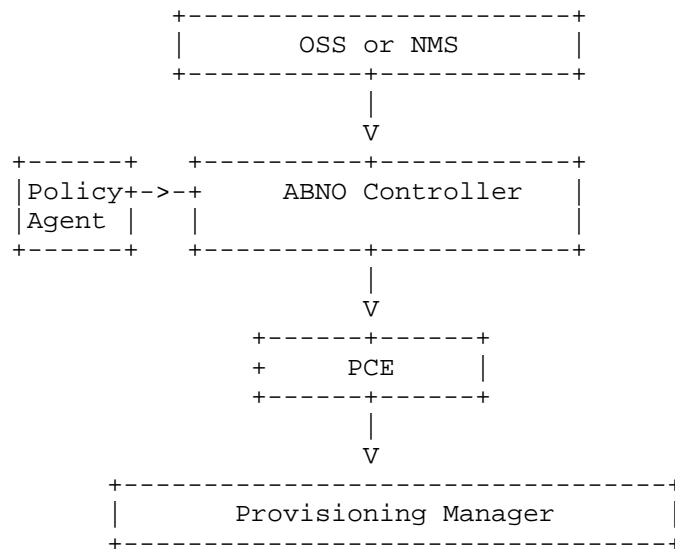


Figure 23: Adaptive Network Management without human intervention

3.6 Pseudowire Operations and Management

Pseudowires in an MPLS network [RFC3985] operate as a form of layered network over the connectivity provided by the MPLS network. The pseudowires are carried by LSPs operating as transport tunnels, and planning is necessary to determine how those tunnels are placed in the network and which tunnels are used by any pseudowire.

This section considers four use cases: multi-segment pseudowires, path-diverse pseudowires, path-diverse multi-segment pseudowires, and pseudowire segment protection. Section 3.6.5 describes the applicability of the ABNO architecture to these four use cases.

3.6.1 Multi-Segment Pseudowires

[RFC5254] described the architecture for multi-segment pseudowires. An end-to-end service, as shown in Figure 24, can consist of a series of stitched segments shown on the figure as AC, PW1, PW2, PW3, and AC. Each pseudowire segment is stitched at a 'stitching PE' (S-PE): for example, PW1 is stitched to PW2 at S-PE1. Each access circuit (AC) is stitched to a pseudowire segment at a 'terminating PE' (T-PE): for example, PW1 is stitched to the AC at T-PE1.

Each pseudowire segment is carried across the MPLS network in an LSP operating as a transport tunnel: for example, PW1 is carried in LSP1. The LSPs between PEs may traverse different MPLS networks with the

PEs as border nodes, or the PEs may lie within the same network such such that the LSPs each only span part of the network.

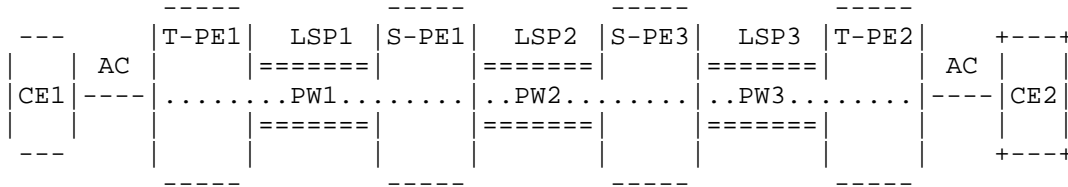


Figure 24 : Multi-Segment Pseudowire

While the topology shown in Figure 24 is easy to navigate, the reality of a deployed network can be considerably more complex. The topology in Figure 25 shows a small mesh of PEs. The links between the PEs are not physical links but represent the potential of MPLS LSPs between the PEs.

When establishing the end-to-end service between CE1 and CE2, some choice must be made about which PEs to use. In other words, a path computation must be made to determine the pseudowire segment 'hops', and then the necessary LSP tunnels must be established to carry the pseudowire segments that will be stitched together.

Of course, each LSP may itself require a path computation decision to route it through the MPLS network between PEs.

The choice of path for the multi-segment pseudowire will depend on such issues as:

- MPLS connectivity
- MPLS bandwidth availability
- pseudowire stitching capability and capacity at PEs
- policy and confidentiality considerations for use of PEs.

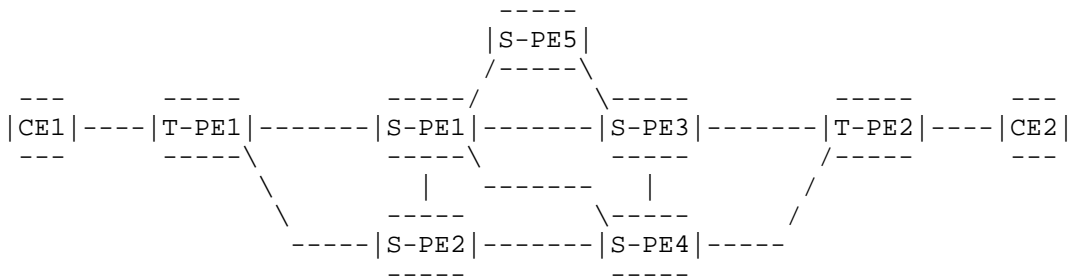


Figure 25 : Multi-Segment Pseudowire Network Topology

3.6.2 Path-Diverse Pseudowires

The connectivity service provided by a pseudowire may need to be resilient to failure. In many cases, this function is provided by provisioning a pair of pseudowires carried by path-diverse LSPs across the network as shown in Figure 26 (the terminology is inherited directly from [RFC3985]). Clearly, in this case, the challenge is to keep the two LSPs (LSP1 and LSP2) disjoint within the MPLS Network. This problem is not different from the normal MPLS path-diversity problem.

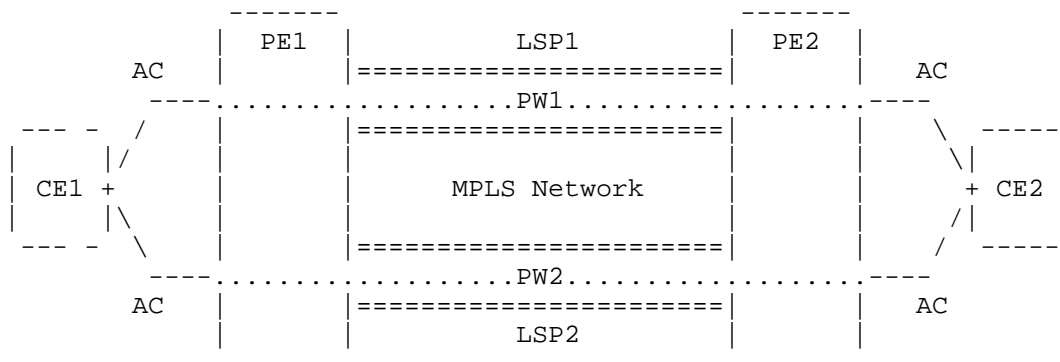


Figure 26 : Path-Diverse Pseudowires

The path-diverse pseudowire is developed in Figure 27 by the "dual-homing" of each CE through more than one PE. The requirement for LSP path diversity is exactly the same, but it is complicated by the LSPs having distinct end points. In this case, the head-end router (e.g., PE1) cannot be relied upon to maintain the path diversity through the signaling protocol because it is aware of the path of the only one of the LSPs. Thus some form of coordinated path computation approach is needed.

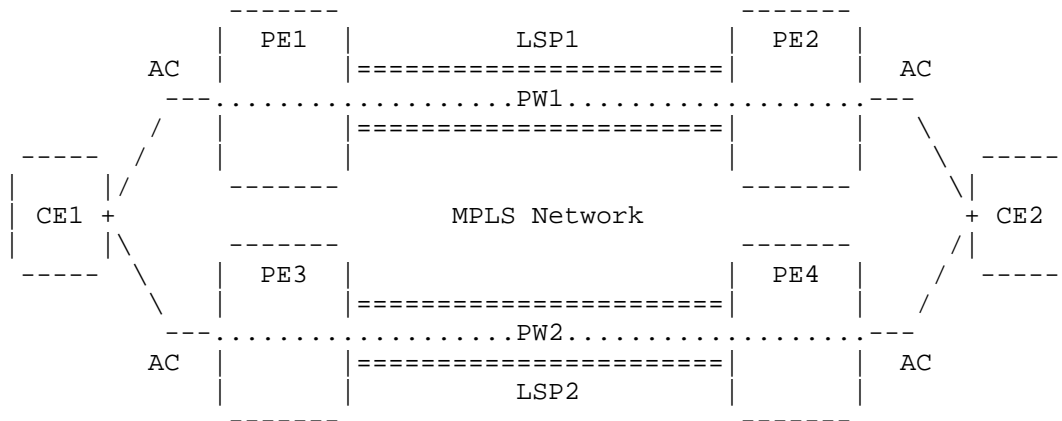


Figure 27 : Path-Diverse Pseudowires With Disjoint PEs

3.6.3 Path-Diverse Multi-Segment Pseudowires

Figure 28 shows how the services in the previous two sections may be combined to offer end-to-end diverse paths in a multi-segment environment. To offer end-to-end resilience to failure, two entirely diverse, end-to-end multi-segment pseudowires may be needed.

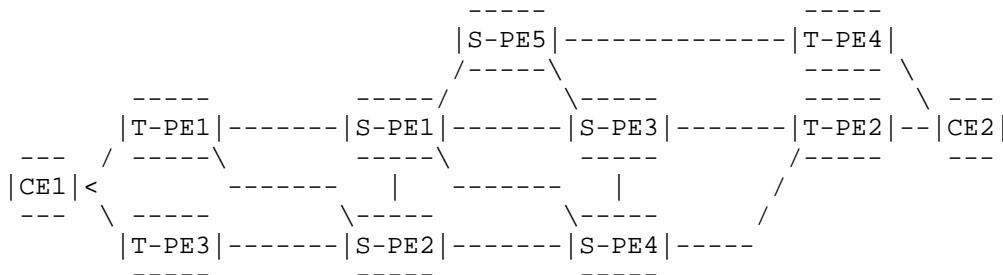


Figure 28 : Path-Diverse Multi-Segment Pseudowire Network Topology

Just as in any diverse-path computation, the selection of the first path needs to be made with awareness of the fact that a second, fully-diverse path is also needed. If a sequential computation was applied to the topology in Figure 28, the first path CE1,T-PE1,S-PE1, S-PE3,T-PE2,CE2 would make it impossible to find a second path that was fully diverse from the first.

But the problem is complicated by the multi-layer nature of the network. It is not enough that the PEs are chosen to diverse because the LSP tunnels between them might share links within the MPLS

network. Thus, a multi-layer planning solution is needed to achieve the desired level of service.

3.6.4 Pseudowire Segment Protection

An alternative to the end-to-end pseudowire protection service described in Section 3.6.3 can be achieved by protecting individual pseudowire segments or PEs. For example, in Figure 28, the pseudowire between S-PE1 and S-PE5 may be protected by a pair of stitched segments running between S-PE1 and S-PE5, and between S-PE5 and S-PE3. This is shown in detail in Figure 29.

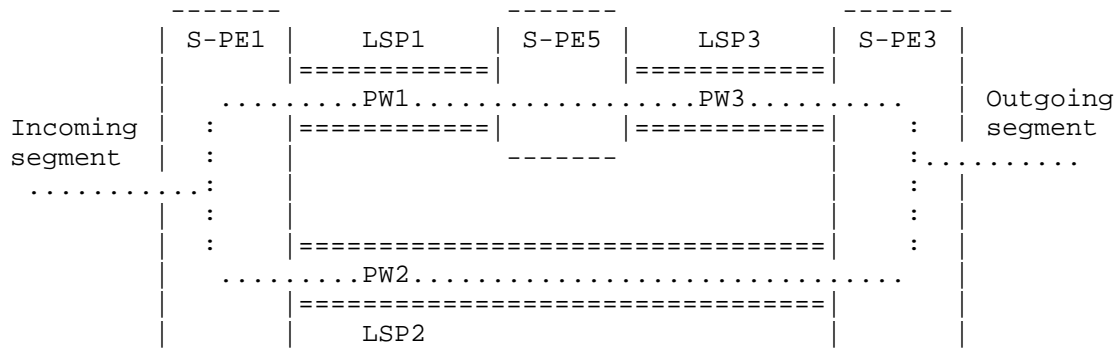


Figure 29 : Fragment of a Segment-Protected Multi-Segment Pseudowire

The determination of pseudowire protection segments requires coordination and planning, and just as in Section 3.6.5, this planning must be cognizant of the paths taken by LSPs through the underlying MPLS networks.

3.6.5 Applicability of ABNO to Pseudowires

The ABNO architecture lends itself well to the planning and control pseudowires in the use cases described above. The user or application needs a single point at which it requests services: the ABNO Controller. The ABNO Controller can ask a PCE to draw on the topology of pseudowire stitching-capable PEs as well as additional information regarding PE capabilities, such as load on PEs and administrative policies, and the PCE can use a series of TEDs or other PCEs for the underlying MPLS networks to determine the paths of the LSP tunnels. Then a number of different provisioning systems can be used to instantiate the LSPs and provision the pseudowires under the control of the Provisioning Manager. The ABNO Controller will use the I2RS Client to instruct the network devices about what

traffic should be placed on which pseudowires, and in conjunction with the OAM Handler can ensure that failure events are handled correctly, that service quality levels are appropriate, and that service protection levels are maintained.

In many respects, the pseudowire network forms an overlay network (with its own TED and provisioning mechanisms) carried by underlying packet networks. Further client networks (the pseudowire payloads) may be carried by the pseudowire network. Thus, the problem space being addressed by ABNO in this case is a classic multi-layer network.

3.7. Cross-Stratum Optimization (CSO)

Considering the term "stratum" to broadly differentiate the layers of most concern to the application and to the network in general, the need for Cross Stratum optimization (CSO) arises when the application stratum and network stratum need to be coordinated to achieve operational efficiency as well as resource optimization in both application and network stratum.

Data center based applications can provide a wide variety of services such as video gaming, cloud computing, and grid applications. High-bandwidth video applications are also emerging, such as remote medical surgery, live concerts, and sporting events.

This use-case for the ABNO architecture is mainly concerned with data center applications that make substantial bandwidth demands either in aggregate or individually. In addition these applications may need specific bounds on QoS related parameters such as latency and jitter.

3.7.1. Data Center Network Operation

Data centers come in a wide variety of sizes and configurations, but all contain compute servers, storage, and application control. Data centers offer application services to end-users such as video gaming, cloud computing and others. Since the data centers used to provide application services may be distributed around a network, the decisions about the control and management of application services, such as where to instantiate another service instance or to which data center a new client is assigned, can have a significant impact on the state of the network. Conversely the capabilities and state of the network can have a major impact on application performance.

These decisions are typically made by applications with very little or no information concerning the underlying network. Hence, such decisions may be sub-optimal from application and network resource utilization and quality of service objectives.

Cross-stratum optimization is the process of optimizing both the application experience and the network utilization by coordinating decisions in the application stratum and the network stratum. Application resources can be roughly categorized into computing resources, (i.e., servers of various types and granularities such as VMs, memory, and storage) and content (e.g., video, audio, databases, and large data sets). By network stratum we mean the IP layer and below (e.g., MPLS, SDH, OTN, WDM). The network stratum has resources that include routers, switches, and links. We are particularly interested in further unleashing the potential presented by MPLS and GMPLS control planes at the lower network layers in response to the high aggregate or individual demands from the application layer.

This use-case demonstrates that the ABNO architecture can allow cross-stratum application/network optimization for the data center use case. Other forms of cross-stratum optimization (for example, for peer-to-peer applications) are out of scope.

3.7.1.1. Virtual Machine Migration

A key enabler for data center cost savings, consolidation, flexibility and application scalability has been the technology of compute virtualization provided through Virtual Machines (VMs). To the software application a VM looks like a dedicated processor with dedicated memory and a dedicated operating system.

VMs offer not only a unit of compute power but also provide an "application environment" that can be replicated, backed up, and moved. Different VM configurations may be offered that are optimized for different types of processing (e.g., memory intensive, throughput intensive).

VMs may be moved between compute resources in a data center and could be moved between data centers. VM migration serves to balance load across data center resources and has several modes:

- (i) scheduled vs. dynamic;
- (ii) bulk vs. sequential;
- (iii) point-to-point vs. point-to-multi-point

While VM migration may solve problems of load or planned maintenance within a data center it can also be effective to reduce network load around the data center. But the act of migrating VMs especially between data centers can impact the network and other services that are offered.

For certain applications such as disaster recovery, bulk migration is required on the fly, which may necessitate concurrent computation and path setup dynamically.

Thus, application stratum operations must also take account of the situation in the network stratum even as the application stratum actions may be driven by the status of the network stratum.

3.7.1.2. Load Balancing

Application servers may be instantiated in many data centers located in different parts of the network. When an end-user makes a request an application request, a decision has to be made about which data center should host the processing and storage required to meet the request. One of the major drivers for operating multiple data centers (rather than one very large data center) is so that the application will run on a machine that is closer to the end-users and thus improve the user experience by reducing network latency. However, if the network is congested or the data center is overloaded this strategy can backfire.

Thus, among the key factors to be considered in choosing the server on which to instantiate a VM for an application include:

- The utilization of the servers in the data center
- The network loading conditions within a data center
- The network loading conditions between data centers
- The network conditions between the end-user and data center

Again, the choices made in the application stratum need to consider the situation in the network stratum.

3.7.2. Application of the ABNO Architecture

This section shows how the ABNO architecture is applicable to the cross-stratum data center issues described in Section 3.7.1.

Figure 30 shows a diagram of an example data center based application. A carrier network provides access for an end-user through provider edge (PE) node PE4. Three data centers (DC1, DC2, and DC3) are accessed through different parts of the network via PE1, PE2, and PE3.

The Application Service Coordinator receives information from the end-user about the services it wants, and converts this to service requests that it passes to the ABNO Controller. The end-user may already know which data center it wishes to use, the Application Service Controller may be able to make this determination, or the task of selecting the data center may be left to the ABNO Controller.

The ABNO controller examines the network resources using information gathered from the other ABNO components and uses those components to configure the network to support the end-user's needs.

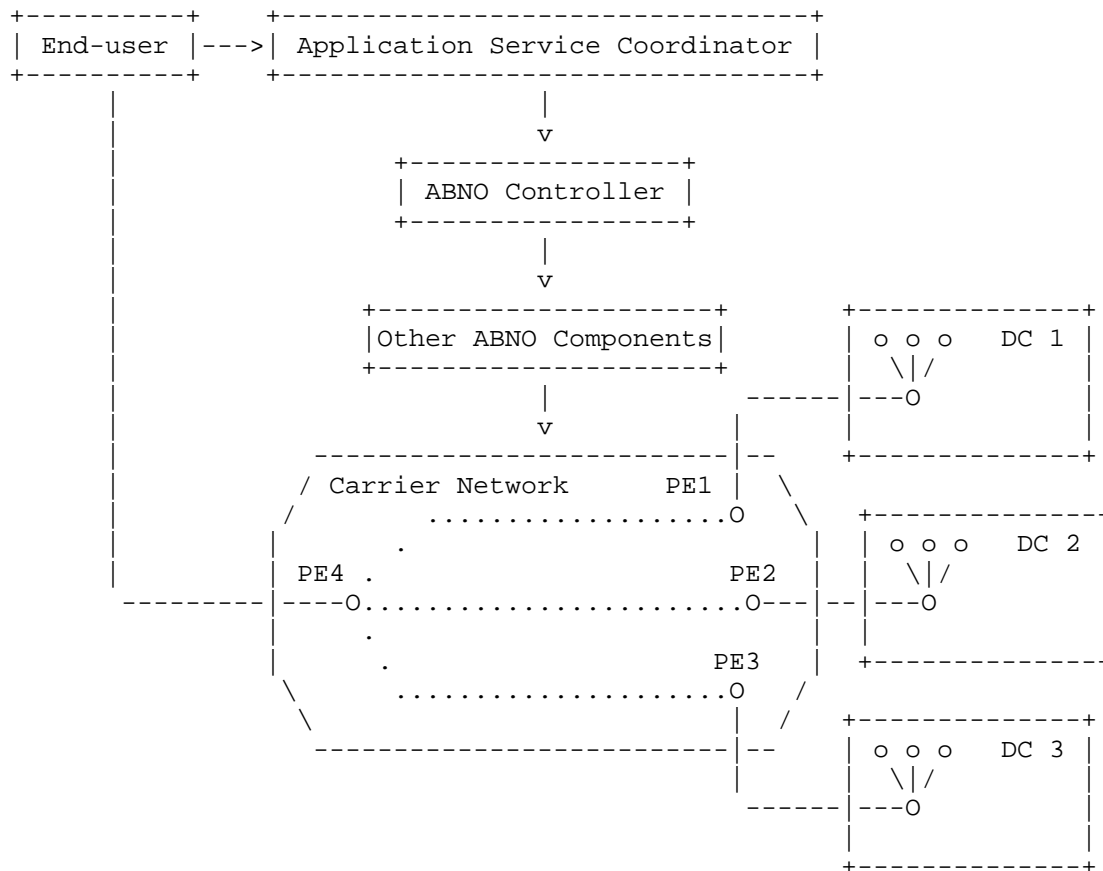


Figure 30 : The ABNO Architecture in the Context of Cross-Stratum Optimization for Data Centers

3.7.2.1. Deployed Applications, Services, and Products

The ABNO controller will need to utilize a number of components to realize the CSO functions described in Section 3.7.1.

The ALTO server provides information about topological proximity and appropriate geographical locations servers with respect to the underlying networks. This information can be used to optimize the selection of peer location which will help reduce the path of IP

traffic or can contain it within specific service providers' networks. ALTO in conjunction with the ABNO Controller and the Application Service Coordinator can address general problems such as the selection of application servers based on resource availability and usage of the underlying networks.

The ABNO Controller can also formulate a view of current network load from the TED and from the OAM Handler (for example, by running diagnostic tools that measure latency, jitter, and packet loss). This view obviously influences not just how paths from end-user to data center are provisioned, but can also guide the selection of which data center should provide the service and possibly even the points of attachment to be used by the end-user and to reach the chosen data center.

As already discussed, the combination of the ABNO Controller and the Application Service Coordinator will need to be able to select (and possibly migrate) the location of the VM that provides the service for the end-user. Since a common technique used to direct the end-user to the correct VM/server is to employ DNS redirection, an important capability of the ABNO controller will be to be able to program the DNS servers accordingly.

Furthermore, as already noted in other sections of this document, the ABNO Controller can coordinate the placement of traffic within the network to achieve load-balancing and to provide resilience to failures. These features can be used in conjunction with the functions discussed above, to ensure that the placement of new VMs, the traffic that they generate, and the load caused by VM migration can be carried by the network and do not disrupt existing services.

3.8 Other Potential Use Cases

This section serves as a place-holder for other potential use cases that might get documented in a future revision of this document.

3.8.1 Grooming and Regrooming

This use case could cover the following scenarios:

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

3.8.2 Bandwidth Scheduling

Bandwidth Scheduling consist of configuring LSPs based on a given

time schedule. This can be used to support maintenance or operational schedules or to adjust network capacity based on traffic pattern detection.

The ABNO framework provides the components to enable bandwidth scheduling solutions.

3.8.3 ALTO Server

A use case describing the ALTO server is needed.]

4. Survivability and Redundancy within the ABNO Architecture

[Editor's note: this section to be written with consideration of how the ABNO system survives the failure of individual components.]

5. Security Consideration

[Editor's note: this section to be written.]

6. Manageability Considerations

[Editor's note: this section to be written.]

7. IANA Considerations

This document makes no requests for IANA action.

8. Acknowledgements

Thanks for discussions and review are due to Ken Gray, Jan Medved, Nitin Bahadur, Diego Caviglia, Joel Halpern, Brian Field, Ori Gerstel, Daniele Ceccarelli, Diego Caviglia Cyril Margaria, and Jonathan Hardwick. Thanks to George Swallow for suggesting the existence of the SRLG database.

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

9. References

9.1. Informative References

[I-D.boucadair-connectivity-provisioning-profile]
Boucadair, M., Jacquenet, c., and N. Wang, "IP/MPLS Connectivity Provisioning Profile",
draft-boucadair-connectivity-provisioning-profile, 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-i2rs-architecture]
Atlas, A., Halpern, J., Hares, S., Ward, D., and T. Nadeau, "An Architecture for the Interface to the Routing System", draft-ietf-i2rs-architecture, work in progress.
- [I-D.ietf-i2rs-problem-statement]
Atlas, A., Nadeau, T., and D. Ward, "Interface to the Routing System Problem Statement", draft-ietf-i2rs-problem-statement, 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.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.
- [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.
- [RFC2753] Yavatkar, R., Pendarakis, D. and R. Guerin, "A Framework for Policy-based Admission Control", RFC2753, 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.
- [RFC3473] L. Berger et al., "Generalized Multi-Protocol Label Switching (GMPLS) Signaling Resource ReserVation Protocol-Traffic Engineering (RSVP-TE) Extensions", RFC 3473, January 2003.
- [RFC3630] Katz, D., Kompella, 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.
- [RFC3985] Bryant, S., Ed., and P. Pate, Ed., "Pseudo Wire Emulation Edge-to-Edge (PWE3) Architecture", RFC 3985, March 2005.
- [RFC4655] Farrel, A., Vasseur, J.-P., and Ash, J., "A Path Computation Element (PCE)-Based Architecture", RFC 4655, August 2006.
- [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] Shiimoto, 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.
- [RFC5254] Bitar, N., Bocci, M. and L. Martini, "Requirements for Multi-Segment Pseudowire Emulation Edge-to-Edge (PWE3)", RFC 5254, October 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, JL., 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.
- [RFC6007] I. Nishioka. and D. King., "Use of the Synchronization VEctor (SVEC) List for Synchronized Dependent Path Computations", RFC 6007, September 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.
- [RFC6120] P. Saint-Andre, "Extensible Messaging and Presence Protocol (XMPP): Core", RFC 6120, March 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.
- [RFC7011] Claise, B., Trammell, B., and Paitken, "Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of IP Traffic Flow Information", STD 77, RFC 7011, September 2013.
- [TL1] Telcorida, "Operations Application Messages - Language For Operations Application", GR-831, November 1996.
- [G.694.2] ITU-T Recommendation G.694.2, "Spectral grids for WDM applications: CWDM wavelength grid", December 2003.
- [G.709] ITU-T, "Interface for the Optical Transport Network (OTN)", G.709 Recommendation, October 2009.

10. Contributors' Addresses

Quintin Zhao
Huawei Technologies
125 Nagog Technology Park
Acton, MA 01719
US
Email: qzhao@huawei.com

Victor Lopez
Telefonica I+D
Email: vlopez@tid.es

Ramon Casellas
CTTC
Email: ramon.casellas@cttc.es

Yuji Kamite
NTT Communications Corporation
Email: y.kamite@ntt.com

Yosuke Tanaka
NTT Communications Corporation
Email: yosuke.tanaka@ntt.com

Ina Minei
Juniper Networks
ina@juniper.net

Young Lee
Huawei Technologies
Email: leeyoung@huawei.com

11. Authors' Addresses

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

Adrian Farrel
Juniper Networks
Email: 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.

- An interface for adding additional information to the Traffic Engineering Database is described in Section 2.3.2.3. No protocol is currently identified for this interface, but candidates include:
 - The protocol developed or adopted to satisfy the requirements of I2RS [I-D.ietf-i2rs-architecture]
 - Netconf [RFC6241]
- The protocol or protocols to be used by the Interface to the Routing System described in Section 2.3.2.8 have yet to be determined. The I2RS working group will make this decision after use cases and protocol requirements have been agreed. Various candidate protocols have been identified although none appears to be suitable without some extensions to the currently-specified protocol elements. The list of protocols supplied here is illustrative and not intended to constrain the work of the I2RS working group. The order of the list is not significant.
 - OpenFlow [ONF]
 - Netconf [RFC6241]
 - ForCES [RFC3746]
- As described in Section 2.3.2.10, the Virtual Network Topology Manager needs an interface that can be used by a PCE or the ABNO Controller to inform it that a client layer needs more virtual topology. It is possible that the protocol identified for use with I2RS will satisfy this requirement.
- 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 as described in Section 2.3.2.11.
 - It is possible that the protocol selected or designed to satisfy I2RS.
 - A potential approach for this type of interface is described in [I-D.boucadair-connectivity-provisioning-profile] for a simple use case.

Routing Area Working Group
Internet-Draft
Intended status: Informational

Katherine Zhao
Jiehui Hu
Huawei Technologies
GuangMing Yong
China Telecom Co.,Ltd

July 9, 2013

Use Cases and Architecture of Central Controlled IP RAN
draft-khy-rtgwg-central-controlled-ipran-00.txt

Abstract

This document introduces the requirements and use cases for IP RAN (Radio Access Networks). To support the requirements, the document provides an architecture with centralized control plane as well as separation of control plane and data plane. The document also describes techniques for IP RAN network initialization and construction; interfacing to management plane and third party applications. This document can be used as a guideline for central controlled IP RAN design and development.

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 January 06, 2014.

Copyright Notice

Copyright (c) 2013 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.

This document may contain material from IETF Documents or IETF Contributions published or made publicly available before November 10, 2008. The person(s) controlling the copyright in some of this material may not have granted the IETF Trust the right to allow modifications of such material outside the IETF Standards Process. Without obtaining an adequate license from the person(s) controlling the copyright in such materials, this document may not be modified outside the IETF Standards Process, and derivative works of it may not be created outside the IETF Standards Process, except to format it for publication as an RFC or to translate it into languages other than English.

Table of Contents

1. Introduction	3
1.1. Requirement Language	3
1.2. Terminology	3
2. Requirements and use case for central controlled IP RAN . . .	3
2.1. Scenarios of traditional IP RAN	3
2.2. Requirements of central controlled IP RAN	4
2.3. Network structure of central controlled IP RAN	5
2.3.1. Mode 1: Integrated Controller	6
2.3.2. Mode 2: Separated Controller for access and aggregation	7
2.3.3. Mode 3: Separated Controller for access, aggregation and metro core	8
2.4. Service carrier requirement for the central controlled IP RAN	9
2.4.1. Requirements for the carrier of mobile back haul . .	9
2.4.2. Requirements for the carrier of enterprises and governments	9
3. Central Controlled IR RAN Architecture	9
3.1. Reference Architecture	9
3.2. Reference Model	11
3.2.1. Built-in IP RAN Control Plane	11
3.2.2. Stand-alone IP RAN Control Plane	13
3.2.3. Functional Block	15
3.2.4. Interface Definition	16
3.2.5. Management Plane	17

3.2.6. Forwarding Plane	18
4. Acknowledgements	18
5. IANA Considerations	18
6. Security Considerations	18
7. References	18
7.1. Normative References	18
7.2. Informative References	19
Author's Address	19

1. Introduction

1.1. Requirement Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

1.2. Terminology

The following terms are used in this document:

ASG: Aggregation Service gateway
BS: Base station
BSC: Base station controller
CSG: Cell site gateway
CE: Customer Edge
CP-DP: Control Plane and Data Plane interface
IP RAN: IP Radio Access Network
IP RNC: IP Radio network controller
NMG: Network management
MME: Mobile management entity
MPLS: Multi-protocol Label Switching
RSC: Radio service controller
RSG: Radio Service Gateway
SGW: Serving gateway
SR: Service router
VLAN - Virtual Local Area Network
VPN - Virtual private network

2. Requirements and use case for central controlled IP RAN

2.1. Scenarios of traditional IP RAN

IP RAN is mainly used to carry the traffic from 3G, LTE base station to provide the carrier of high speed service for IP-based mobile back haul.

The normal topology in IP RAN is based on ring or tree. When L3 service is required in metro, aggregation and access area, IP RAN usually uses ring topology to form the network.

Fig. 1 shows a typical network topology of IP RAN. There are two layers: access layer and aggregation layer. The access devices are the edge of the IP RAN, and are used for the service access; The aggregation devices are used to aggregate the traffic from access devices.

The traffic from base station goes via FE/GE to access device. Access device will establish two redundant pseudowires to two aggregation devices. Aggregation device will terminate the pseudowires and get into the L3VPN. Two aggregation devices will work as the gateway for L3 services, and provide the protection for the base station service.

The service core network element will access the metro core after the aggregation of CE, and are connected with base station through L3VPN.

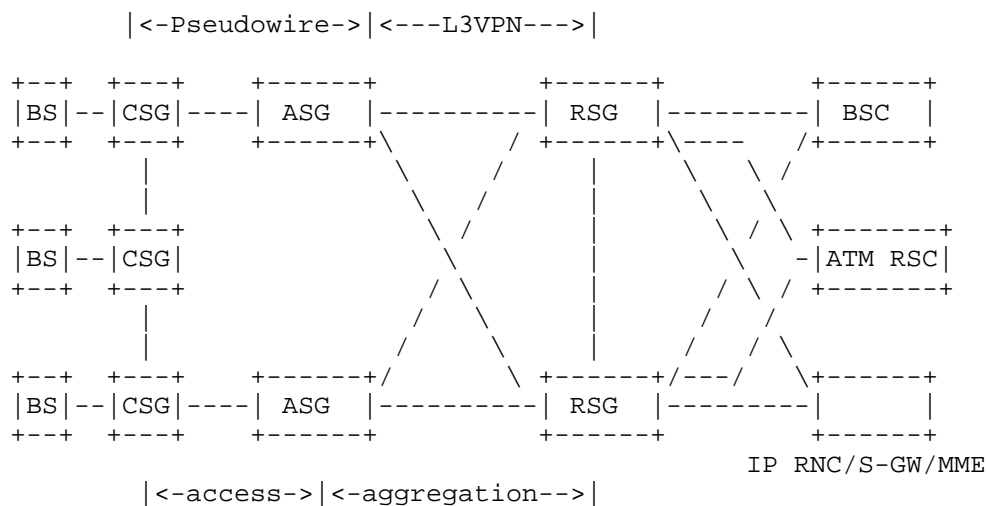


Fig. 1 Traditional IP RAN Network Structure

2.2. Requirements of central controlled IP RAN

IP RAN uses the design ideas from flexible IP network communication, and is based on the architecture of traditional router. It enhances the OAM mechanism, service protection, and the capability of

multiplexing clock transportation. The routing based hardware structure has rich L3 routing functions. It can provide the better multi-service carrier, and support multi-point to multi-point communication scenario in future mobile network. Compared with traditional IP metro network, IP RAN solution will focus on the high availability, simplified operation and management, networking intelligence and OPEX saving from end to end, etc.

The problems in traditional IP RAN network

Network high availability

IP RAN is based on the traditional IP router platform, it uses complicated IP and MPLS protocols to construct the network with huge amount of devices. This is a challenge to the high availability of network deployment

Complexity of network management

IP RAN deployment usually involves a lot of IP MPLS devices, this will introduce very complicated issues in network management and operation. In a typical big metro area, for a local network with the scale of 10k base stations, we need to increase the network with 10500 IP/MPLS equipments. The IP network scale (equipment number) will increase 10 times, and equipment is distributed divergently. This will cause the operation pressure to carriers they nerve faced before. In addition to that, there are many protocols running between access and aggregation devices, network operators must understand and memory lot information regarding to protocols and network structures. It increases the complexity of network operation, and brings more OPEX to carriers.

Low intelligence in network

The time gaps in the mobile network evolving from 2G to 3G and to 4G are decreasing. New value added service and application keep coming up. The question about how to quickly deploy new service and establish the dynamic maintenances will bring in more harsh technical requirements to traditional IP RAN.

Using the architecture of central control, separation of control plane and forwarding plane, open and programmability, we can find out the solution for the network high availability, complexity of operation and management, network intelligence.

2.3. Network structure of central controlled IP RAN

The network structure of central controlled IP RAN is illustrated in Fig. 2 to Fig. 4. The network is still layered as access and aggregation layers. There is central controller to provide the control capability of topology discovery, path calculation and service establishment. There are three modes for central controller, one is built-in controller mode and other two are separated controller mode but for different scope of network. Details for these models and architectures are given in section 3 of the same document.

2.3.1. Mode 1: Built-in Controller

For this mode, the controller is integrated into aggregation routers. There may be multiple controller in a IP RAN.

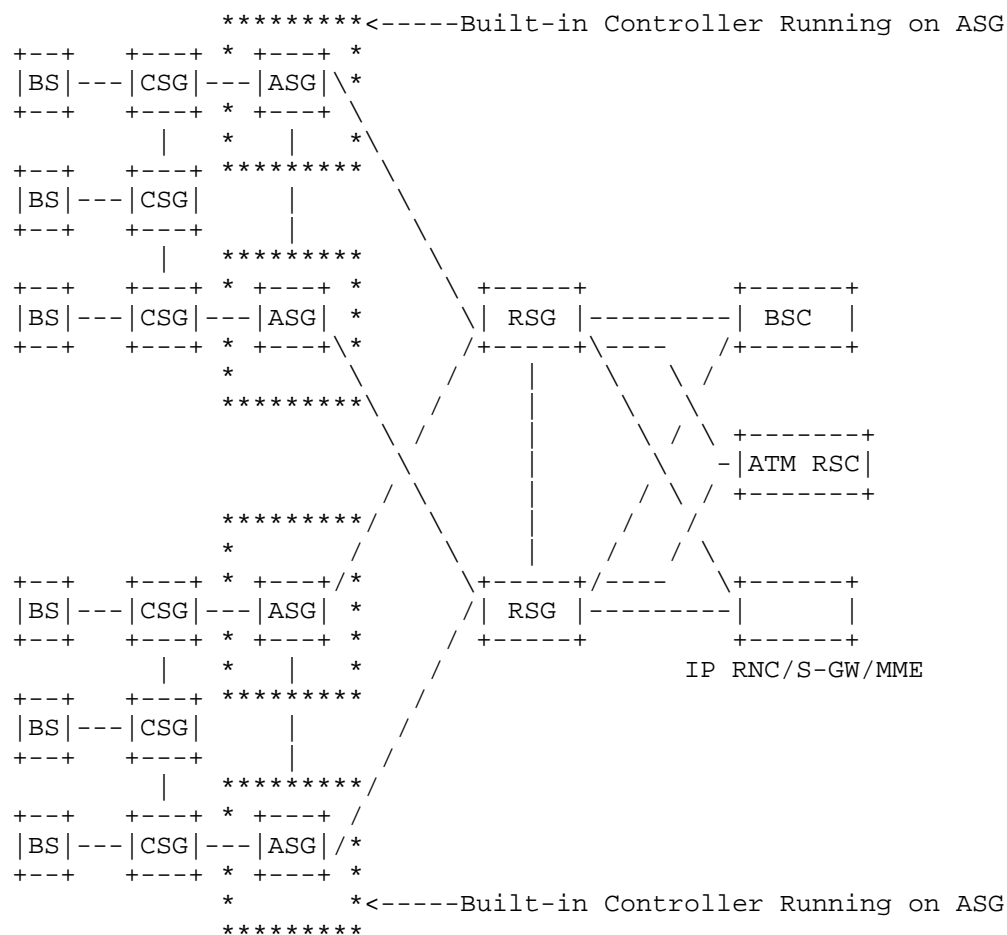


Fig. 2 IP RAN Mode 1: Built-in Controller running on ASG

2.3.2. Mode 2: Stand-alone Controller for Access and Aggregation

For this mode, the stand-alone central controller is used. The control functions in access and aggregation devices are moved to the controller server named IPRAN-CP in this document. In such a way, the access and aggregation devices are only responsible for the forwarding. The controller can control multiple aggregation and access devices.

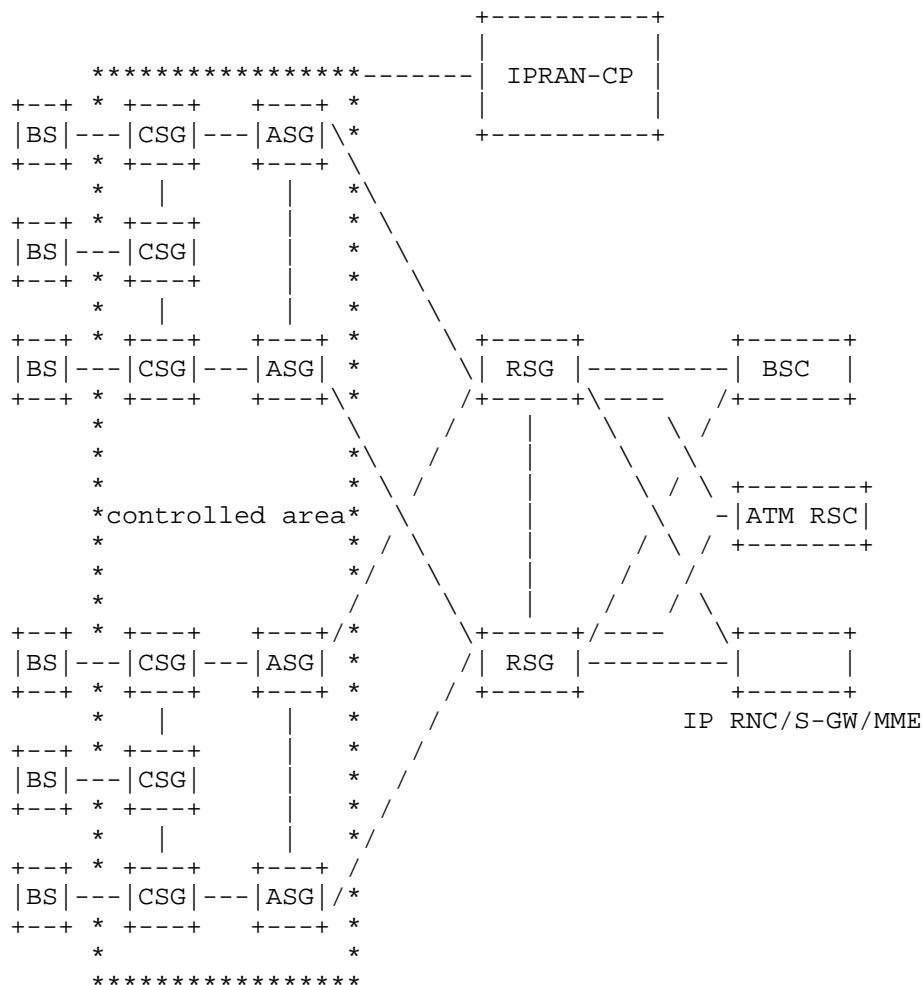


Fig.3 IP RAN Mode 2: Stand-alone Controller Controls CSG and ASG

2.3.3. Mode 3: Stand-alone Controller for Access and Aggregation

For this mode, the stand-alone central controller is used too. The difference from the mode 2 is that the controller will control the access and aggregation layers include RSG. This mode can provide the control capability from end to end.

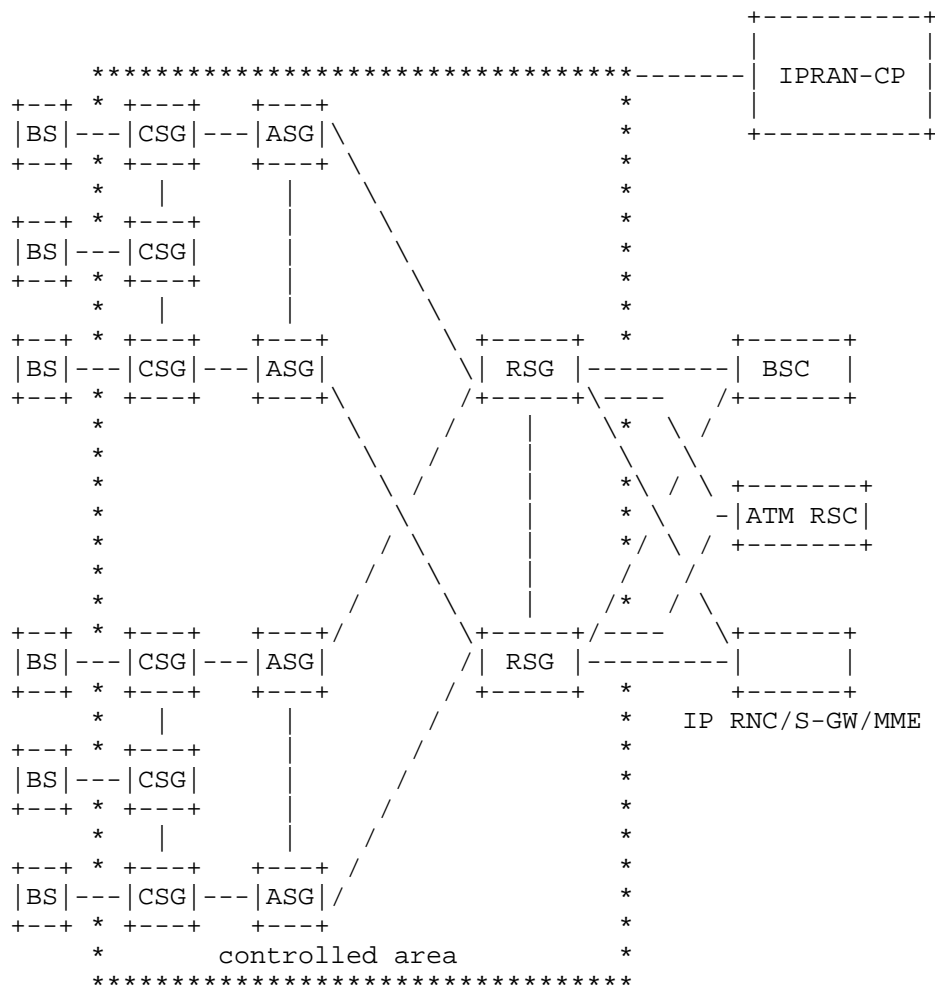


Fig.4 IP RAN Mode 3: Stand-alone Controller controls from CSG to RSG

2.4. Service Carrier Requirement for Central Controlled IP RAN

The central controlled IP RAN must satisfy two major categories of requirements for services. One is for mobile back haul and another one is for enterprises and governments. Services include Ethernet, ATM and TDM.

2.4.1. Requirements for the Carrier of Mobile Backhaul

- 1 Support the access of IP and Ethernet, provide the high available, high throughput carrier for the traffic of mobile back haul
- 2 Provide the carrier for LTE network, support the flexible access between base stations, multi-home base station, and the carrier of multicast traffic for base stations
- 3 Provide the power monitoring, integrated network management for different systems of access services

2.4.2. Requirements for the carrier of enterprises and governments

- 1 Provide the high available, high throughput carrier for the access of L2/L3 VPN services, it includes P2P, P2MP and MP2MP tunnels
- 2 Support the access of high bandwidth Internet leased line, circuit emulation, access of ATM/FR/TDM

3. Central Controlled IR RAN Architecture

3.1. Reference Architecture

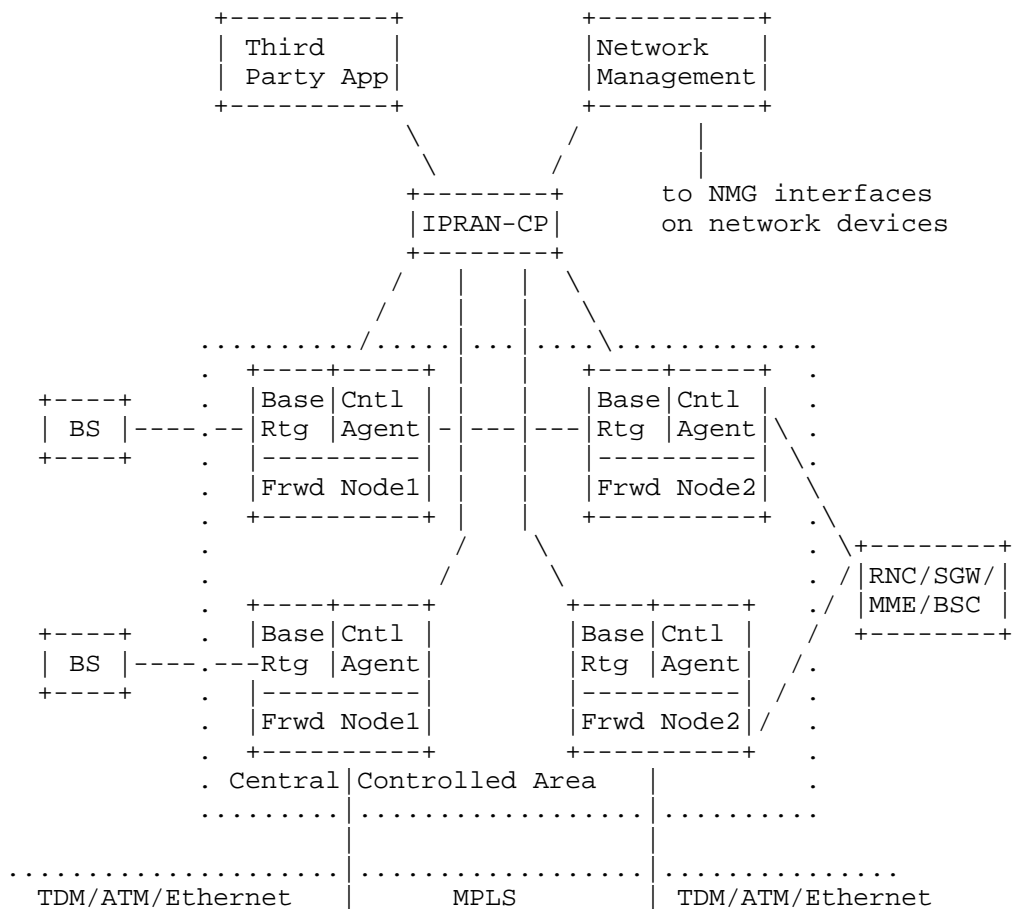


Fig 5. Central Controlled IP RAN Architecture

Above is the central controlled IP RAN architecture, where the control plane is separated from the forwarding plane and moved to an independent entity named IPRAN-CP (IP RAN Control Plane) in this document.

IPRAN-CP, IP RAN Control Plane, is a software system that can be

deployed either in a network device or a separate computer server. IPRAN-CP controls the entire IP RAN network domain, the size of the domain can be defined by Network Operator based on the actual network planning and use cases. IPRAN-CP controls the IP RAN network based on the network topology, actual states and status, which are operated by the network administrator.

IPRAN-CP provides northbound interface to network management system used for service deployment, monitoring, troubleshooting, fault location, and other functions. Netconf could be one of the option for northbound interface implementation. The northbound interface of IPRAN-CP is different from the traditional network equipment interfaces. It is based on the network view and virtual interfaces. IPRAN-CP can provide the interface to network topology and operate on a virtual network.

At the same time, IPRAN-CP also opens to third-party application, allows third party to use RESTful API to program and control network devices. Through IPRAN-CP, Third-party applications can utilize the access to network resources (such as network topology), deliver network diagnosis, fault location, application performance monitoring, add innovative applications in the future and so on.

Because the control plane has been separated out, the network devices become forwarding nodes in the IP RAN domain. Forwarding nodes still keep the basic routing functions in order to establish control and management channel between IPRAN-CP/NMG and all the forwarding nodes. Other control functionality is moved to IPRAN-CP, the forwarding node no longer has the control plane functions (such as protocols needed for establishing services). Forwarding node accepts network resources and states from the controller via Control Agent module and reports itself (such as ports) to the controller.

Forwarding nodes still need traditional northbound management interface. However northbound interface features of the network services and the protocol are no longer included. Instead only a small subset of the features such as power supplies, voltages, veneers and other management features are remained on the network devices. The forwarding plane continues to maintain the existing schema using IP/MPLS forwarding.

3.2. Reference Model

According to different use cases and scenarios, typical reference model can be either in built-in model or stand-alone model. Where the stand-alone model can be divided into another two different sub models

IPRAN-DP supports the network virtualization capabilities, internal CSG topology of an IP RAN is not visible to external network. For external routing device, it only establishes routing neighbor with ASG, and believes that only the ASG routes while the CSG routes are invisible. External routing equipment can be considered an IPRAN-CP controlled domain is a big virtual router.

Using built-in mode, the one can use the existing network equipment to perform software upgrades, apply the new control technology in the access network. Changes to the network architecture and network management model can be limited to small. It is a good strategy to conduct the rapid deployment of central controlled IP RAN.

3.2.2. Central Controlled IP RAN in Stand-alone Model

Built-in IPRAN-CP is easy of deployment and takes advantage of existing network structure, equipments and other investment. However because there is no separation of control plane and forwarding plane on the existing device that IPRAN-CP attached to, for example ASG, the control ability, network performance and the domain scope are limited by the capability of the existing aggregation equipment ASG and its attached network structure.

Stand-alone IPRAN-CP model is designed to resolve above issue. It deploys IPRAN-CP with stand-alone computer server(s) or other IT hardware to eliminate the physical limitation of ASG. The stand-alone control server can use the commodity product with much stronger CPU power and process ability. The controller no longer participates in the forwarding of network traffic like ASG does; it can be located in a remote data center without geographical limitation thus it is able to control more and more network nodes in different locations. Therefore stand-alone IPRAN-CP can provide higher scalability and better performance.

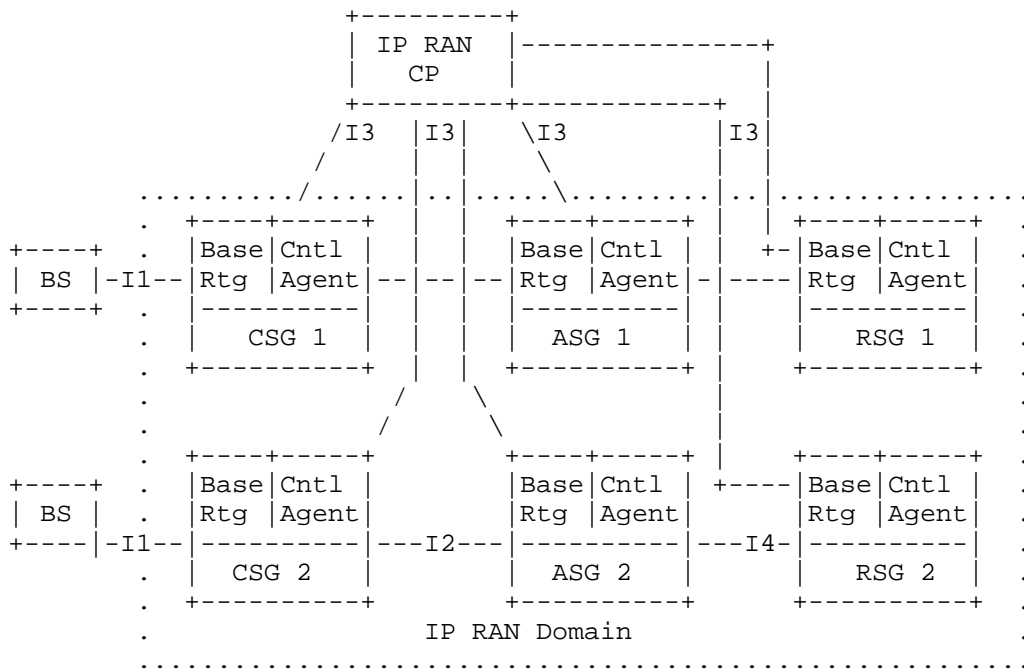
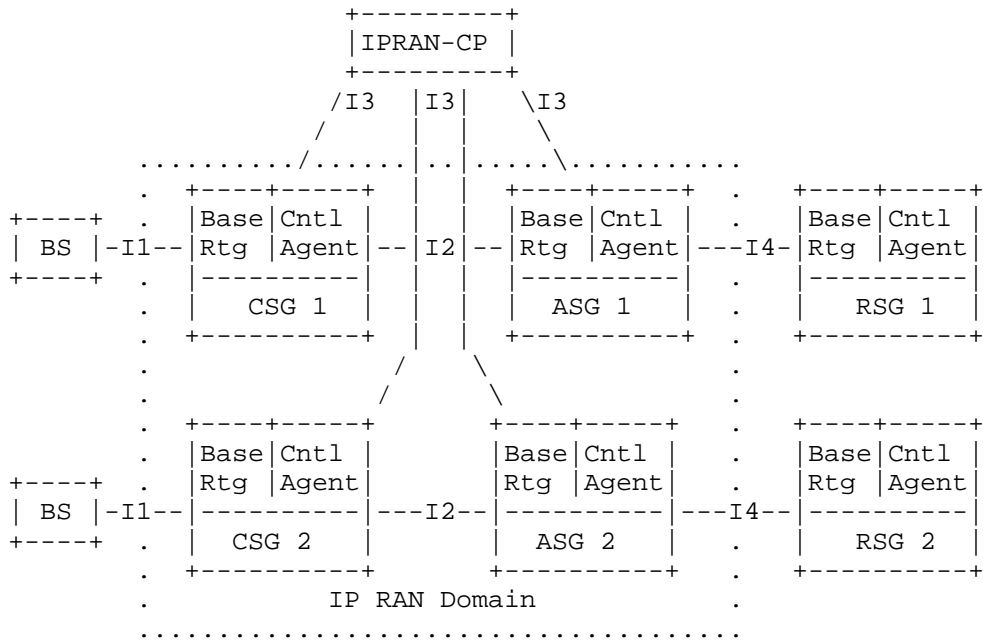


Fig 7. Central Controlled IP RAN in Stand-alone Model

Above diagram shows a use case of stand-alone IP RAN control plane. Where IPRAN-CP software is pulled out from ASG and running on a separate computer server (or in a proprietary device, this document follow-up to the server instead). The servers do not participate in the forwarding of network traffic. The network devices to be controlled by IPRAN-CP can be vary depending on the scope of the actual needs. You can only control access layer equipment, or you can also control the devices include both access layer and aggregation layer.

Different from built-in IPRAN-CP on ASG, a stand-alone control plane can control and manage many CSGs as well as ASG itself, or more than one ASGs with CSGs connected to them. If needed, it can also control even RSGs in an extended IP RAN domain. In this use case, CSG, ASG as well as RSG are integrated with a Control Agent module. Control Agent is responsible for reporting topology, states and statistics to IPRAN-CP; and receiving control information and forwarding tables from IPRAN-CP. The network devices include CSG, ASG and RSG keep basic routing functions for establishing control channel and management channel so that IPRAN-CP can use CP-DP (control plane and data plane) protocol to control these devices.

IPRAN-CP needed to support network virtualization functions, from the view of the external network devices, this IPRAN domain is just a single logical router. If any forwarding device (for example CSG) within the domain has communicated with external network nodes using traditional routing protocol such as BGP,IGP,LDP or RSVP, the central controlled IPRAN-CP must support the same communication channel and support the same routing protocols on behalf of the forwarding devices.

3.2.3. Functional Block

This section describes the functions and features to be supported by the control plane and forwarding plane respectively.

The functions of IPRAN-CP:

- o CP-DP control protocol features: provide forwarding device control information; establish and maintain connections between control plane and the forwarding nodes. CP-DP protocol can be XMPP, Openflow, or a newly defined protocol.

- o Network resource management features: discover and collect the forwarding nodes, the ports on the nodes as well as topology resources in the IP RAN domain. Other software modules can use the resource collection to calculate path, deploying services, and build forwarding tables etc. IPRAN-CP should also open to network management applications and third-party applications to programm and manage the network resources.
- o Network virtualization features: abstract network and shield the physical network details from the service layer. Operational service deployment can be done based on virtual network without concernning how forwarding path for the virtual network to get through at the lower layer. In IP RAN, the transport uses MPLS, the network virtualization here means MPLS virtualization.
- o Network protocol layer: provides traditional routers supported conventional routing protocol stack, which was running on the forwarding nodes. Now these features are moved up to the central controller.
- o Network service layer: Supports IP RAN required L2VPN and L3VPN services. L2VPN including VPWS and VPLS.
- o Supporting module: provides basic tools include fault location, alarm, northbound interface for maintenance and management capabilities.

The functions of forwarding node:

- o Basic routing features: run IGP routing protocol and responsible for establishing connectivity with other devices, setup routes and collect topology. It must be done before a controller can communicate with the network devices.
- o Control protocols: corresponds to the control protocol running on the IPRAN-CP device, responsible for establish and maintain CP-DP connection.
- o Control Agent: Responsible for the registration with IPRAN-CP; reporting its information to the controller; accepting control information and forwarding tables distributed by the IPRAN-DP. Control Agent is also responsible to install the forwarding tables to the data plane that is used to guide the traffic forwarding.
- o Forwarding Plane: forwarding and proccession of messages and user traffics.

- o Support module: provides debug and diagnostics functions on forwarding nodes; supports device management features; provides interface to network management (include power supplies, voltages, etc).

3.2.4. Interface Definition

The interfaces given on above diagrams (refer to Fig 7) are described at below.

- o I1: Interface between base station and forwarding node. It keeps traditional IP RAN network and base station interface, remains no change.
- o I2: Interface between forwarding nodes.
 - * Forwarding nodes run extended ISIS for topology advertisements between nodes. ISIS has been widely used in the existing IP RAN networks to flood link and node information, therefore, ISIS can be leveraged for central controlled IP RAN as part of the topology collection and flood protocol.
 - * With built-in model, because IPRAN-CP is running on an existing network devices such ASG, ISIS is running directly between the ASG and the CSG, IPRAN-CP on ASG can get topology information from ISIS directly.
 - * With stand-alone model, because SDN IPRAN-CP may be deployed outside of the actual physical network, the control plane cannot communication with forwarding plane via ISIS directly. In this case, a selected topology collection agent in the network will be needed. The agent gathers network topology information from CSGs through ISIS protocol, and then reports the information to IPRAN-CP through a CP-DP protocol. An IP RAN may have very large number of network nodes, while the ability of access equipment is limited. Hence it could be required to divide a IP RAN network into smaller ISIS domains to match the capacity of the access equipment. The topology collection agent usually is located at ASBR node of ISIS such as ASG.
- o I3: Interface between IPRAN-CP and forwarding devices. This interface needs to run a CP-DP protocol for the forwarding node's registration, information reporting, and accepting the control information. Options of the CP-DP protocols could be XMPP, OpenFlow or a newly defined protocols. This interface is beyond scope thus left out of this document.

- o I4: Interface between IPRAN-CP and external router. It supports conventional routing protocols such as BGP,ISIS,RSVP and LDP etc.
- o I5: Interface between RSG and base station controllers. It remains unchanged. (I5 and BSC are not shown on Fig 7 due to space limitation)

3.2.5. Management Plane

IPRAN-CP needs to provide northbound interface used for network management and OSS integration. At the same time, IPRAN-CP needs to provide an open programming interface used for third-party application integration and development. For example, when a third-party monitoring application caught a situation of packet dropping and link quality degradation along a path, they can use programming interfaces of IPRAN-CP to control the path and switch to an alternative path. The northbound interface provided by IPRAN-CP can be divided into the following categories:

- o Policy control interface, including routing policy, ACL etc.
- o The service interface, including L2VPN,L3VPN etc.
- o Path control interface, including control of the forwarding path, switching over, definition of bandwidth constraints etc.
- o Monitoring interface, including fault events, network quality threshold, system status monitoring, and statistics etc.
- o Resource interfaces, including bandwidth, network topology, tag resources, interface resources etc.
- o Diagnostic tools, such as ping, tracer, NQA etc.

Forwarding nodes also have a light network management plane. The management uses VPN channel separated from CP-DP channel to manage the forwarding node. Because forwarding node's control plane has been moved up to IPRAN-CP, therefore management functionality provided by the forwarding nodes, are only themselves the equipment management features as follows:

- o Equipment upgrades
- o Device management protocols, such as SNMP, Telnet, and FTP etc
- o Port management

- o Device power, voltage, single-board hardware management features such as alarms, veneer restarts.
- o Environmental monitoring of the device, such as temperature alarms etc.

3.2.6. Forwarding Plane

Forwarding IP/MPLS forwarding service.

4. Acknowledgements

We would like to thank Lin Han and Guoyi Chen who provided valuable contribution to this document.

5. IANA Considerations

This draft requires no changes in IANA specification,

6. Security Considerations

This draft does not add any additional security implications to IP RAN networks. All existing authentication and security mechanisms still apply.

7. References

7.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

7.2. Informative References

Author's Address

Katherine Zhao
Huawei Technologies
2330 Central Expressway
Santa Clara, CA 95050
USA
Email: Katherine.Zhao@huawei.com

Jiehui Hu
Huawei Technologies
Huawei Building, No.156 Beiqing Rd.
Z-park ,Shi-Chuang-Ke-Ji-Shi-Fan-Yuan,
Hai-Dian District , BEIJING 100095
P.R.CHINA
Email:hujiehui@huawei.com

GuangMing Yang
Guangzhou Research Institute of China Telecom Co.,Ltd.
109 West Zhongshan Ave,
Tianhe District, Guangzhou,510630
P.R.China
Email: yanggm@gsta.com

Network Working Group
Internet Draft

Young Lee
Huawei Technologies

Intended status: Informational
Expires: April 2014

Greg Bernstein
Grotto Networking

Ning So
Tata Communications

Luyuan Fang
Cisco

Daniele Ceccarelli
Ericsson

Diego Lopez
Telefonica

Oscar Gonzalez de Dios
Telefonica

October 21, 2013

Network Control Function Virtualization for Transport SDN

draft-lee-network-control-function-virtualization-01.txt

Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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."

The list of current Internet-Drafts can be accessed at <http://www.ietf.org/ietf/lid-abstracts.txt>

The list of Internet-Draft Shadow Directories can be accessed at <http://www.ietf.org/shadow.html>.

This Internet-Draft will expire on April 21, 2014.

Copyright Notice

Copyright (c) 2013 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.

Abstract

This presentation explores the concept of network control function virtualization for transport SDN to help evolve transport networks to provide programmable virtual network services with infrastructure changes to the traditional control plane architecture.

Table of Contents

1. Terminology.....	3
2. Requirements Language.....	3
3. Introduction.....	3
4. Transport SDN Control Architecture.....	4
5. Transport SDN Virtual Network Service.....	6
6. Summary and Conclusion.....	11
7. References.....	11
7.1. Informative References.....	11
8. Contributors.....	12
Authors' Addresses.....	12
Intellectual Property Statement.....	12
Disclaimer of Validity.....	13

1. Terminology

This document uses the terminology defined in [RFC4655], and [RFC5440].

CVI	Client-VNC Interface
PNC	Physical Network Control
VL	virtual Link
VNC	Virtual Network Control
VNE	Virtual Network Element
VNS	Virtual Network Service
VPI	VNC-PNC Interface

2. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

3. Introduction

One of the main drivers for SDN is a physical separation of the network control plane from the forwarding plane. This separation of the network control plane from the forwarding plane has been already achieved with the development of GMPLS/ASON and PCE for transport networks. In fact, in transport networks such separation of data and control plane was dictated at the onset due to the very different natures of the data plane (circuit switched TDM or wavelength) and a packet switched control plane. The physical separation of the control plane and the forwarding plane is a major step toward allowing operators to gain the full control for optimized network design and operation. Another attraction of SDN technology is its logically centralized control regime which allows a global view of the underlying networks under its control. The centralized control of SDN helps improve network resource utilization from a distributed network control. Transport networks have long used this centralized model via network management systems and currently supplement it with topology and resource status information gathered dynamically via GMPLS routing. For transport network control, PCE technology is

essentially equivalent to a logically centralized control for path computation function. By combining the strength of GMPLS/ASON [GMPLS] and PCE [PCE], and open standard interfaces, transport network control plane technology is readily in a position to fully embrace the SDN concepts.

However, the current transport network control plane technology is not suitable for virtualization and client programmability, which are the two of the main drivers for SDN in recent years. Virtualization refers to allowing the clients to utilize a certain amount of network resources as if they own them and thus control their allocated resources in a way most optimal with higher layer or application processes. This empowerment of client control facilitates introduction of new services and applications as the clients are given to create, modify, and delete their virtual network services. The level of virtual control given to the clients can vary from a tunnel connecting two end-points to virtual network elements that consist of a set of virtual nodes and virtual links in a mesh network topology. As part of the VNS, a client control concept is added to the traditional VPN along with a client specific virtual network view. Client control is operated on the view of virtual network resources allocated to the client. This view is called abstracted network topology. Such a view may be specific to the set of client services as well as the particular client. As the client control is envisioned to support a plethora of applications, there is another level of virtualization from the client to individual applications.

4. Transport SDN Control Architecture

To allow virtualization, the network has to provide open, programmable interfaces in which clients/applications can create, replace, modify virtual network services in an interactive manner while having no impact on other network clients. Traditional transport network infrastructure is not suitable for providing programmable interfaces to clients. Direct client control of transport network elements over existing programmable interfaces (control or management plane) is not perceived as a viable proposition for transport network providers due to security and policy concerns among other reasons.

Hence, the need for network control function virtualization where there is a "virtualizer" that interfaces directly with client control and translates/allocates resources (from physical to virtual and vice versa) and creates abstracted network topology for each client and interacts with physical network connection control functions (e.g., GMPLS/ASON for provisioning, PCE for path

computation). The Network control function virtualizer maintains two interfaces: one interface with physical network control functions assumed by GMPLS/ASON and PCE, which is termed as VNC-PNC Interface (VPI); another interface with client control of virtual network, which is termed as Client-VNC Interface (CVI). Figure 1 depicts the overall architecture for transport SDN control in which the virtual network control entity provides network control function virtualization.

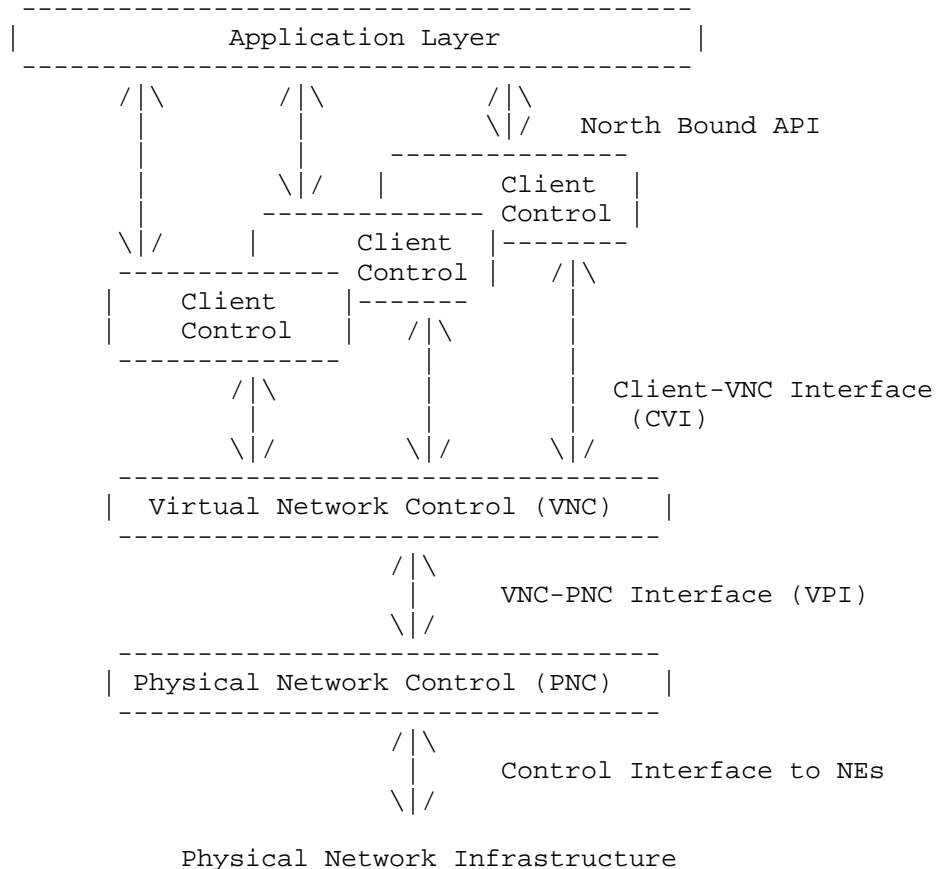


Figure 1: Transport SDN control architecture

Figure 1 shows that there are multiple client controls which are independent to each other and that each client supports various business applications over its NB API. There are layered client-server relationships in this architecture. As various applications are clients to client control, client control itself is also a client to virtual network control; likewise, virtual network control is also a client to physical network control. This layered relationship is important in protocol definition work on NB API, CVI and VPI interfaces as this allows third-party software developers to program client control and virtual network control functions in such a way to create, modify and delete virtual network services.

This architecture in Figure 1 is conceptually in alignment with the Network Functions Virtualization (NFV) architecture [NFV-AF].

5. Transport SDN Virtual Network Service

Virtual Network Service is instantiated by the client control via the CVI. As client control directly interfaces the application stratum, it understands multiple application requirements and their service needs. It is assumed that client control and VNC have the common knowledge on the end-point interfaces based on their business negotiation prior to service instantiation. End-point interfaces refer to client-network physical interfaces that connect client premise equipment to network provider equipment. Figure 2 shows an example physical network topology that supports multiple clients. In this example, client A has three end-points A.1, A.2 and A.3. The interfaces between clients and transport networks are assumed to be 40G OTU links. For simplicity's sake, all network interfaces are assumed to be 40G OTU links and all network ports support ODU switching and grooming on the level of ODU1 and ODU2. Client control for A provides its traffic demand matrix that describes bandwidth requirements and other optional QoS parameters (e.g., latency, diversity requirement, etc.) for each pair of end-point connections.

Figure 2 shows that three independent clients A, B and C provide its respective traffic demand matrices to the VNC. The physical network topology shown in Figure 2 is the provider's network topology created by the PNC's topology creation engine such as the link state database (LSDB) and Traffic Engineering DB (TEDB) based on control plane discovery function. This topology is internal to PNC and not available to the client. What is available to the client is abstracted network topology (aka, virtual network topology) based on the negotiated level of abstraction. This is a part of VNS instantiation between a client control and VNC.

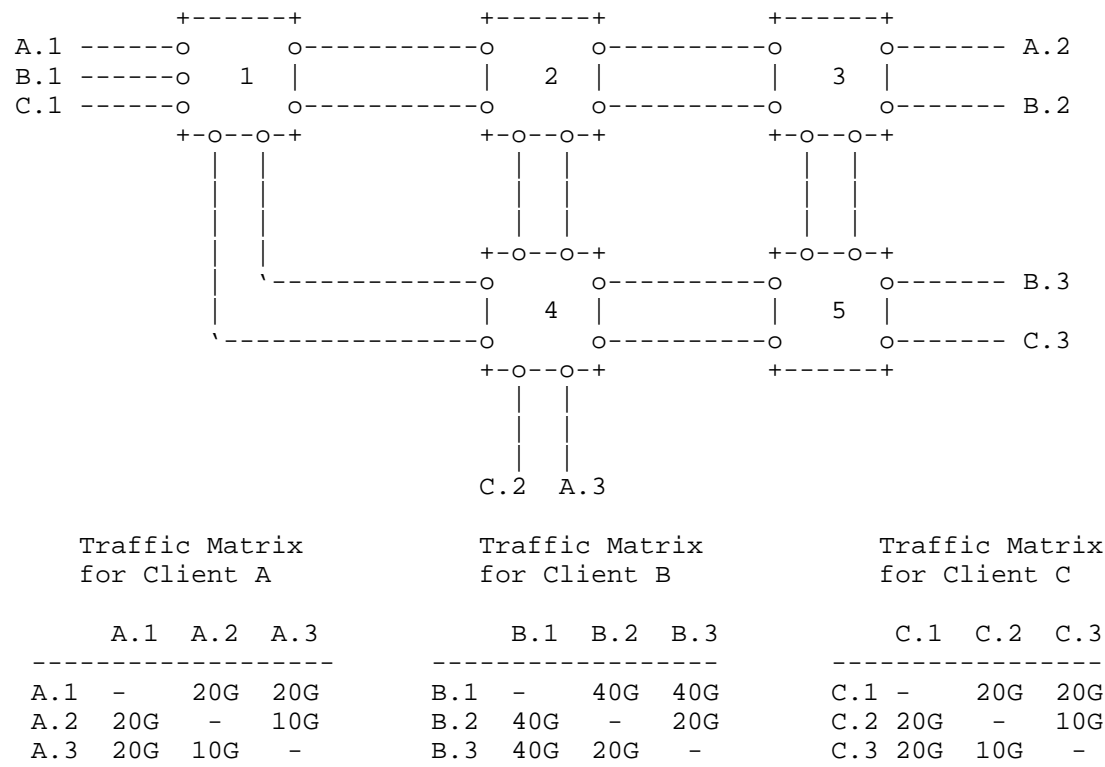
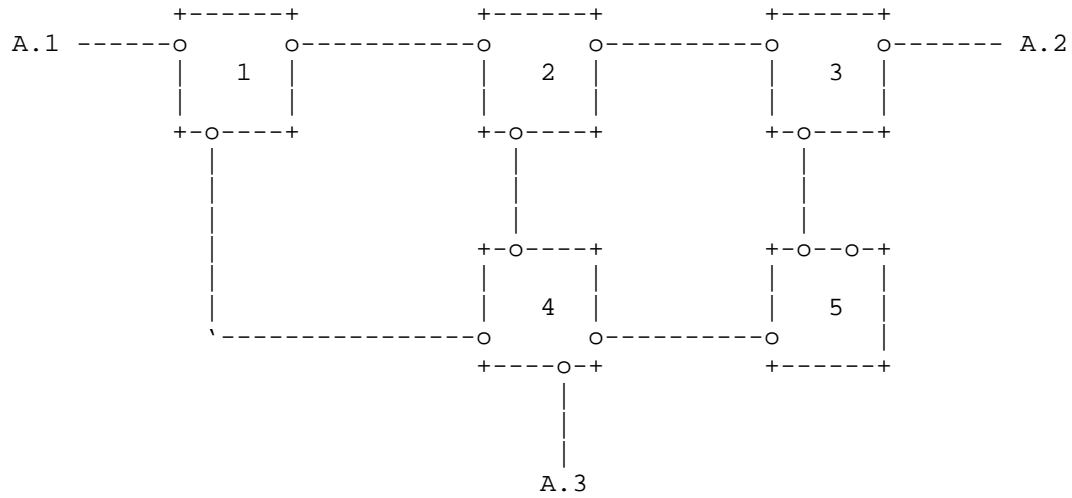


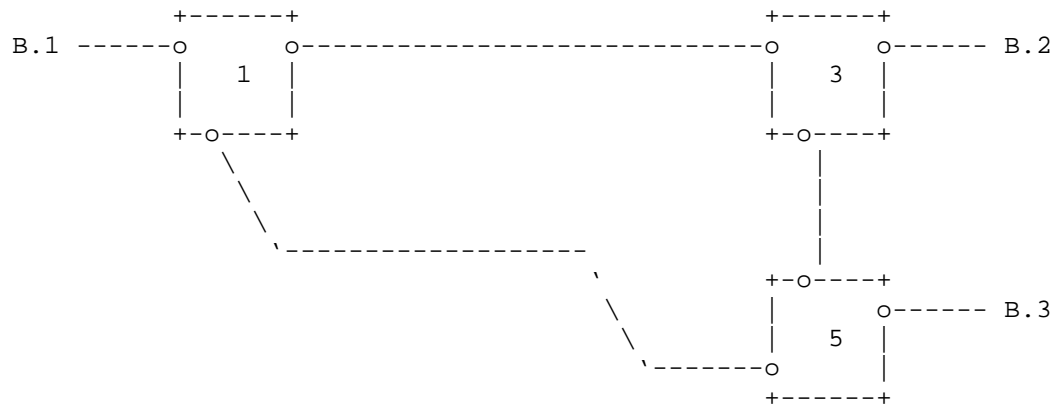
Figure 2: Physical network topology shared with multiple clients

Figure 3 depicts illustrative examples of different level of topology abstractions that can be provided by the VNC topology abstraction engine based on physical topology base maintained by the PNC. The level of topology abstraction is expressed in terms of the number of virtual network elements (VNEs) and virtual links (VLs). For example, the abstracted topology for client A shows there are 5 VNEs and 10 VLs. This is by far the most detail topology abstraction with a minimal link hiding compared to other abstracted topologies in Figure 3.

Abstracted Topology for Client A (5 VNEs and 10 VLs)



Abstracted Topology for Client B (3 VNEs and 6 VLs)



Abstracted Topology for Client C (1 VNE and 3 VLs)

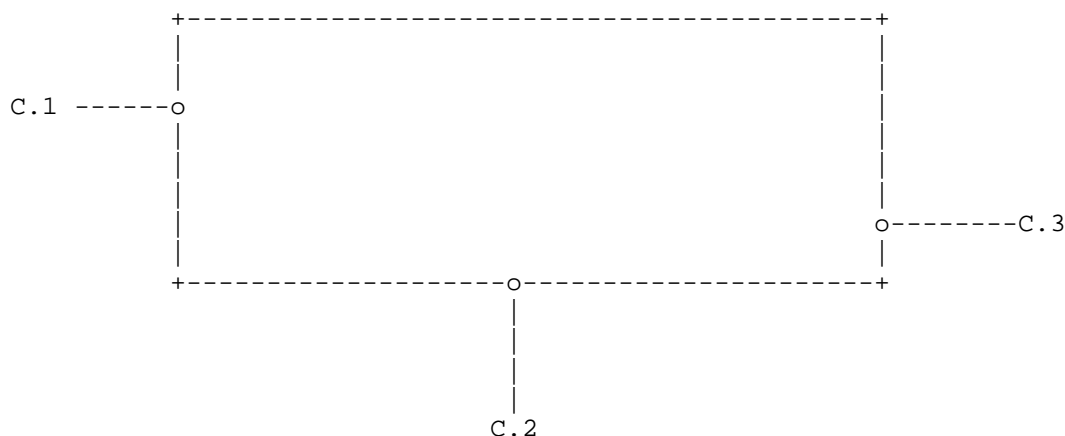


Figure 3: Topology Abstraction Examples for Clients

As different client has different control/application needs, abstracted topologies for client B and C, respectively show much less degree of abstraction. The level of topology abstraction is determined by the policy (e.g., the granularity level) placed for the client and/or the path computation results by the PNC's PCE. The more granular the abstraction topology is, the more control is given to the client control. If the client controller has applications that require more granular control of virtual network resources, then abstracted topology for client A may be the right abstraction level for such client controller. For instance, if the client is a third-party virtual service broker/provider, then it would desire much more sophisticated control of virtual network resources to support differing application needs. On the other hand, if the client were only to support simple tunnel services to its application, then abstracted topology for client C that consists of one VNE and three VLs would suffice.

Figure 4 shows workflows across client control, VNC and PNC for the VNS instantiation, topology exchange, and VNS setup. Client control "owns" a VNS and initiates by providing the instantiation identifier with the traffic demand matrix with path selection constraints for

that instance. This VNS instantiation request from Client Control triggers a path computation request by the virtual PCE (vPCE) agent in the VNC after VNC's proxy's interlay of this request to the vPCE. vPCE requests a concurrent path computation request that is converted based on the traffic demand matrix as part of the VNS instantiation request from Client Control. Upon receipt of this path computation request, the PCE in the PNC block computes paths and updates network topology DB and informs the vPCE agent of the VNC of the paths and topology updates.

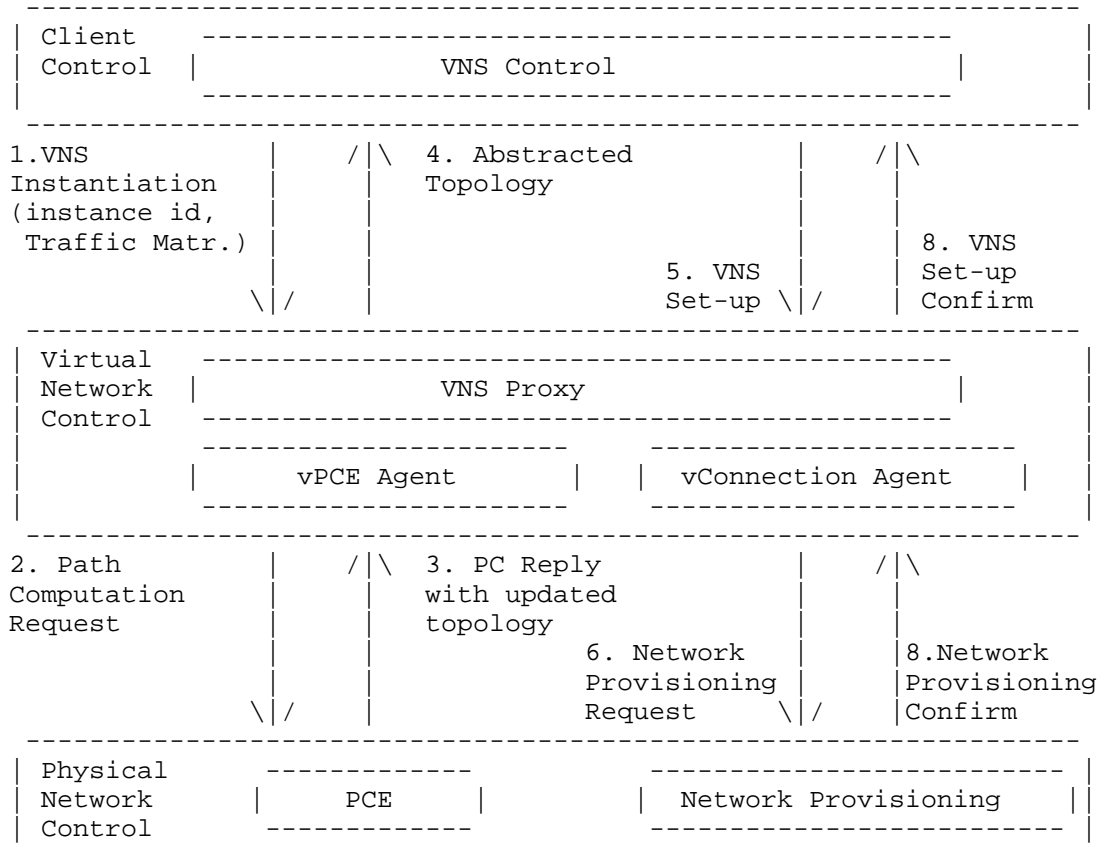


Figure 4. Workflows across Client control, VNC and PNC

It is assumed that the PCE in PNC is a stateful PCE [PCE-S]. vPCE agent abstracts the network topology into an abstracted topology for the client based on the agree-upon granularity level. The abstracted

topology is then passed to the VNS control of the Client Control block. The Client Control's VNS control computes and assigns virtual network resources for its applications based on the abstracted topology and creates VNS setup command to the VNC. The VNC's vConnection module turns this VN setup command into network provisioning requests over the network elements using control plane messages such as GMPLS, etc.

6. Summary and Conclusion

This presentation explores the concept of network control function virtualization for transport SDN to help evolve transport networks to provide programmable virtual network services with infrastructure changes to the traditional control plane architecture. The VNC and its interfaces with the Client Control and the PNC provide control plane function virtualization over programmable interfaces such as virtual network path computation and optimization, topology abstraction hiding details of physical topology while supporting service-specific objectives the clients demand, maintaining virtual network service instances and the states, policy enforcement for virtual network services. With this evolutionary architecture, virtual network services can be readily introduced while re-using physical network control plane functions.

7. References

7.1. Informative References

- [PCE] Farrel, A., Vasseur, J.-P., and J. Ash, "A Path Computation Element (PCE)-Based Architecture", IETF RFC 4655, August 2006.
- [PCE-S] Crabbe, E, et. al., "PCEP extension for stateful PCE", draft-ietf-pce-stateful-pce, work in progress.
- [GMPLS] Manning, E., et al., "Generalized Multi-Protocol Label Switching (GMPLS) Architecture", RFC 3945, October 2004.

[NFV-AF] "Network Functions Virtualization (NFV); Architectural Framework", ETSI GS NFV 002 v1.1.1, October 2013.

8. Contributors

Authors' Addresses

Young Lee
Huawei Technologies
5340 Legacy Drive
Plano, TX 75023, USA
Phone: (469)277-5838
Email: leeyoung@huawei.com

Greg Bernstein
Grotto Networking
Fremont, CA, USA
Phone: (510) 573-2237
Email: gregb@grotto-networking.com

Ning So
Email: Ning.So@tatacommunications.com

Luyuan Fang
Email: luyuanf@gmail.com

Daniel Ceccarelli
Email: daniele.ceccarelli@ericsson.com

Diego Lopez
Email: diego@tid.es

Oscar Gonzalez de Dios
Email: ogondio@tid.es

Intellectual Property Statement

The IETF Trust takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in any IETF Document or the extent to which any license

under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights.

Copies of Intellectual Property disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at <http://www.ietf.org/ipr>

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement any standard or specification contained in an IETF Document. Please address the information to the IETF at ietf-ipr@ietf.org.

Disclaimer of Validity

All IETF Documents and the information contained therein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY, THE IETF TRUST AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION THEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Acknowledgment

Funding for the RFC Editor function is currently provided by the Internet Society.

