Next Generation Control of Transport Networks

Daniel King
Lancaster University

(A PhD Thesis in the Alternative Format)

Contents

ACKNOWLEDGEMENTS

Firstly, I would like to thank my excellent lead PhD supervisor Professor David Hutchison and his constant guidance, enthusiasm, and encouragement. I would also like to thank my additional PhD supervisors, Dr Nicholas Race and Dr Chris Edwards.

A significant portion of my PhD research was conducted with British Telecom Research, specifically at Adastral Park in Martlesham, UK. While fulfilling my research obligations, I spent a significant amount of time at the BT Research site and would like to thank my industrial supervisors, Nektarios Georgalas and Dr Andrew Lord, for their inspiration and support.

I would also like to thank my fellow Lancaster University PhD students and research associates, who contributed to this research with their discussions and challenging questions. I am very grateful to all of you.

Finally, I would like to thank Emiko, simply put this journey would not have been possible without you. Also, Kaylee, Abbey, and Sasha – without whom, my Thesis would have been completed many months ago.

Abstract

It is widely understood by telecom operators and industry analysts that bandwidth demand is increasing dramatically, year on year, with typical growth figures of 50% for Internet-based traffic [5]. This trend means that the consumers will have both a wide variety of devices attaching to their networks and a range of high bandwidth service requirements. The corresponding impact is the effect on the traffic engineered network (often referred to as the "transport network") to ensure that the current rate of growth of network traffic is supported and meets predicted future demands.

As traffic demands increase and newer services continuously arise, novel network elements are needed to provide more flexibility, scalability, resilience, and adaptability to today's transport network. The transport network provides transparent traffic engineered communication of user, application, and device traffic between attached clients (software and hardware) and establishing and maintaining point-to-point or point-to-multipoint connections.

The research documented in this thesis was based on three initial research questions posed while performing research at British Telecom research labs and investigating control of transport networks of future transport networks:

1. How can we meet Internet bandwidth growth yet minimise network costs?
2. Which enabling network technologies might be leveraged to control network layers and functions cooperatively, instead of separated network layer and technology control?
3. Is it possible to utilise both centralised and distributed control mechanisms for automation and traffic optimisation?

This thesis aims to provide the classification, motivation, invention, and evolution of a next generation control framework for transport networks, and special consideration of delivering broadcast video traffic to UK subscribers. The document outlines pertinent telecoms technology and current art, how requirements I gathered, and research I conducted, and by which the transport control framework functional components are identified and selected, and by which method the architecture was implemented and applied to key research projects requiring next generation control capabilities, both at British Telecom and the wider research community.

Finally, in the closing chapters, the thesis outlines the next steps for ongoing research and development of the transport network framework and key areas for further study.

Contributing Publications

This PhD Thesis has been prepared in the Alternative Format, based on the following contributing documents and peer-reviewed publications:

- D. King, “Network Functions Virtualisation: The New Frontier of Telecoms Innovation”, Multi-Service Networking, Science & Technology Facilities Council, Abingdon, UK, July 2013.
- V. Lopez, D. King, et al., "Adaptive network manager: Coordinating operations in flex-grid networks", IEEE 15th International Conference on Transparent Optical Networks (ICTON), Cartagena, July 2013.
- D. King, “Unification of Formal and De Facto Standards for Abstraction and Autonomic Control of the Transport Network”, Layer123 SDN & NFV World Congress, Dusseldorf, Germany, October 2013.
- D. King, "Architecting SDN for Optical Access Networks”, European Conference on Optical Communication (ECOC), September 2014.
- L. Velasco, A. Castro, D. King, O. Gerstel, R. Casellas and V. Lopez, "In-operation network planning," in IEEE Communications Magazine, January 2014.
- D. King, "SDN-based elastic and adaptive optical transport network: findings and future research”, WDM & Next Generation Optical Networking, June 2015.
- D. King, A. Farrel, N. Georgalas, “The role of SDN and NFV for flexible optical networks: Status, Challenges and Opportunities, IEEE Transparent Optical Networks (ICTON), July 2015.
- D. King, A. Farrel, “RFC7491: A PCE-Based Architecture for Application-Based Network Operations”, Internet Engineering Task Force (IETF), March 2015.
- D. King (Editor), V. Lopez, O, Gonzalez de Dios, R. Casellas, N. Georgalas, A. Farrel, “Elastic Optical Networks Architectures, Technologies, and Control: Application-Based Network Operations (ABNO)”, Springer Publishing, 2016.
- R. Casellas, D. King, et al., "A control plane architecture for multi-domain elastic optical networks: the view of the IDEALIST project," in IEEE Communications Magazine, August 2016.
- C. Rotsos D. King, et al., “Network service orchestration standardization: A technology survey”, Elsevier Computer Standards & Interfaces, Volume 54, November 2017.

PhD Thesis Structure

The following section outlines how my contributing publications are applied in relevant sections of this Alternative Format PhD Thesis.

Chapter 1: Introduction

Next generation transport networks are an open subject with a very fast innovation pace. This initial chapter thesis outlines transport architecture design, legacy control architectures and the existing technologies that are used for deploying and operating transport networks. It outlines two new areas of enabling technologies, namely: Software Defined Networks (SDN) and Network Functions Virtualisation (NFV), and their relevance to traffic engineered communication networks, often referred to as "transport networks".

Contributing publications:

- D. King, "Recent Progress in Routing Standardization", UK Network Operators Forum (UKNOF 23), October 2012.
- D. King, "Network Functions Virtualisation: The New Frontier of Telecoms Innovation", Multi-Service Networking, Science & Technology Facilities Council, Abingdon, UK, July 2013.

Chapter 2: Background

The chapter examines the history of transport service management and how new requirements for Cloud services and advance mobile networks present new challenges for transport network operators. This chapter outlines the investigation process, and interviews with leading transport infrastructure operators led to several significant challenges being identified for network architecture and transport service management.

Contributing publications:

- V. Lopez, D. King, et al., "Adaptive network manager: Coordinating operations in flex-grid networks", IEEE 15th International Conference on Transparent Optical Networks (ICTON), Cartagena, July 2013.
- D. King, "Unification of Formal and De Facto Standards for Abstraction and Autonomic Control of the Transport Network", Layer123 SDN & NFV World Congress, Dusseldorf, Germany, October 2013.

Chapter 3: Current Control Architectures

The selection and development of key control plane functions must address the requirements outlined in the previous chapter. This chapter reviewed existing control techniques, strengths, and weaknesses, which outlined the need for infrastructure control flexibility.

Contributing publications:

- D. King, "Architecting SDN for Optical Access Networks", European Conference on Optical Communication (ECOC), September 2014.
- A. Farrel, D. King, "The Role of PCE in an SDN World (Keynote)", European Workshop on SDN (EWSDN), September 2014.

Chapter 4: Transport Network Control Framework Design

This chapter reflects the design methodology, findings, and detailed analysis of requirements from the interviews and information gathering from industry leaders and technology innovators at leading telecom organisations, operating some of the largest telecom transport networks in the world.

Contributing publications:

- D. King, A. Lord, "SDN-based elastic and adaptive optical transport network: findings and future research”, WDM & Next Generation Optical Networking, June 2015.
- D. King, A. Farrel, N. Georgalas, “The role of SDN and NFV for flexible optical networks: Status, Challenges and Opportunities, IEEE Transparent Optical Networks (ICTON), July 2015.

Chapter 5: Framework for Application-Based Network Operations (ABNO)

This chapter outlines a framework and method of control entitled: Application-Based Network Operations (ABNO). It highlights the key control functional components of ABNO, and how this framework and the functional components may be developed and deployed.

Contributing publications:

- D. King, A. Farrel, “RFC7491: A PCE-Based Architecture for Application-Based Network Operations”, Internet Engineering Task Force (IETF), March 2015.
- D. King (Editor), V. Lopez, O, Gonzalez de Dios, R. Casellas, N. Georgalas, A. Farrel, “Elastic Optical Networks Architectures, Technologies, and Control: Application-Based Network Operations (ABNO)”, Springer Publishing, 2016.

Chapter 6: ABNO Framework Implementation and Testing

The ABNO framework has been adopted and applied by numerous European and International research projects, including but not limited to FP7 IDEALIST, FP7 OFERTIE, FP7 DISCUS, FP7 CONTENT, EPSRC TOUCAN, STREP STRAUSS, H2020 ACINO, and most recently the H2020 METRO-HAUL project. This chapter highlights how ABNO was applied and implemented across some of these projects.

Contributing publications:

- R. Casellas, D. King, et al., "A control plane architecture for multi-domain elastic optical networks: the view of the IDEALIST project," in IEEE Communications Magazine, August 2016.
- C. Rotsos D. King, D, Hutchison, et al., “Network service orchestration standardization: A technology survey”, Elsevier Computer Standards & Interfaces, Volume 54, November 2017.

Chapter 7: Conclusions and Areas for Further Research

Finally, this PhD Thesis summarises my conclusions and outlines important areas for further investigation and research.

- D. King, C. Rotsos, I. Busi, F. Zhang and N. Georgalas, "Transport Northbound Interface: The need for Specification and Standards coordination," 2017 International Conference on Optical Network Design and Modeling (ONDM), Budapest, 2017
- J. Ellerton, D. King, D, Hutchison, et al., "Prospects for Software Defined Networking and Network Function Virtualisation in Media and Broadcast," SMPTE 2015 Annual Technical Conference and Exhibition, Hollywood, 2015.

The scope of this PhD Thesis

This Alternative Format PhD Thesis proposes and develops a control framework for traffic engineered communication networks; these are referred to as “transport networks. Traditionally, the transport network was managed using a monolithic management architecture, comprising of an umbrella Network Management System (NMS), with Element Management System per technology domain. Often requiring large teams of specialist technology experts.

More recently, Software Defined Networking (SDN) introduced separation of control and forwarding, coupled with (logically) centralised control, reducing the complexity and skills required for deployment of networking infrastructure.

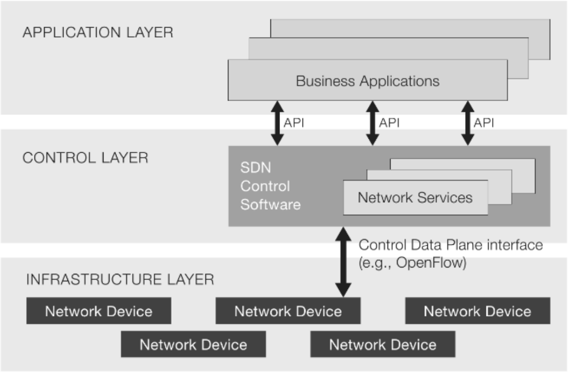


Figure 1 Open Networking Foundation generalised SDN Architecture

The following architecture demonstrates the Application-Based Network Operations (ABNO) Control Layer (developed during my PhD project), on the left side of the future transport network architecture.

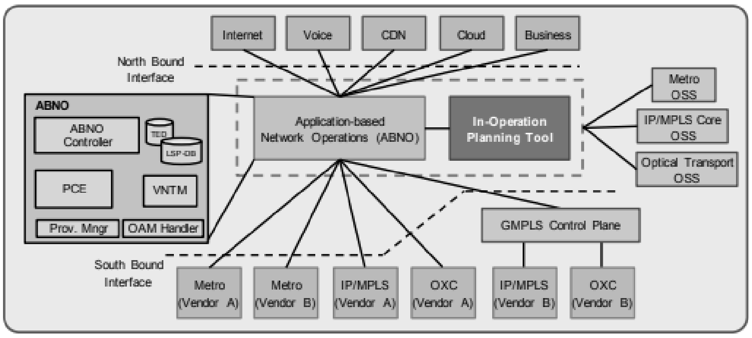


Figure 2 Application-Based Network Operations (ABNO) Control Layer

The research outlined in this thesis began with a review of current art on optical transport network control, documenting key objectives for control of next generation optical networks, an analysis of control plane technologies and a thorough set of interviews with key technologists and network architects at some of the world’s largest network operators.

The culmination of this research and thesis is the Application-Based Network Operations (ABNO) Framework, now an IETF-based Internet Standard (RFC7491). ABNO was developed by the researcher and author of this thesis, and incudes contributions from leading vendors and network operators from around the world. ABNO was then developed further within a number of European projects, in some cases with the direct input of this thesis author.

Thesis Glossary

A list of the abbreviations used in this document is as follows:

3GPP	Third Generation Partnership Project
5G PPP	5G Infrastructure Public Private Partnership
ABNO	Application Based Network Operations
ACTN	Abstraction and Control of Traffic-Engineered Networks
API	Application Programming Interface
ASON	Automatically Switched Optical Networks
BBF	Broadband Forum
BBU	Base Band Unit
BER	Bit Error Rate
BSS	Business Support System
BVT	Bandwidth Variable Transceivers
CAGR	Compound Annual Growth Rate
CDB	Core Engine Database
CDN	Content Delivery Network
CLI	Command Line Interface
CN	Core Network
CNC	Customer Network Controller
CO	Central Office
CoS	Class of Service
CP	Control Plane
CPE	Customer Premises Equipment
DP	Data Plane
DWDM	Dense Wavelength Division Multiplexing
EON	Elastic Optical Network
EPC	Evolved Packet Core
ETSI	European Telecommunications Standards Institute
FEC	Forward Error Correction
GMPLS	Generalized Multi-Protocol Label Switching
gNMI	gRPC Network Management Interface
gRPC	Google's Remote Procedure Call
IaaS	Infrastructure-as-a-Service
IETF	Internet Engineering Task Force
IoE	Internet of Everything
IoT	Internet of Things
IPFIX	Internet Protocol Flow Information Export
IT	Information Technology
JSON	JavaScript Object Notation
L2VPN	Layer2 VPN
L3VPN	Layer3 VPN
LSP	Label Switched Path
ML	Machine Learning
MP	Management Plane
MPLS	Multi-Protocol Label Switching

MPLS-TP	MPLS Transport Profile
NBI	North-Bound Interface
NE	Network Element
NFV	Network Function Virtualisation
NFVI, NFV-I	NFV Infrastructure
NFVO, NFV-O	Network Function Virtualisation Orchestrator
NMS	Network Management System
OF	OpenFlow
ONF	Open Networking Foundation
OSS	Operations and Support System
OSS	Open Source Software
PaaS	Platform-as-a-Service
PCE	Path Computation Element
PNF	Physical Network Function
PoC	Proof-of-Concept
QoE	Quality of Experience
QoS	Quality of Service
REST	Representational State Transfer
ROADM	Reconfigurable Optical Add Drop Multiplexer
RPC	Remote Procedure Call
SaaS	Software-as-as-Service
SBI	South Bound Interface
SDN	Software-Defined Networking
SDO	Standards Defining Organization
SNMP	Simple Network Management Protocol
TAPI	Transport API
TED	Traffic Engineering Database
T-SDN	Transport-SDN
VIM	Virtual Infrastructure Manager
VM	Virtual Machine
VNF	Virtualised Network Function
VNT	Virtual Network Topology
VPLS	Virtual Private LAN Service
VPN	Virtual Private Network
WAN	Wide Area Network
WDM	Wavelength Division Multiplexing

List of Figures

List of Tables

1. Introduction

This chapter outlines the core communication technology used to provide core capacity for Internet-based services. It outlines the current packet transport technology, which is underpinned by optical transport network infrastructures. The rate of Internet growth is exponential and greater bandwidth, at faster data rates and significantly, reduced operational costs and complexity will be required.

1.1 Internet Packet Transport

Current Internet infrastructure includes various types of connectivity structures and representations between connected topologies [1]. The Internet topology should be considered at varying abstraction levels, i.e., IP interface, router, subnetwork, areas, and Autonomous System (AS) levels [2]. The current Internet architecture involves a great number of technologies impacting the Internet, many of these technologies are underpinned by packet services carried over Multiprotocol Label Switching (MPLS) [3]. A significant and mature technology widely deployed for differentiated services, load balancing reasons, resilience, and traffic engineering purposes.

1.2 The Role of Optical Transport Networks for the Internet

To provider high-bitrate connectivity for packet transport, Optical Transport Networks (OTN) are routinely used [4]. OTNs are a set of optical network elements connected by optical fibre links, able to provide the functionality of communication and transport, multiplexing, switching, supervision, and protection of optical channels carrying client signals, which are typically purely photonic.

1.3 Internet Bandwidth Drivers

All agree that Cloud, 5G and Internet of Everything (IoE) services will have a significant traffic impact on existing networks, including types, volume, and dynamicity, all the while being transmitted at unprecedented rates. To facilitate emerging Internet traffic requirements, the optical transport network should become more responsive to the traffic changes as well as to operate more efficiently. Key enablers include, Software Defined Networking (SDN) and Network Function Virtualisation (NFV), combined they promised to increase transport network flexibility and automation.

A survey conducted by Forrester Consulting on behalf of Juniper Network, January 2014 [5] highlights network bandwidth, performance, reliability, automation/programmability as being key demands from customers for DC interactions. A summary of requirements is shown in Figure 3 (Network Features Required by Cloud Customers) below.

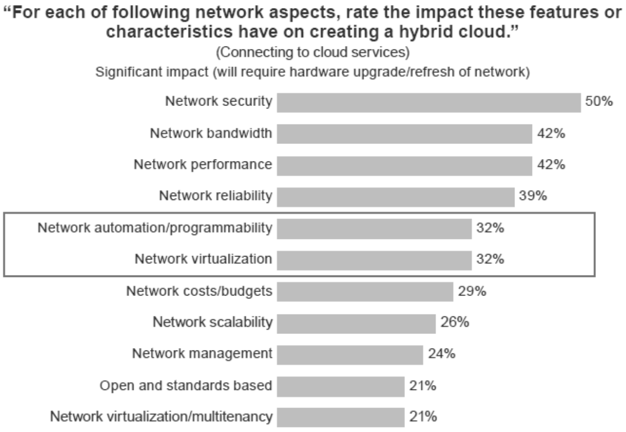


Figure 3 Network Features Required by Cloud Customers [5]

Almost three quarters of the customers interviewed for the usage of private cloud confirmed they would use such inter-connection services (see Figure 4 - Desire for Interconnection of Private and Public Cloud) with the public cloud, while about 73% of the customers said they needed to adjust the underlying network for their inter-cloud service (see Figure 5 – Necessary Network Adjustment for Cloud Services).

Figure 4 Desire for Interconnection of Private and Public Cloud [5]

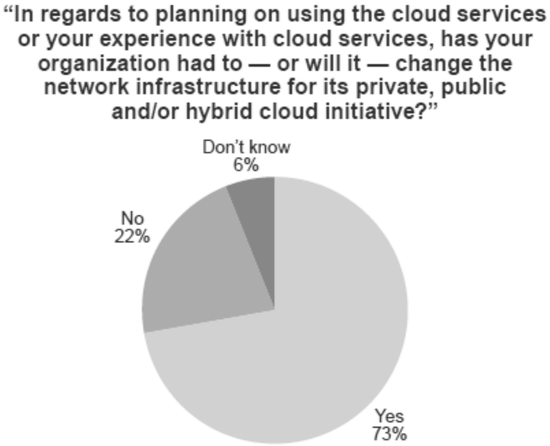


Figure 5 Necessary Network Adjustment for Cloud Services [5]

Given the existing and growing customer profiles, connection and traffic assurance considerations, the research objective was to develop a much more efficient method for operating optical transport network infrastructure capable of end-to-end connection management, without increasing control and management complexity. This includes minimising the deployment of new protocols in the network and reusing existing protocol knowledge where possible.

- Improve flexibility and optical transport services
- Leverage TE at the network edge
- Guarantee constraint optimisations
- Establish traffic differentiation without deploying complex overlay technologies

- Slice available network capacity for Assured Traffic according to Cloud application priority
- Dynamically reallocate bandwidth in the event of oversubscription and ensure that assured traffic is always prioritised.

1.4 Traditional Transport Network Design

Traditionally, an operator may use dedicated physical links or complex distributed control plane mechanisms such as Multiprotocol Label Switching Traffic Engineering (MPLS-TE) to meet the customer and service requirements highlighted in [6]. However, MPLS-TE tunnels separate traffic logically; therefore, traffic across different tunnels may use shared, or dedicated, underlying physical links [2]; meaning that the assurance effect is always non-determined as the logical links inherit the underlying physical link properties.

Methods to eliminate MPLS-TE, and other overlay and tunnelling technologies, dependency on underlying link attributes and map dedicated physical links to logical links, thus providing assured traffic (with guaranteed attributes, including committed bandwidth, total latency, controlled jitter, and explicit network paths) is possible, but such solutions tend to waste large amounts of valuable link resources since the volume of assured traffic, and the normal traffic will vary based on network conditions.

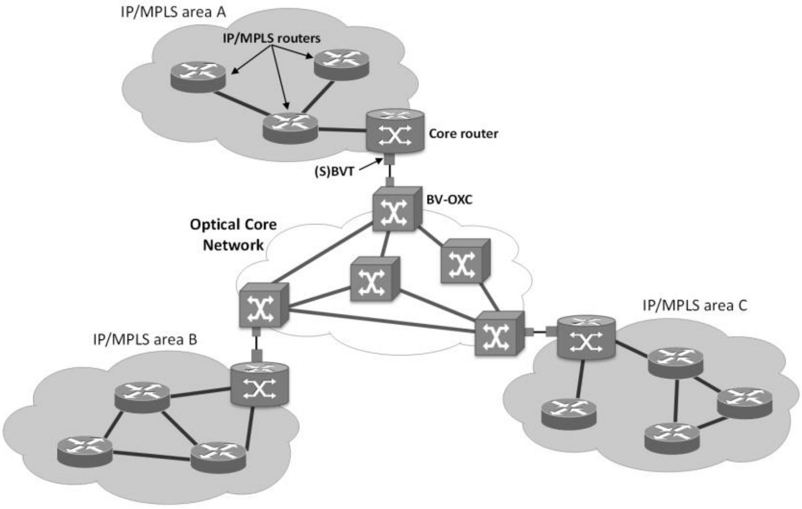


Figure 6 Packet and Optical Transport Network Architecture

With the inception of Software Defined Networking (SDN) and the capability to provide centralised control of resources and using programmatic flow-based technology, like OpenFlow [8], offers an alternative method for traffic assurance objectives described previously.

In this document we will discuss how using SDN-based network principles, it is feasible that specific traffic types may be separated; enabling general traffic to be controlled using the traditional distributed routing protocol, and the assured traffic is controlled and influenced by the centralised controller.

Several technology approaches exist for introducing SDN and related technologies into the network to accomplish the objectives. The complete solution is to use green-field engineering, centralised controllers with OpenFlow-based forwarding technology, which distributes entire flow tables to every network device on the traffic path. However, such solutions are viable in theory but not always applicable in practice, especially within the existing WAN environments of large operator environments, due to existing router deployments and large numbers of deployed nodes and links.

Increasingly traffic engineering is used in transport networks; a traffic-engineered network will use multiple mechanisms to facilitate the split of the data plane and control plane [9]. They also have a range of management and provisioning protocols to configure and activate traffic engineered network resources [10]. These mechanisms represent key technologies for enabling more efficient networking. However, a significant limiting factor of traffic engineering is the skills and time required to design and deploy.

Actual deployment of MPLS-TE relies on offline methods, typically using forecasted traffic demands. Operational tools do not react to real-time traffic changes caused by BGP reroutes, diurnal traffic variations, catastrophic failures, or network attacks [11].

1.5 Legacy Network Control

The connectionless Internet, running of TE-based MPLS services, represents an example of a significant network determinism problem [12]. Vast numbers of administrative regions loosely tied with interconnections that are constantly changing as traffic patterns fluctuate and failures occur. Inherent weaknesses exist some methods exist to minimise these; the Internet is federated with distributed control, where individual nodes participate together to exchange reachability information to develop a localised view of a consistent, loop-free network using IP forwarding. The Internet IP forwarding paradigm, where routes and reachability information are exchanged that later results in data plane paths being programmed, is often sub-optimal and prone to traffic congestion, packet loss and delay, so clearly this approach is not suitable for our end-to-end traffic assurance objectives.

As Internet evolution continued, the trend was that the integrated voice, video, and data should be transported using a converged IP of an MPLS core network. By combining the use of the differentiated services (DiffServ) and MPLS forwarding, operators can deliver a guaranteed quality of service (QoS) [12]. Again, significantly, offline planning is required, and operational tools are not capable of responding to real-time traffic changes or demands.

As network technology evolved and the concepts of SDN were established as applicable for core Internet invented (logically centralised control, separation of control and forwarding, and network programmability), addressing the weaknesses of a federated and distributed IP network. It is much easier to pursue an objective for end-to-end traffic assurance using a centralised management environment with fine-grained control of the forwarding elements.

One early proposal simplifying network hardware, while improving the flexibility of network control using MPLS and centralised control platforms – based on the principles of SDN – was “Fabric” [13]. The authors proposed a network “fabric” based on MPLS, and subsequent labels and encapsulation, it uses the egress edge switch to encode the path that packets must follow to be delivered to the destination host or server. The ingress switches are managed by an “Edge Controller” [13] which would compute the path of the end-to-end services.

The Fabric proposal mentioned above differed significantly from current Internet forwarding architecture, outlined in Figure 7 (Current Control and Forwarding Decision Logic) and discussed below.

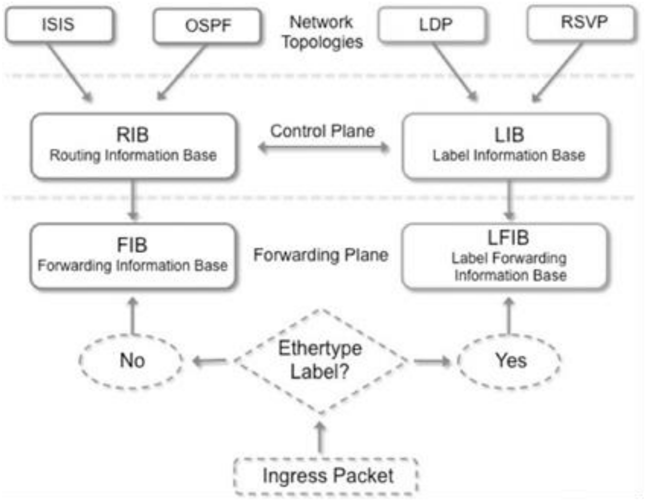


Figure 7 Current Control and Forwarding Decision Logic

Overtime it was identified that TE-based MPLS services often met scaling limitations, these included:

- Network Management: How many connections may an NMS process actually process?
- Protocol Overhead: operating large numbers of connections and subsequent protocols may overload the control plane;
- Node Resources: Depending on the number of connections there are additional memory, and CPU requirements for deploying distributed control planes;
- Service Setup Time: Based on the size, numbers of links and nodes, and overall complexity of the network, there will be a linear degradation of connection setup times, especially when optimising multiple path constraints.

1.5.1 Multi-Domain and Multi-Technology

In general, a domain is defined as “any collection of network elements within a common sphere of address management or path computational responsibility.” Often, these examples would include Interior Gateway Protocol (IGPs) areas and Autonomous Systems (ASes).

In the context of this document, and next generation transport networks, a particularly important example of a domain is an optical network technology environment. These networks do not tend to interoperable between optical venders, as each manufacturer will have a flavour of optical interface and control technique. Thus, we often must consider transport networks are considered multi-domain and multi-technology.

1.5.2 Intra-domain Connectivity

For “intra-domain” connectivity the control plane establishes the network path via routing protocol participation by creating a local rule set used to create the forwarding table entries. The data plane is then programmed to forward traffic between incoming and outgoing ports on a node. The foundation of the current Internet (IP) control plane model is to use an Interior Gateway Protocol (IGP), this IGP may be in the form of a link-state protocol such as Open Shortest Path First (OSPF) [14], or the even older Intermediate-System-to-Intermediate-System (ISIS) [15], which is still the de facto the de facto standard for large service provider backbone networks. Either OSPF or ISIS will provide a method to establish layer-3 reachability between a connected, acyclic graph of IP forwarding elements.

Layer-3 network reachability information primarily concerns itself with the reachability of a destination IP prefix. In our network, layer-3 is used to segment or stitch together layer-2 domains to overcome layer-2 scaling problems for end-to-end services that interconnect data centres. The routing table contains the next hop and destination layer-3 addresses and the outgoing interface(s) associated with them.

Although control plane logic can define certain traffic rules, for priority treatment of specific traffic for which a high quality of service for differentiated services, however, it is occasionally not possible to guarantee service attributes such as minimum bandwidth, overall latency, and acceptable application jitter, due to the fact traffic forwarding is based on the reachability of network addresses.

It is important for a future control plane framework to be agnostic to the underlying connectivity technology and reachability information, as these schemes will generally evolve, and various methods of address abstraction may be applied.

1.5.3 Inter-domain Connectivity

For "inter-domain" connectivity additional control plane mechanisms are required. The predominant inter-domain routing technology is the Board Gateway Protocol (BGP) [16]. Currently, BGP is the predominant protocol for intra-domain Internet routing. Facilitating connectivity between different Autonomous Systems (ASs), often requiring manual policy control of which transit paths to take to destination ASes. The decision tune the BGP configurations to express policies, reflecting how the AS connects to others, to meet operator business requirements is very much a manual process [17] requiring expert BGP engineers and knowledge.

Within the control plane, the computation to create BGP flow rules are defined, typically via manually defined policies using a command line (CLI) to the router or switch. An abstraction of a BGP routing instance is provided below in figure 8 (Abstracted BGP control plane).

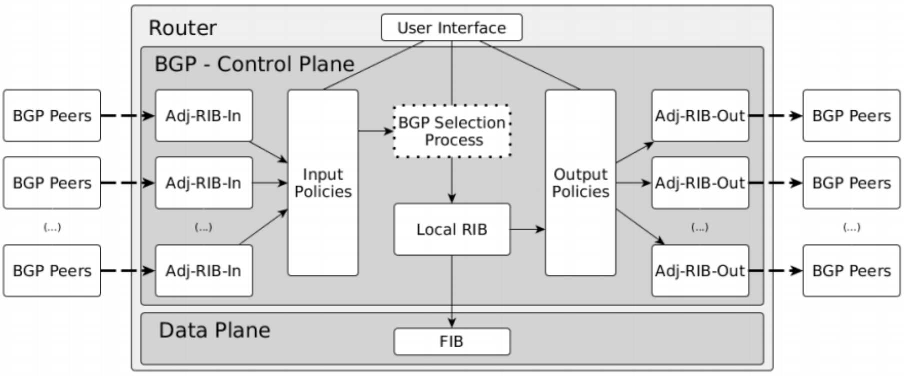


Figure 8 Abstracted BGP Control Plane [20]

The BGP control plane uses various procedures and messages to exchange Network Layer Reachability Information (NLRI) with participating routers and switches to create a network graph, reflecting the desired inter-domain routing policy, i.e., "best path", to a specific AS, or set of ASes.

1.5.4 End-to-End Transport Signaling

In addition to the routing processes and protocols described previously, a Resource Reservation Protocol (RSVP) [18] may be used for end-to-end connectivity management. In a transport network RSVP-TE is used to establish MPLS transport connections (LSPs), when there are traffic engineering requirements, such as minimise cost, include or exclude specific links, nodes or existing services, maximise the use of links with the most available bandwidth (Least Loaded Routing - LLR, furthermore it was adapted to include the ability to control a wide-variety optical network technologies.

It may be considered that RSVP-TE is akin to source routing where the ingress node determines the complete path through the network, overtime this path computation responsibility was delegated to complex NMS platforms, or network design teams.

An RSVP-enabled network would enable Internet applications to be assigned differing qualities of service (QoS) for application data flows; this would allow different applications to be assigned path resources to meet the divergent performance requirements of different application types. In the transport network, RSVP-TE [19] may be used. Thus, allowing the establishment of MPLS connections and taking into consideration constraint parameters such as available bandwidth and explicit hops, providing deterministic control, and forwarding guarantees for bandwidth or latency sensitive applications.

To effectively manage network resources, and establish connections using signaling, operators would typically build large planning teams, in the case of British Telecom when they deployed MPLS-TE in 2004 they had to build out a 300-person planning time and recruit key MPLS-TE experts from around the work.

1.5.5 End-to-End Transport Example – BT Media and Broadcast

In 2005 British Telecom (BT) Media and Broadcast built one of the most advanced networks using MPLS-TE (RSVP-TE) [20]. The BT MPLS network ("Common Network Platform" - CNP) formed their basis of a multi-service platform. An initial use case was to transport real-time broadcast television traffic for the British Broadcasting Company (BBC). Their platform formed the first phase of a new generation strategic IP/MPLS-based national infrastructure, one of the first of its type in the World to carry real-time broadcast video traffic. The CNP topology is shown in figure 9 (British Telecom CNP Topology [19]) below.

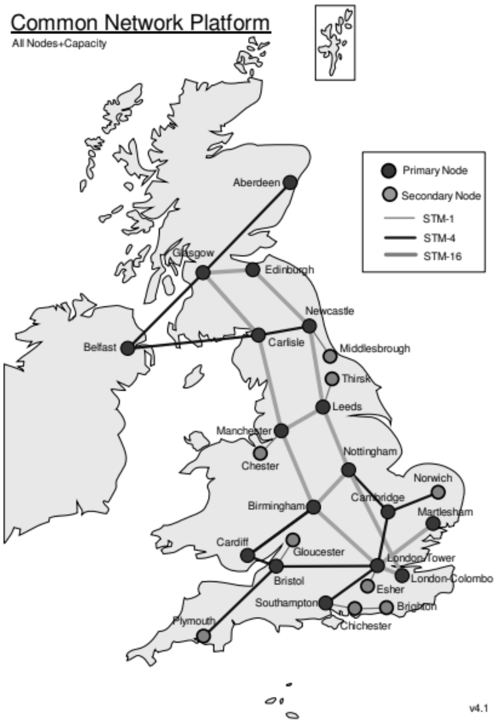


Figure 9 British Telecom CNP Topology [19]

More recently, the obvious benefits of the MPLS-TE concepts led to the development of "generalised" extensions to MPLS-TE, this is known as Generalized MPLS (GMPLS) [21]. GMPLS is an extension to MPLS designed to support optical wavelength management, but instead of using an explicit label to distinguish an LSP at each router, some optical physical property (typically wavelength grid identifier) of the received data connection is used to deduce which LSP is used. A key benefit of GMPLS is that it can be used to establish LSPs for various underlining transport types, including packet, Time Division Multiplexing (TDM) and Wave Division Multiplexing (WDM) based services. Using either a TDM and WDM example, the LSP traffic is switched based on a non-stop data-stream and not switched per single packet, stop, look-up, forward, principle. Thus, providing an extremely efficient implementation in the data plane with zero per-packet lookups, making GMPLS highly suitable for converged high bandwidth networks, including BTs CNP.

The Generalized Multi-Protocol Label Switching (GMPLS) architecture **Error! Reference source not found.** comprises of:

- A link/neighbour discovery/verification protocol, such as the Link Management Protocol (LMP) [22] or Link Layer Distribution protocol (LLDP) that allows neighbouring nodes part of the control plane adjacency to associate data plane adjacencies (e.g. fibre links), correlate identifiers and to assure compatible capabilities;
- Routing protocols. The Open Shortest Path First (OSPF) protocol specification describes the characteristics of nodes and links, so the state and capabilities of the resources are distributed and updated to all nodes, knowing which resources are in use, out of service, or available;
- A signaling protocol. The ReSerVation Protocol with Traffic Engineering extensions (RSVP-TE) is used to set up Label Switched Paths (LSPs). RSVP-TE messages specify the path of the LSP, request specific capacity on the path, and report back the exact allocated network resources to support the LSP.

1.5.6 End-to-End Transport Path Computation

A key aspect is determining what path an LSP should follow. This function can be performed externally (the path is supplied to the control plane) or delegated to the control plane. In either case, the computation can be complex. A method for computing end-to-end paths automatically, without the need for highly skilled engineers, was proposed called the Path Computation Element (PCE) [23]. The PCE is a functional component that can be queried using the Path Computation Element Communication Protocol (PCEP) [24], recently extended to allow the network to delegate control of an LSP to a PCE and allow a PCE to direct the establishment of new LSPs (becoming an active PCE) [25] & [26].

1.5.7 Transport Technology Evolution

As the GMPLS and PCE architecture and protocol suite continues to develop new extensions were introduced for a range of new high-bandwidth optical transport types. Until recently, the large available optical; spectrum provided by optical fiber was expected to offer significantly more bandwidth than required but exponential growth in consumer Internet bandwidth, IoE, Machine-2-Machine (M2M) and Industry 3.0 and beyond, put significant pressure on maximising fibre resources. Adding more capacity to an existing fiber was a simple matter of adding additional wavelengths, making use of the fact that at low enough power levels, multiple waves can be supported on the same fibre. Telecoms research for transport networks has now focused on two related areas; firstly, how to manage the spectrum more effectively, and secondly how to fill the spectrum up as much as possible with light signals.

British Telecom has been at the forefront of transport network research for its network requirements [27], including the use of Wavelength Switched Optical Networks (WSONs) [28]. More recently BT has been developing Variable Bitrate DWDM (Dense Wave Division Multiplexing) Transponders (VBT) called “Flexi-Grid” [29], allowing the slicing optical transport lambdas into bandwidth amounts based on specific user and application demands. This technology is highly anticipated to be of great benefit to BTs networks and services.

1.5.8 Network Planning

In a legacy network the process of network optimisation is a gradual network planning process, where the following process is assumed:

- the Network Management System (NMS) managing the core network, implementing fault, configuration, administration, performance, and security (FCAPS) functions;
- a Planning Department administrating the planning process, i.e. analysing the network performance and finding bottlenecks, receiving potential clients’ needs, evaluating network extensions and new architecture;
- an inventory database containing all equipment already installed in the network, regardless they are in operation or not;
- an Engineering Department, performing actions related to equipment installation and setup;
- network planning tool in charge of computing solutions for each migration step. Since several sub-problems related to network reconfiguration, planning, and dimensioning.

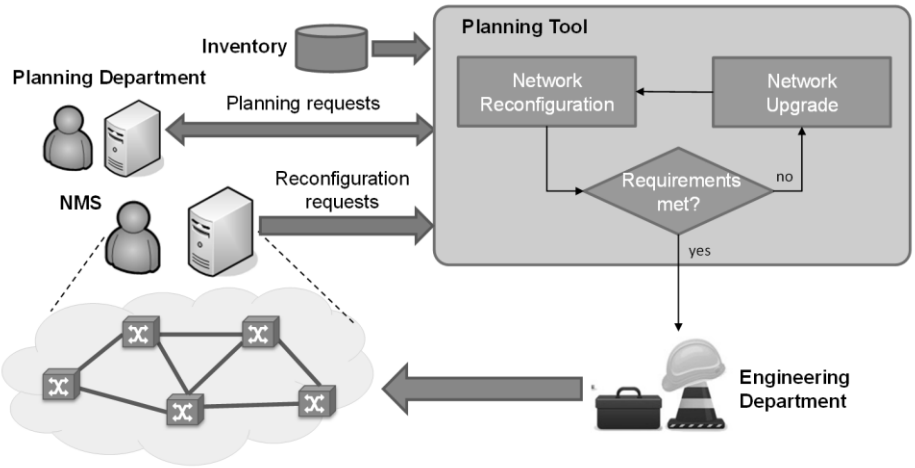


Figure 10 Legacy Network Optimisation

This planning process meant that optimisation of resources might take days, weeks or even months.

1.6 A Need to Redefine Network Control

To support the required dynamicity and flexibility highlighted previously, a new control architecture will be required, providing integration of a wide range of transport technologies. These will have to be controlled using automation schemes and programmability features that will enable disaggregation and virtualisation concepts, the coordination of which will be supported by a purposely designed control plane. This new control plane will be dynamically adapted to meet specific service requirements, exploiting the data plane resource based on relevant data monitoring and heuristic schemes. The control plane will also be responsible for provisioning, not just existing networks, and it needs to support future 5G and Internet of Everything (IoE) applications and ensure the required end-to-end traffic performance and Quality of Experience (QoE) levels for emerging services. Therefore, an evolved control plane will have to leverage the well-established distributed control and signalling methods while utilising emerging SDN, and NFV paradigms, somehow unifying the exploit the benefits of a unified system.

1.6.1 A Question of Scale

A key requirement for future transport networks will be a control framework that is capable of scaling in large multi-domain and multi-technology environments. Whereas traditionally distributed control plane nodes have practical limitations for topology and service state, due to their physical memory and CPU limitations. A centralised control plane that is running in a data center would have a significant amount of general compute and storage that could be added to a virtual machine running the centralised controller. Capable of scaling up, by increasing memory footprint and adding logical CPUs as the number of nodes, links and services increase.

1.6.2 Network Control Objectives

An operator must integrate multiple IP technologies allowing network infrastructure to deliver a variety of services to support the different characteristics and dynamic demands of high bandwidth Internet and Cloud applications. In addition to the end-to-end assurance objective, there is an increasing demand to maximise network resources, provide efficient and responsive service paths and setup, facilitate connections on demand and well-within specific time periods, seconds, minutes or hours, when required. Consider that these goals differ greatly from the established methodology, where services in the network are created in response to CLI commands driven by humans directly, and using a wide variety of Operational Support Systems (OSS), and where networks are typically over-provisioned to ensure minimal traffic loss, even at peak traffic periods.

Our adoption of SDN and specifically a logically centralised controller principle provided the cornerstone for our objective to have traffic assurance, more efficient network usage and provide a foundation for further service innovation in the future. We use the term logically centralised to signify that network control may appear focused in a single entity, independent of its possible implementation in distributed form. The centralised control principle states that resources can be used more efficiently when viewed from a global perspective. A network controller would have to be developed so that it combined several technology components, mechanisms, and procedures. These included:

- Application and OSS requests for network resource availability information and existing connectivity;
- Discovering and disseminating network resource information;
- An analysis of traffic applications, and their mapping to underlying network resources;
- Management and coordination of-of path computation request, computation and response;
- Storing existing resource information, provisioning and reserving network resources;
- Overall verification of connection and resource setup.

The network controller would also need to be capable of orchestrating resources that span several subordinate domains (Data Center, WAN and Access) and in cooperation with other entities, and thereby offer resource efficiency when setting up end-to-end services and overall operation of network resources used to provide those end-to-end services.

Other reasons for adopting a logically centralised control architecture include scale, optimisation of information exchange and minimisation of propagation delay. Given constraints of not being able to deploy green-field networks, it is necessary that a controller co-exists with both native IP forwarding technologies, non-native SDN traffic engineered technology (MPLS-TE), and flow-based technologies (including OpenFlow).

The 2015 SIGCOMM paper: "Central Control over Distributed Routing" [30] proposed a solution for influencing the IGP protocol's decision via introducing pseudo fake nodes into the network. Although the proposal is sound and may be validated via lab simulations, there are significant issues for deploying the proposal into an actual live network because of the topological changes to the entire network and inability to troubleshoot network connectivity problems via a centralised point.

Other methods of proposing centralised a control architecture that utilises the traditional routing protocols and procedures include the Routing Control Platform (RCP) described in [31]. The RCP acts very similarly to a BGP Route Reflector [32] solution that is deployed widely within current networks. By using the BGP protocol to influence the decision of the BGP path selection algorithm. Originally the RR was developed to negate the need for a logical full-mesh requirement of Internal Border Gateway Protocol (I-BGP). RR acts as a central point for I-BGP sessions, allowing for multiple BGP routers to peer with a central router (the RR) acting as a route reflector server, removing the need to for other I-BGP members to peer with every other router in a full mesh. However, although useful (for to address mesh N-squared problems) the RR cannot build dynamic, dedicated paths for the assured traffic, and the potential to resize existing paths (i.e., connection bandwidth elasticity).

One key design objective of a platform to manage end-to-end traffic assured services is to forward packets through a core network that is IP-enabled, but that has no support for MPLS forwarding. A further requirement is that the network should be able to traffic engineer that traffic is sending specific flows down predictable paths and reserving the resources on those paths for just those flows, without requiring the deployment of MPLS-TE.

At the same time, the core network needs to able to operate as a normal IP network in other respects. That is, it must be able to continue to forward IP packets for other traffic and flows that are not part of the Cloud connectivity services or offering bulk connectivity between data centre sites.

Typically a core network must operate normal interior routing protocols (OSPF or IS-IS) as well as external routing (BGP), and it must enable normal IP forwarding on some of the interfaces in the network while other interfaces may be reserved for assured flows. Furthermore, the interface resources may be partitioned so that there are reserved resources for assured flows while another IP forwarding can continue as the best effort service.

1.7. Leveraging Telecommunications Innovation using Software Defined Networking and Network Function Virtualisation

Although SDN technologies and architecture are well known, "NFV" is a relatively recent technological phenomenon that has the potential to disrupt the telecommunications industry, its entire hardware and software supply chain and ultimately its approach to servicing both commercial and domestic consumers around the globe. The development of this new virtualisation technology, which does not yet exist, is initially championed in a collaboratively produced white paper, co-published in October 2012 (NFV White paper, 2012) by 13 of the world's largest telecommunications network operators ('operators'). The head of Network Evolution Innovation at British Telecom stated that NFV is "likely to dramatically change the telecom landscape and industry over the next 2-5 years innovative methods to build and manage networks, spawning a new wave of industry-wide innovation".

1.7.1 SDN for Flexible Transport NFV

Software Defined Networking (SDN) will underpin a dynamic, flexible, cost-effective, and elastic, making it valuable for the high-bandwidth, dynamic nature of emerging Internet applications. The ONF architecture proposal decoupled network control and forwarding functions. A significant step in enabling network state management to be used for the direct programming and the forwarding infrastructure, whilst also capable of being abstracted for applications and network services. Thus, the ONF's OpenFlow (OF) protocol [8] was a foundational element for building SDN architecture:

- **Directly programmable**
    - Network control is directly programmable because it is decoupled from forwarding functions
- **Agility**
    - Abstracting control from forwarding lets administrators dynamically adjust network-wide traffic flow to meet changing needs
- **Centrally managed**
    - Network intelligence is (logically) centralised in software-based SDN controllers that maintain a global view of the network, which appears to applications and policy engines as a single, logical switch
- **Programmatically configured**
    - SDN lets network managers configure, manage, secure, and optimise network resources very quickly via dynamic, automated SDN programs, which they can write themselves because the programs do not depend on proprietary software
- **Open standards-based and vendor-neutral**
    - When implemented through open standards, SDN simplifies network design and operation because instructions are provided by SDN controllers instead of multiple, vendor-specific devices and protocols.

The central challenges and use cases facing operators, including BT, are identified in the NFV white paper (operator written problem statement of current network infrastructure) [33]. Additionally, the NFV Use Cases document (which use cases should be supported by NFV) [34] provides key objectives, including increasing network operating costs, greater physical space and power requirements, and longer deployment times associated with any network growth or increase in functions. These challenges are attributed to the lock-in effect of having multiple layers of proprietary hardware, operated with proprietary software, and the need for physical installations each time new functions, increased capacity or technology developments require it. Within the operators' competitive environment, where technological innovation means ever-shortening lifecycles for hardware and a need to deploy services faster, the costs of running a hardware-dependent network are increasing significantly.

A key goal proposed by BT and the other NFV proponents is to remove proprietary hardware from every point in the network where it is possible to do so, replacing it with software running on commodity hardware located in a small number of centralised data centres.

In the first paragraph of the NFV white paper [33], the operator-authors assert that the status quo is constraining innovation, increasing costs, and leading to ever increasing demands for space, power and physical installations that delay the deployment of new network functions. Acting together, the world's largest operators aim to define a new structure for their industry that allows them to collectively specify the use of commodity hardware (high volume, low-cost servers) to facilitate the growth of an open ecosystem of many more large and small software providers who can create software-based network functions that sit on these servers.

The NFV concept being specifically promoted is to support software-based network functions as described in figure 11 (“Vision for Network Functions Virtualisation”) below.

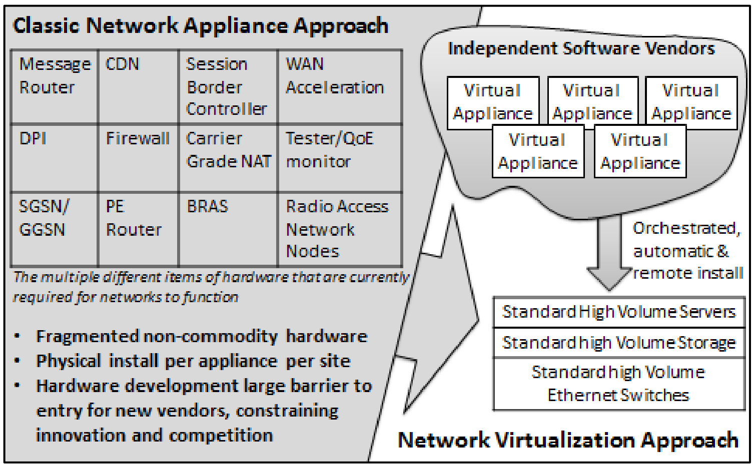


Figure 11 Vision for Network Functions Virtualisation

BT Builds large complex transport network infrastructure so being able to virtualise (via software) fixed and proprietary optical transport devices, would be highly beneficial. By combining NFV and three key aspects of SDN, would yield numerous benefits for BT networks and services.

The key concepts of SDN are outlined in figure 12 (“Open Networking Foundation SDN Architecture [35]”) below.

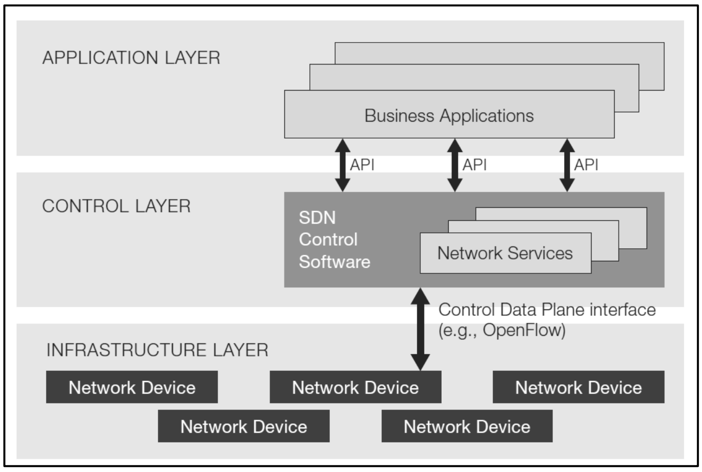


Figure 12 Open Networking Foundation SDN Architecture [35]

Three key concepts of SDN that could be combined with NFV are:

1. **Separation of Control and Forwarding**: this would allow application requests from connection deployment, allowing configuration and management of network state to be transparent from the users, but remains responsive to the application or service;

2. **Network Abstraction**: a process of applying the policy to a set of information about a traffic engineered network to produce selective information that represents the potential ability to connect across the network. The process of abstraction presents the connectivity graph in a way that is independent of the underlying network technologies, capabilities, and topology so that the graph can be used to plan and deliver uniform network services;

3. **Automatic Allocation and Coordination of Resources**: by orchestrating requests across multiple technology layers to provide end-to-end services, regardless of whether the networks use SDN or not.

As networks evolve, the need to provide support for distinct services, separated service orchestration, and resource abstraction have emerged as key requirements for operators. Capable of supporting multiple applications and services, while meeting exponential bandwidth demands.

2. Background

The “data plane” (which is also referred to as the user plane, forwarding plane, carrier plane or bearer plane) is the node component that carries user traffic. Generally, there are three basic components of a telecommunications architecture: data plane, control plane and management plane. Typically, the control plane and management plane concurrently interact with the data plane, which forwards the traffic that the network exists to carry.

There are multiple data plane technologies using a wide variety of physical interface types. The data plane receives and sends packets from the interface and processes them as required based on the transport protocol method, delivering, modifying, or dropping as appropriate.

The “control plane” is the software that controls devices in the network, including switching devices, routers. It typically maintains a real-time view of a “network”. The control plane should react to changes in the network state, and recover from limited failures, without specific human intervention.

The control plane also creates a view of the network state of the network nodes and interfaces and provides a set of useful abstractions for an end-to-end service, hence the notion of network “connections”.

2.1. Evolution of Network Control Architecture

2.1.1 Role of Management Systems in Networks

The advent of network management became prominent for the supporting SONET and SDH based transport systems. Historically, the transport system was based on manual checks and by performing various measurement tests often requiring onsite visits. A key feature of an SDH based transport system was the use of a dedicated management channel for carrying network management data, including configuration, performance, and alarms.

The term Element Management Systems (EMS) was defined with key interfaces into network devices to provide information to and from the transport node, as well as configuration of the systems itself. The EMS performed the functions of Fault Management, Configuration Management, Accounting Management, Performance Management and Security Management, known as “FCAPS” functions.

Typically, an EMS would manage one or more of a specific (vendor or technology) type of Network Element (NE). To facilitate management of the traffic between NEs, the EMS communicates upward to a higher-level component, the Network Management Systems (NMS, this was defined in Telecommunications Management Network (TMN), layered model. The EMS provides the foundation to implement TMN Operations Support System (OSS) architecture, which enabled operators to meet service activation needs for rapid deployment of new services, with defined quality of service (QoS) requirements. The Tele Management Forum (TMF) common object request broker architecture (CORBA). At the time this EMS to NMS interface represented a new era in OSS interoperability and overall network control.

2.2. Network Control Functions

A central principle of SDN is the separation of a network forwarding and control planes. By separating these functions, a set of specific advantages regarding centralised or distributed programmatic control. Firstly, there is a potential economic advantage by using commodity hardware rather than proprietary specific hardware. Secondly, remove the need for a fully distributed control plane with capability often requiring senior engineering experience to deploy and operate, with a wide range of features, which are very often underutilised. Thirdly, the ability to consolidate in one or a few places what a considerably complex piece of OSS software is often to configure and control network resources.

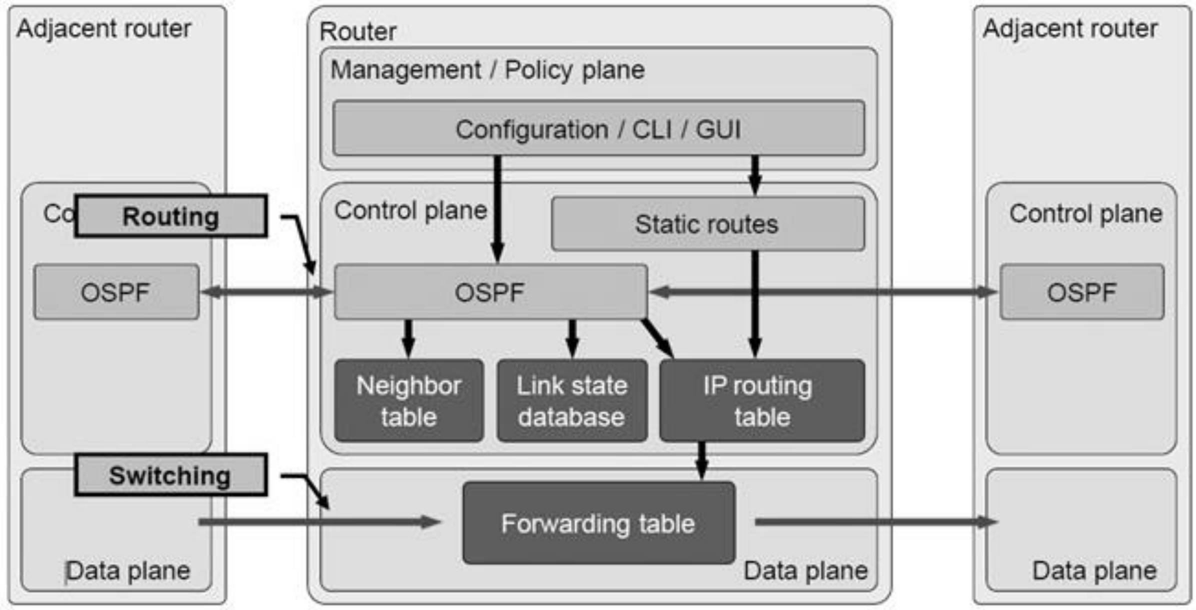


Figure 13 Conventional Router Architecture for Control Decisions

Typically, the network operator has followed a prescribed path for a hardware upgrade to circumnavigate the networking scaling issues. This requires the operator to consider the node forwarding performance versus price-to-performance numbers to pick just the right time to participate in an upgrade. Conversely, as network topologies increase the complexity of the control plane and scalability will also need consideration.

The Internet represents an example of a significant scaling problem — comprising of vast numbers of administrative regions loosely tied with the interconnections constantly changing as traffic patterns fluctuate and failures occur. Therefore, to address the control paradigm, the Internet was designed accordingly. Its structure was federated, where individual nodes participate together to distribute reachability information to develop a localised view of a consistent, loop-free network using IP forwarding. The Internet forwarding paradigm, where routes and reachability information is exchanged that later results in data plane paths being programmed to realise those paths, however, paths are often sub-optimal and prone to traffic congestion, so clearly this approach has weaknesses which might be addressed using a centralised approach.

As network technology evolved and the concepts of SDN were invented (centralised control, separation of control and forwarding, and network programmability), the cycle of growth and scaling management and upgrade in the control plane to accommodate scale, was a clear objective. It is much easier to pursue solutions for a centralised management environment controlling distributed, but simple, forwarding elements.

2.3. Control Plane

The control plane facilitates resource discovery and reachability and builds the network link and node map. Control plane functions include: participation routing protocols processes but are path control elements. This establishes the local rule set used to create the forwarding table entries, interpreted by the data plane, to forward traffic between incoming and outgoing ports on a node.

The main functionalities that the control plane provides include:

Provisioning (set up and tear down) of connections: The control plane automatically configures all necessary devices to create a connection between two (or more) points in the network. The process by which the control plane configures different elements to set up a connection is known as signalling.

Restoration: Upon a failure in some element of the EON, a connection may no longer be able to meet the necessary QoS required for the transmitted service. In this case, through the restoration process, the configuration of the network is changed so that the connection satisfies the desired quality again. The restoration process usually implies a change over the “physical” path of a connection.

Automatic network element discovery: The control plane automatically discovers which elements are present in the network.

Routing: The control plane automatically builds a topological view of the network; it discovers the connections among network elements and keep the information up to date. Based on this discovery, a topological graph comprised of nodes and edges is built as an abstracted view of the topology. Also, traffic engineering (TE) information (e.g., available spectrum and the shared risk link group information of a link representing a fibre) is also added to the graph.

Path Computation: using a network graph, traffic engineering capabilities of both edges (e.g., availability of spectrum) and vertexes, i.e., connectivity matrix between incoming/outgoing edges), the path of service is computed. Constraints (e.g., Shared Risk Group (SRLG)) and optimisation objectives, such as cost, can be applied to the computation.

The typical transport network will use control plane architecture based on GMPLS [1] and PCE based, and we will investigate its details in this chapter. This architecture relies on distributed communication between control elements and occasionally, between control elements and a central element such as in the network with the Path Computation Element (PCE) [6] and Software Defined Networks (SDN) [27], both of which requires communications between all configurable elements and a centralised controller.

The foundation of the current IP control plane model is to use an Interior Gateway Protocol (IGP). This normally is in the form of a link-state protocol such as Open Shortest Path First (OSPF) or Intermediate-System-to-Intermediate-System (ISIS). The IGP will establish layer three reachability between a connected, acyclic graph of IP forwarding elements.

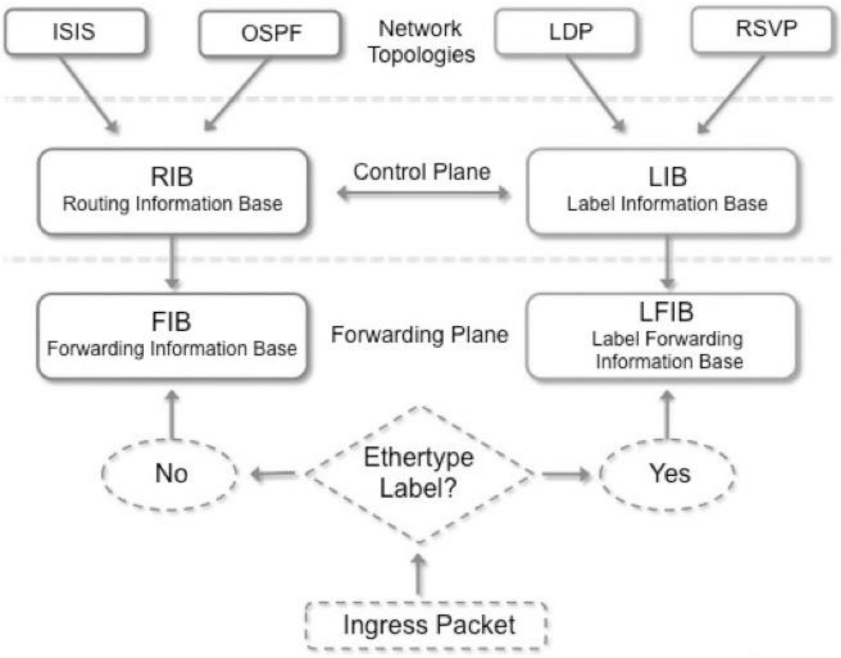


Figure 14 Relationship of Control and Data (Forwarding) Plane

Layer-3 network reachability information primarily concerns itself with the reachability of a destination IP prefix. In all modern uses, layer three is used to segment or stitch together layer two domains to overcome layer-2 scaling problems. Traditionally, the routing table contains a list of destination layer-3 addresses and the outgoing interface(s) associated with them. Control plane logic can define certain traffic rules, for priority treatment of specific traffic for which a high quality of service is defined known as differentiated services. Forwarding focuses on the reachability of network addresses.

The role of the control plane includes:

- Network topology discovery (resource discovery)
- Signaling, routing, address assignment
- Connection set-up/tear-down
- Connection protection/restoration
- Path Computation & Traffic engineering

A few downsides of current control plane technology are the fact they are generally distributed. This requires significant memory and CPU overhead for each device, to implement the necessary protocol mechanisms and procedures, which include: neighbour discovery, keep-alive mechanisms, both internal and external routing protocols. Furthermore, a significant amount of expert knowledge is also required to configure and deploy distributed control plane technology.

2.3.1 Distributed versus Centralised Control

A control plane needs to address common functions like addressing, automatic topology discovery, network abstraction, path computation, and connection provisioning, as stated earlier. For this research and the overall controller design, the continued use of a control plane fulfilled the requirements of reusing IP technology and automatic end-to-end provisioning and rerouting of connections, while supporting different levels of quality of service.

From a high-level perspective and as any software system that automates tasks and processes, the functions of a control plane can, may be distributed or centralised.

This dichotomy applies not only from a functional perspective but also from a resource allocation perspective. Both models were viable in our controller design; both have their strengths and weaknesses and must be extended to meet the emerging requirements. Thus, the selection of a centralised or distributed control plane is conditioned by diverse aspects, such as the desired functions, flexibility and extensibility, availability, etc., as well as by more concrete aspects such as the inherent constraints of the application and service.

Table 1 Analysis of Control Plane Architecture

Architecture	**Features**	**Strengths**	**Weaknesses**
Centralised	• Global view of network resources • Vendor and technology data plane agnostic	• No need for node control plane intelligence or state • New southbound APIs can be supported directly from the centralised controller	• May not reflect rapid state changes in distributed network notes • Service setup scalability in large networks • Single point of failure
Distributed	• Highly-available by design as no single-point-of-failure • Policies can be applied locally at the node level	• Significantly better scalability • Easier to implement protection mechanisms at local node interfaces	• No global network resource view • Computational resources for control plane actions required locally
Hierarchical	• An overall global abstracted view of network resources • Capable of integrating new lower-layer technologies	• Scalable • Delegates technology specific control to child controllers.	• The top-level controller may still represent a single point of failure • System complexity is increased

2.4. Management Plane

The Management Plane interacts directly with the control plane and data plane; it provides management functions. It has several responsibilities, including configuration management and applying policy. It also provides Fault Management, Performance Management, Accounting, and Security Management functions.

In their early deployments, optical transport networks were inherently managed, deployed in a single administrative domain, and locked to a single vendor hardware solution (i.e., arranged into *vendor islands*). Such small and mid-sized networks, regarding some nodes, were relatively homogeneous, thus reducing interoperability issues. A single, vendor-specific Network Management System (NMS) was deployed, being responsible for the management of the optical network, tailored to the underlying hardware, and using proprietary interfaces and extensions.

Those systems were perceived as closed, bundled together as a whole, and with a limited set of functionalities that were dependent on a given release. The provisioning of a network connectivity service involved manual processes, where a service activation or modification could involve human intervention, with a user requesting the service provider, which was then manually planning and configuring the route and resources in the network to support the service.

Several challenges motivated the evolution towards the control plane. First, network operators have continuously specified requirements to reduce operational costs, while ensuring that the network still meets the requirements of the supported services. Second, the manual, long-lasting processes associated with NMS-based networks did not seem adapted for the dynamic provisioning of services with recovery and Quality of Service (QoS). In short, the introduction of a dynamic control plane was justified, from an operational perspective, for the automation of certain tasks, freeing the operator from the burden of manually managing and configuring individual nodes, leading to significant cost reductions.

In this context, the introduction of a control plane aims at fulfilling the requirements of fast and automatic end-to-end provisioning and re-routing of flexi-grid connections, while supporting different levels of quality of service. Regardless, of the actual technology, a control plane needs to address common functions like addressing, automatic topology discovery, network abstraction, path computation, and connection provisioning, as stated earlier in this chapter. From a high-level perspective, and as any software system that automates tasks and processes, the functions of a control plane can, from a simplistic point of view, be distributed or centralised, although we will later see that this separation is becoming blurry. This dichotomy applies not only from a functional perspective but also from a resource allocation perspective. Both models are viable; both have their strengths and weaknesses, and both are being extended to address the new requirements associated to the emerging optical technologies, such as flexible spectrum allocation, efficient co-routed connection setup and configuration of related optical parameters.

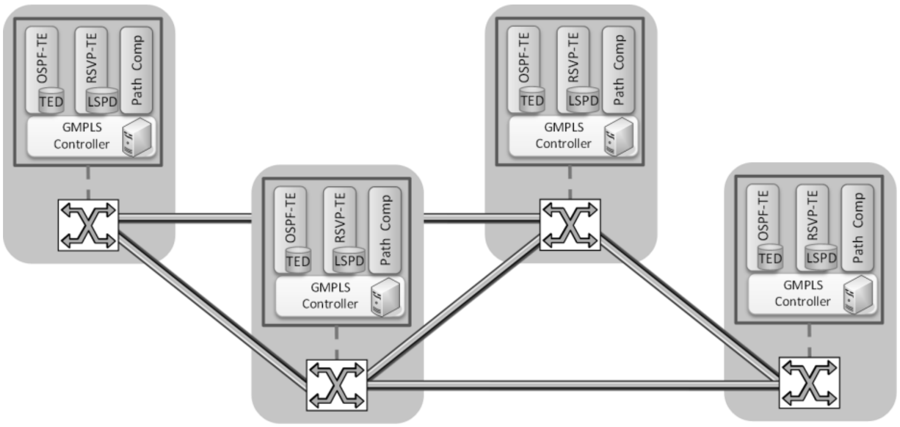


Figure 15 Example of GMPLS-controlled Optical Transport Network

The network elements participating in distributed control plane environment exchange the accumulated advertisements from other nodes in a state database (e.g., OSPF database) and run a Dijkstra (shortest path) algorithm to establish a reachability graph of best paths to destinations. This process uses a distributed flooding algorithm within the IGP protocol procedure to propagate attachment information, thus, all nodes speaking an IGP protocol in the domain remain connected to each other (directly or indirectly) and participate with timely reachability information and establish a network topology, that reports change in connectivity in the event of failure. A key aspect is thus convergence, which is the time it takes from when a network element introduces a change in reachability of a destination due to a network. A variety of methods exist in various IGP mechanisms and procedures to address scaling of the control plane state (memory and CPU) in the network, both for physical and logical design.

2.5. Control Elements for Operating Transport Networks

The Generalized Multi-Protocol Label Switching (GMPLS) architecture [6] and protocol [21] was defined within the IETF Common Control and Measurement Plane Working Group (CCAMP WG), as an extension of the MPLS specification. The GMPLS architecture provides control plane procedures for automated provisioning of network connectivity services with functions for Traffic Engineering (TE) and network resource management. GMPLS also supports specific recovery procedures to retrieve the correct functioning of the transport network when a resource failure involving an established connection is detected [36].

The main requirements needed in a recovery procedure include:

- notification of the failure
- fault isolation,
- reestablishment of the faulty connections.

This latter reconfiguration action may be implemented using two different mechanisms:

- Protection: when the recovery paths are pre-planned, pre-computed, pre-signalled and pre-committed;
- Restoration: when the recovery paths can be either pre-planned or dynamically allocated, but on-demand additional signaling is always needed to establish the restoration path.

The GMPLS architecture was developed to support a variety of traffic engineered transport switching technologies; these included: packet, Layer-2, Time Division Multiplexing (TDM), WDM and DWDM fibre and emerging wavelength switching technologies). Extensions in the GMPLS framework, signaling (RSVP-TE) and routing (OSPF-TE) protocols were developed to support specific technologies like Wavelength Switched Optical Networks (WSON), G.709 Optical Transport Networks (OTN) [37, 38] and Flexi-Grid Networks are currently under specification and discussion in the IETF. More recently, techniques to manage Multi-Layer and Multi-Region Networks (MLN-MRN), have been proposed [39]. This is because transport networks are more complicated and often comprised of multiple types of data plane forwarding, or multiple transport layers, all under managed using a single instance of the GMPLS control plane - early implementations exist of a centralised GMPLS control plane, but generally GMPLS is used in a distributed manner.

2.5.1. Path Computation

A fundamental aspect of GMPLS routing is the path computation process. To this purpose, the IETF PCE WG defines architectures and protocols for path computation. Path computation manages aspects related to finding a physical route between two network nodes, commonly referred to as endpoints. Path computation is a functional component of a control plane, invoked for (dynamic) provisioning, re-routing, restoration, as well as advanced use cases such as overall optimisation, adaptive network planning or, in the case of DWDM flexi-grid networks, spectrum de-fragmentation.

The Path Computation Element (PCE) framework in [23] outlines two main components: the Path Computation Client (PCC) and a Path Computation Element (PCE). Consider the PCC is the initiator of a path computation request, while the PCE is responsible for computing the end-to-end network paths, using a set of constraints and objective functions that may be minimised, maximised, include or exclude.

The PCE will operate looking at topology and Traffic Engineering (TE) information, via a Traffic Engineering Database (TED), for network domain it is responsible for. Extensions to the PCE exist for end-to-end inter-domain path computations, which are then performed through the cooperation of multiple PCEs.

The Internet Engineering Task Force (IETF) PCE WG specifies different models for inter-PCE cooperation in multi-domain scenarios. Firstly, the “crankback”-based Backward-Recursive PCE-based Computation (BRPC) procedure follows a peer-to-peer approach [40]. Alternatively, a hierarchical model specified in [41] called Hierarchical PCE (H-PCE) introduces the concepts of a “Parent” and a

"Child" PCEs: the parent PCE oversees the coordination of an end-to-end path computation operation. This would be based on abstracted views of the transit inter-domain topologies, provided by the cooperating with child PCEs (i.e., the management entities responsible for the internal transit domain path computation.

The PCE communication Protocol (PCEP) is the protocol regulating the interaction between PCC and PCE, or between different PCEs. It is initially defined in [24] and extended in several RFCs and IETF Drafts in support of advanced features, including point-to-multipoint services (such as video) [42], and Global Concurrent Optimization [43] to defragment resources, and network path computation in MLN-MRN environments [44].

The two main PCE models for computing end-to-end transport paths have been defined and standardised for multi-domain computation: BRPC and hierarchical PCE. The next two subsections provide a brief overview of both types of path computation.

2.5.1.1 Backwards Recursive Path Computation

The multiple PCE computation models, where different PCEs cooperate to compute the end-to-end path in multi-domain scenarios, allows limiting the flooding scope within each domain. Following this approach, each PCE has visibility only on the topology of its domain, and inter-domain flooding or neighbour domain knowledge is not required or generally available. For network scalability, a single PCE does nor have visibility of other domains, and therefore unable to compute a path that crosses any transit domain. It must communicate with other PCEs in order to obtain intra-domain path segments that can be combined to provide an end-to-end path, or an engineer must manually stitch together a path, which often takes days to design.

The current Backward Recursive PCE-based Computation (BRPC) mechanism is typically used to automate inter-domain transport paths across a predetermined sequence of transport domains, that must be identified by the operator.

2.5.1.2 Hierarchical PCE

The hierarchical PCE model is a more recent method for the multi-domain path computation problem that the BRPC mechanism described above solves. One of the earliest heuristics and procedures for inter-PCE cooperation, facilitating calculation of an optimum end-to-end path without operator engineers having to define which transit domains should be used, i.e., requiring a-priori known domain path. This model is defined in and is characterised by a hierarchical relationship between domains, each of them controlled by an H-PCE (or "Parent", also known as the broker PCE, with domain topology knowledge and policy rules for domain transport.

2.5.1.3 Impact of PCE within Software Defined Networks

SDN Is emerging as an extensible and programmable open way of operating networks. Its main concept is the decoupling of forwarding and control functions, centralising network intelligence and state information while providing to the upper layers an abstracted and vendor independent view of network state and available resources through well-defined or documented open Application Programming Interfaces (APIs). Compared to prior technologies, SDN allows network providers to build more scalable, agile and easily manageable networks. This resource abstraction, via a software layer, of the physical network facilitates "network programmable" resources.

If we agree SDN supports programmability of the network path by decoupling the data plane and allowing the removal of a distributed control plane (previously required on all forwarding nodes), which are currently integrated vertically in most network equipment's. Separation of control plane and data plane functions, then SDN becomes the underlying principle of heterogeneous technologies for a variety of transport types (e.g. optical layer, carrier Ethernet, and other traffic engineered technologies) and administrative regions.

SDN can abstract the heterogeneous transport technologies employed in the data plane and represent them in a unified way, under the umbrella of a centralised control plane. Well documented open standards, vendor and technology agnostic protocols and procedures will be needed for the SDN controller to communicate with a wide range of devices ("open" or proprietary hardware), in the data plane. These requirements established OpenFlow [8] as the de facto protocol for early SDN deployments.

In this context, the PCE architectures natively offer a solution to decouple the path computation from the forwarding plane, also providing an open standard protocol instead of using OpenFlow. This opens wide opportunities for integration of PCE in an SDN controller pre-imbued with a diverse set of control plane path computation and traffic engineering capabilities even beyond its original MPLS/GMPLS scope, with SDN. On the one hand, PCE can offload path computations to dedicated engines/elements with the aim of assisting SDN controllers for their base services, while natively providing mechanisms and procedures for cooperation among diverse PCEs in multi-domain transport scenarios [46]. The integration of PCE within SDN allows operators to utilise well-defined and well-documented routing and traffic engineering algorithms developed in the scope of PCE for SDN purposes, thus not wasting solid expertise and knowledge (e.g. from network operators) in the PCE area. These PCE models are discussed in "A Survey on the Contributions of Software-Defined Networking to Traffic Engineering" [47].

Numerous PCE and SDN integration models exist, and depending on the specific needs and PCE capabilities available: path setup for point-to-point services, multipoint or point-to-multipoint. While a stateless, stateful, or active stateful PCE may be an external application of the SDN controller, it would utilise LSP information exchanged through a dedicated set of controller northbound APIs. A stateful PCE with LSP initiation (based on application demands or in response to changing network conditions) this capability itself becomes itself a kind of controller application. Moreover, a PCE might be used to manage SDN resources for network virtualisation.

In summary, the PCE is an extremely powerful functional component with three key architectures, a wide variety of applications and use cases. It also has an extensive set of well-standardised extensions developed by the IETF. Therefore, it is likely to play a key role in the development of transport network control platforms for future networks.

2.5.2. Service Provisioning

This would include the node and interface configuration, specifically known as service provisioning — the setup and teardown of connections. The control element would automatically configure the required hops between the source and destination nodes required to create a connection between two (or point to multi-point) points in the network. The procedure and protocols used via the controller to configure different elements to set up a connection are known as either distribute via the signalling mechanisms available (such as RSVP-TE) or directly using a flow provision process (such as OpenFlow).

2.5.3. OAM and Performance Monitoring

Operations, Administration, and Maintenance (OAM) [48] is often used as a general term to describe a collection of tools for fault detection and isolation, and for performance measurement. Many OAM tools and capabilities have been defined for various technology layers.

OAM tools may, and quite often do, work in conjunction with a control plane and management plane. OAM provides analytics via instrumentation protocols and tools. These enable measurement and monitoring the data plane, nowadays OAM is known as “network telemetry”. Often OAM tools are used to record control-plane functions, and to initialise OAM sessions and to exchange various parameters. Specific OAM tools would communicate with the management plane to identify problems, raise alarms, and respond to requests activated by the management plane (as well as by the control plane) and triggered from high-layer OSS, e.g., to locate and localise problems, and initiate performance measurement of an optical segment, or end-to-end service. The role of OAM activity was typically performed by dedicated teams and engineers, and support systems.

2.5.4. Control Plane architecture evolution

In their early deployments, optical transport networks were inherently managed, deployed in a single administrative domain, and locked to a single vendor hardware solution (i.e., arranged into *vendor islands*). Such small and mid-sized networks, regarding some nodes, were relatively homogeneous, thus reducing interoperability issues. A single, vendor-specific Network Management System (NMS) was deployed, being responsible for the management of the optical network, tailored to the underlying hardware, and using proprietary interfaces and extensions.

Those systems were perceived as closed, bundled together as a whole, and with a limited set of functionalities that were dependent on a given release. The provisioning of a network connectivity service involved manual processes, where a service activation or modification could involve human intervention, with a user requesting the service provider, which was then manually planning and configuring the route and resources in the network to support the service.

Several challenges motivated the evolution towards the control plane. First, network operators continuously have specific requirements to reduce operational costs, while ensuring that the network still meets the requirements of the supported services. Second, the manual, long-lasting processes associated with NMS-based networks did not seem adapted for the dynamic provisioning of services with recovery and QoS. In short, the introduction of a dynamic control plane was justified, from an operational perspective, for the automation of certain tasks, freeing the operator from the burden of manually managing and configuring individual nodes, leading to significant cost reductions.

In this context, the introduction of a control plane aims at fulfilling the requirements of fast and automatic end-to-end provisioning and re-routing of flexi-grid connections, while supporting different levels of quality of service. Regardless, of the actual technology, a control plane needs to address common functions like addressing, automatic topology discovery, network abstraction, path computation, and connection provisioning, as stated earlier in this chapter. From a high-level perspective, and as any software system that automates tasks and processes, the functions of a control plane can, from a simplistic point of view, be distributed or centralised, although we will later see that this separation is becoming blurry. This dichotomy applies not only from a functional perspective but also from a resource allocation perspective. Both models are viable; both have their strengths and weaknesses, and both are being extended to address the new requirements associated to those above emerging optical technologies, such as flexible spectrum allocation, efficient co-routed connection setup and configuration of related optical parameters. The selection of a centralised or distributed control plane is conditioned by diverse aspects. This choice may include the desired functions, flexibility and extensibility, availability, etc., as well as by more concrete aspects such as the inherent constraints of the optical technology (e.g., the need to account for physical impairments which are collected from monitoring systems and not standardized), already installed deployments, and actual network size and scalability.

For example, the Internet represents an example of a significant scaling problem. Vast numbers of administrative regions are loosely tied with the interconnections constantly changing as traffic patterns fluctuate and failures occur. To address the Internet control paradigm was designed to be distributed. On the other hand, SDH/Optical core transport networks, while geographically spanning national or continental regions, are still relatively small in size /number of elements when compared to IP networks, and are commonly under the control of a single entity or operator. Services offered were relatively stable, characterised by long holding times, coupled to slow traffic dynamics, and service provisioning delays of the order of days/ weeks were acceptable. Such deployments models were, arguably, best addressed with a centralised control paradigm.

While the need for a control plane does not seem to present significant opposition, the choice of the technology is still debatable. From a historical perspective, the evolution of the control plane for optical networks started augmenting NMS based networks with a distributed control plane, based on the ASON (Automatically Switched Optical Networks) [49] architecture with Generalized Multi-Protocol Label Switching GMPLS suite of protocols, as detailed next. Recently, the application of Software Defined Networking (SDN) principles to the control of optical networks are presented as a means to enable the programmability of the underlying network (in any case, the formal separation of the data and control planes is a key concept in optical network control). To some extent, there is an analogy between a transport network SDN control architecture and a legacy centralised NMS (umbrella system for transport specific EMSes), although the former insists on using modern system architectures, open and standard interfaces, and flexible and modular software development.

2.5.5. Distributed Control

In a distributed control plane model each network node has the necessary logic (a control plane entity) to communicate with other network nodes (with logic components). These logic components combine resource discovery, reachability, signaling and often connection, or link management, functions.

Each distributed node is responsible for the dissemination of resources under its control (e.g., its links) so the network view is built cooperatively. Once a connection between needs is required, a service setup is requested. The ingress node is typically responsible for the path computation function based on the topology obtained and for triggering the signaling process by which resources are reserved for the connection setup. Note there is no central authority that coordinates the network operation in a distributed control plane environment.

In this setting, the control plane is implemented by a set of cooperating entities (control plane controllers) that execute processes that communicate. Control plane functions such as topology management, path computation or signaling are distributed (for the first one, each node disseminates the topological elements that are directly under its control, and the IGP routing protocol enables the construction of a unified view of the network topology. Path computation is carried out by the ingress node of the connection and signaling is distributed along the nodes involved in the path). The protocols ensure the coordination and synchronisation functions, autonomously (although commonly, the provisioning of a new service is done upon request from an NMS).

The reference architecture is defined by the ITU-T, named ASON enabling dynamic control of an optical network, automating the resource and connection management. ASON relies on the GMPLS set of protocols defined by the IETF (with minor variations). In short, the ASON/GMPLS architecture defines the transport, control and management planes. In particular, the control plane is responsible for the actual resource and connection control and consists of Optical Connection Controllers (OCC), interconnected via Network to Network Interfaces (NNIs) for network topology and resource discovery, routing, signaling, and connection setup and release (with recovery). The Management

Plane is responsible for managing and configuring the control plane and fault management, performance management, accounting and security.

Within ASON the main involved processes are the Connection Controller (CC) and the Routing Controller (RC), and optionally a path computation component. A data communication network, based on IP control channels (IPCC) to allow the exchange of control messages between GMPLS controllers, is also required, which can be deployed in-band or out-of-band (including, for example, a dedicated and separated physical network). A GMPLS-enabled node (both control and hardware) is named Label Switched Router (LSR). Each GMPLS controller manages the state of all the connections (i.e., Label Switched Path - LSPs) originated, terminated or passing-through a node, stored in the LSP Database (LSPDB), and maintains its network state information (topology and resources), collected in a local Traffic Engineering Database (TED) repository.

The network elements participating in distributed control plane environment exchange the accumulated advertisements from other nodes in a state database (e.g. OSPF database) and run a Dijkstra (shortest path) algorithm to establish a reachability graph of best paths to destinations. This process uses a distributed flooding algorithm within the IGP protocol procedure to propagate attachment information, thus, all nodes speaking a particular IGP protocol in the domain remain connected to each other (directly or indirectly) and participate with timely reachability information and establish a network topology, that reports change in connectivity in the event of failure. A key aspect is thus convergence, which is the time it takes from when a network element introduces a change in reachability of a destination due to a network change, such as a failure. A variety of methods exist in various IGP mechanisms and procedures to address scaling of the control plane state (memory and CPU) in the network, both for physical and logical design.

2.5.6. Centralised Control

In a centralised control plane, a controller interacts with the nodes directly, the logic (and topology) remains in the controller, addressing the complexity and cost of distributed control planes. While this architecture simplifies the implementation of the control logic, it has scalability limitations as the size and dynamics of the network increase.

A central control architecture is conceptually simpler, a single point of deployment of policies and business logic, easier to deploy, and requires less state synchronisation. It may also present a bottleneck or single point of failure, with latent fault-tolerance issues.

Network functions requiring local knowledge (dynamic restoration, fast rerouting) are harder to achieve in a centralised model, where a distributed model is potentially faster (capable of responding to local knowledge) and more robust and mature, although implementations usually need to conform to a wider set of protocols.

In an SDN centralised controlled network, a single entity, usually called controller, is responsible for the control plane functions, commonly using open and standard protocols, such as those defined by the SDN architectures and protocols, e.g. OpenFlow protocol (OF/OFP). The controller performs path computation and service provisioning and proceeds to configure the forwarding and switching behaviour of the nodes. A centralised control plane provides a method for programmatic control of network resources and simplification of control plane process. Deployment and operation of connections require interaction with control points to establish the forwarding rules for specific traffic. These are not recent innovations, separation of the control and data planes occurred with the development of ForCES [50, 51] and Generalized Switch Management Protocol (GSMP) [52] many years ago.

By deploying the control plane intelligence in the controller, resources allocated in hardware nodes for control plane functions are reduced significantly. Moreover, such solutions involve deploying hardware (computational and storage) in a centralised location which is orders of magnitude more powerful than individual controllers are. Although a centralised controller does not seem significantly different from an NMS, it is worth noting aspects such as the automation of processes, and programmability, as well as the use of open interfaces and standard architectures, terminology, models, and protocols. Note that a logically centralised controller may, itself, be implemented as a distributed system, while appearing, programmatically, as a single entity.

Finally, a key conclusion is that any transport network controller must be forwarding technology agnostic, capable of computing and programming a wide variety of existing and future transport technologies.

2.5.7. Selecting a Distributed or Centralised Control Plane?

In a distributed control approach, individual nodes participate together to distribute reachability information in order to develop a localised view of a consistent, loop-free network. Routes and reachability information is exchanged that later results in data plane paths being programmed to realise those paths, however, paths are often sub-optimal and prone to traffic congestion, so clearly this approach has weaknesses which might be addressed using a centralised approach. Mainly, a distributed control plane is affected by the latencies in the propagation and synchronisation of data. Changes occurring at a given network element need to be propagated, and the transitory may affect network performance.

On the other hand, in a distributed model, each node element is mainly self-sustained. There is no bottleneck or single point of failure, such as an SDN controller, and is the model that seems most appropriate when there is no central authority, and functional elements need to cooperate. Each node can survive failures at other nodes as long as the network remains connected.

The benefits of a centralised model are lower capital and operational cost, involving, in the case of a control plane, minimal control plane hardware and software at each node, while enabling computational scaling at the controller location. A centralised controller may be easier to implement, given the tight coupling of components, and the less stringent requirements of internal interfaces not subject to interoperability issues. It simplifies automation and management, enables network programmability, and it is less subject to latencies and out-of-date information due to the need of synchronising entities. It provides more flexibility, a single point of extension for operators' policies and customisations, and improved security. There is less control plane overhead, and arguably, network security is increased, with less complexity and greater control over potential risk areas. The downside is that centralised elements are always points of failure.

The control plane (definition of routing and traffic engineering policy) remains a significant operational task in Transport SDN, and control of resources via centralised platform would provide a global network view and efficient use of resources. However, any changes to physical optical network parameters would need to be reflected the central controller quickly, or it may suffer from scalability problems and compute paths on outdated information.

Distributed control planes adapt quickly to changing conditions so provide high survivability, fast recovery and would maintain accurate state accuracy, utilising protocol methods that advertise state changes (such as link or node failures), and advertising reachability to specific networks. However, there is a need to have better configuration management, a clear separation of configuration and operational data for the network slicing objectives outline earlier in the document while enabling high-level constructs more adapted to future transport services and supporting network-wide transactions such as concurrent global optimisations.

The question of which is best, distributed, or centralised, when designing control planes is no longer clear cut. The design considerations should now include how we might blend control plane architectures and principles for optimal transport network operation and utilise the best of distributed and centralised control plane methodologies. Clearly, there are benefits of using a distributed control plane for resource discovery and recovering from local failures, and then global network resource optimisations might then be performed by a centralised control plane residing in a transport controller. The centralised controller may also manage end-to-end connection setup, especially when services traverse multi-domain and multi-technology environments.

2.5.8. Hybrid Control plane models

Given the current trends and evolutions of control plane architectures, it seems too simplistic to tag a control plane as distributed or centralised. Control plane architectures are evolving towards hybrid control- plane models, in which some elements may be centralised and some elements may be distributed, sometimes following the mantra "distribute when you can, centralise when you must". Even if a given control plane entity is centralised, it can be *logically* centralised, where a system is implemented regarding the composition of functional components that appear as one. A given function can be centralised in a given domain (e.g. the path computation function can be centralised in a Path Computation Element (PCE) assuming a single PCE per domain deployment model, but the same function can be distributed amongst several children PCE in Hierarchical PCE (H-PCE) architecture [53] within a multi-domain scenario.

New use cases, such as remote data centre interconnection, highlight the need for multi-domain service provisioning and heterogeneous CP interworking, potentially requiring an overarching control (see figure 16 – "Overarching Control of Heterogeneous Technologies").

Additionally, network operators aim at addressing the joint control and allocation of network and IT resources (e.g. networking, computing, and storage resources), or the joint optimisation of different network segments, such as access, aggregation, and core. The adaptation of one control model to the other or more advanced interworking requiring the definition of common models (e.g. a subset of attributes for network elements) and coordination and orchestration functions. Such orchestrator may, in turn, be (logically or physically) centralised while delegating specific functions, to subsystems that may be distributed (such as the provisioning of connectivity delegated to a GMPLS control plane).

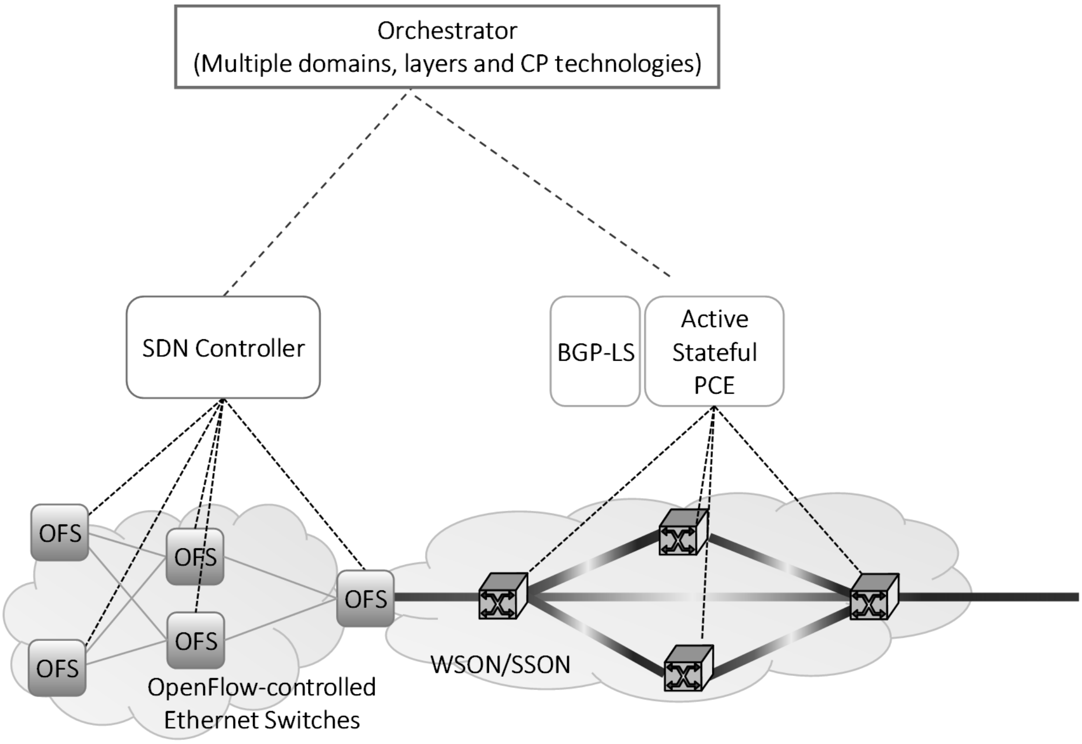


Figure 16 Overarching Control of Heterogeneous Technologies

We should mention that the adoption of new computing and interworking models, and concepts, such as those of server consolidation, host virtualisation or Network Function Virtualisation (NFV), are challenging common approaches and existing practice: for example, a GMPLS control plane could be run as a Virtual Network Function running in a data centre, for legacy purposes, in which a distributed system could run on a centralised physical infrastructure.

The control plane (definition of routing and traffic engineering policy) remains a significant operational task in Transport SDN, and control of resources via centralised platform would provide a global network view and efficient use of resources. However, any changes to physical optical network parameters would need to be reflected the central controller quickly, or it may suffer from scalability problems and compute paths on outdated information.

Distributed control planes adapt quickly to changing conditions so provide high survivability, fast recovery and would maintain accurate state accuracy. However, there is a need to have better configuration management, a clear separation of configuration and operational data for the network slicing objectives outline earlier in the document while enabling high-level constructs more adapted to 5G services and supporting network-wide transactions such as global concurrent optimisations.

Therefore, as discussed earlier it is not a question of which is best, distributed, or centralised control? The question is how we might blend control plane architectures and principles for optimal transport utilising features that could be implemented on a central component per domain, or globally on a super-controller or parent controller, as well as a capability that is delegated locally.

3. Transport Network Control Framework Design

In this chapter, we outline the requirements gathering, documentation, design and development of a next generation transport network control framework. This framework was developed by conducting extensive and detailed interviews within British Telecom and with other key operator architects and technology decision makers. These interviews highlighted the key requirements and objectives faces by some of the largest network operators in the world.

The requirements gathering was conducted over a one-year period, from 2012 to mid-2013. There are extensive records available as surveys were conducted as interviews and encoded in NVivo for analysis.

3.1 Requirements Gathering

The following table outlines the series of interviews conducting during this initial PhD research for requirements gathering and network strategy.

Table 2 Schedule of Research Interviews Related to NFV & SDN Architecture Development

Interview No.	**Format**	**Interviewee: Position and Company**
1	Individual	Chief Network Services Architect, British Telecom
2	Individual	Chief Data Networks Strategist, British Telecom
3	Individual	Head of Network Evolution Innovation, British Telecom
4	Panel	Head of Core Optics Research, British Telecom
4	Panel	Core Optics Research, British Telecom
4	Panel	Core Optics Research, British Telecom
5	Individual	Senior Research Officer, ETSI
6	Individual	Head of Technology Exploration, Telefonica
7	Individual	Senior Expert Standardization, Deutsche Telekom
8	Individual	Director of Network Architecture, Verizon
9	Individual	Principal Member of the Technical Staff, Verizon
10	Individual	Principal Member of the Technical Staff, Verizon
11	Panel	Technical Manager, NTT Labs
11	Panel	Senior Network Engineer, NTT Labs
11	Panel	Senior Network Engineer, NTT Labs
11	Panel	IP Engineer, NTT Labs
12	Individual	Technical Manager, KDDI
13	Panel	Technical Manager, NTT docomo
13	Panel	Engineer, NTT DoCoMo
13	Panel	Engineer, NTT DoCoMo
14	Individual	Network Architect at Colt Technology Services, Colt
15	Individual	Lead Member of Technical Staff, AT&T
16	3rd Party	Network Architect & Research Scientist, Orange
17	3rd Party	Distinguished Network Architect, AT&T
18	Individual	Technology Specialist, Telefonica
19	Individual	Technology Specialist, Telefonica
20	Individual	Member of Technical Staff, AT&T

3.1.1 Technical Drivers for Transport Network Innovation

The interviews conducted highlighted that operators must balance their desire to innovate and create value through new services or cost savings, with an understanding of the available methods and technologies, and the limitations imposed on them. Operators face a second paradox: they must innovate to create improved network flexibility and performance because consumers and applications demand it, but they must not innovate to the extent that they risk overall network control and stability.

3.1.1.1 Reducing the use of Propriety Hardware Platforms

With these twin challenges of increasing capacity demands and regulatory pressure, the need for operator-driven innovation is focussed on finding more cost-efficient ways of moving high volumes of data, and the need to address the current dependence on expensive, dedicated hardware and processors. A leading organisation in this search for solutions based on cheaper, generic hardware was British Telecom, working with Intel initially but then a growing group of other operators from around the world.

> *"I had various discussions with colleagues going back over many years about the potential for generic processors to shift packets and got into various discussions as to what sort of packets; you know packet performance was the main parameter of interest. We then got into a more detailed discussion with Intel about 2½ or 3 years ago and initiated a study for them which they grew into a wider set of partners."*
> **Chief Data Networks Strategist, British Telecom**

The development of these exploratory collaborations between operators and a chip manufacturer was a significant precursor to the current move towards NFV and SDN. In these early years, the main motivation was to use innovative methods for cheaper, and more generic hardware running the latest Intel chips as an alternative to the costlier dedicated network hardware, running proprietary chips and proprietary software. These current provisions were costly in part because the vendors could lock-in operators due to the lack of interoperability of their hardware, and the onus on learning and using proprietary software solutions from a specific vendor.

> *"At the end, all of us agreed that at first, it is about reducing, well, the direct hour costs, if you are buying normal standard servers it is much cheaper than buying expensive dedicated boxes… because one of the things that organisations like mine hate are what we; you are always talking about vendor lock-in, you do not want to be caught by a single vendor."*
> **Head of Technology Exploration, Telefonica**

This lock-in effect is a legacy of the layering that evolved since privatisations took place and the Layer 1 vendors took an increasingly important role in R&D. The rapid improvements in generic processors and their proven, cost-effective use in large data centres is a compelling alternative, assuming that the required performance is acceptable.

> *"thanks to Moore's Law with respect to processor speed, and power and storage costs coming down, being able to take advantage of that, which you can do much more in a data centre environment."*
> **Principal Member of the Technical Staff, Verizon**

The reasoning is that if telecommunications networks can begin to look more like data centres, with centralised commodity hardware managing the networks in place of distributed, specialist hardware, the costs of operating such networks will tumble.

More recently focus appears to be on developing new commodity hardware, and rapid reduction requiring proprietary hardware, for legacy equipment (that was highly specialised) equipment might often be junked after a certain time period, rather than being reused:

> "[it's] as much about decommissioning as commissioning savings. We [currently] simply leave equipment at customer sites, it's cheaper than collecting and disposing of"
> **Chief Network Services Architect, British Telecom**

NFV-based functions are delivered in software form to data centres, so there is no longer a need to physically move an engineer and a piece of equipment to each location to install network services or to remove or repurpose them. Under these new conditions, the full-life cost of hardware drops significantly.

3.1.1.2 Flexibility of Virtual Network Functions

In addition to hardware cost considerations, there are long-term service implications that the new NFV approach will allow. As well as shifting the primary technological core of network infrastructures to data centres there would be a shift towards the use of software-based virtual network functions, in place of hardware reliant functions.

> "It will bring flexibility, agility and automation and a much faster time-to-market cycle, where the latter is something that we, as operators, lack today."
> **Network Architect & Research Scientist, Orange**

> "Since it is software only, the composition or decomposition of functions allows us to be more flexible in responding to the marketplace."
> **Distinguished Network Architect, AT&T**

If physical infrastructure no longer needs to be installed at or near a customer's premises when new telecoms functionality is required but can instead be remotely installed into servers located at a data centre, the benefits to both operator and customer will be significant.

The importance of deployment speed is emphasised by BT, who use this as an important internal driver for change by providing a clear indication of just how much faster and more responsive they want to be to customer needs, through NFV:

> "One of the taglines we've used was 'from 90 days to 90 seconds' that our lead time to deploy a box to wherever in the world the customer premises happens to be"
> **Chief Data Networks Strategist, British Telecom**

In addition to this aspect of flexibility, they also see real benefits to both operators and customers of being able to delay purchasing decisions.

> "There's a real option which is being able to defer a decision on what you deployed because the hardware is exactly as you say, generic, so you've not committed to the particular functionality at the time you deployed the hardware."
> **Chief Data Networks Strategist, British Telecom**

Customers and operators will have the ability to select and install software-based functions at the time they are needed, without having to try and predict what might be needed ahead of time. In addition, functionality can be scaled up, scaled down or repurposed as evolving demands deprived of the need to redeploy engineers or incur both the financial and the ecological cost of hardware decommissioning.

The long-term flexibility goals stated within the NFV White Paper (2012) include a desire to create a true software market for telecommunications functions, where smaller firms can compete with the very large and well-established Layer 1 players on a software-only basis. For operators, separation of hardware and software eliminates the de-facto lock-in associated with proprietary hardware, and at the same time creates a potentially much larger, more international, and more competitive service-based marketplace for functions software.

When considered together these drivers for developing NFV are compelling: To be able to save costs, improve service speed and flexibility, create a new market that provides greater innovation and opens up competition amongst suppliers to the operators, whilst enforcing interoperability of their different products; in sum, this looks like a kind of strategic nirvana. However, certain challenges must be overcome regarding the development of the technology itself, as well as the management of the multi-organizational collaboration that is required to achieve this new industrial vision. These organisational challenges are discussed in the following sections.

3.2 Motivation and Aims

Previously transport networks were typically static, lacked flexibility, and required long planning times when deploying new services. Operators have embraced technologies that allow separation of data plane and control plane, distributed signaling for path setup and protection, and centralised path computation for service planning and traffic engineering.

Although these technologies provide significant benefits, they do not meet the growing need for network programmability, automation, resource sharing, and service elasticity necessary for meeting operator's requirement for their virtual network operators.

Virtual network operation may be categorised as the creation of a virtualised environment allowing operators to see a simplified view (via abstraction) of the underlying multi-admin/multi-vendor/multi-technology network. It would also allow the operator to control and manage these multiple networks as if a single virtualised network. Another dimension of virtual network operation is associated with the use of the common core transport network resource by multi-tenant service networks as a way of providing a virtualised infrastructure, thus enabling a flexible method to offer new services and applications.

3.2.1 Development of the NGN Controller Framework

The research documented in this thesis is the culmination of five years of research by the thesis author and led the development of Application-Based Network Operations (ABNO) framework by the researcher. ABNO was firmly grounded in requirements identified by the thesis author and derived from leading operators who wanted to leverage the emerging field of Software Defined Networks (SDN) and Network Functions Virtualisation (NFV).

3.2.2 New Generation of Transport Services

A notable recent research project called IDEALIST (Industry-Driven Elastic and Adaptive Lambda Infrastructure for Service and Transport Networks) [67] and [69] have developed a control plane to meet the evolving requirements for managing elastic optical infrastructure. Each supported a set of basic functions, including i) element addressing; ii) dynamic resource discovery (e.g. local interfaces and device ports and capabilities); iii) automatic topology and reachability discovery and management (by which a control plane may discover the topology without explicit pre-configuration), iv) path computation and v) actual service provisioning with recovery (protection and restoration) ensuring efficient resource usage

3.2.3 Virtualisation Transport Networks

Users demand new services that flexible and time-based (Pay As You Go billing models). These services are provided to customers from the operators and service providers , and facilitate a variety of applications. They offer operators new revenue generation opportunities, and these services are Cloud-based and have different traffic characteristics from established services. Deploying and operating these emerging applications using traditional network technologies and architectures is not feasible, and has significant network performance, resource, scalability, and elasticity (i.e., capable of adapting to customer and application demands) limits.

Network virtualisation is clearly an important innovation towards providing the demands from customers and enabling next generation applications and services. New requirements, methods and capabilities for the deployment and operation of next generation transport infrastructure resources, may be summarised as:

- Coordination and abstraction of underlying transport network resources to higher-layer applications and customers (note that higher-layer applications and customers could be internal users of the core transport network resource such as various service networks);
- Multi-domain virtual network operation that facilitates multi-admin, multi-vendor, multi-technology networks as a single virtualised network;
- Multi-tenant virtual network operation that consolidates different network services and applications to allow slicing of network resources to meet specific service, application and customer requirements;
- Provision of a computation scheme and virtual control capability, via a data model, to customers who request virtual network services (note that these customers could be service providers themselves.

3.3 Framework Component Considerations

We already identified today's networks are heterogeneous, i.e., integrate multiple technologies allowing network infrastructure to deliver a variety of applications, services and bandwidth to support the different characteristics and dynamic demands of applications.

Increasingly, a need to make the transport network more responsive to service requests issued directly from the application layer and high-layer client interfaces. It should be considered that this differs from the established archetypal network, where services in the network are instantiated in reply to business platforms, CLI commands driven by a human engineer, using a plethora of Operational Support Systems (OSS) components (NMS, EMS, et al.), due to the inflexible nature of traditional networks they are also typically over-provisioned thereby ensuring minimal traffic loss, even during network failure and at peak traffic periods.

An idealised network resource controller would be based on an architecture that combines several technology components, mechanisms and procedures. These include:

- Policy control of entities and applications for managing requests for network resource information and connections;
- Retrieve information on available network resources;
- Consideration of multi-layer resources and how topologies map to underlying network resources;

- Handling of path computation requests and responses;
- Provisioning and reserving network resources;
- Verification of connection and resource setup.

Based on the requirements discussed we must develop a control and management architecture of transport networks to allow network operators to manage their networks using the core principles of Software Defined Networks to allow high-layer applications and clients to request, reconfigure and optimise the network resources in near real time, and in response to fluid traffic changes and network failures.

3.3.1 Network Abstraction

A major purpose of Software Defined Networks (SDN) is to bury complexity and make service deployment and overall network operation simpler without invoking the management and provisioning software of the many manufacturers deployed in the network. Consequently, allowing higher-layer applications to automate requests and creation of services simpler and more direct.

A control framework for next generation transport networks will need several technology components, mechanisms, and procedures to enable abstraction of underlying resources.

At a minimum, the following requirements must be met to provide network resource abstraction:

- Generation of a network graph, using links and nodes
- Computation engine for optimisation of the network graph
- Definition of objective functions, with the ability to apply link and node constraints
- Service definitions, including flow or connection types, for end-to-end connection setup and management

3.3.2 Logically Centralised Control

We use the term “logical centralised” to signify that network control may appear focused in a single entity, independent of its possible implementation in distributed form. The centralised control principle states that resources can be used more efficiently when viewed from a global perspective.

A centralised SDN controller would be able to orchestrate resources that span some subordinate domains or in cooperation with other elements, and maximising resource efficiency when creating new services and overall operation of existing services and network resources. Other reasons for logically centralised control include scale, optimisation of information exchange and minimisation of propagation delay.

Given constraints of not being able to deploy greenfield networks, in some situations, it is necessary that a controller co-exists with both native SDN forwarding technologies (OpenFlow) non-native SDN traffic engineered technology (MPLS and GMPLS).

3.4 Application Driven Use-Cases

Dynamic application-driven demands and the services they create specific requirements on the management of transport network infrastructure, these new requirements include:

- a need for on-demand and application-specific assignment of network connectivity, which is reliabile
- optimise resources (such as bandwidth) constraints in a variety of network application topologies (such as point-to-point connectivity
- provide network virtualisation, also known as network slicing

- supporting a range of traffic engineered transport technologies including packet (IP/MPLS) and optical transport networks, to Software Defined Networks (SDN) forwarding technologies,

Additionally, to the general requirements above, a set of application-driven use cases must also be considered:

- **Virtual Private Network (VPN) Planning** –Support and deployment of new VPN customers and resizing of existing customer connections across packet and optical networks;
- **Optimization of Traffic Flows** – Applications with the capability to request and create overlay networks for communication connectivity between file sharing servers, data caching or mirroring, media streaming, or real-time communications;
- **Interconnection of Content Delivery Networks (CDN) and Data Centers (DC)** – Establishment and resizing of connections across core networks and distribution networks;
- **Automated Network Coordination** – Automate resource provisioning, facilitate grooming and grooming, bandwidth scheduling, and concurrent resource optimisation;
- **Centralised Control** – Remote network components allowing coordinated programming of network resources through such techniques as Forwarding and Control Element Separation (ForCES) OpenFlow (OF);

3.4.1 BT Media and Broadcast

At British Telecom a specific network "BT Media and Broadcast", needs significant change on the design, deployment and operation of broadcast and contribution video services is conducted.

The number of media consumption devices and consumers continues to increase exponentially, whether to watch live television or on-demand content, the pressure on the broadcast network operator to deliver fast, secure, and reliable connective capacity across the contribution and distribution infrastructure increases.

Although the contribution and distribution network share common technology requirements, distinct objectives must still be defined. Contribution networks need to support seamless, resilient uncompressed and real-time transmission of multi-format production content. Distribution networks must also scale, but to support a wide variety of low bit-rate streams, as consumer electronics manufacturers push 4K Smart TVs into the home, and sell High Dynamic Range-equipped TVs, creating consumer demand for Ultra High Definition (UHD) content to view on Internet-connected TVs.

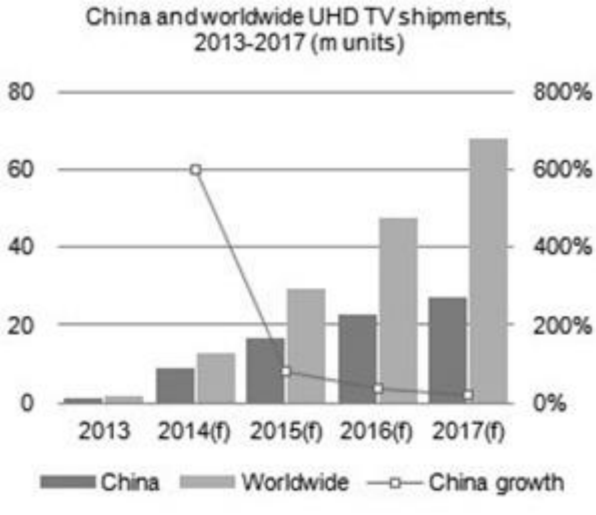


Figure 17 UHD Shipments from DIGITIMES Research 2014

This section provides insight into British Telecom's Media and Broadcast organisation, and specifically the contribution, and distribution test laboratory efforts. The section outlines how the technology and economics of "Software Defined Networking" and "Network Function Virtualisation" discussions, both buzzwords of broadcast shows and conferences, are already impacting the way BT consider requirements, design and deploy network infrastructure.

BT has multiple use cases and depending on the type and scale of Media and Broadcast customer application; we will see that each have a specific set of requirements and capabilities for the transport network, depending on the type of media transport and delivery required. We may summarise core requirements across most use cases:

- Aggregation of multiple flows and formats across studio infrastructure;
- Broadcast industry native interface support;
- High-bandwidth connections for Content Distribution Network (CDN) video.

Each broadcast or contribution flows have their formats, underpinned by the use of Serial Digital Interfaces (SDI). There is a Standard Definition (SD), High Definition (HD), and Ultra High Definition (UltraHD, which is also known as 4K), 8K is sometimes als0 required for transport. These formats are based on well-defined protocols based on published standards. HD-SDI can be multiple format streams, i.e., 1080i, 1080p, 720p, or 480p. A format type specifies the encoding, and vertical and horizontal resolution, quality, speed and aspect ratio, pixel aspect ratio, scanning and frame rate of the content.

Moreover, there is increasing use of 4K as UltraHD, and 8K UltraHD which translates into a considerable increase in bandwidth consumption, and often these services are temporary, so they have to be placed efficiently and created and torn down automatically so not to waste transport network resources. This trend will only continue with further bandwidth demands based on growth in frame rates, colour depth, and number and quality of sound channels, only compounds the need to provide scalable high-capacity bit-rate services. Additional video application requirements and future (expected demands) are outlined in the following sub-sections.

3.4.1.1 Content Capture and Encoding

In some situations, SDI must be encoded to a broad spectrum of formats for live or production content. One critical consideration for selecting the media format is its intended use or delivery platform, and the path and bandwidth required. Upon captured it may be encoded, and then forwarded across the network to its desired destination (production studio, content server, or even live broadcast), and often require some path engineering. Network functions including a production switcher, or directly to a production server are also often required. Typically, a Media Manager handles this decision. It is worth noting that in some cases, the greater the resolution of content it may have multiple outputs at the camera for specific uses and will need to be encoded multiple times and recompiled and synchronised at the router, production switcher, and encoder.

3.4.1.2 Content Transport

In addition to encoding, media will be ingested directly from other sources as files or flows and as mentioned may require encoding to traverse IP infrastructure, often from a Serial Data Transport Interface (SDTI) source.

The SDTI source is a method for transmitting data packets over a Serial Digital Interface data stream. It has been developed to provide a variety of compressed video standards, including DV, DVCPRO, and

MPEG2. There are several well-defined additional standards and protocols, which allow video media to be encapsulated and transported across network infrastructure, including:

- SMPTE – SD-SDI SMPTE 259M;
- HD-SDI SMPTE 292M;
- ETSI – ASI- TR 101 891;
- MPEG2 – ISO/IEC 13818;
- MPEGTS – ISO/IEC 13818-1;
- MPEG4 – ISO/IEC 14496;
- MPEG4 H.264 – ISO/IEC 14496-10.
- JPEG2000 – ISO/IEC 15444-12

3.4.1.3 Bandwidth, Compute & Storage

Studio environments typical contain nodes with HD-SDI interfaces and 10Gb/s network cards. Allowing to receive, transmit, encode, and decode services, with centralised management.

Both multicast and unicast may be used to distribute UHD (4K) compressed video at 2160p 50fps, using H.264 encoding this would require between 800Mb/s to 1.2Gb/s per service. Computing point-to-point and multipoint-to-multipoint trees are not-trivial.

Demands by content consumers for increased video resolution, frame rate, colour depth & sound channels, all add to bandwidth consumption for services. As indicated by the British Broadcasting Corporation (BBC), contribution network uses are requesting a move to near lossless or uncompressed video streams, these equate to:

- HD 1080p 8bit 4:2:2 59.94fps uncompressed bit rate @ 3Gb/s;
- 4K UHD 2160p 12bit 4:2:2 59.94fps uncompressed bit rate @ 10Gb/s;
- 8K SHV 4320p 12bit 4:2:2 59.94fps uncompressed bit rate @ 48Gb/s.

3.4.1.4 Studio Media IP Evolution

Our objective is to facilitate IP Studio media production. This would require a mass migration from dedicated synchronous interfaces to generic IP networks. The rationale for migration to an all IP network, running over a high-capacity commodity-based optical infrastructure with an automated control platform, is extremely compelling:

- Leverage the flexibility and operational experience of traffic engineered networks;
- Support varying types of video, audio and data from a variety of sources and formats over the network with low latency, and minimal jitter;
- Efficiently utilise network resources, resource sharing where applicable;
- Elastic control of the network, setting up and tearing down occasional-use services, links for optimal cost-effectiveness.

If the studio production is live or recorded, it will have different requirements and may need near-real-time setup. Typically, scheduling, content encoding, format decisions and network path decisions have already been made.

During production workflow, media files may need to be accessible to various production applications and processes and possibly need to move between storage locations. Normally the applications (hardware or software) for production workflow are dedicated and fixed and may only be used part-time. If functions were entirely software based and could be efficiently deployed in a “just in time” manner and scaled accordingly, it would provide significant cost savings and flexibility. However, different layers of automation to manage these applications and processes, with the capability to handle the file movement would also be required.

3.4.1.5 Linear Contribution and Content Transport

Our initial use cases for the lab were based on a linear contribution service (pre-consumer), some requirements for broadcast media networks. These type of content services tend to have the following transport requirements:

- End-to-end Automation: the request, computation, setup, a teardown of the end-to-end service;
- Initial support for 4K contributions, but capable of scaling up to 8k and 16k;
- Integrate encoding functions, scale-out storage, durability, adaptive performance, self-healing capabilities;
- Supports high frame rates and other developing formats that exceed client expectations and requirements.

The media flows are expected to be IP-based and support both live, linear TV programs and transport of media content files for production.

Whereas current broadcast video IP links are based on permanent data connections via Ethernet, with variable data rates up to 200Mb/s compressed, or 3Gb/s uncompressed. We designed our infrastructure to support anything from a few 100Mb/s to 10Gb/s, based on a control architecture capable of evolving beyond 100Gb/s.

3.4.1.6 British Telecom Media and Broadcast Laboratory

BT Has built a research laboratory to explore the potential impact of SDN & NFV on networks required to carry high bandwidth broadcast video traffic. The layout is depicted in the figure below which shows our intentions to research the various aspects of building end-to-end video contribution networks. Video creation at HD and UHD rates produces multi-Gb/s SDI formats that require network signal compression and conversion into traffic engineered connections.

For BT Media and Broadcast traditional NMS platforms lack the flexibility, they needed large network engineering and planning teams. Looking towards the architecture and principles defined by the Software Defined Networking (SDN) architecture developed and ratified by the Open Networking Foundation (ONF) creates a new value proposition. The core SDN architectural principles offer a variety of options when looking to plan, control, and manage flexible network resources both centrally and dynamically, that is simply not available to BT currently.

The advent of Network Functions Virtualisation (NFV) has also provided the ability to deploy network functions (media encoding, storage, load balancing) for BT Media and Broadcast on virtualised infrastructure hosted on commodity hardware, decoupling dedicated network function from proprietary hardware infrastructure. Consequently, this allows network function to be instantiated from a common resource pool and to exploit performance predictability where dimensioning remains stable whatever the use of virtualised hardware resources. Emboldened with the suitable control and orchestration tools, these virtual and on-demand capabilities could have a significant impact on how telecom infrastructure is managed.

A commodity-based optical platform comprises a combination of optical switches, amplifiers and fibre. The switches here are Reconfigurable Optical Add-Drop Multiplexers (ROADM) which have at their heart Wavelength Selective Switch (WSS) technology. Using a central controller would provide the capability to compute and route wavelength channels from any input to any output fibre, on demand and without the current weeks of network planning by human engineers.

In a grand design for BT Media and Broadcast, there would be the capability to manage multiple controllers, as BT operates multiple transport domains. These domain-specific controllers provide inputs to an orchestrator who has now a centralised view of all the network resources. Applications can take advantage of this SDN-based network orchestration, and we have demonstrated a Scheduler application that can request on-demand large bandwidth pipes set up at specific times and durations.

The figure below presents our initial view of this idealised architecture and a candidate architecture to meet the idealised view is provided later in this document.

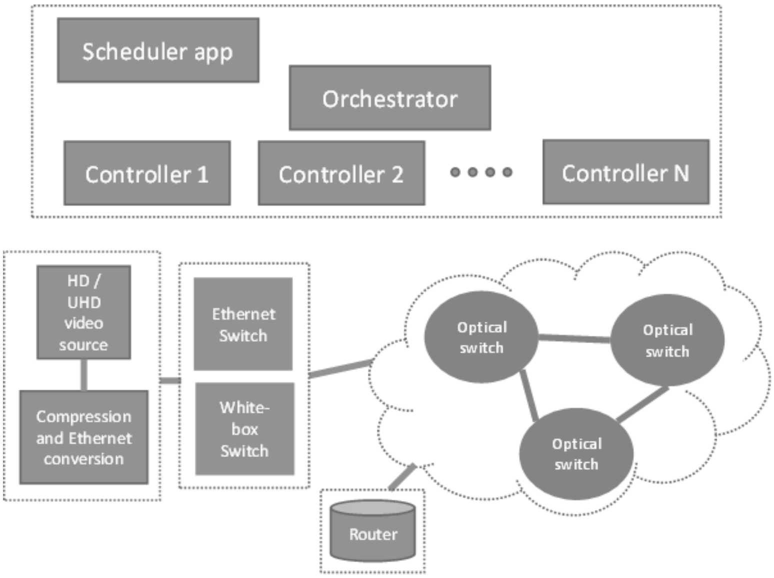


Figure 18 British Telecom Media & Broadcast Idealised View

4. Framework for Application-Based Network Operations (ABNO)

This chapter outlines the core network control principles required for application-based network operations of transport networks, discusses key control plane principles and architectures. It introduces the Application-Based Network Operations (ABNO) Framework [54], and how this framework and functional components and how they are combined for Adaptive Network Manager (ANM), used to address the requirements for operating next generation transport networks.

The three tenants of SDN are programmability, the separation of the control and data planes, and the management of ephemeral network state in a centralised control model.

Application-Based Network Operations (ABNO) was designed using set architectural principles gathered during the requirements discussion with operators for transport network evolution, and British Telecom research discussions, specifically for the Media and Broadcast transport network:

1. **Loose Coupling**: For ease of operation and rapid, yet agile, development, and tightly integrate the functional components of the network controller, the use of well-defined APIs and protocol mechanisms must be used.
2. **Low Overhead**: ensure that resource management and network control functions are not duplicated, reducing overall platform overhead.
3. **Modular**: A modular design enables easier composition of existing features into new capabilities.
4. **Intelligent**: Designing the framework around the Path Computation Element and Traffic Engineered principles, leveraging years of existing protocol development for managing heterogeneous technologies and efficient resource utilisation.
5. **Resource Management**: The framework allows for various network and node state to be discovered and stored. This state information is collected using the protocol mechanisms provided by traditional and already existing network and service management tools.
6. **Dynamic Management**: A key goal of an SDN controller is actuate dynamic control based on application demands and other network events.
7. **Policy Control**: implement policy management mechanisms for specifying connection requirements (e.g., QoS, security) based on applications demands and constraints. It also allows operators to meet the varying service levels they provide to customers .
8. **Technology Agnostic**: communicates with a wide range of network nodes using varying forwarding technology, and using a variety of Southbound APIs and protocols.

It should also be possible to utilise both a distributed control plane as well for local policy decisions and leveraging years of protocol development and function, thus providing the best practices of centralised control, and distributed control plane for ephemeral state management.

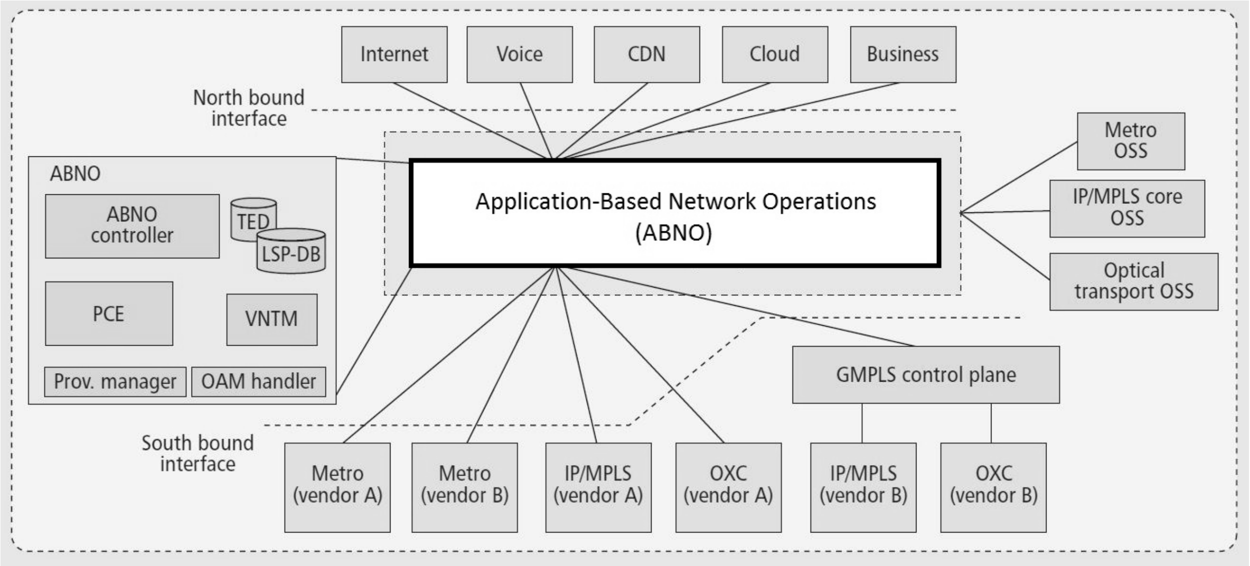


Figure 19 Application-Based Network Operations (ABNO) Framework [54]

Current networks consist of switches and routers using traditionally distributed control planes and data plane technologies. Ensuring network efficiency is limited in such networks as intelligence is distributed across many switches or routers and often involves complex protocols and procedures. In contrast, an SDN network with OpenFlow will use a centralised control plane (or “controller”). This will be the entity receiving application and customer requests directly, and then responsible for establishing the transport paths or flows directly, and data planes at nodes to perform packet matches, forwarding, copying or dropping actions.

The ABNO-based architecture [54] allowed a controller to be data plane technology agnostic, a significant difference compared to SDN Controllers, which are typically OpenFlow based. An ABNO Controller, per domain (administrative or technology), discovers, organises, and layers multiple services across the infrastructure. This programmable control feature facilitated automation techniques to be used to set up end-to-end services. Allowing for far more flexibility beyond the customer requested service, and with the capability to modify paths and network function nodes to be modified (torn down, resized, relocated) at any time particularly in response to changing network conditions of the operational network state. This was a direct solution to the BT Media and Broadcast issue of having to build in significant network capacity and lack of adaptability to fluctuations in the resource location, types or changing availability, and in recovering from partial or catastrophic failure.

The advent of NFV is also used within ABNO to leverage IT virtualisation techniques to migrate entire classes of network functions (the BT example might include media encoding and storage) requiring proprietary hardware onto virtual platforms based on general compute and storage servers, at a far cheaper cost point. These virtual function nodes are often known as a Virtualised Network Function (VNF), and typically executed on a single VM, or collection of Virtual Machines (VMs), and more recently Containers (light-weight Linux machines).

Furthermore, this virtualisation allows multiple isolated VNFs or unused resources to be allocated to other VNF-based applications during weekdays and business hours, facilitating overall IT capacity to be shared by all content delivery components, or even other network function appliances. Industry, via the European Telecommunications Standards Institute (ETSI), has defined a suitable architectural framework and has also documented a number of resiliency requirements and specific objectives for virtualised media infrastructures.

4.1. ABNO Functional Components

The research in this document culminates in the development of the ABNO framework, a standards-based reference framework for flexible control of transport resources. The ABNO framework was published by the Internet Engineering Task Force (IETF) and represents an industry acceptance of an SDN and NFV capable control framework to meet the requirements of future networks and services.

The ABNO architecture builds on the establish SDN principles for on-demand and application-specific provisioning of network resources, supporting a wide range of applications (e.g. point-to-point and point-to-multipoint connectivity in transport networks, capable of providing optimisation of traffic paths. The ABNO approach is disruptive when compared to traditional network provisioning model, where services are created based on management requests and deployed by network planners. Above all, ABNO addresses key requirements gathered during discussions with the worlds largest transport network operators, and the challenges of BT's Media and Broadcast networks. It was designed to integrate multiple technologies and need to provide a wide variety of services in the response of direct requests from the customer and application layer.

A main principle of the ABNO architecture was to leverage several existing technologies for discovering and disseminating information about the resources available in a network, regarding topologies and their mapping to network resources, for requesting path computations and for provisioning/reserving application-aware network services. Therefore, ABNO may be considered as a composition of existing components but enhanced with new elements and interfaces. The PCE is a key element and performs the role of the "brain" in the ABNO architecture. Its usage is extended to provide application-aware path computations and policy enforcement for the set of services supported in ABNO. The deployment of stateful PCE is of particular interest in the context of ABNO, mainly for proactive control and operation of underlying networks. Further PCE developments to fully utilise the ABNO ambitions will be required.

ABNO consists of nine functional blocks, presented in figure 20 ("Key Functional Blocks of the ABNO Architecture") below.

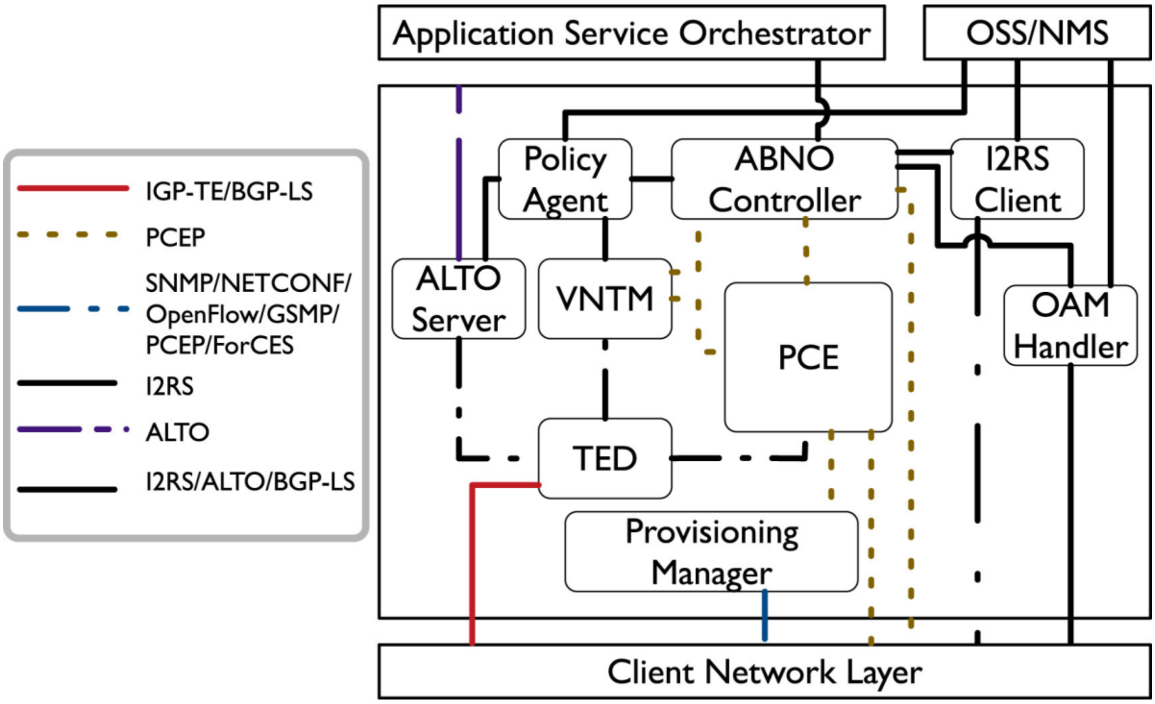


Figure 20 Key Functional Blocks of the ABNO Architecture [55]

The core of the ABNO architecture is the ABNO controller itself. The controller allows applications and NMS/OSS to specify end-to-end path requirements and access path state information. A path request triggers the controller to inspect the current network connectivity and resource allocations, and to provision a path which fulfils the resource requirements and does not violate the network policy. Also, the controller is responsible to re-optimise paths at run-time, taking into consideration other path requests, routing state and network errors. The architecture contains an OAM handler to collect network error from all network layers. The OAM handler monitors the network and collects various performance, alarms, and health notifications from network devices, using OAM protocols like IPFIX [56] and NETCONF, which are correlated to distil high-level error reports for the ABNO controller and the NMS.

It is worth noting that the ABNO architecture integrates with the network routing policy through an Interface to the Routing System (I2RS) client, this allows direct modification of the control plane and applies policy for candidate paths, which could then filter down to the data plane.

Legacy NMS and OSS

A Network Management System (NMS) or an Operations Support System (OSS) can be used to control, operate, and manage a network. Within the ABNO framework, an engineer, NMS or OSS may require a high-level service directly to the ABNO Controller.

The NMS and OSS may also need to be consumers of network events reported through the OAM Handler, especially relevant when ABNO is used in a legacy network. ABNO could also be used to react to OAM reports as well as displaying them to users and raising alarms. It certain situations the NMS and OSS can also access the Traffic Engineering Database (TED) [57] and Label Switched Path Database (LSP-DB), hosted by the ABNO instance, to show the users the current state of the network.

Finally, the NMS and OSS may utilise a direct programmatic or configuration interface to interact with the network nodes within the network, circumventing ABNO entirely. However, any node state change will eventually be discovered by ABNO.

Application Service Coordinator

The Application Service Coordinator communicates with the ABNO Controller to request operations on the network. Requests may be initiated from entities such as the NMS and OSS, application specific interface, and services in the ABNO architecture may be requested by or on behalf of applications themselves.

In the context of this section, the term "application" is a broad one, and defined in RFC7491 and quoted below:

- “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 that a user uses to make requests to the network to set up specific services such as end-to-end connections or scheduled bandwidth reservations.”

Furthermore, an application may be a sophisticated control system that is responsible for arranging the provision of more complex tasks, such as a virtual private network or inter-data centre connectivity. For the sake of ABNO architecture discussion, all of these concepts of an application are grouped and shown as the Application Service Coordinator (ASC). In reality (an implementation), the function of the Application Service Coordinator may be distributed across multiple applications or servers, for scale, speed and resiliency.

ABNO Controller

The ABNO Controller component is the main interface to the network for the NMS, OSS, and Application Service Coordinator. It manages the provisioning request and other advanced network coordination and functions. The ABNO Controller oversees the behaviour of the network in response to changing network conditions and by application network requirements and policies. It instantiates the required components, in a correct sequence, and applies policies where applicable.

Policy Agent

The policy is a very important aspect of the control and management of the transport network. Provisioning high bandwidth connections are costly. It is, therefore, significant in deciding how the key capabilities and components of the ABNO architecture function. The Policy Agent is responsible for propagating those policies into the other components of the system. Simplicity in this discussion necessitates leaving out many of the policy interactions that will take place. In our example, the Policy Agent is only discussed interacting with the ABNO Controller, in reality, it will also interact with some other components and the network elements themselves. For example, the Path Computation Element (PCE) will be a Policy Enforcement Point (PEP) [58], and additionally, the Interface to the Routing System (I2RS) Client (where applicable) will also be a PEP as noted in [59].

OAM Handler

During discussions with operators and BT, it became clear that Operations, Administration, and Maintenance (OAM) [48] plays a pivotal role in the health of the network and overall efficiency. Its required for detecting faults, and taking the necessary action to react to problems in the network. Therefore, these capabilities must be represented within the ABNO architecture. The ABNO OAM Handler is responsible for receiving notifications from the network about potential problems or testing newly setup connections, for correlating alerts and alarms, and for instantiating other required components of the ABNO platform, for resilience and recover connections that were established by the ABNO Controller, based on application requests. The OAM Handler also reports network problems to high-layer OSS and BSS, especially for 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 [4], such as monitoring and testing.

Path Computation Element

As discussed previously the PCE is already a highly capable functional component that services requests to compute paths across a network graph deployed already in key transport networks for managing traffic engineered services. In particular, it can manage a variety of traffic engineered MPLS and GMPLS Label Switched Paths (LSPs), and supports optimisation functions. By leveraging the PCE within ABNO, we inherit key capabilities. The ABNO PCE may receive these requests from the ABNO Controller, from the Virtual Network Topology Manager (VNTM), or from network elements themselves.

As discussed, the PCE operates on a view of the network topology, to be accurate and provide relevant paths it must be updated to reflect actual state, is stored in the Traffic Engineering Database (TED) [57]. A more sophisticated computation may be provided by a Stateful PCE that enhances the TED with a database (the LSP) containing information about the LSPs that are provisioned and operational within the network.

Numerous additional functionality developed by the IETF, including the 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 [25,26]. This function may directly access the network elements or channelled supported via the ABNO Provisioning Manager. This component also provides coordination between multiple PCEs (possible transit domain management entities) each operating on a local TED. This proves very useful for automating (and reducing the time for) performing path computation in multi-domain or multi-layer networks. Reducing or negating entirely, the need for human engineers to traffic engineer a path across multiple transit domains especially if the transit domains are operated by different teams or even organisations.

In the latter case, the ABNO controller will need to request an optimal path for the service. If the domains (ASes) require path setup to preserve confidentiality about their internal topologies and capabilities, they will not share a TED, and subsequently, each domain (AS) will operate its PCE. In such a situation, the Hierarchical PCE (H-PCE) architecture, described in [53], is necessary.

Network Database

The ABNO architecture includes some databases that contain information stored for use by the system. The two main databases are the TED and the LSP Database (LSP-DB), but there may be some other databases used to contain information about topology (ALTO Server), policy (Policy Agent), services (ABNO Controller), etc.

Typically, the IGP (like OSPF-TE or IS-IS-TE) are responsible for generating and disseminating the TED within a domain. Often in multi-domain and multi-layer environments, it may be necessary to export the TED to another control element, such as a PCE, which can perform more complex path computation and optimisation tasks.

Virtual Network Topology Manager

A Virtual Network Topology (VNT) [60] is defined as a group of one or more LSPs in one or more lower-layer (server) networks that provide information for efficient path handling in an upper-layer (client) network. An example might be: using a set of LSPs in a transport wavelength division multiplexed

(WDM) network (server layer), which may provide connectivity as virtual links (client yet) in a higher-layer IP/MPLS packet switched network.

The creation of virtual topology within ABNO for inclusion in a network is not a simple activity and will require further development. Consideration and selection of which nodes in the upper-layer are best to connect, in which lower-layer network to provision LSPs to provide the connectivity, and how to route the LSPs.

Provisioning Manager

The ABNO Provisioning Manager is responsible for making or directing requests for the establishment of connections. Instructions to the control planes running in the network (via signalling methods such as RSVP-TE) or the direct programming of individual network nodes via provisioning protocol, or both methods simultaneously.

South Bound Interfaces

ABNO Should support both management of existing (legacy) nodes, or where the network devices will need to managed (configured) directly from the legacy OSS platforms. Many protocols already exist which are capable of performing programming functions, and these must be supported by ABNO, examples include:

- SNMP [61]
- Network Configuration Protocol (NETCONF) [62]
- REST-based Configuration (RESTCONF) [63]
- ForCES [50]
- OpenFlow Wire Protocol [64]
- PCEP [24]

The role of the protocols described is to assign a state to the forwarding element, either by programming each node individually or via a distributed signalling mechanism. Indeed, the previous list is not an exhaustive representation of protocol methods and procedures available, and over time, new forwarding mechanisms will be developed. Therefore, the ABNO framework has been designed to be forwarding mechanism-agnostic, and able to support future, yet unknown forwarding technologies.

5. ABNO Architecture Implementation and Testing

This chapter highlights an important instantiation of ABNO further developed for control of flexible optical bitrate services. This ABNO-based control platform was called Adaptive Network Manager (ANM) and is described in more detailed in the following sub-sections.

Often, the primary purpose of a functional architecture is to decompose a problem space and separate distinct and discrete functions into capabilities. These can then be evaluated against a requirement document and use cases. It is critical that we consider the core requirements and use cases, to ensure we are solving the right problem. It may also be noted:

- Architecture is not a blueprint for implementation;
- Each component are abstract functional units;
- Functions can be realised as separate software blobs on different processors;
- Depending on resiliency requirements, functions may be replicated and distributed, or centralised;
- A protocol provides a realisation of the interaction between architectural, functional components.
- Not all interfaces require protocols; often an interface may be internal.

Various acadamic and industry attempts to define and document candidate SDN, and NFV network architectures exist, but these are use case specific (mostly enterprise and campus networks) and very limited research has been published on large-scale operator use cases. ABNO was one of the first control frameworks that truly met the emerging requirements of real-world operators. The following sections outline some success stories for ABNO implementation.

5.1 Adaptive Network Manager (ANM)

The European Commission funded project "IDEALIST" (BT was a major partner led by Andrew Lord) identified the need for a control architecture [56] to combine the best of distributed routing and signaling protocols. The ABNO architecture provided real-time adaption and to survive against failures, and a centralised intelligence that, on the one hand, provides a point for optimization (e.g. interfacing with the planning tool), and capable of interfacing with the higher-applications, including cloud platforms and data centre (WAN) inter-connections.

The control plane functions are based on the well-known GMPLS architecture, while the centralised intelligence and interface with applications follow an SDN approach. Thus, the "Adaptive Network Manager" (ANM) was the pivotal network controller (underpinned on the ABNO framework), that considers not only the Flexi-grid Network but a wider scope, a multi-layer IP/MPLS over optical Network.

Several initial feasibility studies were conducted to ascertain the suitability of the ANBO-based ANM platform. The scope and outcomes from these early tests are documented in key papers and journals, and my own paper:

- R. Casellas, R. Muñoz, J.M. Fabrega, M.S. Moreolo, R. Martinez, L. Liu, T. Tsuritani, and I. Morita, "Design and Experimental Validation of a GMPLS/PCE Control Plane for Elastic CO-OFDM Optical Networks," Selected Areas in Communications, IEEE Journal on, vol.31, no.1, pp.49,61, January 2013. [66]
- Aguado, et al., "ABNO: a feasible SDN approach for multi-vendor IP and optical networks," in OFC, Th3I.5, March 2014. [67]

- L. Velasco, D. King, O. Gerstel, R. Casellas, A. Castro, and V. López, “In-Operation Network Planning,” IEEE Communications Magazine, vol. 52, pp. 52-60, 2014. [68]

5.1.1 ANM Interfaces

As the ABNO architecture was generic in its intent, most of the interfaces are defined as concepts. In the ANM architecture some modules whose interfaces are not already defined, then HTTP/JSON interfaces will be used in these interfaces. There are two reasons: easy development and flexibility for the workflows definition. These interfaces will help to have a modular design, which can be adapted to the future requirements that may come during the project. If during the project, there are some other solutions in the standardisation fora, this has been assessed and where applicable, included in the ANM architecture.

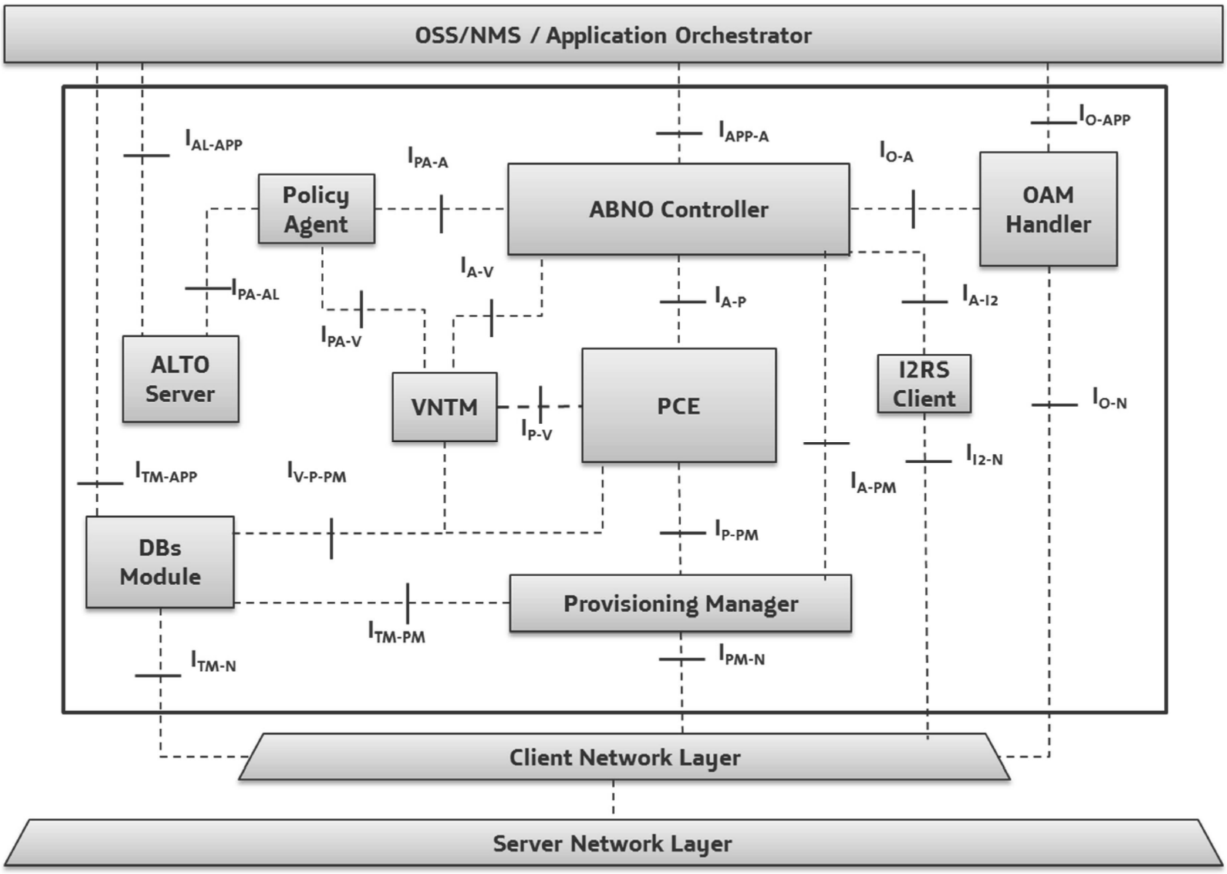


Figure 21 Adaptive Network Manager Functional Components and Interfaces [67]

- IN-APP - This is the interface between the application layer/NMS/OSS and the ABNO controller. Application layer makes requests to set up connections or to trigger any other workflow using HTTP/JSON. This interface is currently under development in the Internet Engineering Task Force (IETF). The parameters of the requested change depending on the workflow, but the operation type is always mandatory;
- IAL-APP - This is the interface between the ALTO Server [70] and Application layer/NMS/OSS, where the Application layer acts as an ALTO Client [70]. They communicate using the ALTO Protocol [69]. They communicate over HTTP/JSON. An information model has to be defined for this interface to support TED, LSPs and inventory requests;
- IA-I2, II2-N - The Interface to the Routing System (I2RS) [59];
- IPA-A, IPA-V, IPA-AL - All the interfaces between the Policy Agent and the modules that request it for permission using an HTTP/JSON request;

- IA-P - This is the interface between the ABNO controller and the PCE. The ABNO controller queries the PCE using PCE, Stateless and Stateful PCEs may be used this interface will support requests for both PCEs;
- IA-V - This interface connects the ABNO controller and the VNTM [60]. They communicate through PCEP.

5.1.2 Adaptive Network Manager (ANM) Network Optimization

While most networks are designed to survive single failures without affecting customer service level agreements (SLAs), they are not designed to survive large-scale disasters, such as earthquakes, floods, wars, or terrorist acts, simply because of their low failure probability and the high cost of overprovisioning to address such events in today's network.

Since many systems might be affected, large network reconfigurations are necessary during large-scale disaster recovery. The disaster recovery process is like that of the virtual topology reconfiguration after a failure. However, multiple optical systems, IP links, and possible routers and OXCs (assuming central offices are affected) may be taken offline during the disaster. Several additional planning and operation requirements in response to largescale disasters are highlighted below:

- Consideration of potential IP layer traffic distribution changes, either using MPLS-TE tunnels or by modification of IP routing metrics, and evaluating benefits based on the candidate topology

- It may be impossible to reach the desired network end state with one-step optimisations. Therefore, two or more step optimisations may be necessary, for example, to reroute some other optical connections to make room for some new connections

- The system must verify that the intermediate configuration after each such step is robust and can support the current traffic and possibly withstand additional outages

- Based on pre-emption and traffic priorities, it might be desirable to disconnect some virtual links to reuse the resources for post-disaster priority connections and traffic

We have described the creation of one disaster recovery plan, but in a real network, there may be several possible plans, each with its pros and cons. The tool must present all these plans to the operator so that the operator can select the best plan, and possibly modify it and understand how it will behave.

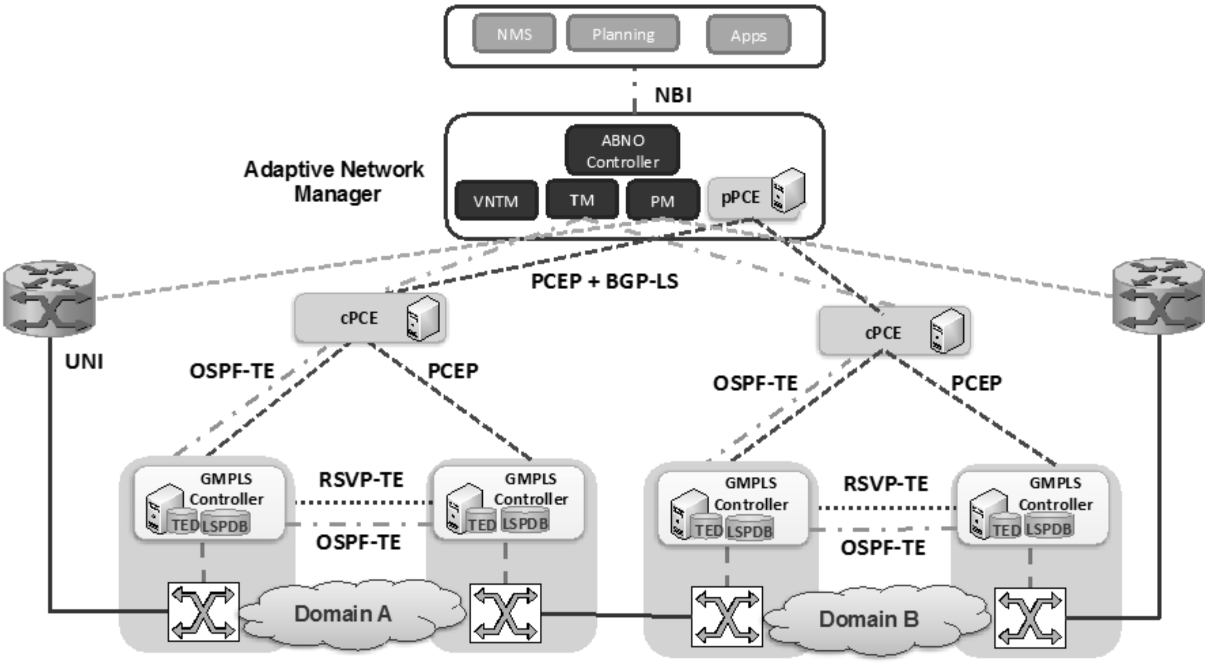


Figure 22 Control Plane Architecture showing a Multi-Domain Network Using an ABNO-based Controller

To summarise, the above process consists of several steps:

1. Immediate action by the network to recover some of the traffic;
2. Dissemination of new or updated network state;
3. The root cause analysis to understand what failed and why;
4. An operator-assisted planning process to come up with a disaster recovery plan;
5. Execution of the plan, possibly in multiple steps;
6. Re-convergence of the network after each step and in its final state.

This scenario for recovering from catastrophic network failures may also be known as "In-Operation Network Planning".

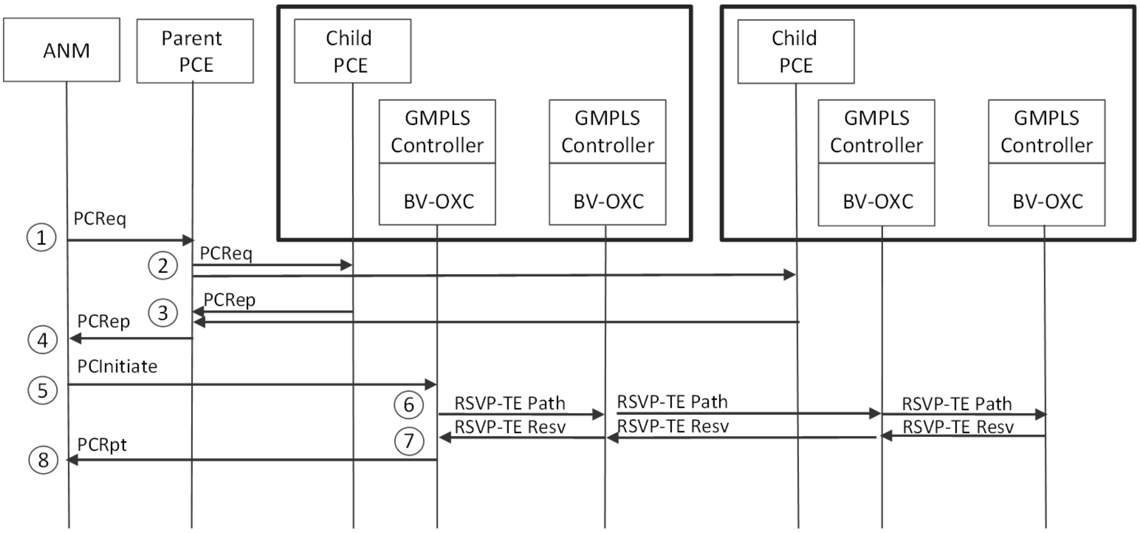


Figure 23 Control Messaging in the ABNO-based Controller Environment [68]

5.1.3 Applicability of ABNO to BT Media and Broadcast

Although ABNO was developed in cooperation with BT applicability to BT Media and Broadcast is currently work in progress. A core design principle for BT Media and Broadcast is to create a contribution and content network that can be deployed rapidly and in a scalable way. The first element to be virtualised is the cache node itself, and then required services such as content monitors and load balancers. OpenSource software-based (virtualised) CDN (vCDN) platforms are available, and at BT we used the Lancaster University developed OpenCache [72] platform, for our lab testing.

A key requirement of the vCDN is reconfigurable bandwidth as the content we move from HD content at 1080p to 4k streams, and demands change based on time of day and week. Deploying the various infrastructure elements of a CDN as a collection of virtual appliances (VNFs) and connecting content and access (user networks) with a flexible optical network infrastructure offers significant benefits.

The following figure describes how an ABNO-enabled network controller would integrate with an NFV-based CDN and shows its capability to future BT Media and Broadcast CDN network infrastructure.

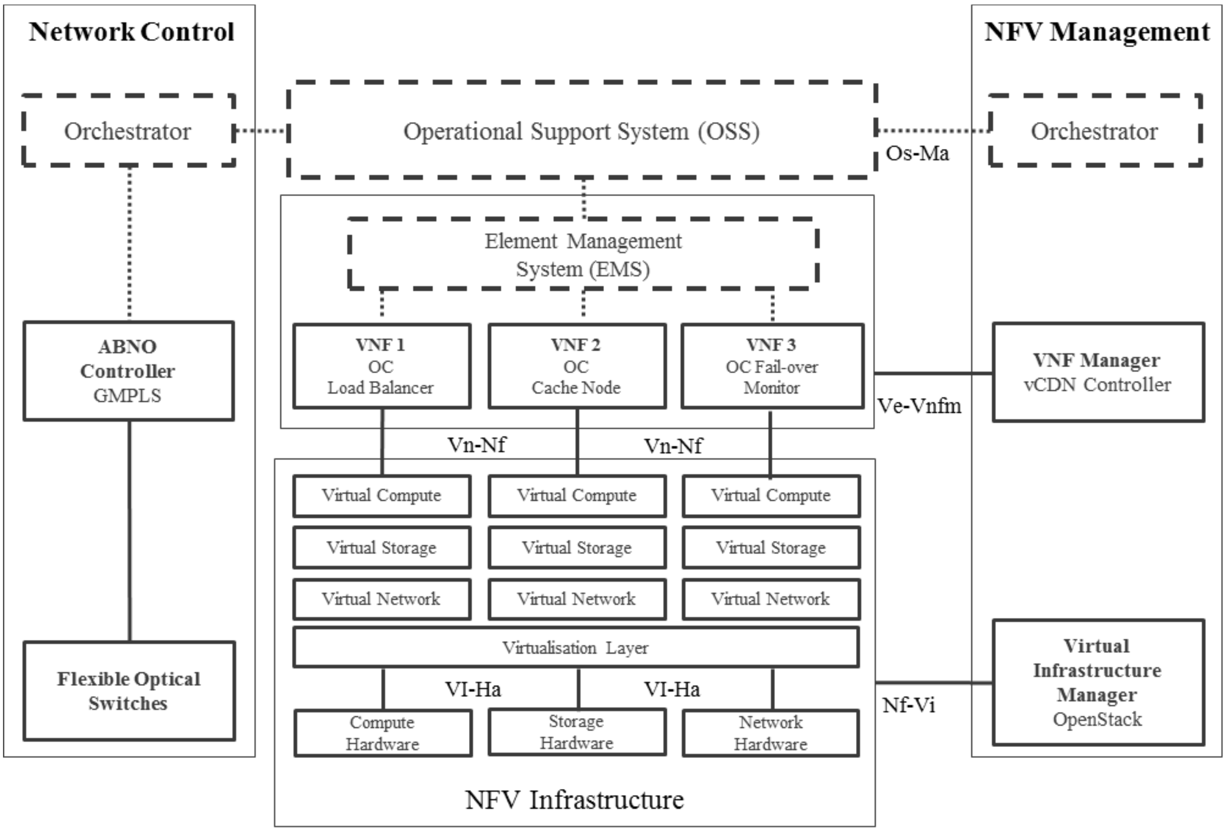


Figure 24 Candidate SDN & NFV Framework based on the ETSI NFV ISG Model using ABNO for Contribution Video Distribution [73]

The functional components and interfaces identified in figure 23 ("Candidate SDN & NFV Framework based on ETSI NFV ISG Model using ABNO for Contribution Video Distribution") above were identified to deliver a workable architecture for BT Media and Broadcast [73]. The interfaces are further described below:

1. **Os-Ma**: an interface to OSS and handles network service lifecycle management and other functions
2. **Vn-Nf**: represents the execution environment provided by the Vim to a VNF (e.g. a single VNF could have multiple VMs)

3. **Nf-Vi**: interface to the Vim and used for VM lifecycle management
4. **Ve-Vnfm**: interface between VNF and Vnfm and handles VNF set-up and tear-down
5. **Vi-Ha**: an interface between the virtualisation layer (e.g. hypervisor for hardware compute servers) and hardware resources

Using the ABNO-based controller in conjunction with the NFV Management and Infrastructure itself would provide the VNFs connectivity over a high-bitrate optical infrastructure, and similar flexibility in the IP and Ethernet layer, which until recently with the advent of Elastic Optical Networks, was simply not previously available in the optical transport domain.

This proposal highlighted, and the interfaces identified, are now being considered by BT for development into a further “Phase 3 – Trial”. If successful, would form the basis of an operational platform for future BT Media and Broadcast services.

6. Conclusion

The efforts described in this thesis to design and build a control platform for next generation transport networks have proven very successful. The author of this thesis – who has taken the lead throughout these efforts over the past several years – has used his work to directly impact the telecoms development of next generation transport controller architectures. Key achievements have been the ABNO development effort, the related ABNO documents and papers, standardisation of the framework in the relevant industry standards forum, and also coordination of BT's network operator contributions to the framework. The author, as lead researcher, also facilitated ABNO acceptance by industry and academia dissemination via papers, resource models used by ABNO, applicability statements for ABNO, and also ABNO-based tutorials and workshops. The feasibility of ABNO and its subsequent adoption by numerous industry partners, research projects and also within the relevant part of the academic world, demonstrates its novelty, relevance and timely adoption.

Emerging optical technologies are providing a compelling answer for exponential bandwidth consumption, and a variety of European Commission projects have utilised ABNO as a solution to the lack of automation, service elasticity and reduction in operational complexity and costs when compared to traditional techniques. We have identified that Elastic Optical Networks (EON) and the flexi-grid (flexible bit rate) technology offers important benefits and capabilities, including wavelength slicing from 100Mb/s up to 200Gb/s, and beyond. Again, ABNO-based controllers have proven more than capable and will generate innovative research for many years to come.

The BT organisation sponsoring the researcher provided the environment for the thesis author to develop their ABNO framework and facilitate its application to several use cases, both within BT but also within other European operator environments, and telecoms labs. The ABNO framework is being considered for BT Media and Broadcast network, utilising on the principles on SDN, NFV and related technologies, and initial thoughts are the ABNO framework will offer exciting results. These benefits should manifest themselves as new service capabilities and flexibility while reducing costs across multiple layers for the transport of BT's Media and Broadcast services.

Using an ABNO-based control platform, BT will be able to set up and tear down end-to-end connections, via a centralised controller, significantly faster with less protocol complexity compared to existing transmission and IP/MPLS networks. Furthermore, using protocol agnostic south-bound interfaces and commodity routers and switches will offer a significant reduction in capital costs.

The feasibility of ABNO and subsequent adoption by numerous industry partners, research projects and wider academia, demonstrates its relevance and timely adoption. Furthermore, ABNO was instantiated as an Adaptive Network Manager (ANM) in the H2020 IDEALIST project. Supported by key industrial vendors including Ericsson, ADVA and Alcatel (now Nokia) which underscores ABNOs industrial usage, as well as academic. Other challenges remain for ABNO, as highlighted in the following sub-section 6.2 ("Areas for Further Research").

6.2 Research Questions-Findings

The overall objectives and research described in this document were firmly anchored around three initial research questions, these were:

1. **How can we meet Internet bandwidth growth yet minimise network costs?**

Transport networks are used to aggregate traffic pipes from multiple users and services among different cities, regions, or continents. Typically, the operation of this infrastructure has been complex and was not capable of adapting to significant traffic changes without significant manual input. The

ABNO framework approach will help operators to reduce the CAPEX and OPEX in the networks, thanks to the optimisation of the resources and the reduction of the complexity in the operation of the network.

2. **Which enabling network technologies might be leveraged to control network layers and functions cooperatively, instead of separated network layer and technology control?**

The ABNO framework is well-adapted to heterogeneous network environments, avoiding vendor lock-in (solutions in the market are typically mono-vendor), support for a variety of SDN (including Open Flow (OF) networks), facilitating edge-to-edge multi-domain path setup.

3. **Is it possible to utilise both centralised and distributed control mechanisms for automation and traffic optimisation?**

A key consideration of the research was to consider if it was feasible and useful to blend both distributed and centralised control planes. While the initial findings on the functional benefits of the ABNO framework look very promising, adopting an approach where both the hierarchical centralised and distributed models may be utilised and exploited is a complex process. The current findings discussed in this document highlight that a hybrid control plane deployment model would yield the greatest benefits.

The ABNO-based centralised controller may act as a consistent global database and specific network mechanisms to ensure new traffic or service requests are handled consistently. A cluster of ABNO-based controllers may be deployed, to improve partition tolerance, but the potential issue of network resource inconsistency must be considered, and some form of global network state synchronisation needs to be provided between controllers.

By its nature, a distributed control plane will be dynamic, with any link or service state change being propagated via the distributed communication mechanisms. If we consider convergence after the partition of the network, a traditional distributed control-plane operation provides high survivability, fast recovery, and can maintain an accurate state. The centralised controller element may then be used to compute end-to-end services that are built across multi-domain and multi-technology environments, facilitate network-wide transactions such as specific application grooming and global concurrent optimisations.

Several challenges will stem from stitching heterogeneous environments across multiple technological and administrative domain-levels, spanning multiple resource segments. These challenges include scaling the control architecture, addressing the potential system complexity of maintaining state synchronisation between the SDN Orchestrator and Child Controllers and adapting YANG resource models for control of end-to-end services.

6.3 Areas for Further Research

The following sub-sections outline key opportunities to continue the investigation and research of the ABNO architecture, with a focus on applying ABNO to future networking, including network slicing.

6.3.1 Applicability of ABNO to Slicing as a Service (Saas) and Beyond

The advent of 5G to serve large-scale deployment of networked sensors, mission-critical services, and evolved residential and business applications is an exciting prospect. Automating the provisioning of 5G services, deployed over a heterogeneous infrastructure (regarding domains, technologies, and management platforms), remains a complex task, yet driven by the constant need to provide end-to-end connections at network slices at reducing costs and service deployment time. At the same time, such services are increasingly conceived around interconnected functions and require allocation of computing, storage, and networking resources.

The provisioning of 5G services (network connectivity, services involving heterogeneous resources) and network slicing will require automated connection setup using specific requirements regarding quality of service, latency, bandwidth, enabling recovery (protection and restoration), across multiple domain and technology layers. This makes ABNO a highly suitable control and orchestration architecture.

Two large European Commission funded projects (H2020 5G “CROSSHAUL” and H2020 5G “METRO-HAUL” are actively investigating and using ABNO for 5G services and network resource control. While the initial findings on the functional benefits of varying control plane deployment scenarios, adopting a common approach where both distributed and centralised models can be utilised and exploited, would yield the greatest benefits. However, several challenges will stem from stitching heterogeneous environments across multiple technological and administrative domains, spanning multiple network segments.

Therefore, significant research work will be required for METRO-HAUL to provide complete integration in which constrained 5G services (including end-to-end connections and network slices) are allocated in environments spanning multiple administrative domains, supported by heterogeneous control planes, while ultimately requiring flexible control and monitoring by the instance controller. Furthermore, it is expected that advances related to data analytics (telemetry) and machine learning are also required for improved control of 5G services managed by, most likely, a hierarchical control system.

The following sub-sections outline key areas for further ABNO investigation and development for 5G networks and services.

6.3.1.1 Requirements for Network Slicing

A platform managing network slicing will have to provide the following capabilities, as defined by the 5G PPP discussion [77], [78]:

- **Resource Slicing:** For network slicing, it is important to consider both infrastructure resources and service functions, allowing a flexible approach to delivering a range of 5G services both by partitioning (slicing) the available network resources to present them for use by an application or consumer. It would also provide instances of service and network function at the right locations and in the correct chaining logic, with access to the necessary hardware, including specific compute and storage resources. Mapping of resources to slices may be 1-to-1, or resources might be shared among multiple slices;

- **Network and Function Virtualization**: Virtualization is the abstraction of resources where the abstraction is made available for use by an operations entity, for example, by the Network Management Station (NMS) of a high-layer network. The resources to be virtualised can be physical or already virtualised, supporting a recursive pattern with different abstraction layers. Therefore, virtualisation will be critical for network slicing as it enables effective resource sharing between network slices;

  Just as server virtualisation makes virtual machines (VMs) independent of the underlying physical hardware, network virtualisation will facilitate the creation of isolated (virtual) networks, which are then decoupled from the underlying physical transport network;

- **Resource Isolation**: Isolation of data and traffic is a major requirement that must be satisfied for certain applications to operate in concurrent network slices on a common shared underlying infrastructure. Therefore, isolation must be understood regarding;

- **Performance**: It is critical that each virtual slice is created to meet specific service objectives and performance requirements. These are usually identified as operator Key Performance Indicators (KPIs). Furthermore, performance isolation per slice is required. No network slice should be adversely impacted by application or use congestion on other slices;

- **Security**: Attacks or faults occurring in one slice must not have an impact on other slices, or service flows are not only isolated on network edge, but multiple customer traffic is not mixed across the core of the network.

- **Slice Management**: Each slice must be independently viewed, utilised, and managed as a separate network.

6.4.1 Orchestration of ABNO-based Controllers in 5G

Large network operators, like British Telecom, must integrate multiple transport domain technologies for next generation transport networks, including 5G. By allowing a single converged network infrastructure to deliver multiple service types, with varying characteristics and meeting the dynamic demands of large bandwidth, low latency, applications.

It has been demonstrated that ABNO may directly manage a variety of network devices using multiple programming methods, as well as coordinate several control plane instances (such as SDN controllers or PCE to provide end-to-end connectivity across multiple transport domains that may be comprised of varying technologies or managed by different administrative zones, even within a single operation like British Telecom.

However, a multi-domain network coordination mechanism between ABNO controllers would need to be developed. This might sit on top of the ABNO architecture and provide an abstracted and virtual view of the tenant's virtual infrastructure exposing topological information.

6.4.1.1 Reliability of the ABNO Controller

Any future 5G network will be carrying mission-critical services and connectivity across the network will have to be reliable for such services. There are primarily two types of failure recovery mechanisms: restoration and protection for the network element and link failures. Restoration is a reactive strategy, while protection is a proactive strategy.

The recovery paths can be either pre-planned or dynamically allocated, but resources are not reserved until failure occurs. Additional signaling is required to establish the restoration path when a failure occurs. Protection: The paths are pre-planned and reserved before a failure occurs. When a failure occurs, no additional signaling is needed to establish the protection path. Compared to the restoration scheme, the protection scheme can enable faster recovery without the involvement of the network controller when failures are detected.

Moreover, the required bandwidth and latency during failures can be considerably reduced because no interactions are required between switches and the controller. Therefore, for large-scale SDN systems, path protection solutions are more favourable to achieve fast failure recovery

6.4.1.2 Network Telemetry and ABNO

Dynamic resource setup and reallocation is critical for 5G operators; however, these capabilities are heavily dependent on the ability to measure and collect transport network performance information [79], then evaluate network and service quality using a very small set of metrics (including KPIs), then providing a network or service diagnosis, or root cause analysis for service disruptions. In parallel, the ABNO controller must support network resource scheduling which can adapt to real-time connection setup or resizing demands.

Work has begun on developing telemetry models to support ABNO-based management and recovery. Generally, this work would use a YANG-based [76] telemetry model, and a set of candidate models and how they would be used have been recently submitted to the IETF as a standardisation activity [80].

6.4.1.3 Securing the ABNO Controller and Network Resources

Securing the transport network when using a centralised controller and distributed forwarding nodes poses significant challenges. By its nature, an ABNO-enabled transport network will encounter multiple threat vectors and may be more vulnerable than traditional network architecture. Traditional security techniques and solutions may not be applicable, as the transport topology changes and the network, forwarding fabric, is reconfigured, new security policies will be inserted, and multiple security services must be enabled and monitored. Furthermore, the significant challenge of managing the trade-offs between network security, performance and flexibility must also be evaluated, to ensure 5G services and networks are hardened to cyber-attacks.

7. Impact

The impact of ABNO is summarised here, covering practical dissemination including academic research and collaborative project use (sub-section 7.1 "Research Projects using ABNO") and industrial use (sub-section 7.2 "Industrial Uptake"). The industrial dissemination includes the adoption of ABNO as an IETF Internet Standard – RFC7491 [54], which then spawned numerous additional work items within the IETF for ABNO-related resource models and interfaces.

7.1 Research Projects using ABNO

The ABNO framework and subsequent architecture have been adopted and exploited in numerous collaborative research projects; in chronological order, these include:

FP7 IDEALIST – ABNO Is used as the central management framework for an industry-driven elastic and adaptive optical network infrastructure for transport networks. The platform that integrated ANBO design principles was Adaptive Network Manager (ANM), which was the network management and operation platform developed by IDEALIST for control of the Elastic Optical Networks (EON). The ANM platform provided Multi-Layer Path Provisioning, Multi-layer Restoration and Network Optimization after Restoration.

EC FP7 OFERTIE – In the OFERTIE (OpenFlow Experiment in Real-Time Internet Edutainment) project we were researching the use of software-defined networking (SDN) to improve delivery of an emerging class of distributed applications for the Future Internet known as Real-Time Online Interactive Applications (ROIA).

ABNO was used to enhance the OFELIA testbed facility to allow researchers to request, control and extend network resources dynamically.

EC FP7 DISCUS – The DISCUS project demonstrated a complete end-to-end architecture and technologies for an energy efficient and environmentally sustainable optical network. It provided a revolution in communications networks applicable across Europe and the wider world exploiting to the full the opportunity offered by LR-PONS technology and flat optical core networks to produce a simplified and economically efficient infrastructure. The ABNO controller platform was used for the distributed DISCUS core, providing high-bandwidth services for all users and services.

EC FP7 CONTENT – CONTENT developed the next generation ubiquitous converged network to support the future Infrastructure as a Service (IaaS) platforms. It provided a technology platform interconnecting geographically distributed computational resources that can support a variety of Cloud and mobile Cloud services. The connectivity required between mobile and fixed end-users and the IT resources was provided by an advanced multi-technology network infrastructure, where computational resources are shared and accessed remotely on an on-demand basis in accordance to the cloud computing paradigm.

The ABNO platform facilitated the convergence of wireless and optical network and IT resources in support of CONTENT IaaS Cloud services.

EC FP7 STRAUSS – The STRAUSS project developed highly efficient and global (multi-domain) optical infrastructure for Ethernet transport. It's architecture leveraged SDN principles for flexible optical circuit and packet switching technologies beyond 100 Gbps. It used ABNO for dynamic virtual network reconfiguration over SDN orchestrated multi-technology optical transport domains.

EC FP7 LIGHTNESS – Developed a metro/core network orchestration platform using a centralised ABNO-based decision point responsible for inter-data centre network resources allocation.

FI-PPP XIFI – One of the key points XIFI was to use OpenNaaS with network services provided by European NRENs and GEANT, enabling an effective orchestration of network resources to facilitate the deployment of several Future Internet application scenarios.

ABNO – Provided a controller for creating a multi-DC community cloud across Europe. It was used to facilitate on-demand and application-specific reservation of network connectivity, reliability, and resources.

TOUCAN – The TOUCAN project aims are bold "to achieve ultimate network convergence enabled by a radically new technology agnostic architecture targeting a wide range of applications and end users", this required a radically different approach to network resource management, i.e., ABNO.

H2020 ACINO – Providing infrastructure for application-centric optical and IP network orchestration based on ABNO.

H2020 ORCHESTRA – Using ABNO OAM Handler for optical performance monitoring for enabling dynamic networks using a holistic cross-layer, self-configurable approach.

More recently, ABNO has been adopted to address 5G network control and orchestration requirements:

H2020 5G CROSSHAUL – Developing a 5G integrated backhaul and fronthaul transport network enabling a flexible and software-defined reconfiguration of all networking elements in a multi-tenant and service-oriented unified management environment. The control platform is ABNO-based.

H2020 5G METRO-HAUL – Providing all the elements of the transmission, switching, networking, compute, and storage, orchestrating dynamic solutions for next generation 5G applications and services. The control platform is ABNO-based.

7.2 Industry Uptake of ABNO

Contribution to Internet Standards was a key objective during the researchers PhD research period. A relevant organisation for ABNO was the Internet Engineering Task Force (IETF). The main method of participation is via mailing lists: there is one mailing list for each working group where all topics relevant to the working group are discussed. They also attended several IETF meetings where they contributed directly using research developed during my PhD.

The most direct method of contribution is proposing an Internet Draft of a technical solution and using a working group to develop the documents that will be progressed towards becoming an RFC. Thus, it would be classed as an Internet Standard, Informational Standard, Best Practice Standard, or an Experimental Standard.

An RFC proposal is reviewed by IETF participants, typically engineers from vendors or network operators, by researchers, or by scientists in the form of a document describing methods, behaviours, research, or innovations. They take many forms: requirements, architecture, protocol specifications, and best practices. All apply to the working of the Internet and Internet-connected systems. Each proposal is submitted either for review by a working group tasked with a specific technology topic or challenge or convey new concepts, information.

7.3 Personal ABNO Publications

The following list of the researcher's conference presentations, publications, book chapters, and peer-reviewed journals further highlights the impact that ABNO has had on research and industry.

- **“Software Defined Networks”**
  Jan 9, 2012
  UKNOF 21 London
- **“Blending SDN with PCE for Scalable Data Center Service Deployment”**
  May 31, 2012
  iPOP (IP over Optical) Tokyo
- **“Computing Protection and Recovery Paths for Data Center Services and Applications”**
  Oct 28, 2012
  ISOCORE MPLS Washington
- **“A PCE-based Architecture for Application-based Network Operations”**
  Feb 25, 2013
  Internet Engineering Task Force
- **“A Critical Survey of Network Functions Virtualisation (NFV)”**
  May 30, 2013
  iPOP (IP over Optical) Tokyo
- **“Using the Path Computation Element to Enhance SDN for Elastic Optical Networks (EON)”**
  May 31, 2013
  iPOP (IP over Optical) Tokyo
- **“Adaptive Network Manager: Coordinating Operations in Flex-grid Networks”**
  June 23, 2013
  IEEE Transparent Optical Networks (ICTON), 2013 15th International Conference
- **“Network Functions Virtualisation: The New Frontier of Telecoms Innovation”**
  Jul 11, 2013
  Multi-Service Networking, Science & Technology Facilities Council, Abingdon, UK
- **“Unification of Formal and De Facto Standards for Abstraction and Autonomic Control of the Transport Network**”
  Oct 13, 2013
  Layer123 SDN & NFV World Congress
- **“An Architecture for Application-based Network Operations”**
  Nov 18, 2013
  MPLS & SDN Washington 2013
- **“In-operation Network Planning”**
  Jan 22, 2014
  IEEE Communications Magazine
- **“SDN Testbed Experiences, Challenges and Next Steps”**
  Jan 30, 2014
  FP7/FIRE SDN Workshop
- **“NFV: A Real Options Analysis for vEPC”**
  Mar 20, 2014
  SDN World Congress & NFV Summit
- **“The Role of PCE in an SDN World”**
  Sep 1, 2014
  European Workshop on SDN (EWSDN)
- **“Architecting SDN for Optical Access Networks”**
  Sep 28, 2014
  European Conference on Optical Communication (ECOC)

- **“NFV Impact on European Research and Education”**
  Oct 17, 2014
  Layer 123 SDN & NFV World Congress
- **“RFC7491: A PCE-Based Architecture for Application-Based Network Operations”**
  March 2015
  Internet Engineering Task Force (IETF)
- “**OpenCache: A Software-defined Content Caching Platform”**
  Apr 14, 2015
  IEEE NetSoft
- **“Evolution of OpenCache: an OpenSource Virtual Content Distribution Network (vCDN) Platform”** May 7, 2015
  Cambridge Wireless, The End of Network Architecture
- **“SDN-based elastic and adaptive optical transport network: findings and future research.”**
  Jun 24, 2015
  WDM & Next Generation Optical Networking
- **“The role of SDN and NFV for flexible optical networks: Status, Challenges and Opportunities.”**
  July 5, 2015
  IEEE Transparent Optical Networks (ICTON)
- **“Using YANG for the dissemination of the Traffic Engineering Database within a software-defined Elastic Optical Networks.”**
  July 5, 2015
  IEEE Transparent Optical Networks (ICTON)
- **“Prospects for the Software Defined Network and Network Function Virtualisation in Media and Broadcast”**
  Oct 22, 2015
  Society of Motion Picture & Television Engineers (SMPTE)
- **“Elastic Optical Networks: Application-Based Network Operations (ABNO)”** (Book Chapter)
  Jun 14, 2016
  Springer Publications
- **“The Software Defined Transport Network: Fundamentals, Findings and Futures”**
  July 13, 2016
  Multi-Layer Network Orchestration (NetOrch)
- **“Network-based Telemetry to Facilitate the Programmable Management Plane for Optical Transport Infrastructure”**
  July 14, 2016
  IEEE Transparent Optical Networks (ICTON)
- **“Baguette: Towards end-to-end service orchestration in heterogeneous networks.”**
  Oct 25, 2016
  16th International Conference on Algorithms and Architectures For Parallel Processing
- **“Network Service Orchestration Standardization: A Technology Survey”**
  Feb 7, 2017
  Elsevier, Computer Standards & Interfaces
- **“Transport Northbound Interface: The Need for Specification and Standards Coordination”**
  May 16, 2017
  IEEE Transparent Optical Networks (ICTON)

- **“A Yang Data Model for WSON Tunnel”**
  June 2017
  Internet Engineering Task Force (IETF)
- **“A Yang Data Model for WSON Optical Networks”**
  July 2017
  Internet Engineering Task Force (IETF)
- **“YANG data model for Flexi-Grid media-channels.”**
  July 2017
  Internet Engineering Task Force (IETF)
- **“YANG data model for Flexi-Grid Optical Networks.”**
  July 2017
  Internet Engineering Task Force (IETF)
- **“YANG models for ACTN TE Performance Monitoring Telemetry and Network Autonomics.”**
  July 2017
  Internet Engineering Task Force (IETF)
- **“Applicability of Abstraction and Control of Traffic Engineered Networks (ACTN) to Network Slicing”**
  July 2017
  Internet Engineering Task Force (IETF)
- **“Transport Northbound Interface Use Cases”**
  July 2017
  Internet Engineering Task Force (IETF)

7.4 ABNO Open Source Software

Various OpenSource instantiations exist of ABNO implementations; these include:

Netphony Suite

The Telefonica Netphony suite is a set of Java-based libraries that enable to create an ABNO-based centralised control plane. It comprises a set of components, distributed as JAR files, which are hosted in publicly available GitHub repositories.

https://github.com/telefonicaid/netphony-abno

iONE

An implementation of the ABNO architecture, named as iONE. The iONE platform consists of a single generic configurable module and a set of dynamically linkable workflows. The main application is optical spectrum defragmentation and used to experimentally demonstrate iONE's key functions.

https://ieeexplore.ieee.org/document/7329043/

7.5 Transport Network Resource Models and North-bound API

An ancillary activity that was spawned by the research was the need to develop additional resource modes for transport network resources, that would be controlled using an ABNO-based controller. It was important that the wider industry accepted these resource models. Therefore the IETF was the forum used, and the following models are now being developed towards a further set of RFCs:

“A YANG Data Model for Flexi-Grid Media-Channels”
https://tools.ietf.org/html/draft-ietf-ccamp-flexigrid-media-channel-yang

“YANG data model for Flexi-Grid Optical Networks”
https://tools.ietf.org/html/draft-ietf-ccamp-flexigrid-yang

“A Yang Data Model for WSON Tunnel”
https://tools.ietf.org/html/draft-ietf-ccamp-wson-tunnel-model

“A Yang Data Model for WSON Optical Networks”
https://tools.ietf.org/html/draft-ietf-ccamp-wson-yang

Transport network domains (OTN and WDM), managed by ABNO would benefit from a well-defined open-source interface (API) to each transport network domain controller. This is required for operators to facilitate control automation and orchestrate end-to-end services across multi-domain networks. These functions may be enabled using standardised data models (e.g. aforementioned resource models), and appropriate protocol (e.g., NETCONF and RESTCONF).

“Transport Northbound Interface Applicability Statement”
https://tools.ietf.org/html/draft-ietf-ccamp-transport-nbi-app-statement

This document analyses the applicability of the resource YANG models discussed and being defined by the IETF to support control and orchestration across transport domains via a North-bound Interface (NBI).

8. References

[1] B. Carpenter, “Architectural Principles of the Internet. Internet”. IETF RFC 1958, June 1996.

[2] P. Zave, and J. Rexford., “The compositional architecture of the Internet”, In Proceedings of ACM Conference, Washington, DC, USA, July 2017.

[3] Xipeng Xiao and L. M. Ni, "Internet QoS: a big picture," in IEEE Network, March-April 1999.

[4] M. J. O'Mahony, D. Simeonidou, D. K. Hunter and A. Tzanakaki, "The application of optical packet switching in future communication networks," in IEEE Communications Magazine, March 2001.

[5] Hybrid Cloud Places New Demands On the Network, Forrester Consulting, Commissioned by Juniper Networks, April 2014

[6] 2017 (Conference’17) E. Mannie (Ed.), “Generalized Multi-Protocol Label Switching (GMPLS) Architecture”, IETF RFC 3945, October 2004.

[7] Awduche, D.O. and Bijan J., “Internet Traffic Engineering Using Multi-Protocol Label Switching (MPLS)”, Computer Networks, 2002.

[8] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner. “OpenFlow: Enabling Innovation in Campus Networks”, ACM SIGCOMM Computer Communication Review, Volume 38, Number 2, April 2008.

[9] E. Rosen, A. Viswanathan, R. Callon, “Multiprotocol Label Switching Architecture”, IETF RFC 3031, January 2001

[10] Awduche, D., et al., “Overview and Principles of Internet Traffic Engineering”, IETF RFC 3272.

[11] Srikanth Kandula, Dina Katabi, Bruce Davie, and Anna Charny, “Walking the tightrope: responsive yet stable traffic engineering” SIGCOMM, August 2005.

[12] D. Zhang and D. Ionescu, "QoS Performance Analysis in Deployment of DiffServ-aware MPLS Traffic Engineering," Eighth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, 2007.

[13] Martin Casado, Teemu Koponen, Scott Shenker, and Amin Tootoonchian. “Fabric: a retrospective on evolving SDN”, Hot Topics in Software Defined Networks (HotSDN). 2012.

[14] J. Moy, “OSPF Version 2”, IETF RFC 2328, April 1998

[15] Oran, D., Ed., "OSI IS-IS Intra-Domain Routing Protocol", IETF RFC 1142, February 1990.

[16] Rekhter, Y.; Li, T.; Hares, S., “A Border Gateway Protocol 4”, IETF RFC 4271, 2005.

[17] A. Feldmann and J. Rexford, "IP network configuration for intradomain traffic engineering," IEEE Network, vol. 15, no. 5, Sept.-Oct. 2001.

[18] R. Braden, Ed., L. Zhang, S. Berson, S. Herzog, and S. Jamin, “Resource ReSerVation Protocol (RSVP)”, IETF RFC 2205, September 1997.

[19] Andy Rayner, Michael Firth, “Broadcast Video & Data Over MPLS”, http://www2.bt.com/static/i/media/pdf/campaigns/media_broadcast/mpls_white_paper.pdf, 2005.

[20] Silva, Walber & Sadok, Djamel, “A Survey on Efforts to Evolve the Control Plane of Inter-Domain Routing. Information”, Information and Communications Technology, 2018.

[21] L. Berger (Ed.), “Generalized multi-protocol label switching (GMPLS) resource reservation protocol-traffic engineering (RSVP-TE) extensions”, IETF RFC 3473, January 2003.

[22] J. Lang (Ed.), “Link Management Protocol (LMP)”, IETF RFC 4204, Oct. 2005.

[23] A. Farrel, J.P. Vasseur, and J. Ash, “A Path Computation Element (PCE)-Based Architecture,” IETF RFC 4655, Aug. 2006.

[24] J. P. Vasseur Ed. and JL. Le Roux Ed., “Path computation element (PCE) communication protocol (PCEP)”, IETF RFC 5440, March 2009.

[25] E. Crabbe, I. Minei, J. Medved and R. Varga, “PCEP extensions for stateful PCE,” IETF draft-ietf-pce-stateful-pce (work in progress).

[26] E. Crabbe, I. Minei, S. Sivabalan and R. Varga, “PCEP extensions for PCE-initiated LSP setup in a stateful PCE model,” IETF draft-ietf-pce-pce-initiated-lsp (work in progress).

[27] Lord, Andrew, et al. "Evolution from Wavelength-Switched to Flex-Grid Optical Networks." Elastic Optical Networks. Springer International, 2016.

[28] Y. Lee (Ed.), “Framework for GMPLS and Path Computation Element (PCE) Control of Wavelength Switched Optical Networks (WSONs)”, IETF, RFC 6163, April 2011.

[29] O. Gonzalez de Dios (Ed.) and R. Casellas (Ed.), “Framework and Requirements for GMPLS based control of Flexi-grid DWDM Networks”, IETF draft-ietf-ccamp-flexi-grid-fwk, work in progress.

[30] S. Vissicchio, O. Tilmans, L. Vanbever, J. Rexford, “Central control over distributed routing, SIGCOMM Computer Communications”, 2015.

[31] M. Caesar, D. Caldwell, N. Feamster, J. Rexford, A. Shaikh, and J. van der Merwe. Design and implementation of a routing control platform. In NSDI’05, 2005.

[32] Bates, T., Chen, E., and R. Chandra, "BGP Route Reflection: An Alternative to Full Mesh Internal BGP (IBGP)", RFC 4456, April 2006.

[33] Chiosi, Margaret & Clarke, Don & Willis Cablelabs, Peter & Donley, Chris & Johnson Centurylink, Lane & Bugenhagen, Michael & Feger, James & Khan, Waqar & China, Chunfeng & Cui, Hui & Chen China Deng, Clark & , Telecom & Baohua, Lei & Zhenqiang, Sun & Wright, Steven. “Network Functions Virtualisation (NFV) Network Operator Perspectives”, ETSI, 2013.

[34] ETSI GS NFV 001. Network Functions Virtualization (NFV); Use Cases, 2013.

[35] Software Defined Networking Architecture Overview, ONF TR-504, 2014.

[36] J. Lang, B. Rajagopalan, D. Papadimitriou, “Generalized Multi-Protocol Label Switching (GMPLS) Recovery Functional Specification”, IETF RFC 4426, March 2006

[37] ITU-T Recommendation G.872, Architecture of optical transport networks

[38] ITU-T Recommendation G.709/Y.1331, Interface for the optical transport network (OTN)

[39] K. Kompella, Y. Rekhter, “Label Switched Paths (LSPs) Hierarchy with Generalized Multi-Protocol Label Switching (GMPLS) Traffic Engineering (TE)”, IETF RFC 4206, October 2005

[40] J.P. Vasseur, R. Zhang, N. Bitar, J.L. Le Roux, “A Backward-Recursive PCE-Based Computation (BRPC) Procedure to Compute Shortest Constrained Inter-Domain Traffic Engineering Label Switched Paths”, IETF RFC 5441, April 2009

[41] D. King, A. Farrel, “The Application of the Path Computation Element Architecture to the Determination of a Sequence of Domains in MPLS and GMPLS”, IETF RFC 6805, November 2012

[42] Q. Zhao et al., "Extensions to the Path Computation Element Communication protocol (PCEP) for Point-to-Multipoint Traffic Engineering Label Switched Paths", IETF RFC 6006, September 2010.

[43] 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", IETF RFC 5557, July 2009.

[44] Oki, E., Takeda, T., Le Roux, JL., and A. Farrel, "Framework for PCE-Based Inter-Layer MPLS and GMPLS Traffic Engineering", IETF RFC 5623, September 2009

[45] Gredler, H., Ed., Medved, J., Previdi, S., Farrel, A., and S. Ray, "North-Bound Distribution of Link-State and Traffic Engineering (TE) Information Using BGP", IETF RFC 7752, March 2016.

[46] R. Muñoz, R. Casellas and R. Martínez, "PCE: What is it, how does it work and what are its limitations?", Optical Fiber Communication Conference and Exposition and the National Fiber Optic Engineers Conference (OFC/NFOEC), 2013.

[47] A. Mendiola, J. Astorga, E. Jacob and M. Higuero, "A Survey on the Contributions of Software-Defined Networking to Traffic Engineering," in IEEE Communications Surveys & Tutorials, 2017.

[48] N. Sprecher, et al. “An Overview of Operations, Administration, and Maintenance (OAM) Tools”, RFC 7276, June, 2014.

[49] ITU-T Recommendation G.8080/Y.1304, Architecture for the automatically switched optical network (ASON)

[50] Yang, L., Dantu, R., Anderson, T., and Gopal, R., "Forwarding and Control Element Separation (ForCES) Framework", IETF RFC 3746, April 2004.

[51] Halpern, J. and J. Hadi Salim, "Forwarding and Control Element Separation (ForCES) Forwarding Element Model", RFC 5812, March 2010.

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

[53] King, D., Ed., and A. Farrel, Ed., "The Application of the Path Computation Element Architecture to the Determination of a Sequence of Domains in MPLS and GMPLS", RFC 6805, November 2012,

[54] D. King, A. Farrel, “A PCE-based Architecture for Application-based Network Operations,” IETF Internet RFC 7491, March, 2015.

[55] Charalampos Rotsos, Daniel King, Arsham Farshad, Jamie Bird, Lyndon Fawcett, Nektarios Georgalas, Matthias Gunkel, Kohei Shiomoto, Aijun Wang, Andreas Mauthe, Nicholas Race, David Hutchison, Network service orchestration standardization: A technology survey, Computer Standards & Interfaces, 2017.

[56] Sadasivan, G., Brownlee, N., Claise, B., and J. Quittek, "Architecture for IP Flow Information Export", IETF RFC 5470, March 2009.

[57] O. Dugeon, et al., “Path Computation Element (PCE) Database Requirements,” IETF Internet Draft draft-dugeon-pce-ted-reqs-03, February 2014, work in progress.

[58] I. Bryskin, et al. “Policy-Enabled Path Computation Framework”, RFC 5394, December 2008.

[59] Atlas, A., Ed., Nadeau, T., Ed., and D. Ward, "Interface to the Routing System Problem Statement", Work in Progress, draft-ietf-i2rs-problem-statement, March 2015.

[60] Andriolli, F. Cugini, L. Valcarenghi, P. Castoldi and A. Welin, "Virtual network topology manager (VNTM) and path computation element (PCE) cooperation in multi-layer GMPLS networks", Conference on Optical Fiber Communication, 2009.

[61] Case, J., Harrington, D., Presuhn, R., and B. Wijnen, "Message Processing and Dispatching for the Simple Network Management Protocol (SNMP)", STD 62, RFC 3412, December 2002,

[62] R. Enns, et al., “Network Configuration Protocol (NETCONF),” IETF RFC 6241, 2011.

[63] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", IETF 8040, 2017.

[64] Open Networking Foundation, "OpenFlow Switch Specification Version 1.4.0 (Wire Protocol 0x05)", October 2013.

[65] M. Bjorklund, Ed., “YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF),” IETF Request or Comments 6020, 2010.

[66] R. Casellas, R. Muñoz, J.M. Fabrega, M.S. Moreolo, R. Martinez, L. Liu, T. Tsuritani, and I. Morita, "Design and Experimental Validation of a GMPLS/PCE Control Plane for Elastic CO-OFDM Optical Networks," Selected Areas in Communications, IEEE Journal on, vol.31, no.1, pp.49,61, January 2013.

[67] A. Aguado, et al., “ABNO: a feasible SDN approach for multi-vendor IP and optical networks,” in OFC, Th3I.5, March, 2014.

[68] L. Velasco, D. King, O. Gerstel, R. Casellas, A. Castro, and V. López, “In-Operation Network Planning,” IEEE Communications Magazine, vol. 52, pp. 52-60, 2014.

[69] R. Muñoz, R. Vilalta, R. Casellas, R. Martínez, T. Szyrkowiec, A. Autenrieth, V. López, D. López, Integrated SDN/NFV management and orchestration architecture for dynamic deployment of virtual SDN control instances for virtual tenant networks, Journal of Optical Communications and Networking, Vol. 7, No. 11, pp. B62-B70, November 2015.

[70] Seedorf, J. and E. Burger, "Application-Layer Traffic Optimization (ALTO) Problem Statement", RFC 5693, October, 2009.

[71] Alimi, R., Ed., Penno, R., Ed., Yang, Y., Ed., Kiesel, S., Previdi, S., Roome, W., Shalunov, S., and R. Woundy, "Application-Layer Traffic Optimization (ALTO) Protocol", IETF RFC 7285, 2014.

[72] Broadbent, M.; King, D.; Baildon, S.; Georgalas, N.; Race, N., "OpenCache: A software-defined content caching platform," in Network Softwarization (NetSoft), April, 2015.

[73] J. Ellerton, A. Lord, P. Gunning, K. Farrow, P. Wright, D. King, D. Hutchison, “Prospects for software defined networking and network function virtualization in Media and Broadcast”, SMPTE, Oct 2015.

[74] A. Aguado et al., "Dynamic Virtual Network Reconfiguration Over SDN Orchestrated Multi-Technology Optical Transport Domains," in Journal of Lightwave Technology, 2016.

[75] D. King, C. Rotsos, A. Aguado, N. Georgalas and V. Lopez, "The Software Defined Transport Network: Fundamentals, findings and futures," 2016 18th International Conference on Transparent Optical Networks (ICTON), Trento, 2016.

[76] M. Bjorklund, “YANG - A data modeling language for the network configuration protocol (NETCONF)”, IETF RFC 6020, 2010.

[77] 5G Vision. The 5G Infrastructure Public Private Partnership: the next generation of communication networks and services. 2015. U Siddique et al., Wireless backhauling of 5G small cells: Challenges & solution approaches, IEEE Wireless Comm. vol.22, no.5, pp.22-31, 2015.

[78] J. Ordonez-Lucena, P. Ameigeiras, D. Lopez, J. Ramos-Munoz, J. Lorca, and J. Folgueira, “Network Slicing for 5G with SDN/NFV: Concepts, Architectures, and Challenges,” IEEE Communications Magazine, vol. 55, pp. 80-87, 2017.

[79] D. King and Y. Lee, “Applicability of Abstraction and Control of TE Networks (ACTN) to Network Slicing”, IETF, draft-king-teas-applicability-actn-slicing, work-in-progress, 2018.

[80] Y. Lee, D. Dhody, S. Karunanithi, R. Vilalta, D. King, D. Ceccarelli, “YANG models for ACTN TE Performance Monitoring Telemetry and Network Autonomics”, IETF, draft-leeteas-actn-pm-telemetry-autonomics, 2017.

Appendix

A1. Contributing Publications

This PhD Thesis has been prepared in the Alternative Format, based on the following contributing documents and peer-reviewed publications:

- D. King, “Network Functions Virtualisation: The New Frontier of Telecoms Innovation”, Multi-Service Networking, Science & Technology Facilities Council, Abingdon, UK, July 2013.
- V. Lopez, D. King, et al., "Adaptive network manager: Coordinating operations in flex-grid networks", IEEE 15th International Conference on Transparent Optical Networks (ICTON), Cartagena, July 2013.
- D. King, “Unification of Formal and De Facto Standards for Abstraction and Autonomic Control of the Transport Network”, Layer123 SDN & NFV World Congress, Dusseldorf, Germany, October 2013.
- D. King, "Architecting SDN for Optical Access Networks”, European Conference on Optical Communication (ECOC), September 2014.
- L. Velasco, A. Castro, D. King, O. Gerstel, R. Casellas and V. Lopez, "In-operation network planning," in IEEE Communications Magazine, January 2014.
- D. King, "SDN-based elastic and adaptive optical transport network: findings and future research”, WDM & Next Generation Optical Networking, June 2015.
- D. King, A. Farrel, N. Georgalas, “The role of SDN and NFV for flexible optical networks: Status, Challenges and Opportunities, IEEE Transparent Optical Networks (ICTON), July 2015.
- D. King, A. Farrel, “RFC7491: A PCE-Based Architecture for Application-Based Network Operations”, Internet Engineering Task Force (IETF), March 2015.
- D. King (Editor), V. Lopez, O, Gonzalez de Dios, R. Casellas, N. Georgalas, A. Farrel, “Elastic Optical Networks Architectures, Technologies, and Control: Application-Based Network Operations (ABNO)”, Springer Publishing, 2016.
- R. Casellas, D. King, et al., "A control plane architecture for multi-domain elastic optical networks: the view of the IDEALIST project," in IEEE Communications Magazine, August 2016.
- C. Rotsos D. King, et al., “Network service orchestration standardization: A technology survey”, Elsevier Computer Standards & Interfaces, Volume 54, November 2017.

These presentations, conference papers and peer reviewed journals may be found below.

Network Functions Virtualisation:
The New Frontier of Telecoms Innovation

Daniel King
PhD Student – Lancaster University
d.king@lancaster.ac.uk

Research – Team

- LU Team
  - PhD Supervisors:
    - Professor David Hutchinson
    - Dr Christopher Edwards
    - Dr Nicholas Race
  - Research Partner
    - Chris Ford, Lancaster University Management School
- Academic Rationale
  - Opportunity to investigate an emerging area in computer science and telecommunications research.
  - Provide useful data and evidence to industry and standards development organisations.
- My Industry Experience
  - Bell Labs, Cisco Systems, Redback Networks, Movaz (ADVA), Aria Networks
  - IETF WG Secretary of ROLL, L3VPN, CCAMP and PCE.
    - Author: RFC4687, RFC5557, RFC6006, RFC6007, RFC6163, RFC6639, RFC6805.
    - Currently progressing 7 WG documents and 7 individual drafts.

Research – Network and Function Virtualisation

Classical Network Appliance Approach

BRAS

Firewall

Carrier Grade NAT

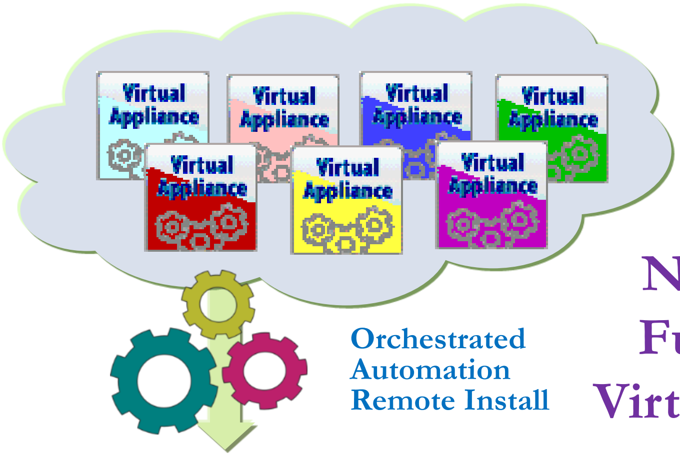
WAN Acceleration

DPI

Session Border Controller

Access Router

Network Function Virtualisation



Orchestrated Automation Remote Install

Hypervisors

Generic High Volume Servers

Generic High Volume Storage

Generic High Volume Ethernet Switches

Research – Investigating the Problem Space

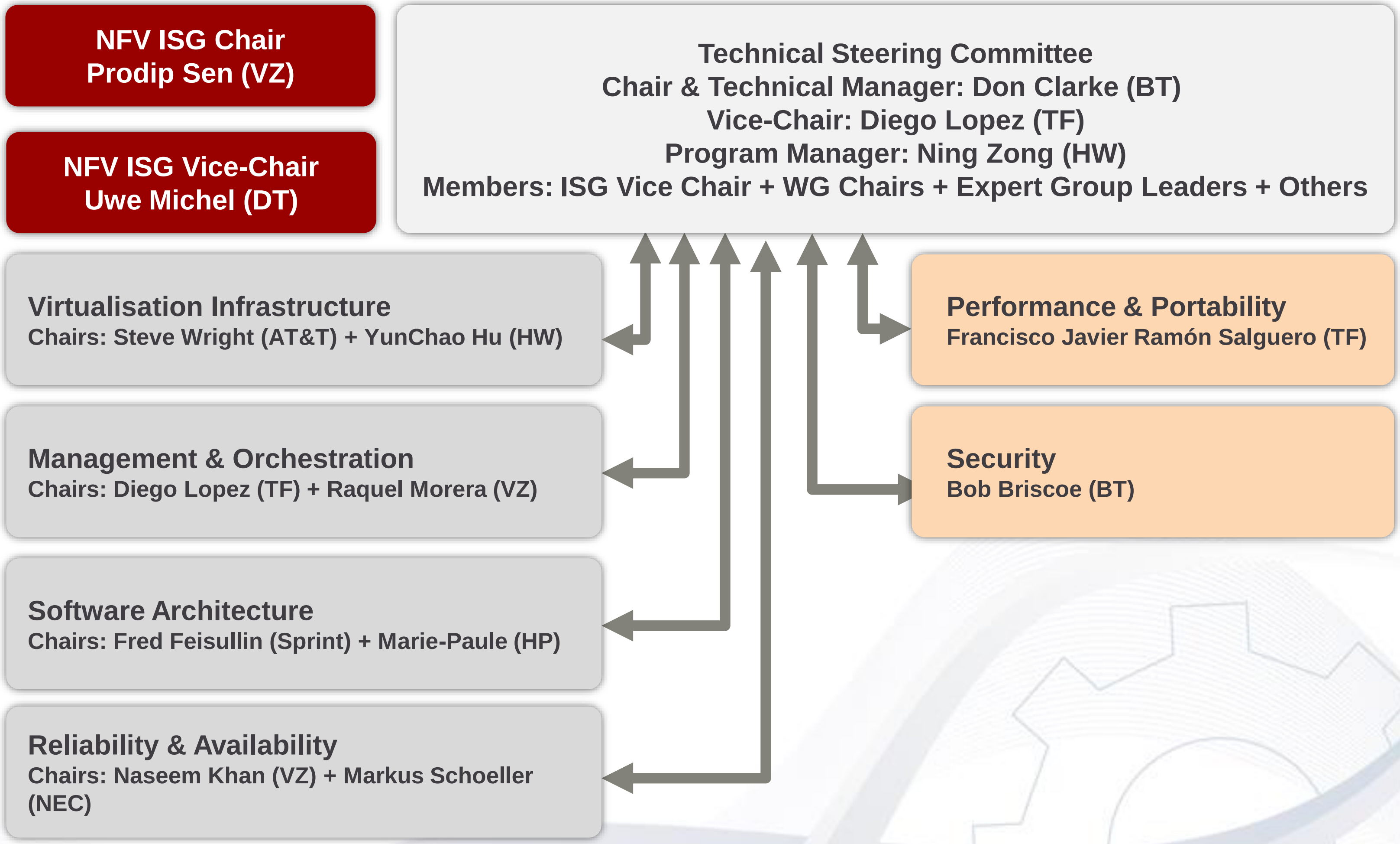
- Evidence gathering
- “**A Critical Survey of Network Functions Virtualization**” to help define the problem space
    - Qualitative and exploratory study (Eisenhardt 1989, Yin 2009, Thomas 2011)
    - Inductive, hypothesis-generating approach
    - Guided by tenets of Grounded Theory (Glaser and Strauss 1967, Charmaz 2006, Corbin and Strauss 2008, Suddaby 2006)
- Analysis (Miles and Huberman 1994)
- Detailed coding of interview transcripts (nVivo).
    - Development of concepts and their dimensions.
    - Intensive review around each concept.
- Interpretation
- Combining memos & concepts into cohesive whole.
    - Establishing cross-user connections.
    - Identifying industry comparatives to inform analysis (e.g., Human Genome Mapping)
- Writing up
- Develop substantive model and frameworks.
- Construct authentic & plausible arguments (economic and technical) based on evidence.
- Publishing findings and conclusions documents (including IETF informational I-Ds and ETSI contributions).

Research – NFV Concept Development

- European Telecommunication Standards Institute (ETSI)
  - Role has been to provide an environment to develop the problem space.
  - Responsibility to publish problem statements, requirements and recommendations.

- ETSI NFV History
  - Whitepaper "Network Functions Virtualisation - An Introduction, Benefits, Enablers, Challenges & Call for Action", October 2012.
  - Initial concepts discussed at the end of 2012 in ETSI Future Networks Workshop.
  - Formal Industry Specification Group (ISG) session in January, 2013.
  - NFV ISG has met twice in 2013, with a third session planned for Bonn in July 2013.

Research – ETSI NFV ISG Structure



NFV ISG Chair
Prodip Sen (VZ)

NFV ISG Vice-Chair
Uwe Michel (DT)

Technical Steering Committee
Chair & Technical Manager: Don Clarke (BT)
Vice-Chair: Diego Lopez (TF)
Program Manager: Ning Zong (HW)
Members: ISG Vice Chair + WG Chairs + Expert Group Leaders + Others

Virtualisation Infrastructure
Chairs: Steve Wright (AT&T) + YunChao Hu (HW)

Management & Orchestration
Chairs: Diego Lopez (TF) + Raquel Morera (VZ)

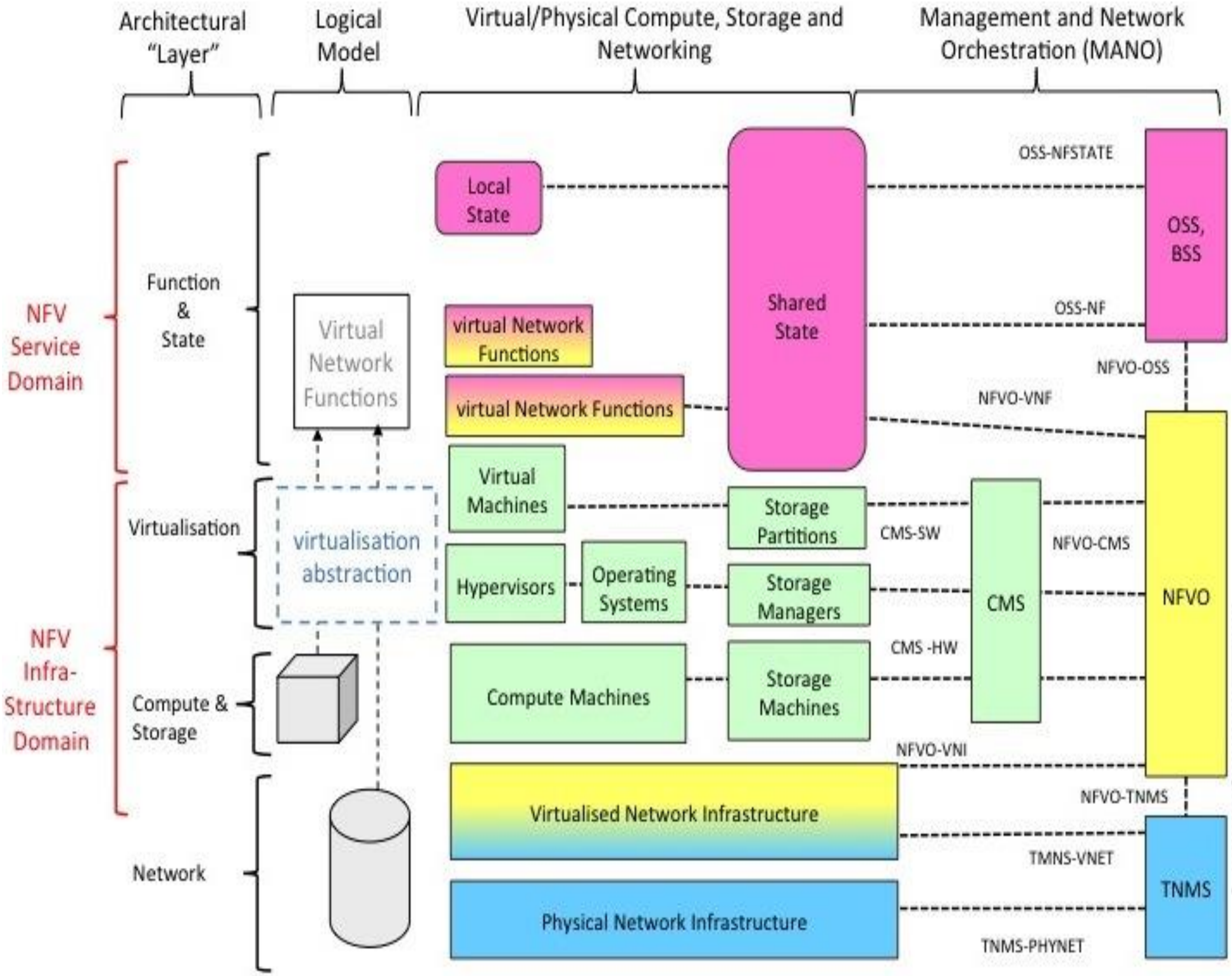
Software Architecture
Chairs: Fred Feisullin (Sprint) + Marie-Paule (HP)

Reliability & Availability
Chairs: Naseem Khan (VZ) + Markus Schoeller (NEC)

Performance & Portability
Francisco Javier Ramón Salguero (TF)

Security
Bob Briscoe (BT)

Research –NFV ISG Work Contributions



Research – NFV Interviewees

- A total of Twenty (20) CSPs have been identified and targeted.
- Discussions and interviews to date:
    - British Telecom
    - Verizon
    - KDDI
    - AT&T
    - Telefonica
    - Telstra
    - NTT docomo
    - France Telecom
    - Deutsche Telekom
- Initial focus on CSPs to gain rich data and develop initial concepts.
- Second round includes vendors and other stakeholders.

Findings – So Far (1)

- Operators have been independently researching network and function virtualisation with hardware and software vendors for years.
- "Enablers for NFV?"
  - Open Innovation during early stages of process and technology development
  - Performance of commodity hardware
  - Success of previous Hosted and Cloud Services
- Most interviews highlighted that industry cooperation is required to:
  - Sanity check use cases.
  - Apply pressure on vendors.
  - Provide the economy of scale for commercial development, deployment and operation of NFV-enabled services.

Findings – So Far (2)

- Infrastructure Complexity
  - Increasing variety of proprietary hardware and dedicated function.
  - Current nodes are fragmented with disparate operation and management.
- Energy Consumption
  - Sites are expanding while operators and customers are being directed to reduce CO2 emissions.
- Service Deployment
  - The time to specify, procure, integrate and deploy needs to be radically reduced.
  - Increased automation of service deployment.
- Rationalisation of Operation Support Systems
  - Physical presence and consequent operations per component and site.
  - Too many disparate OSS and NMS entities in the network.

Findings – Network Functions Virtualisation

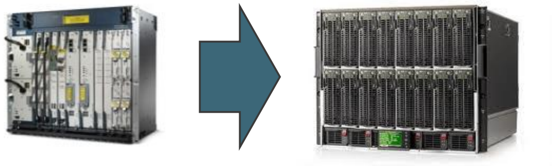
- BT Virtualisation Testing from 2012 [1]
- Combined BRAS & CDN functions on Intel® Xeon® Processor 5600 Series HP c7000 BladeSystem using Intel® 82599 10 Gigabit Ethernet Controller sidecars
    - BRAS chosen as an “acid test”
    - CDN chosen as architecturally complements BRAS
- BRAS created from scratch so minimal functionality:
    - PPPoE; only PTA, priority queuing; no RADIUS, VRFs
    - CDN COTS – fully functioning commercial product

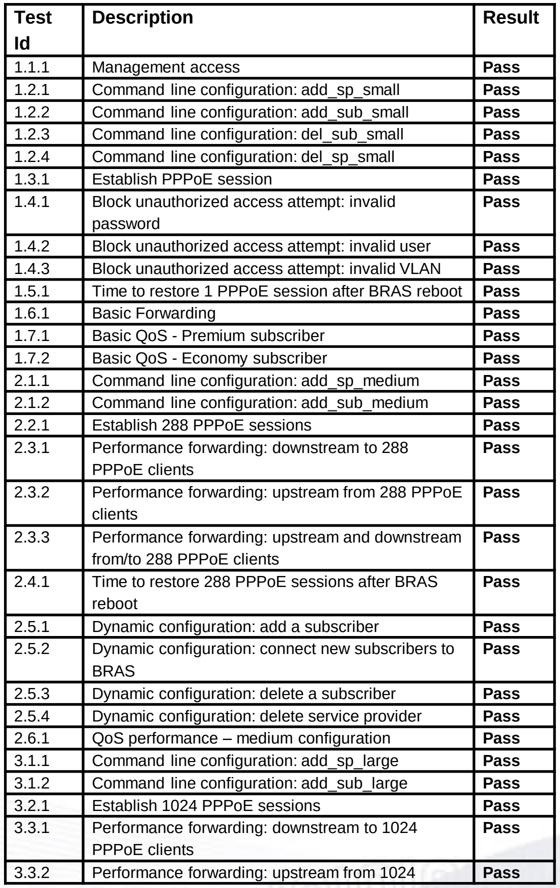
BT
hp
intel
WIND RIVER
verivue
tail-f

---

[1] Bob Briscoe, Don Clarke, Pete Willis, Andy Reid, Paul Veitch, “Network Functions Virtualisation” http://www.ietf.org/proceedings/86/slides/slides-86-sdnrg-1.pdf

Findings – Network Functions Virtualisation





Test Id	Description	Result
1.1.1	Management access	**Pass**
1.2.1	Command line configuration: add_sp_small	**Pass**
1.2.2	Command line configuration: add_sub_small	**Pass**
1.2.3	Command line configuration: del_sub_small	**Pass**
1.2.4	Command line configuration: del_sp_small	**Pass**
1.3.1	Establish PPPoE session	**Pass**
1.4.1	Block unauthorized access attempt: invalid password	**Pass**
1.4.2	Block unauthorized access attempt: invalid user	**Pass**
1.4.3	Block unauthorized access attempt: invalid VLAN	**Pass**
1.5.1	Time to restore 1 PPPoE session after BRAS reboot	**Pass**
1.6.1	Basic Forwarding	**Pass**
1.7.1	Basic QoS - Premium subscriber	**Pass**
1.7.2	Basic QoS - Economy subscriber	**Pass**
2.1.1	Command line configuration: add_sp_medium	**Pass**
2.1.2	Command line configuration: add_sub_medium	**Pass**
2.2.1	Establish 288 PPPoE sessions	**Pass**
2.3.1	Performance forwarding: downstream to 288 PPPoE clients	**Pass**
2.3.2	Performance forwarding: upstream from 288 PPPoE clients	**Pass**
2.3.3	Performance forwarding: upstream and downstream from/to 288 PPPoE clients	**Pass**
2.4.1	Time to restore 288 PPPoE sessions after BRAS reboot	**Pass**
2.5.1	Dynamic configuration: add a subscriber	**Pass**
2.5.2	Dynamic configuration: connect new subscribers to BRAS	**Pass**
2.5.3	Dynamic configuration: delete a subscriber	**Pass**
2.5.4	Dynamic configuration: delete service provider	**Pass**
2.6.1	QoS performance – medium configuration	**Pass**
3.1.1	Command line configuration: add_sp_large	**Pass**
3.1.2	Command line configuration: add_sub_large	**Pass**
3.2.1	Establish 1024 PPPoE sessions	**Pass**
3.3.1	Performance forwarding: downstream to 1024 PPPoE clients	**Pass**
3.3.2	Performance forwarding: upstream from 1024	**Pass**

- Average 3 Million Packets Per Second per Logical Core for PPPoE processing.
  - Equivalent to 94 M PPS/97 Gbps per Blade = 1.5 G PPS/1.5 Tbps per 10 U chassis[1].
  - Test used 1024 PPP sessions & strict priority QoS
  - Test used an Intel® Xeon® E5655 @ 3.0 GHz, 8 physical cores, 16 logical cores (not all used).
- Scaled to 9K PPPoE sessions per vBRAS.
  - Support of 3 vBRAS per server.
- Subsequent BT research:
  - Implemented & testing software Hierarchical QoS.
  - Results so far show processing is still not the bottleneck.
  - Also tested vCDN performance & video quality.

“Performance potential to match the performance per footprint of existing BRAS equipment.”

[1] Using128 byte packets. A single logical core handles traffic only in one direction so figures quoted are half-duplex.

[2] http://www.btplc.com/Innovation/News/NetworkVirtualization.htm

Next Steps – Management & Orchestration

- Management & Service Orchestration
  - Discovery of network resources.
  - Routing and path computation.
  - Network resource abstraction, and presentation to application layer.
  - Multi-layer coordination and interworking.
  - Multi-domain & multi-vendor network resources provisioning through different control mechanisms (e.g., Optical, OpenFlow, GMPLS, MPLS).
  - Policy Control.
  - OAM and performance monitoring.
- Leveraging existing technologies
  - What is currently available?
  - Integrate with existing and developing standards!

Next Steps – Management & Orchestration

- Application-Based Network Operations
  - A PCE-based Architecture for Application-based Network Operations
  - draft-farrkingel-pce-abno-architecture
- “Standardised” components
  - Policy Management
  - Network Topology
    - LSP-DB
    - TED
  - Path Computation and Traffic Engineering
    - PCE, PCC
      - Stateful & Stateless
      - Online & Offline
      - P2P, P2MP, MP2MP
  - Multi-layer Coordination
    - Virtual Network Topology Manager
  - Network Signaling & Programming
    - RSVP-TE
    - ForCES and OpenFlow
    - Interface to the Routing System (I2RS)

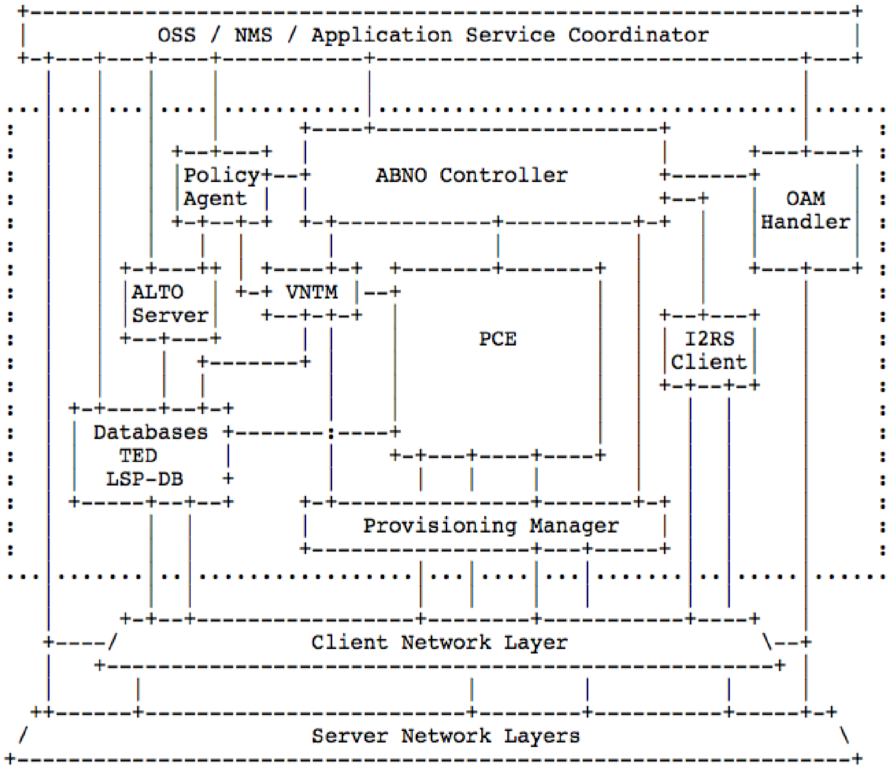


Figure 1: Generic ABNO Architecture

Next Steps – Currently

- Publish “Survey” results and findings.
- Developing orchestration and provisioning architecture and components for NFV applications
  - “Application-Based Network Operations (ABNO)” as an IETF Standard
- Documenting technical gaps for resiliency and restoration across use cases:
  - “Use cases and Requirements for Virtual Service Node Pool Management”
  - “An Overview of Reliable Service Nodes Discovery and Provision Protocols”
- Build Something!

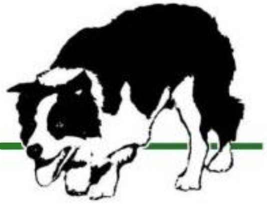
Thank You!

Any comments or questions are welcome.

Daniel King
PhD Student – Lancaster University
d.king@lancaster.ac.uk

Recent Progress in Routing Standardization

An IETF update for UKNOF 23



Old Dog Consulting

Adrian Farrel adrian@olddog.co.uk
afarrel@juniper.net
IETF Routing Area Director
Daniel King daniel@olddog.co.uk
IETF Working Group Secretary
(CCAMP, L3VPN, PCE, ROLL)

What Is Interesting and New?

- Secure Inter-domain Routing (SIDR)
  - A long-standing effort making progress
- Network Virtualization Overlays (NVO3)
  - A new working group starting to focus
- Interface to the Routing System (IRS)
  - A new proposal with a meeting planned for IETF-85 in November

SIDR

- Inter-domain routing is fragile
  - *"99% of mis-announcements are accidental originations of someone else's prefix"* – Google
  - It is possible some mis-announcements are malicious!
- SIDR aims to address
  - Is an AS authorized to originate an IP prefix?
  - Is the AS-Path represented in the route the same as the path through which the NLRI travelled?
  - Is the BGP protocol exchange secure?
- Non-goal is to prevent all malicious attacks

Resource Public Key Infrastructure (RPKI)

- Public ***and*** private key
  - Encrypt with one; decrypt with the other
- Public key issued by certifying authority
- X.509 certificates used
  - Tree of certification following address allocation
  - Address prefix is signed and announced with public key
- Route Origin Authorization
  - A signed prefix and AS number
  - Some support for aggregation
  - BGP advertisement checked against signed ROAs
- NB.Compute load much less than ACLs

SIDR Progress

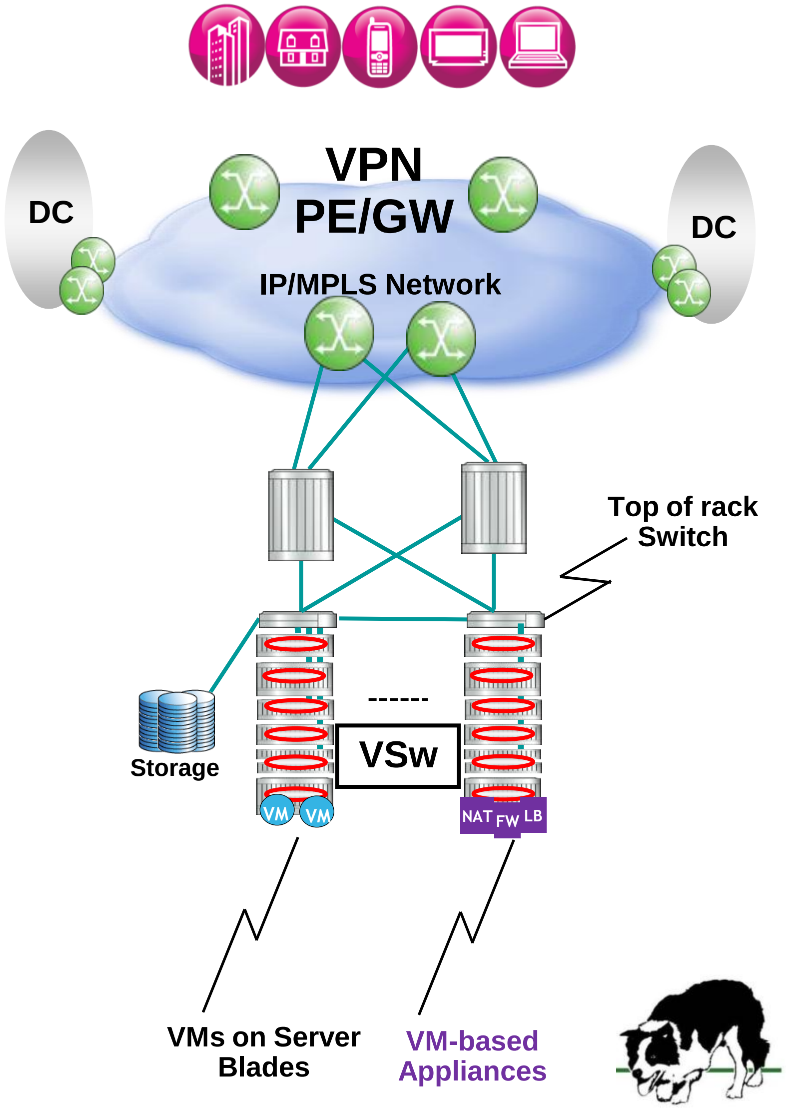
- Completed frameworks for RPKI and ROAs
- Completed core infrastructure for RPKI/ROA
- Mature/completed
  - Protocol for exchanging information between RPKI and routers
  - Advertisement validation mechanism
- Work in progress
  - Security enhancements to BGP
    - Specifically secure the AS-PATH attribute

SIDR References

- SIDR Working Group
  http://datatracker.ietf.org/wg/sidr/charter/
- RFC 6480
  An Infrastructure to Support Secure Internet Routing
  http://datatracker.ietf.org/doc/rfc6480/
- Endless presentations at nanog and ripe
  - http://www.nanog.org/presentations/archive/index.php
    - Search for SIDR
  - https://ripe64.ripe.net/programme/meeting-plan/tutorials/

Multi-tenant DC Networking

- Gateway to the outside world.
- DC Interconnect and connectivity to Internet and VPN customers.
- High capacity core node, usually a cost effective Ethernet switch; may support routing capabilities.
- Top of Rack (ToR) hardware-based Ethernet switch; may perform IP routing.
- Virtual Switch (VSw) software based Ethernet switch running inside the server blades.



NVO3 Overview

- Multi-tenancy has become a core requirement of data centers
  - Including for Virtualized Machines (VMs) and VM multi-tenancy
- Three key requirements needed to support multi-tenancy are
  - Traffic isolation
  - Address independence
  - Fully flexible VM placement and migration
- NVO3 WG considers approaches to multi-tenancy that reside at the network layer rather than using traditional isolation (e.g., VLANs)
  - An overlay model to interconnect VMs distributed across a data center
- NVO3 WG will determine which types of connectivity services. are needed by typical DC deployments (for example, IP and/or Ethernet)
- NV03 WG Will Not develop service provider solutions for wide-area interconnect of data centers

NVO3 WG Progress

- NVO3 Working Group
  - First meeting IETF-84 July 2012
  - http://datatracker.ietf.org/wg/nvo3/charter/

- Problem Statement: Overlays for Network Virtualization
  - Describes issues associated with providing multi-tenancy that require an overlay-based network virtualization approach to addressing them
  - Adopted by working group September 2012
  - http://tools.ietf.org/html/draft-ietf-nvo3-overlay-problem-statement

- Framework for DC Network Virtualization
  - Provides a framework for NVO3. It defines a logical view of the main components with the intention of streamlining terminology and focusing the solution set
  - Adopted by working group September 2012
  - http://tools.ietf.org/html/draft-ietf-nvo3-framework-00

NVO3 has loads of buzz

- Internet-Drafts include:
  - Data and Control Plane Requirements
  - Framework
    - Overlay Architecture
    - Addressing
  - Use Cases
    - VPN Applicability
    - Mobility Issues
  - Operational Requirements
  - Security Framework

Related Active Documents (not working group documents):

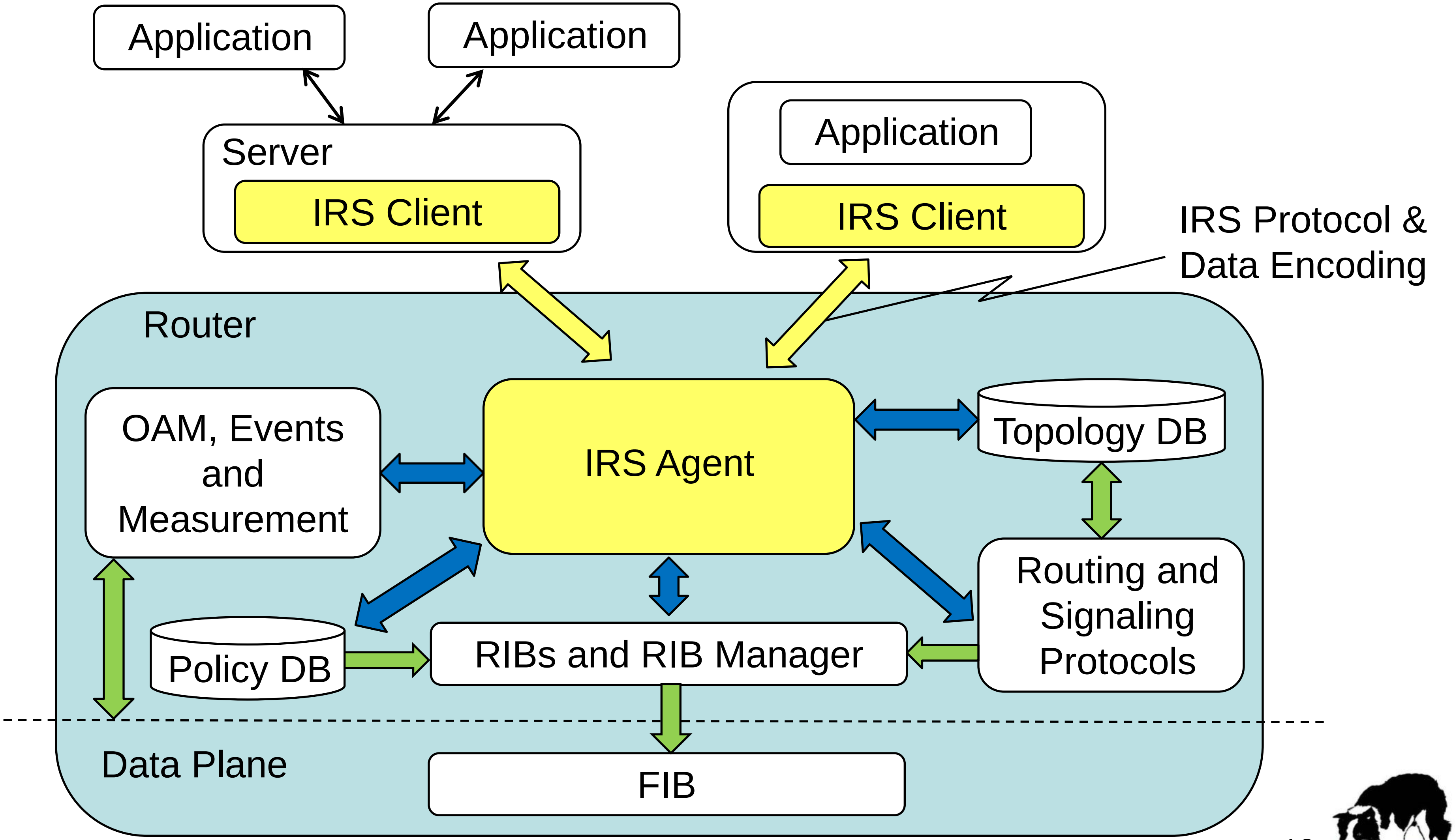
(To see all nvo3-related documents, go to nvo3-related drafts in the ID-archive)

draft-ashwood-nvo3-operational-requirement	-00	2012-06-14
draft-bitar-nvo3-vpn-applicability	-00	2012-08-30
draft-bl-nvo3-dataplane-requirements	-01	2012-06-26
draft-carpenter-nvo3-addressing	-00	2012-07-05
draft-drake-nvo3-evpn-control-plane	-00	2012-09-17
draft-dunbar-nvo3-overlay-mobility-issues	-00	2012-06-28
draft-gu-nvo3-overlay-cp-arch	-00	2012-07-09
draft-gu-nvo3-tes-nve-mechanism	-00	2012-07-06
draft-hy-nvo3-vpn-protocol-gap-analysis	-01	2012-09-10
draft-kj-nvo3-encapsulation-reqt	-00	2012-09-25
draft-kj-nvo3-pion-architecture	-00	2012-05-11
draft-kompella-nvo3-server2nve	-00	2012-07-09
draft-kreeger-nvo3-overlay-cp	-01	2012-07-16
draft-maino-nvo3-lisp-cp	-01	2012-09-20
draft-mity-nvo3-use-case	-03	2012-08-30
draft-rekhter-nvo3-vm-mobility-issues	-02	2012-09-27
draft-wei-nvo3-security-framework	-01	2012-07-16
draft-xu-nvo3-lan-extension-path-optimization	-00	2012-07-09

IRS

- Configuration access to routers tends to be
  - Non-dynamic
  - Granular
  - Non-standard
- Existing programmatic interfaces target
  - Data plane
  - FIB
- Need a way to provide high-level input to routing and to extract data
  - Make entries in RIBs
  - Control routing protocols
  - Set policies
    - For policy-based routing QoS, OAM, etc.
    - Security, firewalls, etc.
    - Route import/export
  - Read topology and routing information

IRS Framework



Questions to Be Answered

- What is an IRS Application?
- How does IRS interact with Configuration?
- Are there already existing protocols and encoding languages?
- How does this relate to OpenFlow?
- What’s it all for?

IRS Use Cases

- Core routing system manipulation
  - Injection of static routes
  - Control of RIB-to-FIB policy
  - Extraction of RIBs and other data
- Topology manipulation
  - Extraction of topology and traffic engineering info
  - Creation of virtual links and tunnels
- BGP policy
  - Import and export policies
  - Route reflector control
  - Flowspec definition and configuration
- Firewalls
  - Injection of policies

IRS Plans

- Post some Internet-Drafts and discuss the idea
- BoF meeting IETF-85 in Atlanta (November)
  - Assess level of focus and support
- Maybe form a working group
  - Start with framework, use cases, requirements
  - Write ***abstract*** information models
  - Continue to evaluate existing protocols and encoding languages
  - Maybe develop new protocols/languages
  - Write data models

IRS References

- IETF-85 BoF Proposals
  http://trac.tools.ietf.org/bof/trac/
- IRS discussion mailing list
  http://www.ietf.org/mailman/listinfo/irs-discuss
- IRS Problem Statement
  http://datatracker.ietf.org/doc/draft-atlas-irs-problem-statement/
- IRS Framework
  http://datatracker.ietf.org/doc/draft-ward-irs-framework/

Adaptive Network Manager: Coordinating Operations in Flex-Grid Networks

Victor Lopez[1], **Ori Gerstel**[2], **Ramón Casellas**[3], **Adrian Farrel**[4], **Daniel King**[4], **Sergio López-Buedo**[5], **Antonio Cimmino**[6], **Roberto Morro**[7] **and Juan Fernandez-Palacios**[1]
[1]Telefónica I+D, Calle Emilio Vargas Nº6,Madrid, 28043, Spain
Tel: (34) 913128872, e-mail: vlopez@tid.es
[2]Cisco Systems Inc., Israel
[3]CTTC, Av. Carl Friedrich Gauss 7, Castelldefels, 08860, Spain
[4]Old dog Consulting, Graythwaite, LA12 8BA, UK
[5]NAUDIT HPCN, c/ Faraday 7, Madrid, 28049, Spain
[6]Alcatel Lucent Italia, Via Bosco Primo, Battipaglia, Italy
[7]Telecom Italia, Via Reiss Romoli 274, Torino, 10148, Italy

ABSTRACT

Transport networks provide reliable delivery of data between two end points. Today's most advanced transport networks are based on Wavelength Switching Optical Networks (WSON) and offer connections of 10Gbps up to 100Gbps. However, a significant disadvantage of WSON is the rigid bandwidth granularity because only single, large chunks of bandwidth can be assigned matching the available fixed wavelengths resulting in considerable waste of network resources. Elastic Optical Networks (EON) provides spectrum-efficient and scalable transport by introducing flexible granular grooming in the optical frequency domain. EON provides arbitrary contiguous concatenation of optical spectrum that allows creation of custom-sized bandwidth. The allocation is performed according to the traffic volume or user request in a highly spectrum-efficient and scalable manner.

The Adaptive Network Manager (ANM) concept appears as a necessity for operators to dynamically configure their infrastructure based on user requirements and network conditions. This work introduces the ANM and defines ANM use cases, and its requirements, and proposes an architecture for ANM that is aligned with solutions being developed by the industry.

Keywords: Elastic optical networks, control plane, network automation, multi-layer.

1. INTRODUCTION

Transport networks provide reliable delivery of data between two end points. Elastic Optical Networks (EON) offers a scalable solution for transport networks thanks to the introduction of spectral adaptation in the optical frequency domain [1]. EON provides arbitrary contiguous concatenation of optical spectrum that allows creation of custom-sized bandwidth. This bandwidth is defined in slots of 12.5 GHz. EON allows allocating appropriate-sized, as opposed to fixed-sized, optical bandwidth to an end-to-end optical path. The allocation is performed according to the traffic volume or user request in a highly spectrum-efficient and scalable manner.

The existing transport network architectures were conceived and designed having in mind both the characteristics and the traffic demands of the classic services (e.g. Internet access or VPNs), which are predictable. Traditional carriers' networks operation is very complex and is neither readily adaptable nor programmable to flexible traffic requirements. Multiple manual configuration actions are needed in metro and core network nodes (e.g. hundreds of thousands of nodes configurations per year in mid-size network operators). Furthermore, network solutions from different vendors typically use vendor-specific Network Management System (NMS) implementations. Such complex architecture (depicted in Fig. 1) derives in complex and long workflows for network provisioning (e.g. up to two weeks for Internet service provisioning and more than six weeks for core routers connectivity services over photonic mesh).

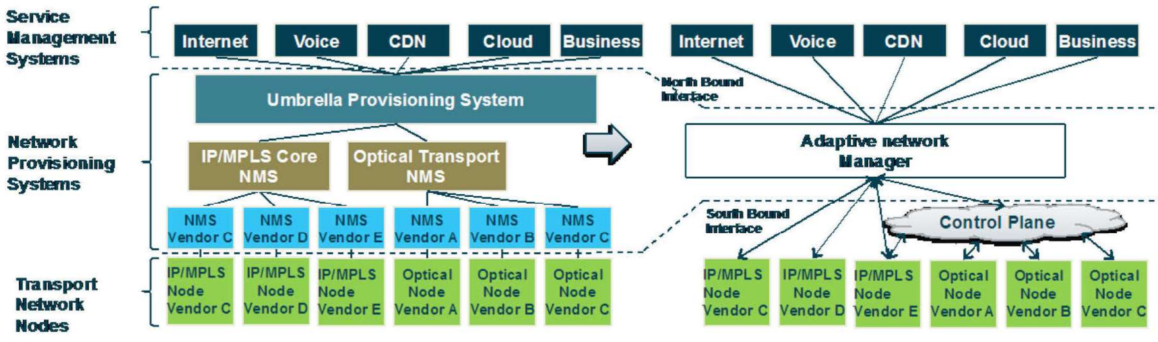


Figure 1. Evolution towards an Adaptive Network Manager.

978-1-4799-0683-3/13/$31.00 ©2013 IEEE

There is a number of problems with the current transport network provisioning approach. First, the interfaces between the service management systems and the umbrella provisioning system are typically proprietary, non-programmable and closed interfaces that prevent new applications from a rapid and automated introduction. Second, the orchestration capabilities across different NMSes (e.g., IP/MPLS NMS and Optical Transport NMS) are very difficult to achieve as each NMS is a highly specialized vendor element that lacks interoperability with other vendors' elements especially on the NMS to NMS communication. Third, there is little standardization on interface for upper layer applications or services. With the current approach, it is not easy to provide an abstracted topology view or service-specific view of the network to the application in a fairly generic fashion, or to allow application to request and/or control virtual network resources.

ANM proposed in IDEALIST project should improve provisioning process of legacy NMSs (Fig. 1). Current approach does not allow a common interface to support deploying multiple services. ANM architecture would require a network-service interface, which is a common standard interface for multiservice provisioning. On the other hand, NMS has multiple vendor-specific interfaces, which creates great problems in terms of tools integration. ANM would use standard network configuration interfaces, which will trigger automated standard control plane for multidomain/vendor/layer operation. Key building blocks of such unified network provisioning architecture are: (1) network elements interface must be standard, (2) service layer and network coordination is required, (3) common Network-Service interface enabling a common entry point to provision multiple services.

ANM enables the dynamic and automated control of server layer (EON) transport resources. However, based on Fig. 1, ANM looks like a black box with multiple functionalities inside of it. Within IDEALIST project, the architecture of ANM will be defined and standardized in multiple boxes with defined standard interfaces.

2. ADAPTIVE NETWORK MANAGER

Adaptive Network Manager (ANM) monitors network resources, and decides the optimal network configuration based on the status, bandwidth availability and user service. It is important that an ANM provides a set of standard interfaces, which facilitates communication with other network elements and key network components. These components include the Operation Support Systems (OSS), Network Management Systems (NMS) or Path Computation Elements (PCE), to provide additional capabilities, including automated network configuration and resource optimization. The main task of ANM is to coordinate, or orchestrate, network procedures based on received requests. ANM starts processes after receiving triggers from the operator via NMS, failures, measurements or periodical requests. After a trigger is received, ANM process it and starts a workflow or queues it for later analysis. Once a workflow is run, ANM can return the answer to the operator so network configuration can be accepted, rejected or modified. There are other workflows that do not require human involvement. Finally, ANM can be focus just on elastic optical networks or it can take into account the impact of client layers like IP/MPLS. Table 1 shows a classification for the different scenarios where ANM operates.

Table 1: Classification for the different ANM scenarios.

Triggers	Processing Triggers	Human involvement	Network Scope
• Human • Failure • Measurement • Periodic	• Start process • Queue for correlation	• Automatic Configuration • User Assisted Configuration	• Single Layer • Multi Layer

3. USE CASES

3.1 Automatic IP Link provisioning

The first use case describes how the ANM framework can be applied to the provisioning of an IP link between two routers. In this example, the photonic meshed network is composed of (elastic) ROADMs providing connectivity to several IP routers.

IP link provisioning is a basic operation done by network operators. This operation is used to provide customer services, including Internet connectivity, VPN or IPTV. When operators deploy additional capacity, new IP link equipment may be installed in the network. This process typically requires manual intervention and is scheduled and deployed periodically. Once equipment is installed in the network and operator receives a request to create a new IP link between two locations, there is a dialog between the IP and transport department to complete the configuration of both layers. This configuration process may take days to complete, even when network elements are already set-up in the network. ANM is intended to automate the configuration process, and in specific cases dynamically, by utilizing control plane technologies, and using an interface to configure IP routers (like OpenFlow or NetConf) to configure individual network elements. Also, the optical layer can be directly configured from the router using either User Network Interface (UNI) or PCE Protocol (PCEP) to trigger control plane mechanisms [6].

3.2 Dynamic Bandwidth Allocation based on traffic changes

Current network provisioning of packets over circuits is done in a static manner. Network operators are willing to provide services to end-users (Internet access, VPN, etc.). In aggregation networks, traffic from multiple sources is multiplexed so large traffic streams are sent to backbone networks. There are monitoring probes in the network, which provide periodical information to network operators, but modifications of circuits is not done. Typically new connections are created yearly or at specific time intervals (six months) in the network.

ANM can deal with this dynamic information and decide on the bandwidth adaptation of the connections thanks to the elasticity of BVT. ANM requires retrieve information from routers (such as SNMP) or monitoring probes depending on the traffic patterns in the network. Based on this information, ANM would decide modifications in the parameters of the connections and apply changes to the configuration of the router or BVT. This use case is shown in Fig. 2.

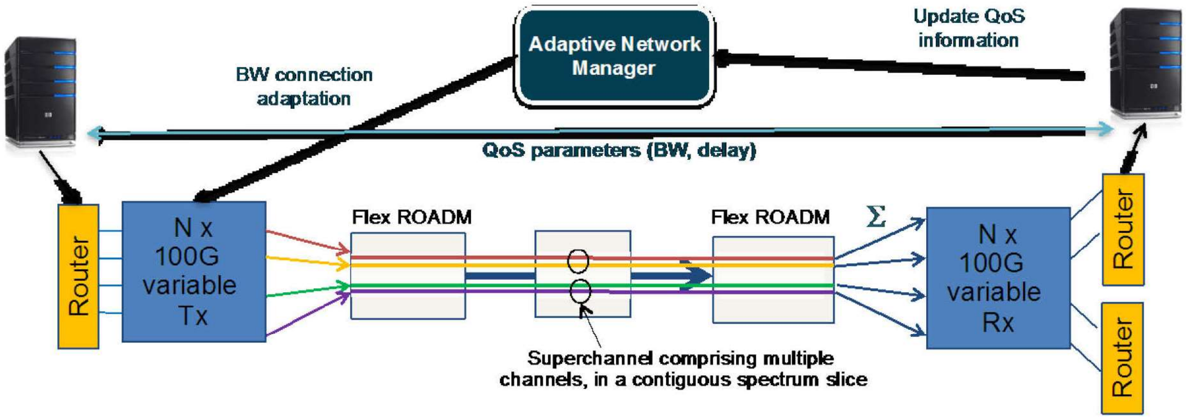


Figure 2. Example scenario for dynamic bandwidth allocation use case.

Previous example focused on the parameter modification of an already established link. Another scenario is the case of the creation of a by-pass link when all the existing bandwidth on an intermediate link between two routers has already been entirely used up (or crosses a pre-defined threshold). Based on monitoring information, ANM would start an Automated IP Link Provisioning workflow as defined in previous section. If there are Sliceable BVTs (SBVT) in the network, ANM can split the interface's bandwidth in two (or more) fragments, reducing the bandwidth of the original connection (the one to the next IP hop) and using the new available bandwidth for a new direct connection to the destination router.

3.3 Periodic defragmentation to improve bandwidth allocation

The reoptimization (defragmentation) process is roughly defined as the process by which an ANM affects the state of currently active connections in the network by changing some of their attributes. Such attributes typically correspond to the actual reserved resources and changing them may involve, for example, shifting the nominal central frequency of the frequency slot allocated to a connection and/or adjusting its allocated frequency slot width (i.e., due to a change of modulation formats or bitrate) or even the physical routes that were assigned to the connections during path computation. In general, the main purpose of the reoptimization process is to improve the utilization of the network resources, since the main observable result is a sub-optimal throughput. This process can be triggered either manually by a network operator or based on automated maintenance process.

3.4 Network reoptimization after network failure recovery

In optical transport networks, operators are commonly required to deploy some form of resilience when transporting client data. Such resilience can be implemented by means of either dynamic restoration of failed connections (i.e., a new path is computed and established after a failure is localized) or dedicated/shared protection by establishing at the same time, e.g., for a given traffic demand, the corresponding working and backup paths.

In both cases, if network connections are flagged with elasticity (i.e., their properties and attributes can be dynamically adjusted) of the physical path, bitrate or modulation format, such elasticity can be exploited to improve the network survivability by dynamically adapting those attributes to the network state. As there are dynamic control plane mechanisms, which run after each failure, they can lead to an inefficient network configuration. Hence, after multiple failures, ANM can check using an algorithm in a PCE or an external tool if current network configuration is optimal or not. Based on this information, ANM can alert operator, who decide if this new configuration should be loaded in the network.

3.5 Multi-layer restoration

Multi-layer restoration is the process of restoring a fail of any element in the IP/MPLS or optical layer between two client nodes in a coordinated manner. Unlike single layer restoration (i.e., pure optical restoration), the multi-layer restoration process involves the negotiation of the best possible path properties between the optical layer and the IP/MPLS layer, given a failure in the network. There are two scenarios where coordination is beneficial: failure in the optical layer or failure in the IP/MPLS layer.

Existing approaches to optical restoration do not focus on the constraints that must be met for the restoration path. Often these approaches implicitly assume that any viable restoration path is good. This is not a valid assumption in the event the failure takes a long time to repair since the client layer must return to a relatively normal state. Therefore the most optimal approach is to allow the client to define different constraints for the restoration path versus the constraints that have been defined for the working path. With this negotiation between layers, it is possible to dynamically adapt to the requirements of the client layer.

The second scenario where multi-layer restoration can be interesting is when there is a failure in the IP layer. In case there is a failure on a router, ANM can look for a candidate back-up router at any location of the network, because there is an underlying optical layer. Once a suitable path is found, ANM start the Automated IP Link Provisioning use case.

4. REQUIREMENTS

ANM enables the dynamic and automated control of server layer (EON) transport resources. However, based on Fig. 1, ANM looks like a black box with multiple functionalities inside of it. ANM must have enough functionalities to cover use cases defined in previous sections. Figure 3 shows the functional blocks identified in the IDEALIST project. Each of the building blocks will be assessed during the project so a proof-of-concept will be done at the end of the project.

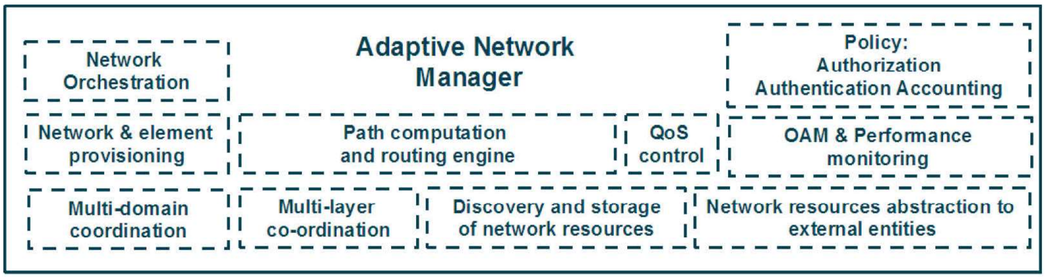


Figure 3. ANM functional building blocks.

One of the key issues in ANM is the utilization of standard technologies, so ANM can operate in existing networks. From this perspective, there are three architectures related to concepts presented in the ANM: Active PCE [2], which is capable of set-up and tear down LSPs, SDN controller [3], which is defined mainly for OpenFlow controlled network elements and Application-Based Network Operations (ABNO) controller, recently proposed in IETF [4]. These architectures will be assessed when defining the functional blocks of ANM.

5. CONCLUSIONS

This paper presents the definition of Adaptive Network Manager (ANM), its use cases and the requirements in terms of functional blocks identified in the project. The Adaptive Network Manager (ANM) concept appears as a necessity for operators to dynamically configure their infrastructure based on user requirements and network conditions. Three architectures may fit with ANM requirements, but they will be evaluated as future work.

ACKNOWLEDGEMENTS

This work was partially funded by the European Community's Seventh Framework Programme FP7/2007-2013 through the Integrated Project (IP) IDEALIST under grant agreement nº 317999.

REFERENCES

[1] O. Gerstel *et al.*, "Elastic optical networking: A new dawn for the optical layer?", *IEEE Communications Magazine*, vol. 50, no. 2, pp. s12-s20, Feb. 2012.
[2] E. Crabbe *et al.*, "PCEP Extensions for Stateful PCE", IETF draft, draft-ietf-pce-stateful-pce-02.
[3] Ping Pan, "Efficient inter-data center transport within SDN framework", in *Proc. iPOP 2012*.
[4] D. King and A. Farrel, "A PCE-based Architecture for Application-based Network Operations", IETF Draft, draft-farrkingel-pce-abno-architecture-02.

Unification of Formal and De Facto Standards for Abstraction and Autonomic Control of the Transport Network

Daniel King - d.king@lancaster.ac.uk
Senior Researcher, Lancaster University (British Telecom & Intel Co-Lab)
Research Associate at the Open Networking Foundation (ONF)
Co-Chair Simplified Use Policy Abstractions (SUPA) Working Group, IETF
Co-Chair Software Defined Networks (SDN) Research Group, IRTF

UK EPSRC-funded Project
” TOUCAN”

- A UK Funded project
  - £6M from the UK Research Council
  - £6M from industry partners
  - Duration is 5 years from August 2014
- Towards Ultimate Network Convergence (TOUCAN)
  - Define technology agnostic architecture for convergence based on SDN & NFV primitives
  - Facilitate optimal interconnection of any transport technology domains, networked devices and data sets with high flexibility, resource and energy efficiency
- Industry partners, include:

– BCC, Broadcom, BT, Janet, NEC, Innovate UK, Plextek, Samsung

TOUCAN
ULTIMATE NETWORK CONVERGENCE

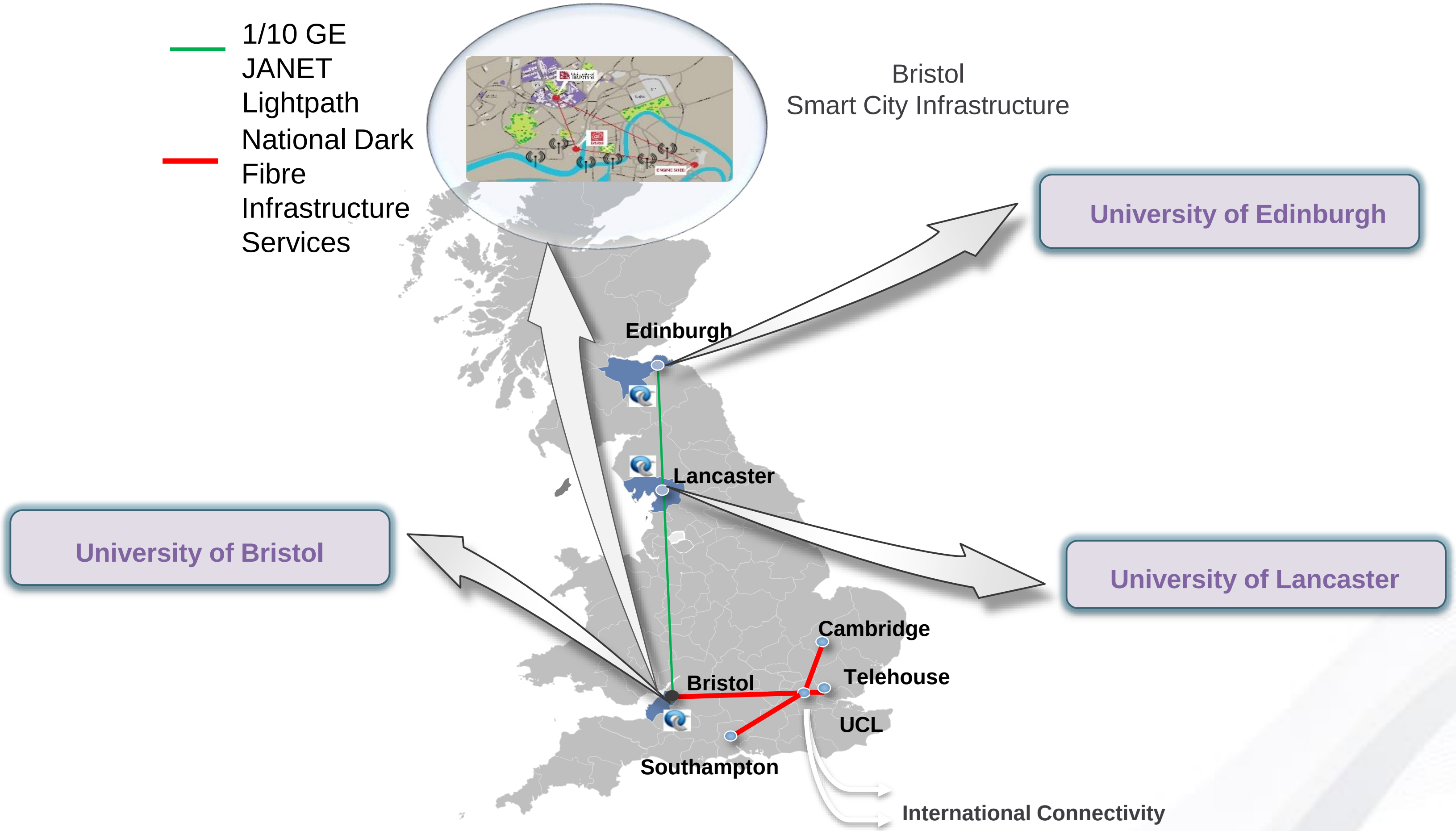
Lancaster University

HERIOT WATT UNIVERSITY

THE UNIVERSITY OF EDINBURGH

University of BRISTOL

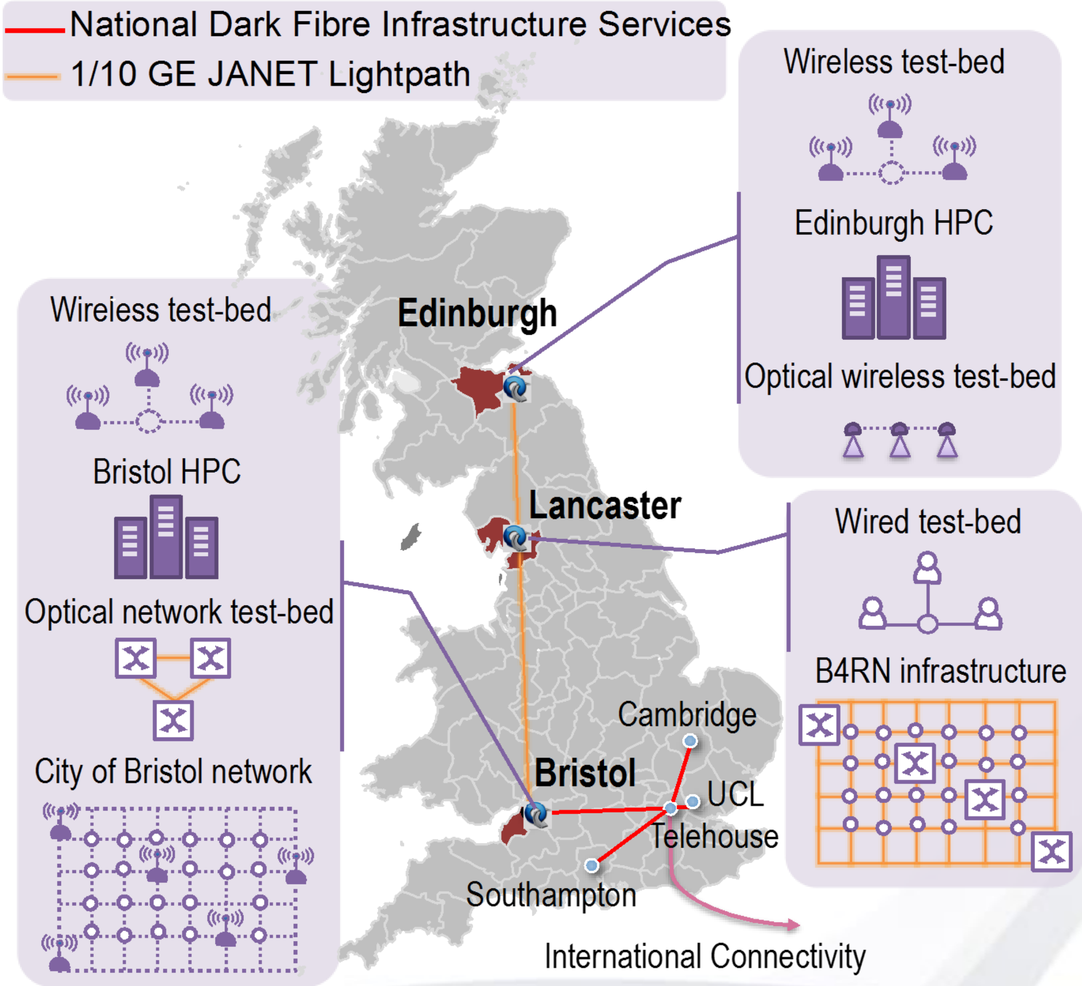
TOUCAN Network
Transport Infrastructure



- TOUCAN Transport network
  - JANET Lightpaths and Dark Fibre
  - Multi-Datacenter interconnections
  - 1/10/40 Gigabyte transport infrastructure
  - Regional connectivity
  - Ongoing deployment of evolving transport technologies

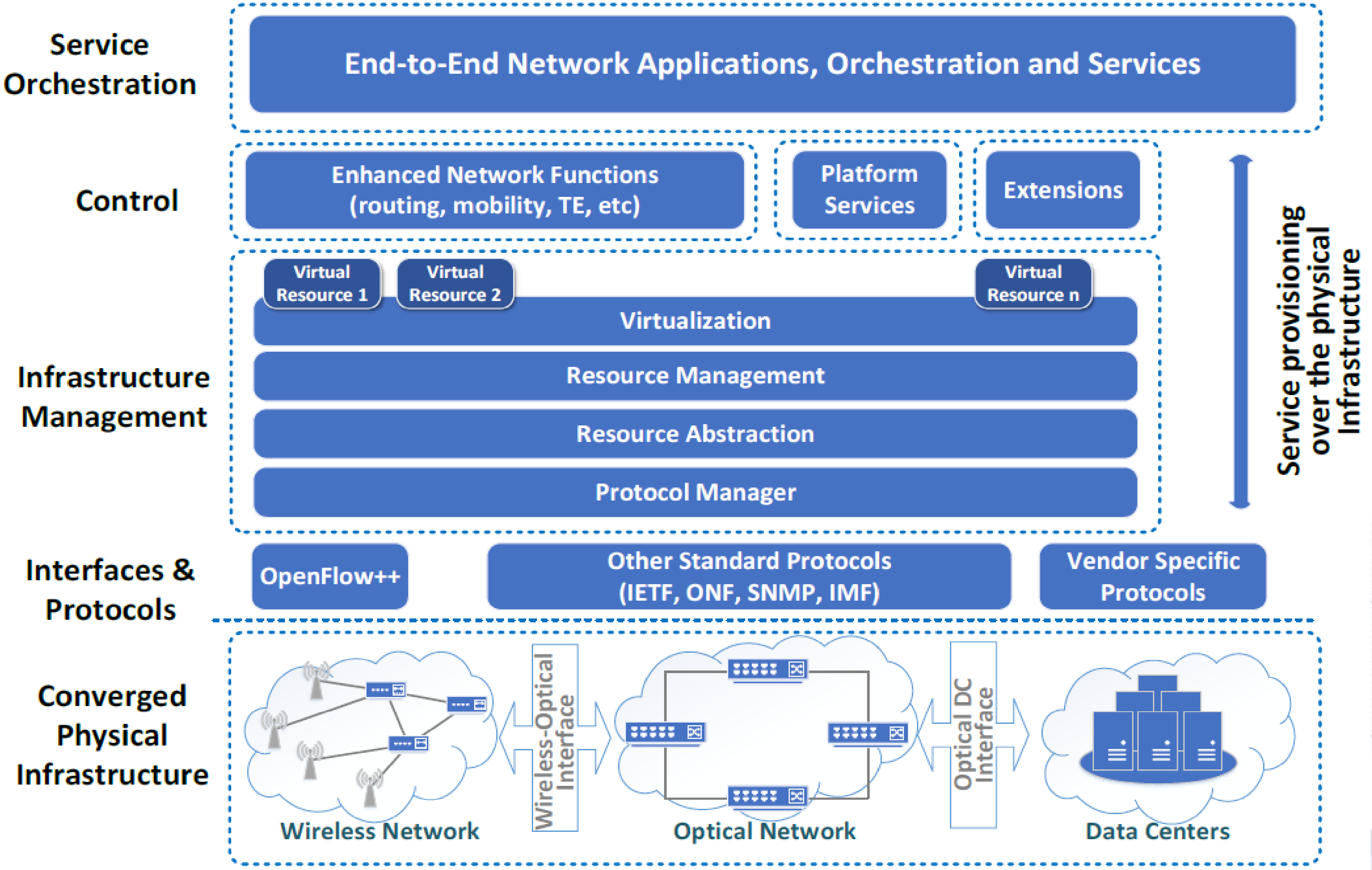
TOUCAN Applications

Testbeds & Services



- Wide range of end-devices and applications
  - Smart city infrastructure
  - Fiber-based broadband
  - Cloud applications
  - NFV-based Services
  - Academic campus and testbed connectivity and experimentation

TOUCAN High-level Architecture
Functional Components and Relationships



TOUCAN Interoperability Challenges

A need for “Open Standards” and applied solutions

- How to ensure interoperability within TOUCAN, and beyond the project?
  - Software has come to dominate what we perceive as "the Internet" and the "agile” development model has created an exponential curve in the rate of innovation
- Standards Development Organisations (SDOs)
  - We found SDOs appear incapable of defining and maintaining their boundaries, and new technology study groups are exploding across them
  - Most organisations are self-perpetuating
- Code is King
  - Although code is “coin of the realm” in Open Source Software (OSS) projects, code is not always normative
- Conclusion? We still need Standards, but they must be applied!
  - A question of relevance and NOT existence
  - More coordination between SDOs, avoiding dilution of effort and resources, and CONFUSION

TOUCAN Transport Infrastructure Objectives
Abstraction and Control of Transport Networks (ACTN)

- TOUCAN Transport network control goals
  - Facilitate seamless interconnection, abstraction and slicing, of transport network technology domains
  - Extreme flexibility in data throughput, high adaptability, and underlying transport resource efficiency
  - Enable seamless application-level infrastructure programmability via automated APIs
- ACTN Design principles
  - Agnostic Resource Sharing
    - Efficient resource sharing for multiple underlying forwarding and function technologies
  - Programmability
    - Pragmatic approach to repurpose existing and well-defined technologies, and underpinning them with SDN principles
  - Automation
    - Enables heterogeneous transport domain networking, management technologies (e.g., GMPLS, ASON, PCE, NMS/EMS,) while allowing logically centralised control and orchestration of resources
  - Slicing
    - Virtual network automation using abstraction, slicing and in-operation optimisation, of underlying resources for higher-layer services, independent of how the underlay domain resources are managed or controlled

Ensuring TOUCAN Platform Interoperability

ACTN Building Blocks

Resource Descriptions

WSON YANG

FLEXI YANG

LIFI YANG

Resource Discovery & Abstraction

PCEP-LS

BGP-LS

Rest/YANG

Controller Hierarchy

Stateful Hierarchical PCE

Virtual Network Control

PCEP VN Association

Controller State Synchronization

PCEP State Synchronization

Resource Models for TOUCAN

Optical Resource Modeling

- Objectives
  - To provide automated interfaces (models) of optical and transport resources to controllers and orchestration layers (including the TOUCAN platform)

– Effort so far

  - Define requirements of TOUCAN architecture for optical transport resource modeling
  - Survey of existing work in IETF and other industry groups for transport service modeling
  - Coordinate proposals with leading vendors to adopt ideas and suggestions from TOUCAN into IETF for industry standardisation
  - Proposed a new service model for connection-orientated SDN transport, being discussed in the Traffic Engineering and Signaling (TEAS) Working Group

– Success so far

  - Proposed a new data model for WSON, which has been accepted by the IETF
    - Wavelength Selective Optical Networking YANG Model https://tools.ietf.org/html/draft-ietf-ccamp-wson-yang
  - Proposed a new data mode for Flexi-Grid, under consideration by the IETF
    - Flexi-Grid YANG Model https://tools.ietf.org/html/draft-vergara-ccamp-flexigrid-yang

Ensuring TOUCAN Platform Interoperability
Abstraction and Control of Transport Networks (ACTN)

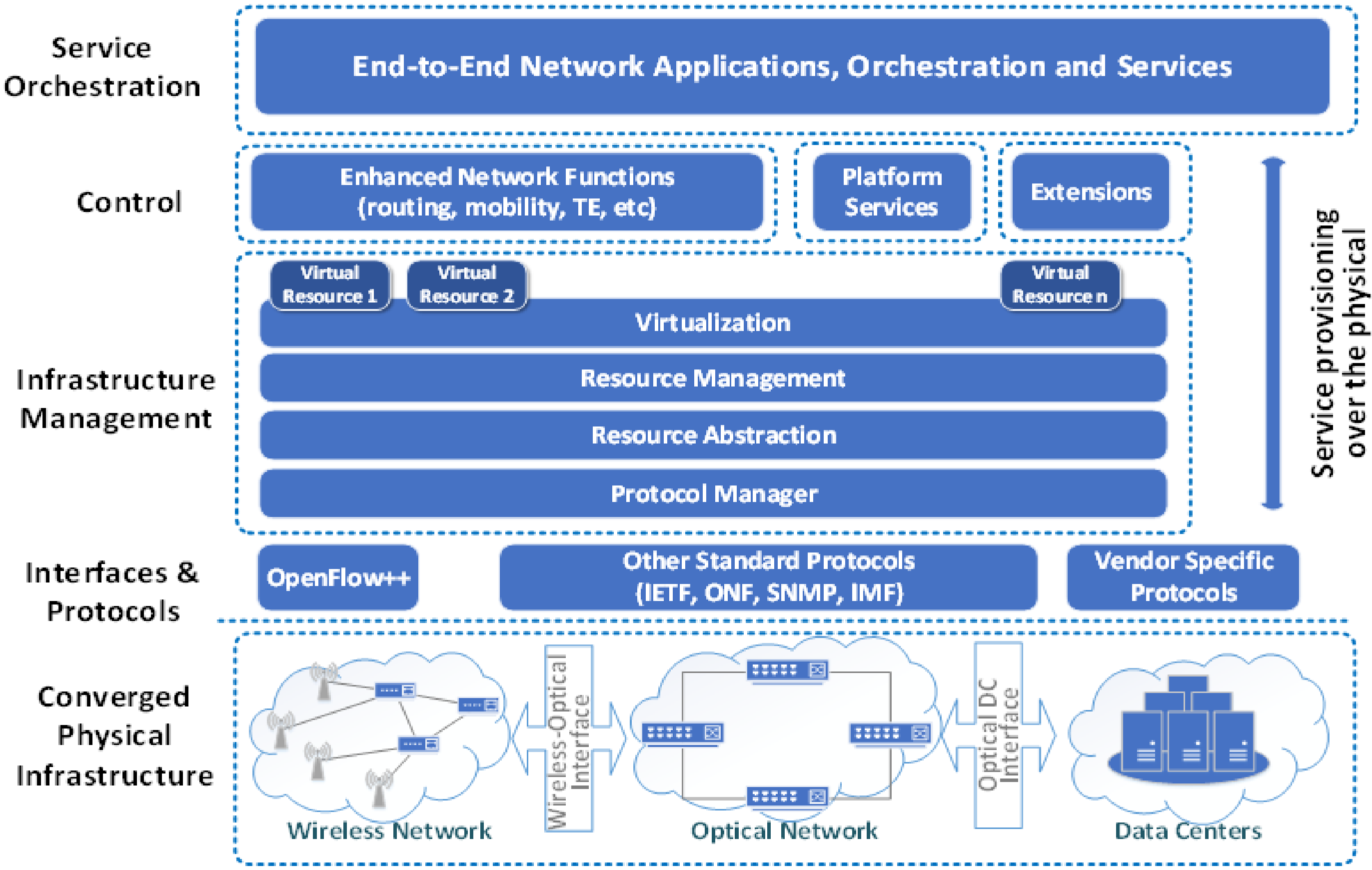
- Objectives
  - To take principles and ideas from TOUCAN and coordinate within industry to facilitate virtual network operation, creation of a virtualized environment allowing network operators to view, control, and partition, multi-domain networks
  - As transport networks evolve, the need to provide network abstraction has emerged as a key requirement for operators, underlying the industry impact of TOUCAN research objectives
- Effort so far
  - Developed a problem Statement for Abstraction and Control of Transport Networks tools.ietf.org/html/draft-leeking-actn-problem-statement
  - Agree a framework for Abstraction and Control of Transport Networks with industry technology leaders: Young Lee (**Huawei**), Daniele Ceccarelli (**Ericsson**), Daniel King (**University of Lancaster**), Sergio Belotti (**Alcatel-Lucent)**, Luyuan Fang (**Microsoft**), Dhruv Dhody (**Huawei**), Diego Lopez (**Telefonica**), Gert Grammel (**Juniper**) tools.ietf.org/html/draft-ceccarelli-actn-framework
- Success so far
  - ACTN Framework proposal has been accepted by the IETF Traffic Engineering And Signaling (TEAS) working group, providing a cornerstone for convergence of transport networks

Northbound Interface for TOUCAN

Service Modeling

- Objectives
  - Using to TOUCAN investigations and findings to facilitate standardisation of the Northbound Interface from the Controller to the Orchestrator
- Effort so far
  - Survey, Requirements and Functions of YANG Models for the Northbound Interface of a Transport Network Controller https://tools.ietf.org/html/draft-zhang-ccamp-transport-yang-gap-analysis
  - https://tools.ietf.org/html/draft-zhang-ccamp-transport-ctrlnorth-yang
  - Proposed a YANG Model for Connection-oriented Transport Services https://tools.ietf.org/html/draft-zhang-teas-transport-service-model
- Success so far
  - No proposal has yet be formally adopted by the working group but we expect to make progress at IETF 97 (November)

TOUCAN Architecture
Data Models and Info Models



IETF Transport Service Model

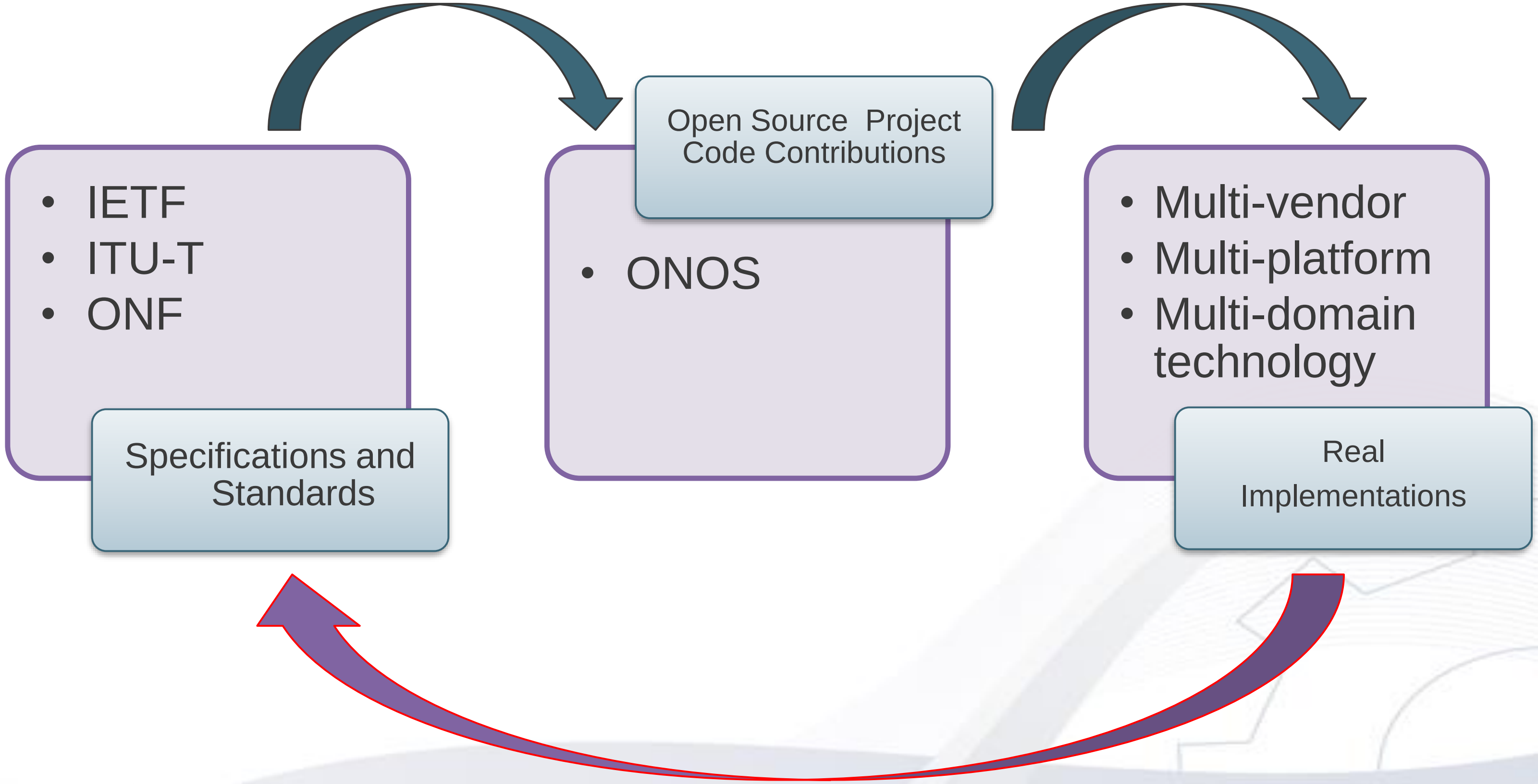
IETF Abstraction and Control of Transport Network (ACTN) Architecture

IEEE LiFi Data Plane Specification

IETF WSON YANG Model
IETF Flexi-Grid YANG Model

Ensuring TOUCAN Platform Interoperability
Abstraction and Control of Transport Networks (ACTN)

- Blending: Standards, Open Source and Interoperability Testing
  - Creating a feedback loop for development and deployment



- IETF
- ITU-T
- ONF

Specifications and Standards

Open Source Project Code Contributions

- ONOS

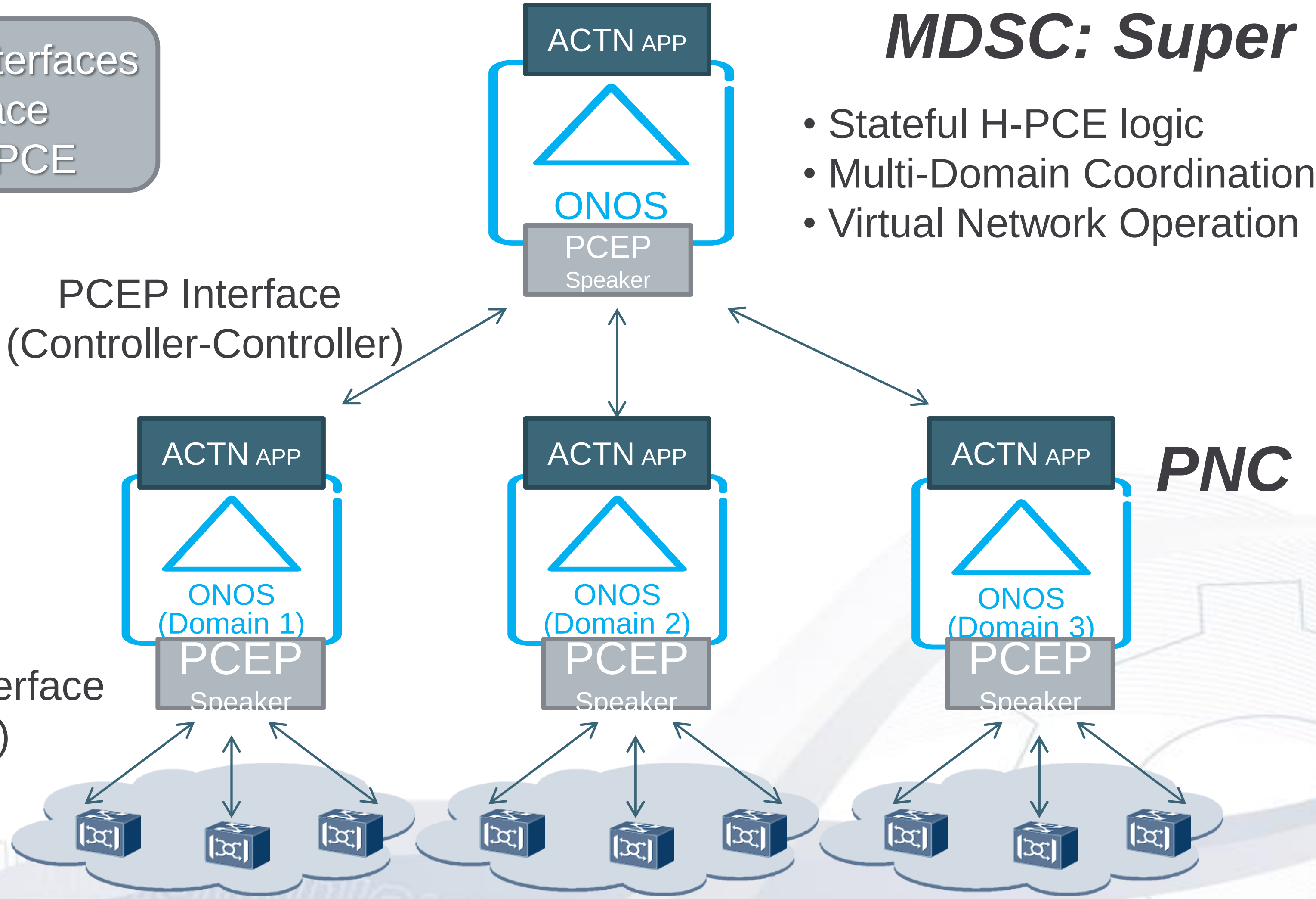
- Multi-vendor
- Multi-platform
- Multi-domain technology

Real Implementations

ACTN Building Blocks

ONOS PCEP Implementation

- Recursive Interfaces
- Binary interface
- Hierarchical PCE



MDSC: Super Controller

- Stateful H-PCE logic
- Multi-Domain Coordination
- Virtual Network Operation

ACTN APP

ONOS

PCEP Speaker

PCEP Interface
(Controller-Controller)

ACTN APP

ONOS (Domain 1)

PCEP Speaker

ACTN APP

ONOS (Domain 2)

PCEP Speaker

ACTN APP

ONOS (Domain 3)

PCEP Speaker

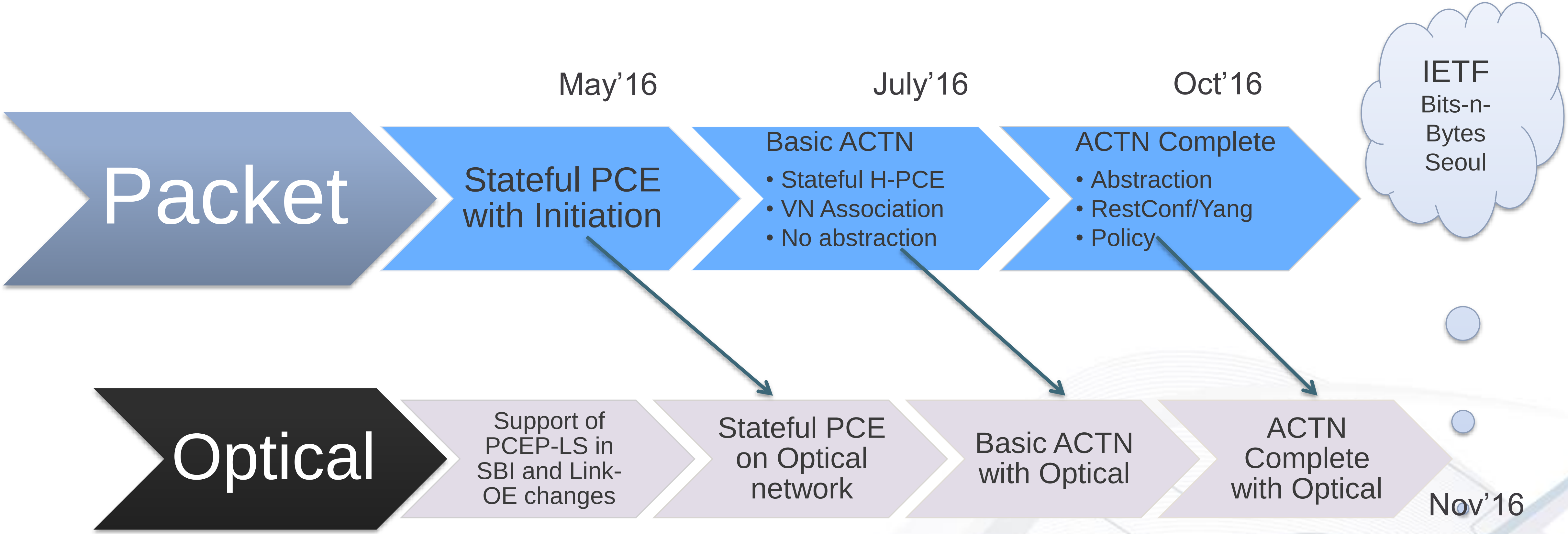
PNC

PCEP Interface
(SBI)

- Resource Discovery
- Resource Reservation

ACTN Code Contributions

ONOS Timeline



May'16

July'16

Oct'16

Packet

Stateful PCE with Initiation

Basic ACTN

- Stateful H-PCE
- VN Association
- No abstraction

ACTN Complete

- Abstraction
- RestConf/Yang
- Policy

IETF
Bits-n-Bytes
Seoul

Optical

Support of PCEP-LS in SBI and Link-OE changes

Stateful PCE on Optical network

Basic ACTN with Optical

ACTN Complete with Optical

Nov'16

Abstraction and Control of Transport Networks (ACTN)

ACTN Summary & Code Current Status

- Working together with SDOs , Open Source projects and PoC demos for early, and often, implementations
- Open ACTN wiki: https://sites.google.com/site/openactn/ for specification and reference information
- YANG Models GitHub
- ONOS GitHub:
  - https://github.com/opennetworkinglab/onos/tree/master/protocols/pcep
  - https://github.com/opennetworkinglab/onos/tree/master/protocols/bgp
- Support from vendors, operators and research/academia: Ericsson, Huawei, ALU, Infinera, KDDI, CMCC, China Telecom, Telefonica, SKT, KT, Microsoft, U. of Lancaster, U. of Bristol, BUPT, ETRI, CATR, etc.
- First industry multi-layer, multi-domain packet optical demo across multiple platform is planned in November 2016 (IETF 97, Seoul, Korea.)

Thank You!

Any comments or questions are welcome.

Daniel King
Lancaster University
d.king@lancaster.ac.uk

TOUCAN
ULTIMATE NETWORK CONVERGENCE

Architecting SDN for the Optical Access Network

ECOC Workshop
September 21, 2014

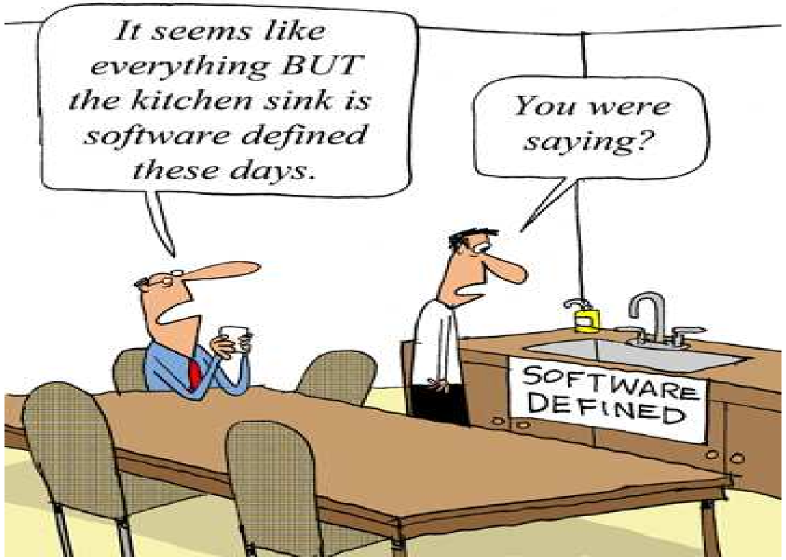
Daniel King
Lancaster University
d.king@lancaster.ac.uk

Recent Optical Network Developments
The Elastic Optical

- Elastic Optical Networks
  - Photonic Integrated Circuit (PIC) technology
    - Paving the path towards cost effective transmission schemes beyond 100Gbps.
  - Digital Coherent and SuperChannel technology solutions
    - Variable >100Gbps client signals and cost effective >100Gbps transponders
    - Capable of long reach up to 400Gbps without regeneration
  - Cost effective and flexible transponders
    - The Sliceable-Bandwidth Variable Transponder (SBVT).
      - Reduce bandwidth to extend reach
      - More spectrum to extend reach
      - More bandwidth over short reach
- Flexi-grid
  - A variable-sized optical frequency range.
  - ITU-T Study Group 15 (www.itu.int/rec/T-REC-G.694.1)

What do we mean by SDN?

- <u>**S**oftware</u>
  - It’s all software!
  - We are looking for automation
  - Tools and applications
- <u>**D**riven or **D**efined</u>
  - Does it matter?
- <u>**N**etworks</u>
  - Management of forwarding decisions
  - Control of end-to-end paths
  - Whole-sale operation of network



- The goals of commercial SDN networks
  - Make our networks better
  - Rapidly provide cool services at lower prices
  - Reduce OPEX and simplify network operations
  - Enable better monitoring and diagnostics
  - Make better use of deployed resources

- Converged services are the future
- Converged infrastructure is the future
- There is a significant element of centralisation

Building a Functional Architecture

- The purpose of a functional architecture is to decompose a problem space
  - Separate distinct and discrete functions into separate components
  - Identify the functional interactions between components
- An architecture is not a blue-print for implementation!
  - Components are *abstract* functional units
  - They can be realized as separate software blobs on different processors
  - Or they can all be rolled into one piece of spaghetti code
  - And they can be replicated and distributed, or centralized
- A protocol provides a realization of the interaction between two functional components
  - You only need to use it when the components are separated
- There have been many useful attempts to document architectures for SDN and NFV
- Our work has tried to present a wider picture
  - Address a range of network operation and management scenarios
  - Encompass (without changing) existing profiles of the architecture
  - Embrace SDN and NFV without becoming focused or obsessed with them
  - Highlight existing protocols and components

Application-Based Network Operation (ABNO)

- Application-Based Network Operations
  - A PCE-based Architecture for Application-based Network Operations
  - draft-farrkingel-pce-abno-architecture
- Network Controller Framework
  - Avoiding single technology domain “controller” architecture
  - Reuse well-defined components and protocols
    - Discovery of network resources and topology management.
    - Routing and path computation
    - Multi-layer coordination and interworking
    - Policy Control
    - OAM and performance monitoring
- Support a variety of southbound protocols
  - Leveraging existing technologies, support new ones
- Integrate with defined and developing standards, across SDOs

ABNO
Functional Components

- “Standardized” components
- Policy Management
- Network Topology
    - LSP-DB
    - TED
    - Inventory Management
- Path Computation and Traffic Engineering
    - PCE, PCC
    - Stateful & Stateless
    - Online & Offline
    - P2P, P2MP, MP2MP
- Multi-layer Coordination
    - Virtual Network Topology Manager
- Network Signaling & Programming
    - Optical (GMPLS/RSVP-TE)
    - ForCES
    - OpenFlow
    - Interface to the Routing System
    - Future technologies: Segment Routing & Service Function Chaining

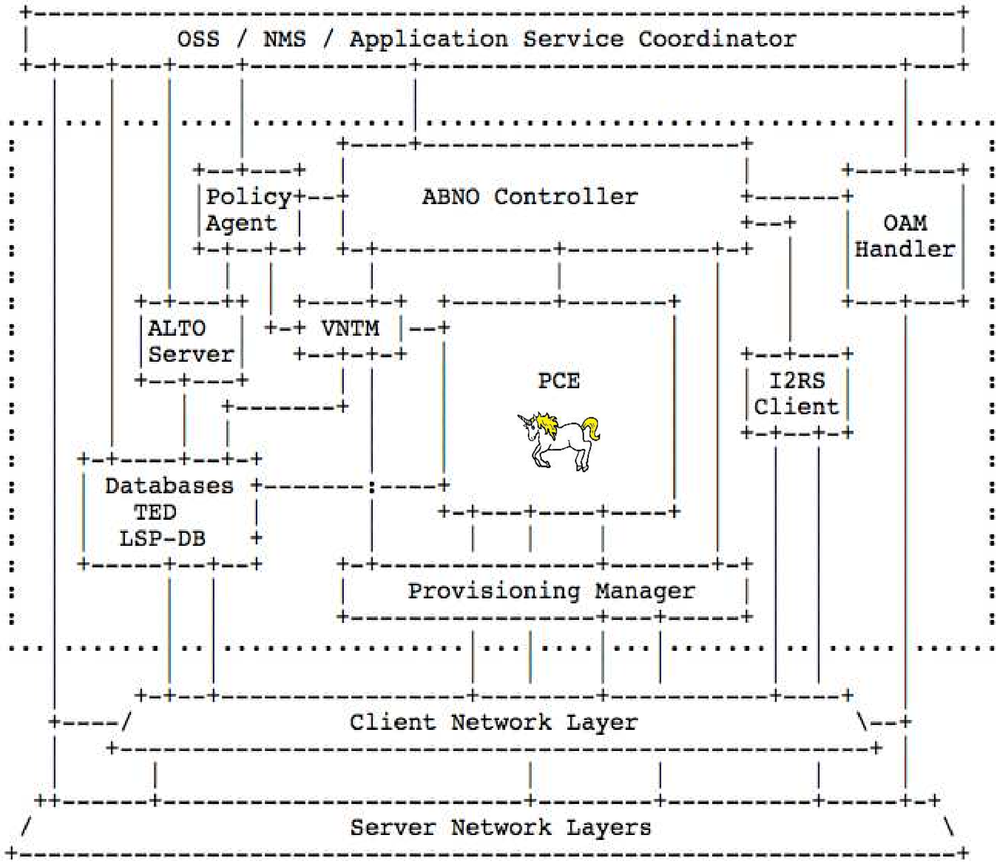
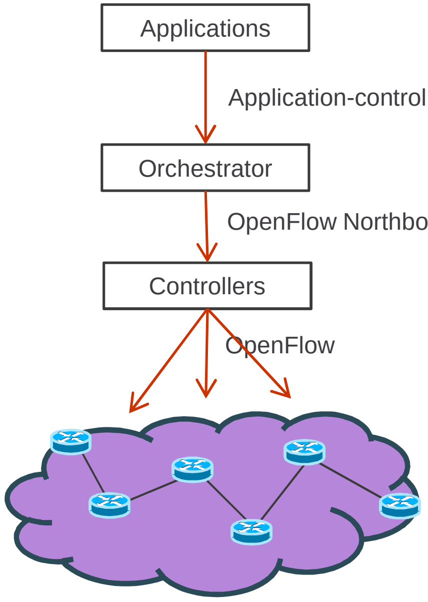


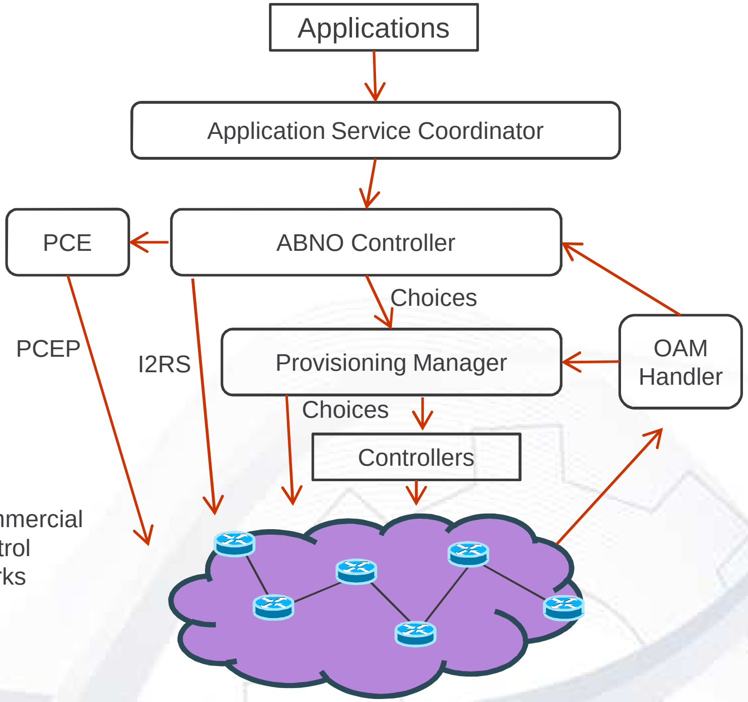
Figure 1: Generic ABNO Architecture

Compare ABNO with SDN Architecture

- A richer function-set based on the same concepts
- Enables the use of OpenFlow and other protocols
- There are implementation/deployment choices to be made

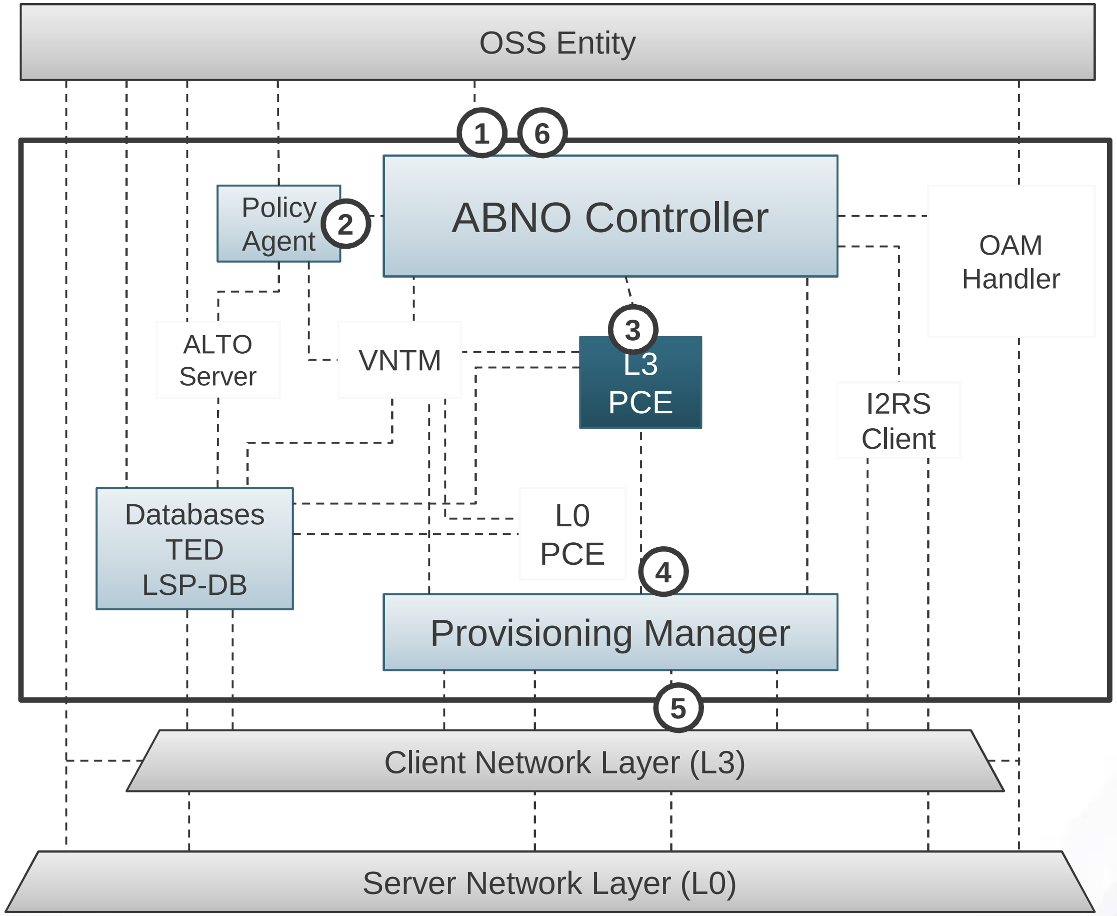
Minimum required for SDN controller of infrastructure





What is required for commercial deployment of SDN control platforms for real networks

FP7 IDEALIST Adaptive Network Manager
Based on an ABNO architecture



ABNO Operation

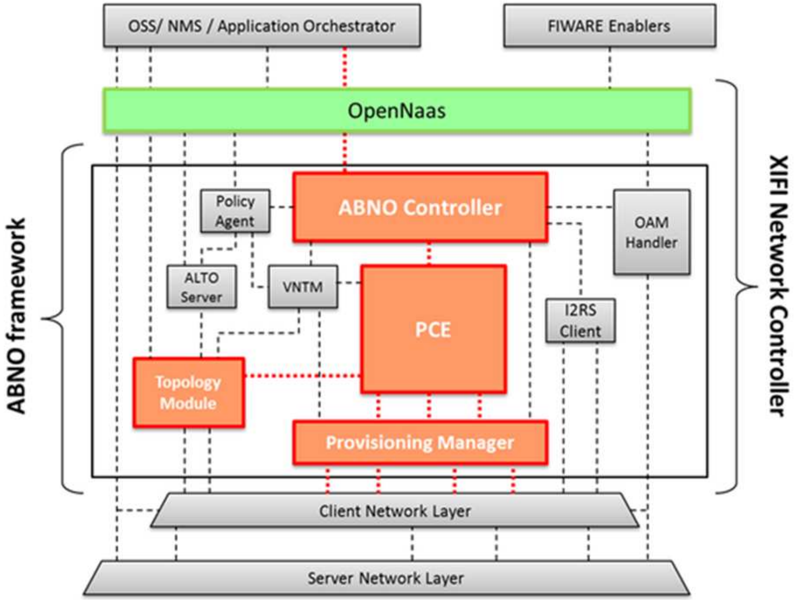
1. **OSS Entity** requests for a path between two L3 nodes.
2. **ABNO Controller** verifies **OSS Entity** user rights using the **Policy Manager**.
3. **ABNO Controller** requests to **L3-PCE** (active) for a path between both locations.
4. As **L3-PCE** finds a path, it configures L3 nodes via the **Provisioning Manager**.
5. **Provisioning Manager** configures L3 nodes using the required interface (RSVP-TE)
6. Response of successful path setup sent to **ABNO Controller**
7. **ABNO Controller** notifies the **OSS Entity** that the connection has been set-up.

FP7 IDEALIST Findings
ABNO Related Articles & Developments

- Publications (just a few)
  - In-Operation Network Planning
    IEEE Communications Magazine
  - Experimental Demonstration of an Active Stateful PCE performing Elastic Operations and Hitless Defragmentation
    ECOC European Conference on Optical Communications
  - Planning Fixed to Flexgrid Gradual Migration: Drivers and Open Issues
    IEEE Communications Magazine
  - Dynamic Restoration in Multi-layer IP/MPLS-over-Flexgrid Networks
    IEEE Design of Reliable Communication Networks (DRCN)
  - A Traffic Intensity Model for Flexgrid Optical Network Planning under Dynamic Traffic Operation
    OSA Optical Fiber Communication (OFC)
  - Full list of IDEALIST publications: www.ict-idealist.eu/index.php/publications-standards
- Standards Input
  - A PCE-based Architecture for Application-based Network Operations
    draft-farrkingel-pce-abno-architecture
  - Unanswered Questions in the Path Computation Element Architecture
    tools.ietf.org/html/draft-ietf-pce-questions

Additional EC Projects
ABNO Actively being investigated and developed

- **FI-PPP XIFI** (wiki.fi-xifi.eu) Creating a multi-DC community cloud across Europe.
  - **Flexible User Interface**
  - **Federated Cloud and Service Management**
  - **Dynamic Network Management**
  - **Resource Monitoring**



- **FP7 OFERTIE** (www.ofertie.org) Enhances the OFELIA testbed facility to allow researchers to request, control and extend network resources dynamically.
- **FP7 DISCUS** (discus-fp7.eu) Distributed Core for unlimited bandwidth supply for all Users and Services
- **FP7 CONTENT** (content-fp7.eu) Convergence of Wireless Optical Network and IT Resources in Support of Cloud Services
- **FP7 PACE** (ict-pace.net) - Next Steps for the Path Computation Element

Thank You!

Any comments or questions are welcome.

Daniel King
Lancaster University
d.king@lancaster.ac.uk

EWSDN 2014
European Workshop on
Software Defined Networks

The Role of PCE in an SDN World

Adrian Farrel – Old Dog Consulting
adrian@olddog.co.uk

Daniel King – Lancaster University

d.king@lancaster.ac.uk

Old Dog Consulting

LANCASTER
UNIVERSITY

Supported by pace

What shall we talk about?

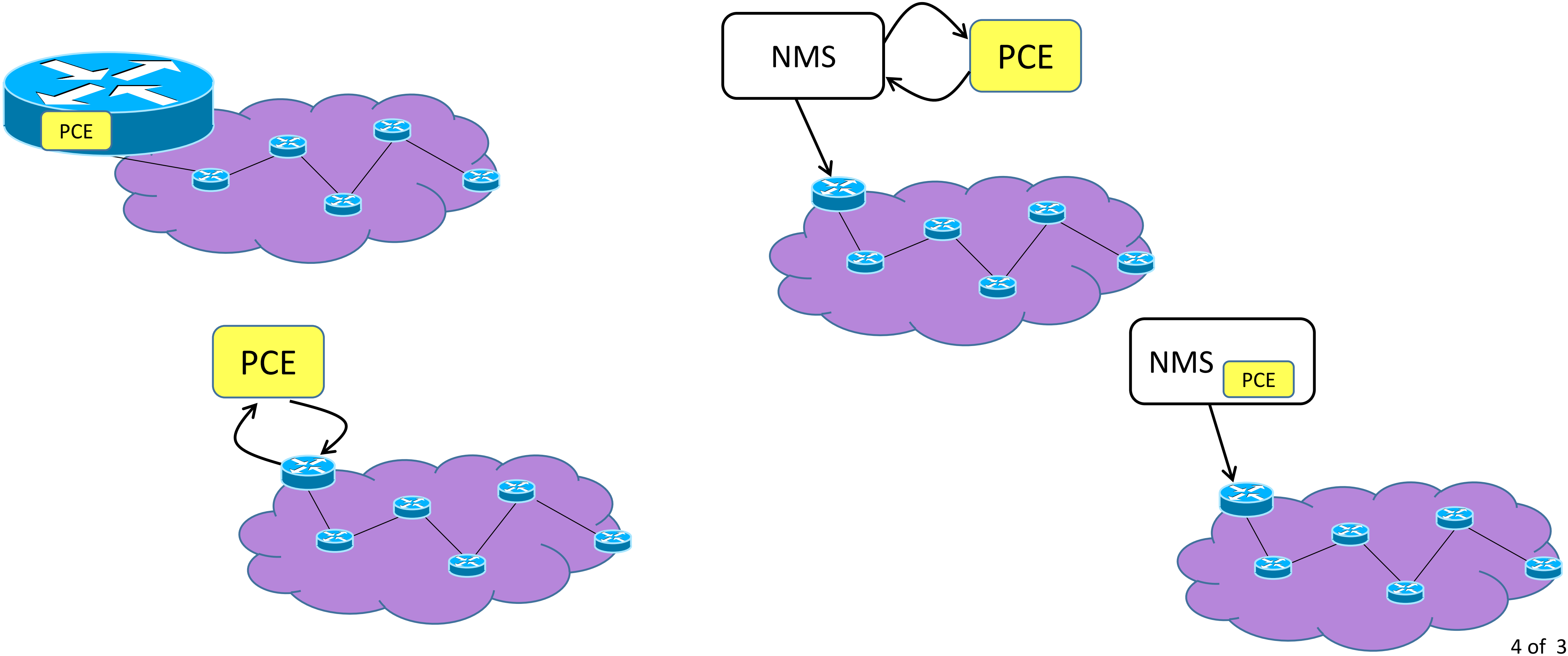
- The Path Computation Element (PCE)
  - What it is and where it comes from
  - How it is being used and what are the future plans
- SDN and NFV
  - What do we mean with these terms?
  - Is there a need for path computation?
- Application-Based Network Optimization (ABNO)
  - An “all-embracing” architecture or SDN and NFV
  - Where does PCE fit in ABNO?
  - What further work is needed?
- ABNO-centric implementations and research

The PCE – A short history

- PCE: Path Computation Element - “*An entity (component, application, or network node) that is capable of computing a network path or route based on a network graph and applying computational constraints.*” from RFC 4655
- That means that a PCE is a *functional component* in an abstract architecture.
    - It’s purpose is to determine paths though a network
    - It operates on a topology map (the Traffic Engineering Database – TED)
        - Nodes and links == connectivity graph
        - Node constraints and link constraints == metrics and capabilities
        - Learned from the routing protocol in the network, or from the inventory database, or direct from the network nodes
    - It can be realised as a component of an existing device (NMS, router, switch) or as a dedicated server (or virtualised service)
- Benefit of identifying PCE as a separate service…
    - Offload CPU-heavy computations
        - Provide advanced and sophisticated algorithms
    - Coordinate computation across multiple paths
    - Operate on an enhance TED
- Primary initial purpose was for Traffic Engineered MPLS LSPs
    - Rapidly picked up for optical transport networks

Deployment Models for PCE

- The Path Computation Client (PCC) may be co-located with the PCE or separate



Deployment models can be seen as theology

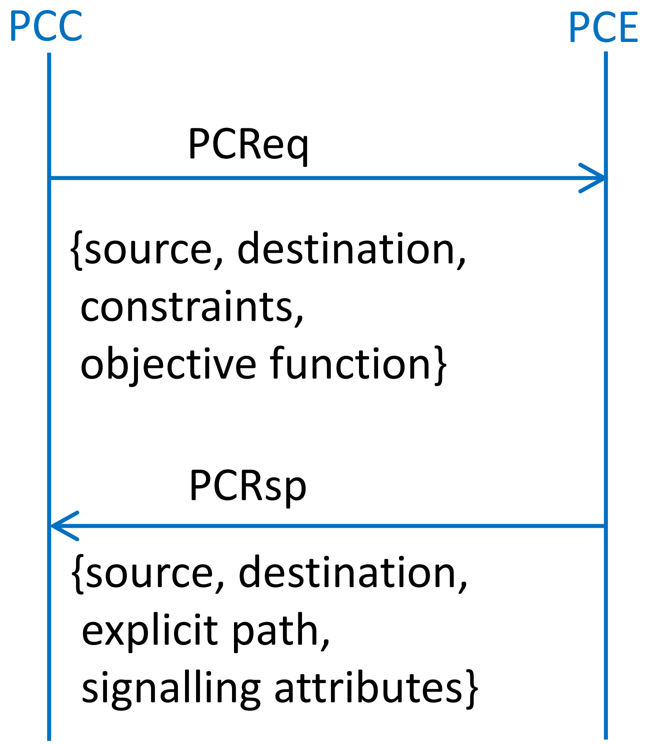
- How you choose to use PCE depends on how you like to operate your network

- There is a range of theologies
  - There is one God who sees and controls everything
  - There is a single God who answers prayer, but you have free choice
  - There are many gods each with different responsibilities
  - We all contain an element of God

- PCE can be placed in a number of places
  - In a central provisioning server (NMS)
  - In a dedicated computation server
  - There may be multiple PCEs with different capabilities in different parts of the network
  - The PCE function can be distributed into the routers

The PCE Protocol (PCEP)

- The PCE architecture originates in the IETF
  - The main focus of the IETF is to specify protocols
- PCEP is the request/response protocol for accessing the services of a PCE
  - Session-based carried over TCP
- Like PCE, PCEP had a very narrow purpose
  - Simple path computation request/response for MPLS-TE LSPs
- Initial proposals and early implementations
  - Used RSVP-TE Path messages
    - It is “kind of obvious”: that is exactly what we will signal
    - Just use the TCP session to give context to the usage
  - It really worked
- But was that really extensible?
  - Even in the MPLS-TE context we needed multiple extensions
  - RSVP has a lot of baggage
- Result:
  - A new container protocol and re-use of RSVP objects

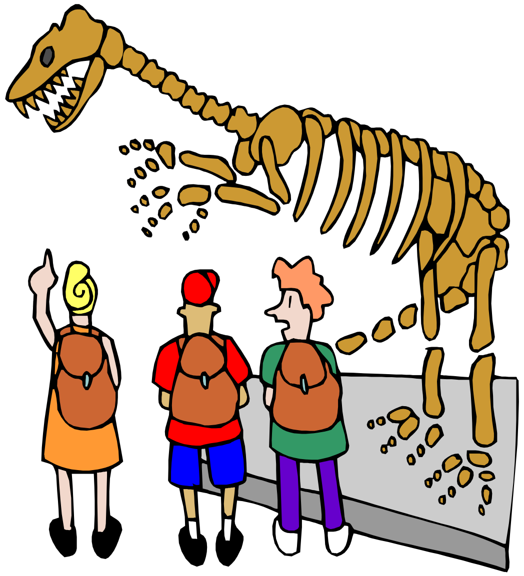


{source, destination,
constraints,
objective function}

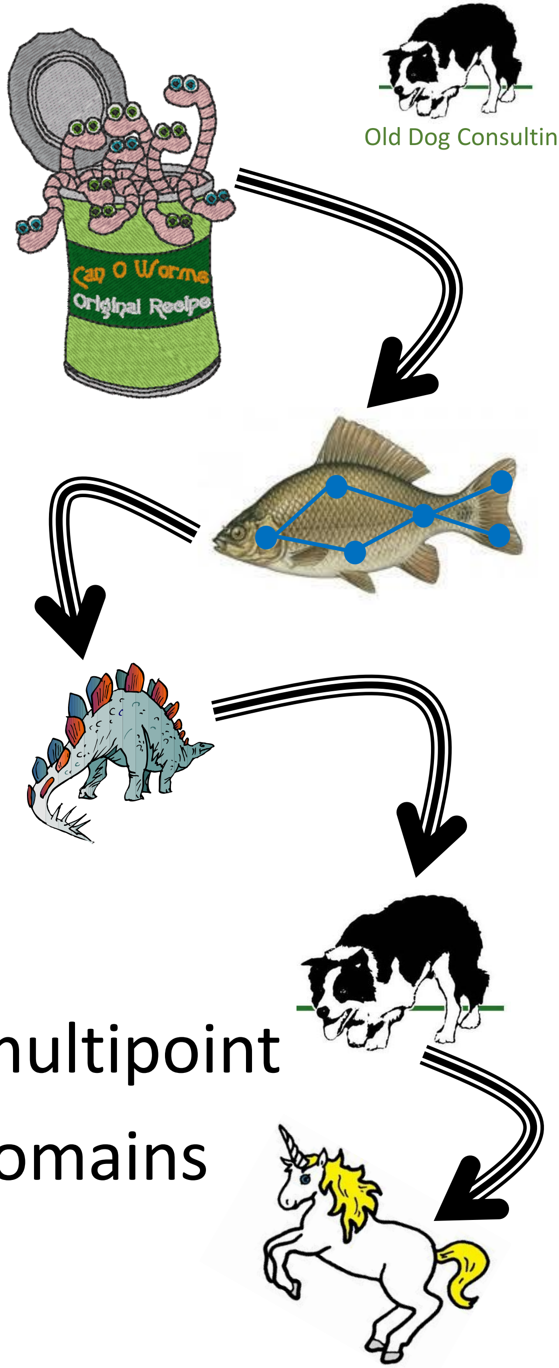
{source, destination,
explicit path,
signalling attributes}

The PCE – some more history

- Packet networks have not been a roaring success for PCE
  - Initially, only Cisco implemented
  - It is implemented and deployed
  - Main use cases are
    - Dual-homed IGP areas
    - Centrally controlled TE domains
- There is a huge amount of research and experimentation
  - More than 20 projects funded by the EU have PCE as a core component
  - A number of operators have in depth experimentations
- Commercial and Open Source Implementations
  - Software stacks from Metaswitch and Marben
    - But these are PCEP implementations, not full PCEs
  - Several Open Source implementations exist
  - Hardware vendors
  - Network operators
- The best take-up for PCE so far is in optical networks



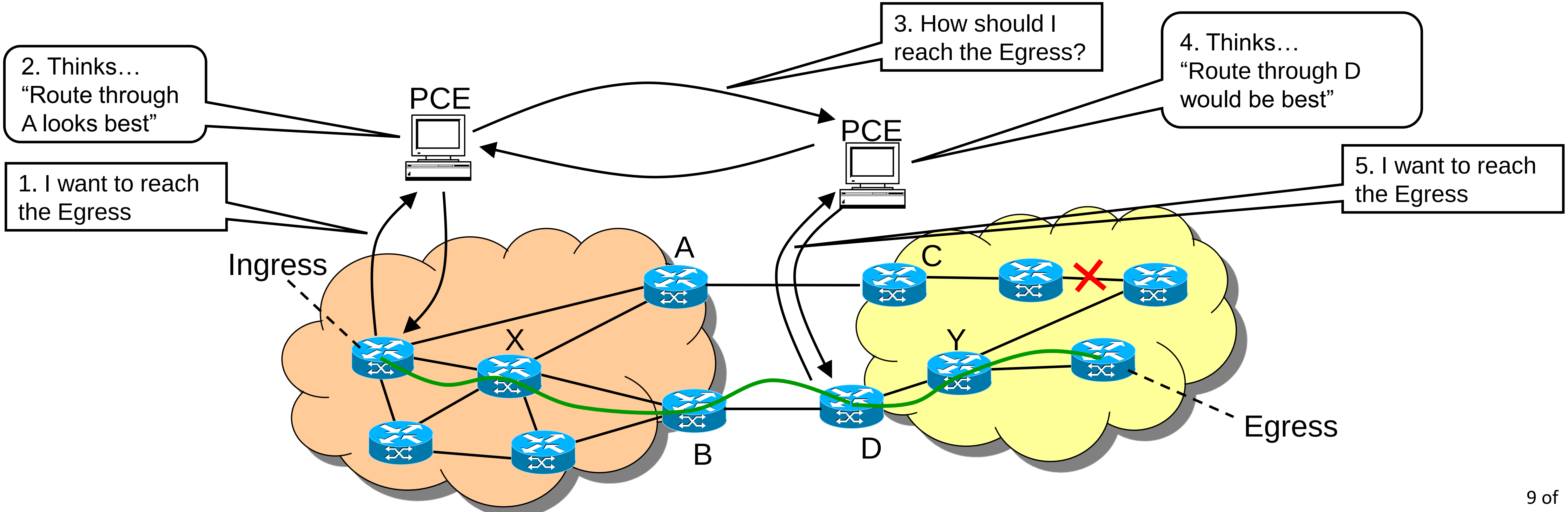
Evolution



- PCE evolved very quickly after it was invented
- Advanced PCEP encodings for non-packet environments
- PCEP extensions for coordinated path computations
  - Path protection
  - Network re-optimisation
- Cooperating PCEs for multi-domain applications
- Applicability to sophisticated services such as point-to-multipoint
- Hierarchical PCE for selection of paths across multiple domains
- And evolution continues today

Cooperating PCEs

- The first “interesting” problem for PCE was inter-domain TE
  - *“A domain is any collection of network elements within a common sphere of address management or path computation responsibility.”* RFC 4655
  - An IGP area or an Autonomous System
  - An optical island
- Nodes in one network cannot see into other networks
  - PCEs must ask each other for advice



2. Thinks…
“Route through A looks best”

1. I want to reach the Egress

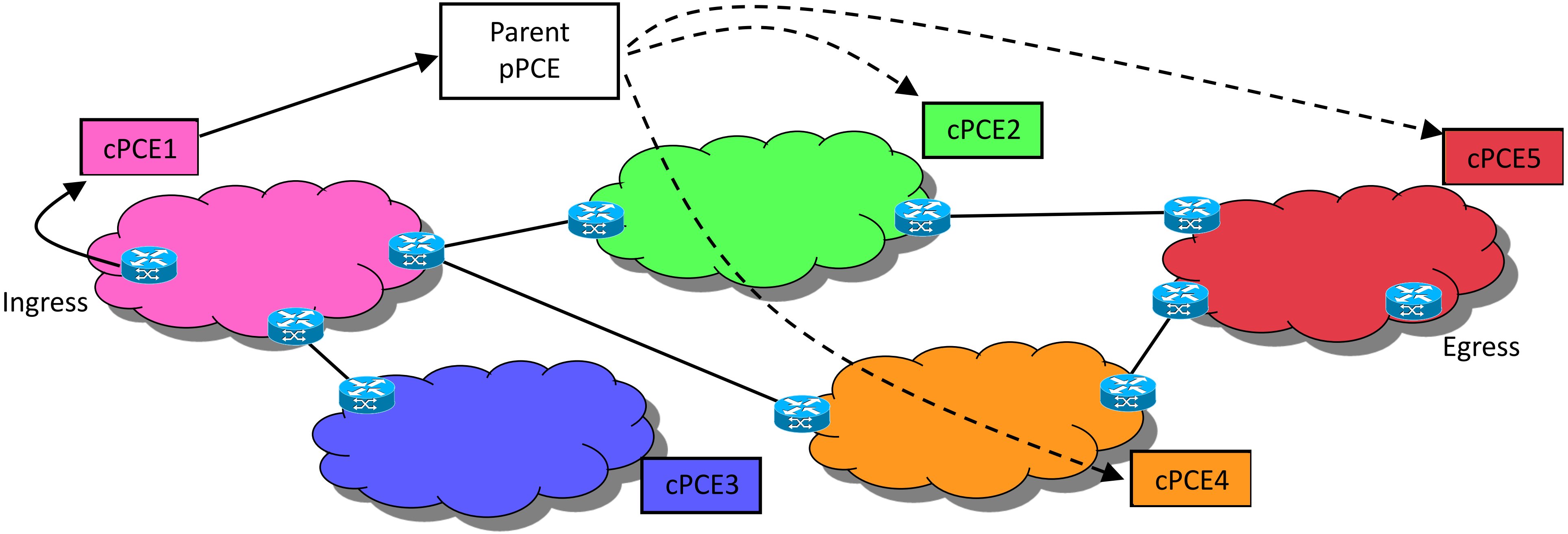
3. How should I reach the Egress?

4. Thinks…
“Route through D would be best”

5. I want to reach the Egress

Hierarchical PCE

- How do I select a path across multiple domains?
- Parent PCE (pPCE) has
  - An overview topology showing connectivity between domains
  - Communications with each Child PCE (cPCE)
- Parent can selectively and simultaneously invoke children to assemble an end-to-end path

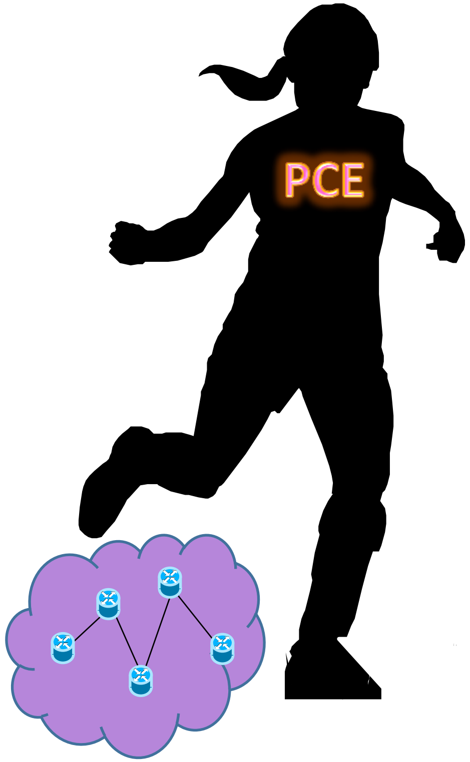


The Stateful PCE

- The “classic” PCE uses network state stored in the TED
  - This information may be gathered from the network
    - Passive participation in the IGP
    - Export from the network using BGP-LS
  - Or it may be gathered by “other mechanisms” (RFC 4655)
    - Inventory, management systems, configuration export
- There is also transitory per-computation state in the PCE
  - This allows bulk computation or “Please compute a path considering this other LSP”
- A Stateful PCE is aware of other LSPs in the network
  - A PCE could retain knowledge of paths it previously computed
  - Or it may gather information about LSPs as exported from the network
    - BGP-LS
    - PCEP
      - “Yes, I used that path you gave me”
      - “Here are some other LSPs I know about”
- A Stateful PCE is able to do more intelligent path computation

The Active PCE

- An Active PCE is able to advise the network
  - About more optimal paths
  - When congestion is a problem
- As far as the protocol is concerned, it is only a small step
  - The PCC can say “Please worry about these LSPs for me.”
    - Delegation of LSPs from the PCC to the PCE
  - The PCE can say “Here is a path you didn’t ask for.”
    - For delegated LSPs or for new LSPs
- This enriches PCEP
  - From a request/response protocol
  - To become *almost* a configuration / provisioning protocol
- Architecturally it is “interesting”
  - PCEP used to be the language spoken by the computation engine (PCE)
  - Now it is the language spoken by the network management system (NMS) that has a computation component
  - That doesn’t make it wrong. It does make it different
- It also opens up PCEP as an SDN protocol as we will see later



New Networks and PCE

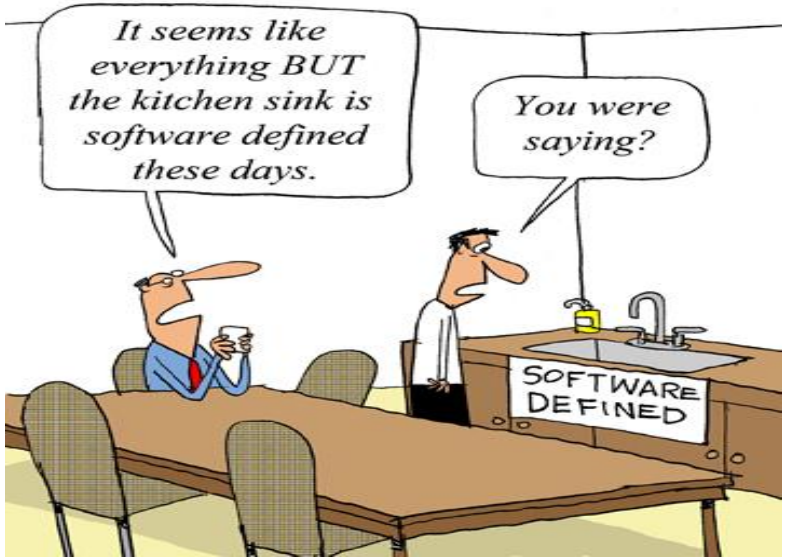
- New IETF effort : SPRING Working Group
  - Source Packet Routed Networking
  - Path through the network is predetermined for each packet
  - Path is encoded in the packet header as a series of hops
  - Some form of path computation is required
    - Could be as simple as SPF
    - May achieve load balancing
    - Might assign flows to different quality paths (delay, jitter, reliability, etc.)
- Service Function Chaining
  - Another new IETF effort : SFC Working Group
  - A Service Function Chain is an ordered list of service functions and servers
    - That means some form of path computation is necessary
- Deterministic wireless networks
  - For example Timeslotted Channel Hopping (TSCH) - IEEE802.15.4e
  - Path planning is an important aspect of operating these networks
- PCE is being investigated as a tool for these new networks
  - What that really means is that PCEP extensions are being proposed

What do we mean by “SDN”?

- **Software**
  - It’s all software!
  - We are looking for automation
  - Tools and applications
- **Driven or Defined**
  - Does it matter?
- **Networks**
  - Management of forwarding decisions
  - Control of end-to-end paths
  - Whole-sale operation of network

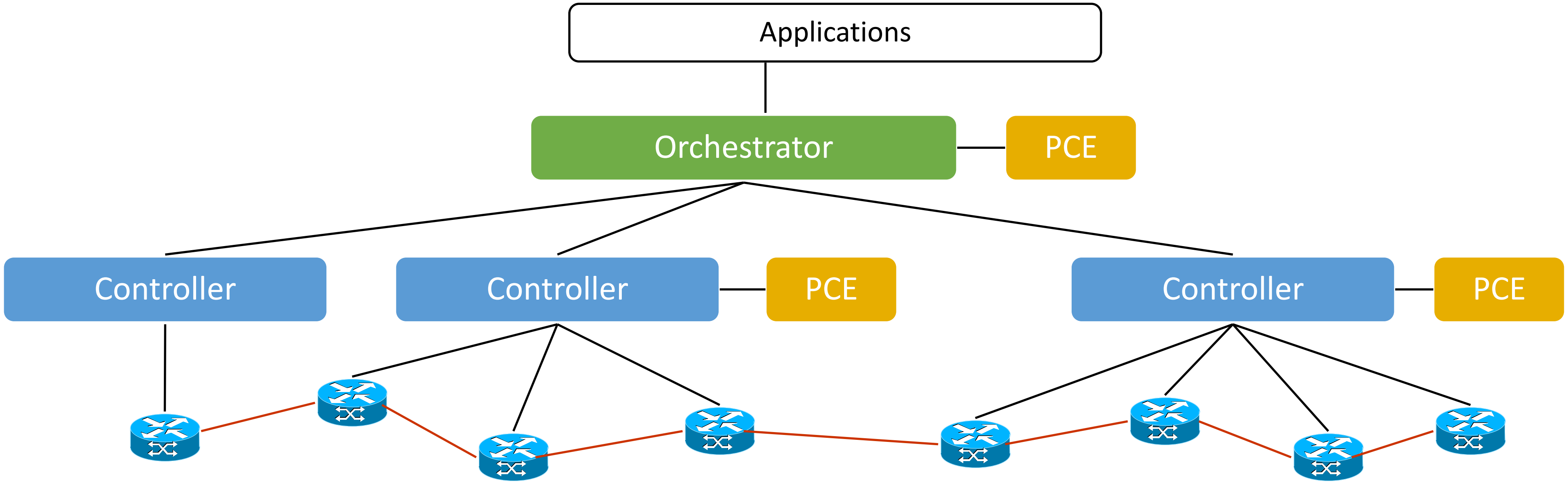
- The goals of commercial SDN networks
  - Make our networks better
  - Rapidly provide cool services at lower prices
  - Reduce OPEX and simplify network operations
  - Enable better monitoring and diagnostics
  - Make better use of deployed resources
- Converged services are the future
- Converged infrastructure is the future

- There is a significant element of centralisation



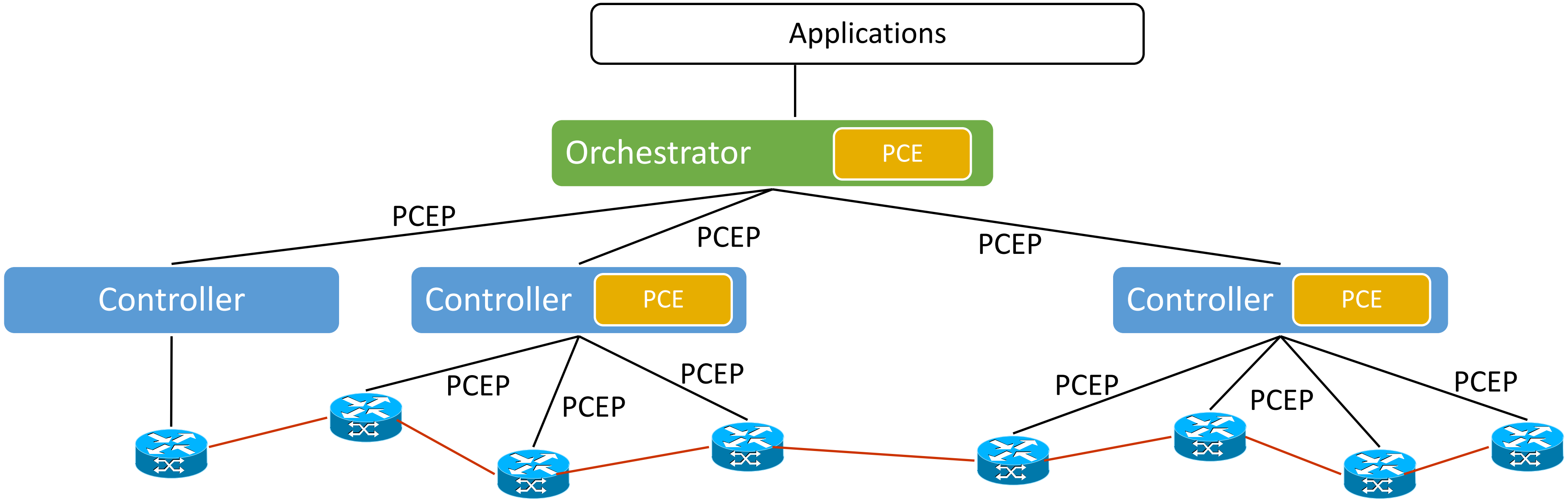
Bringing PCE to the SDN Feast

- PCE is an essential element for planning services in *any* network
- An Orchestrator cannot orchestrate without determining how traffic will flow through the network
  - And that means that an Orchestrator needs path computation function
  - Whether the PCE is built into the Orchestrator or lives as a separate component is an implementation choice
- A Controller cannot control more than a single node without determining how traffic will flow through a set of nodes
  - And that means that a Controller may need path computation function
  - Whether the PCE is built into the Controller or lives as a separate component is an implementation choice



PCEP as an SDN Protocol

- It is a simple step beyond an Active, Stateful PCE
    - Instead of suggesting LSPs, a PCE can provision LSPs
- Now PCEP can be seen as a full-scale provisioning protocol
    - I can provision anything for which I might have asked for a path
        - End-to-end LSPs
        - A fragment or segment of an LSP
        - The forwarding instructions on a single node
- Now PCE can be integral to the SDN components
    - I can use PCEP as an SDN Controller protocol
    - And/or as the Orchestrator-to-Controller protocol
- This raises the question of “competition” with OpenFlow which we will address later



Applications

Orchestrator PCE

PCEP

PCEP

PCEP

Controller

Controller PCE

Controller PCE

PCEP

PCEP

PCEP

PCEP

PCEP

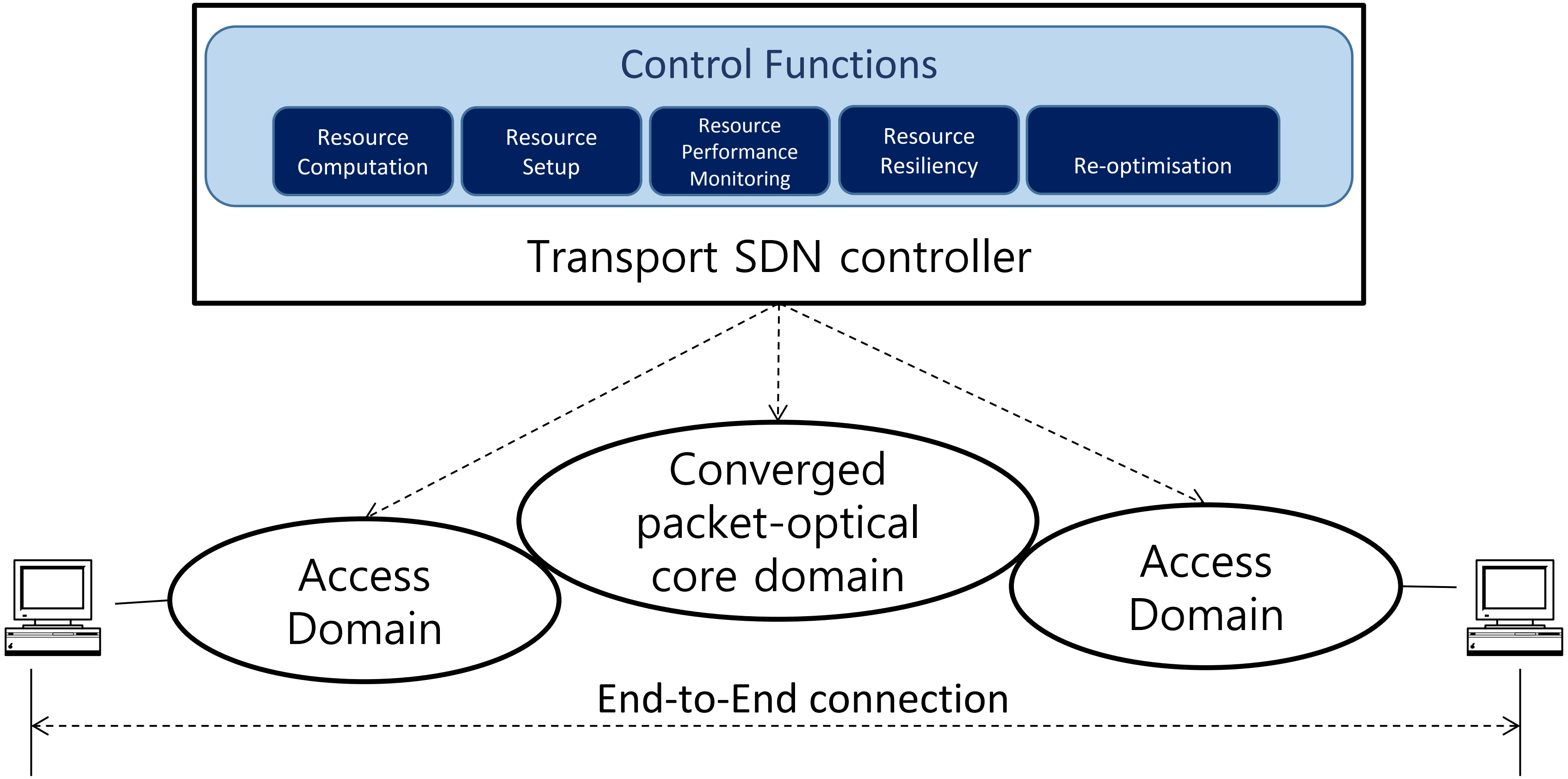
PCEP

Can we define “NFV”?

- Operators use a variety of proprietary appliances to provide network functions when delivering services
- Deploying a new network function often requires new hardware components
    - Integrating new equipment into the network takes space, power, and the technical knowledge
    - This problem is compounded by function and technology lifecycles which are becoming shorter as innovation accelerates
- The concept of virtualization is well-known and has been used for many years
    - Operating system virtualization (Virtual Machines)
    - Computational and application resource virtualisation (Cloud Computing)
    - Link and node virtualisation (Virtual Network Topologies)
    - Data Center Virtualisation (Virtual Data Center)
- Network Function Virtualization
    - Virtualize the *class* of network function
    - Replace specialist hardware with instances of virtual services provided on service nodes in the network
    - Enables high volume services and functions on generic platforms
- Virtualizing network connectivity for services and applications is just another facet of NFV

SDN & PCE as enablers for Network Virtualization

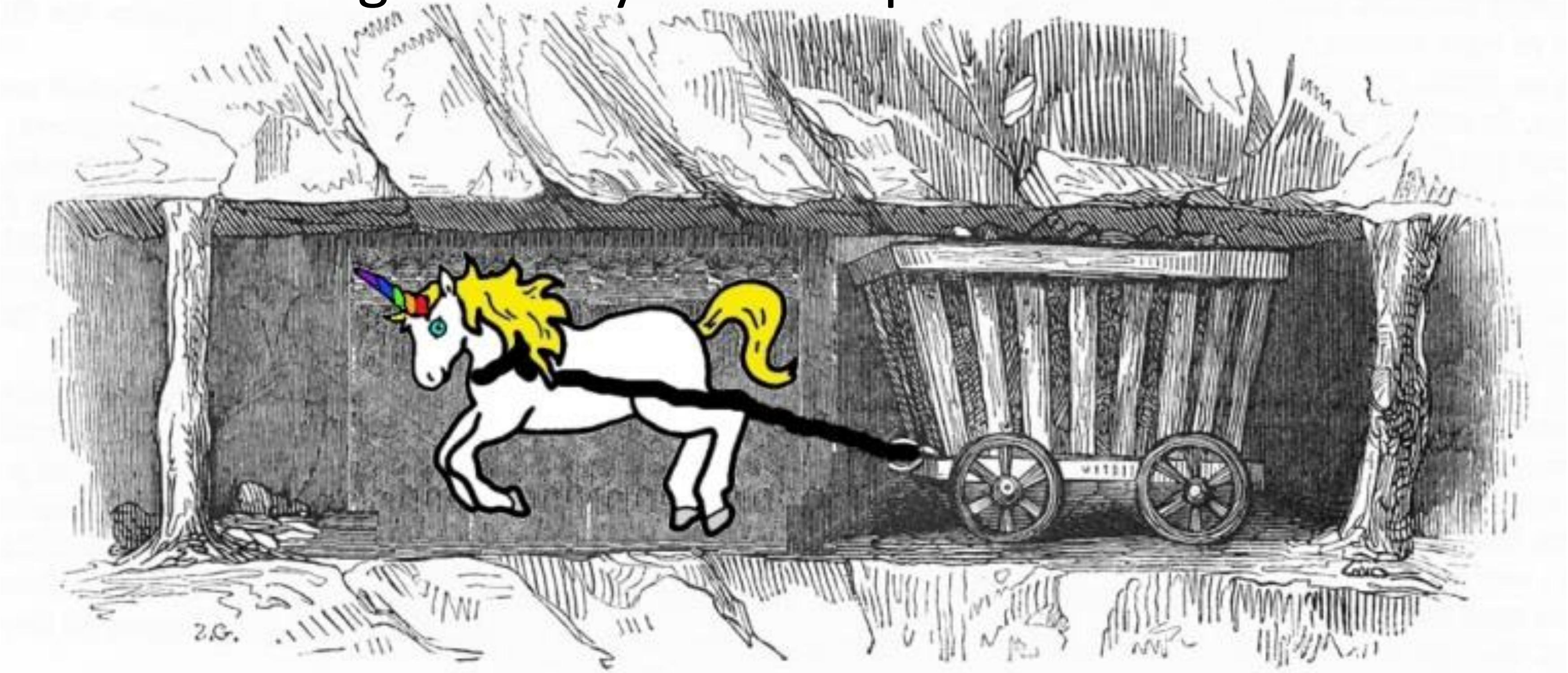
- Consider Transport SDN as an example
  - Integrates Packet, TDM, and Optical Layer into a single converged network
  - Requires centralized control functions including resource computation
  - Uses southbound control interface



Old Dog Consulting

Harnessing the Unicorn

- We’ve established that PCE is a wonderful thing
- We know that SDN and NFV offer a bright future for networking
- How do we bring PCE fully into the picture and make it work for us?



Building a Functional Architecture

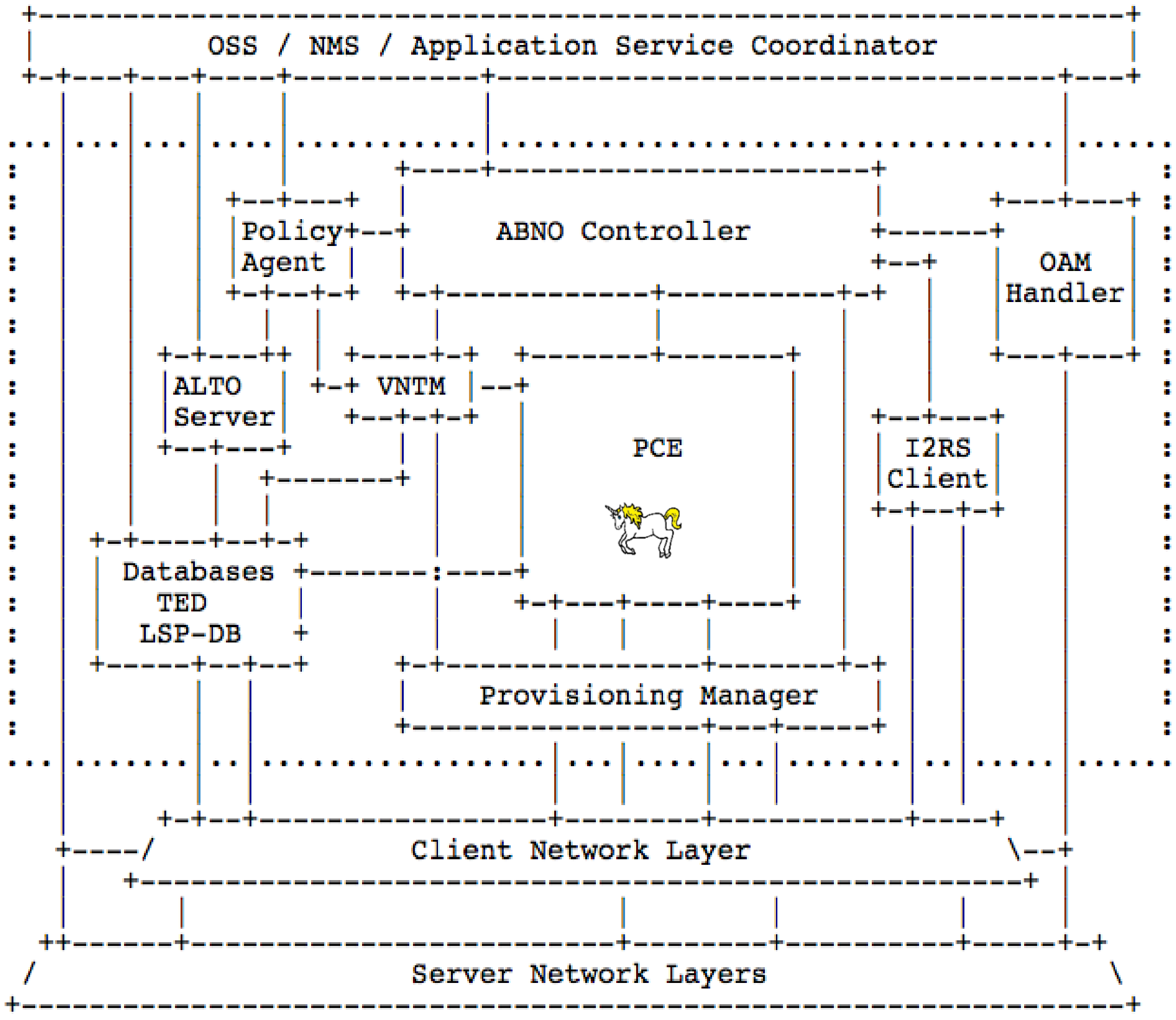
- The purpose of a functional architecture is to decompose a problem space
    - Separate distinct and discrete functions into separate components
    - Identify the functional interactions between components
- An architecture is not a blue-print for implementation!
    - Components are *abstract* functional units
    - They can be realized as separate software blobs on different processors
    - Or they can all be rolled into one piece of spaghetti code
    - And they can be replicated and distributed, or centralized
- A protocol provides a realization of the interaction between two functional components
    - You only need to use it when the components are separated
- There have been many useful attempts to document architectures for SDN and NFV
- Our work has tried to present a wider picture
    - Address a range of network operation and management scenarios
    - Encompass (without changing) existing profiles of the architecture
    - Embrace SDN and NFV without becoming focused or obsessed with them
    - Highlight existing protocols and components

Application-Based Network Operation (ABNO)

- Application-Based Network Operations
  - A PCE-based Architecture for Application-based Network Operations
  - draft-farrkingel-pce-abno-architecture
- Network Controller Framework
  - Avoiding single technology domain “controller” architecture
  - Reuse well-defined components and protocols
    - Discovery of network resources and topology management.
    - Routing and path computation
    - Multi-layer coordination and interworking
    - Policy Control
    - OAM and performance monitoring
- Support a variety of southbound protocols
  - Leveraging existing technologies, support new ones
- Integrate with defined and developing standards, across SDOs

ABNO – Functional Components

- “Standardized” components
- Policy Management
- Network Topology
    - LSP-DB
    - TED
    - Inventory Management
- Path Computation and Traffic Engineering
    - PCE, PCC
    - Stateful & Stateless
    - Online & Offline
    - P2P, P2MP, MP2MP
- Multi-layer Coordination
    - Virtual Network Topology Manager
- Network Programming and Signallling
    - ForCES
    - OpenFlow
    - Interface to the Routing System
    - PCEP
    - RSVP-TE


OSS / NMS / Application Service Coordinator

Policy Agent

ABNO Controller

OAM Handler

ALTO Server

VNTM

PCE

I2RS Client

Databases TED LSP-DB

Provisioning Manager

Client Network Layer

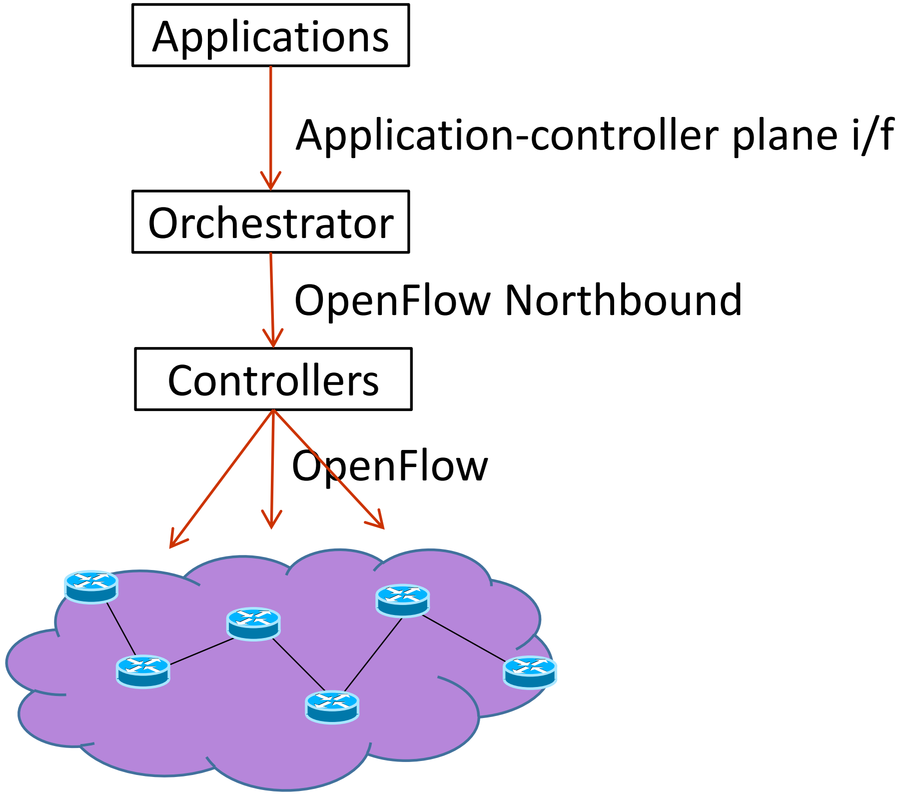
Server Network Layers

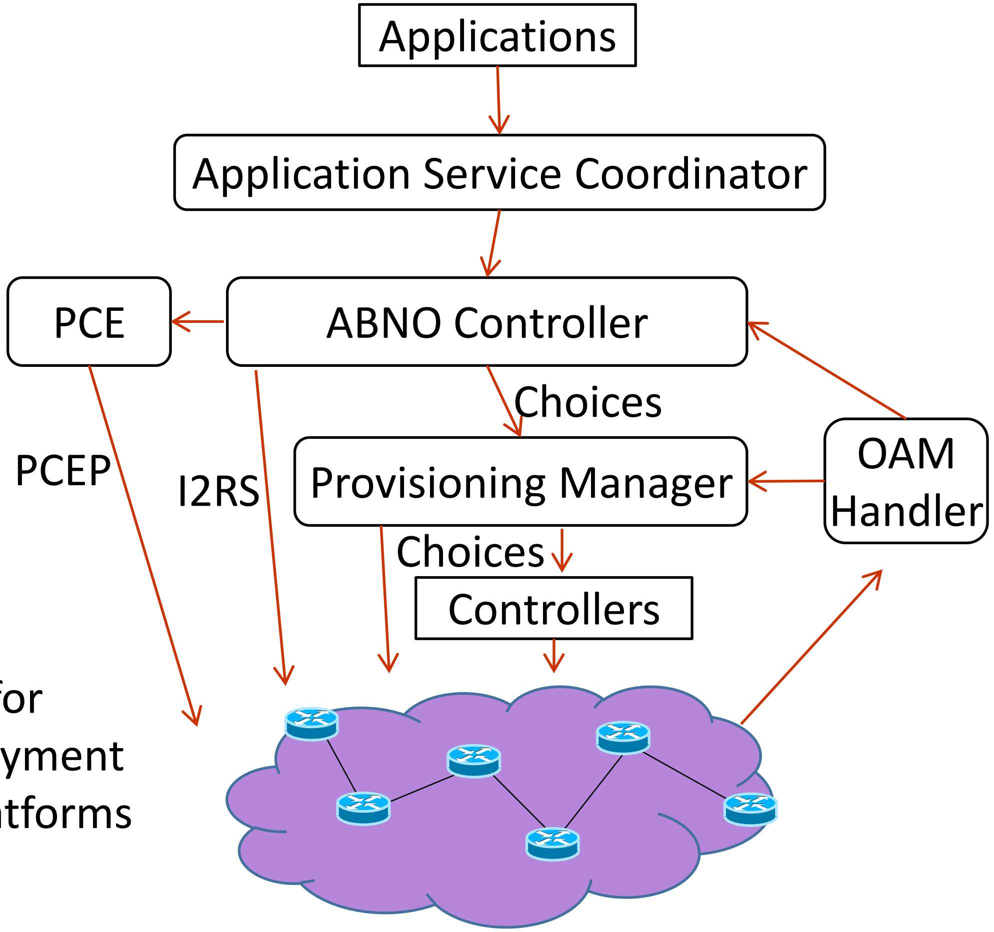
Figure 1: Generic ABNO Architecture

Compare ABNO with SDN Architecture

- A richer function-set based on the same concepts
- Enables the use of OpenFlow and other protocols
- There are implementation/deployment choices to be made

Minimum required for SDN controller of infrastructure





What is required for commercial deployment of SDN control platforms for real networks

ABNO Implementation and Research

- There are a number of experimental implementations of ABNO
    - Most notable was a demonstration of Packet-Transport Integration
        - Packet devices from Juniper Networks
        - Optical devices from Infinera
        - ANBO-based Transport SDN from Telefonica
        - Telefonica has also tested with ADVA and Ciena
- Multiple research projects examining the use of ABNO…

FP7 “IDEALIST” Project

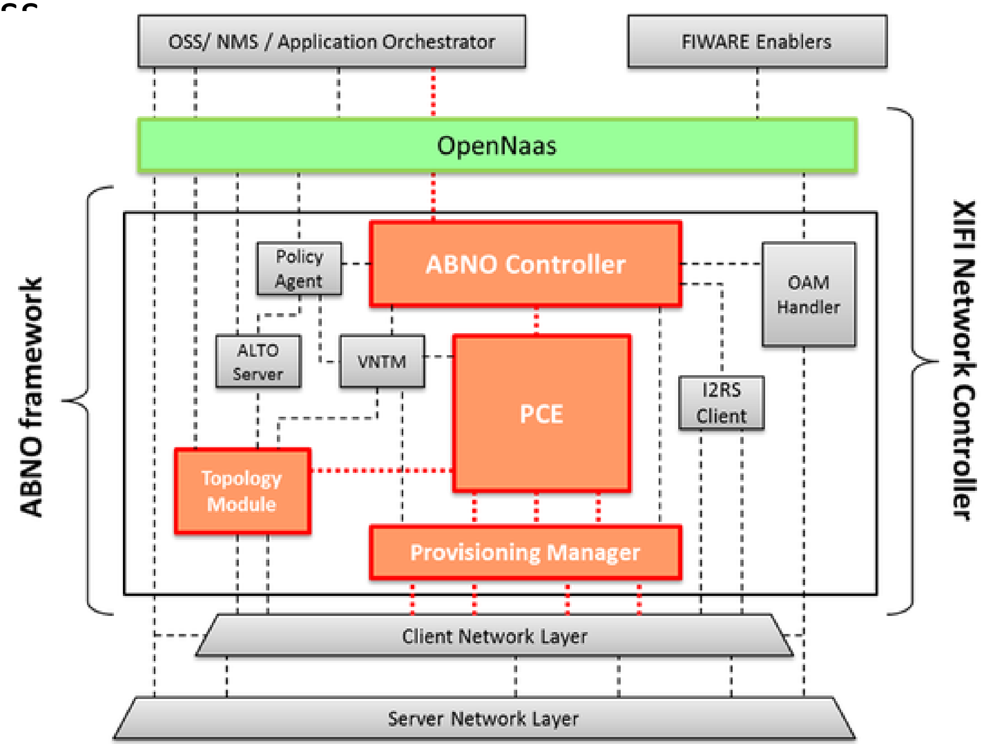
- Industry-Driven Elastic and Adaptive Lambda Infrastructure for Service and Transport (IDEALIST) Networks
  - The work is partially funded by the European Community’s Seventh Framework Programme FP7/2007-2013 through the Integrated Project (IP) IDEALIST under grant agreement nº 317999.
  - www.ict-idealist.eu
- The network architecture proposed by IDEALIST is based on four technical cornerstones:
  - An optical transport system enabling flexible transmission and switching beyond 400Gbps per channel
  - Control plane architecture for multi-layer and multi-domain optical transport network, extended for flexi-grid labels and variable bandwidth.
  - Dynamic network resources allocation at both IP packet and optical transport network layer
  - Multilayer network optimization tools enabling both off-line planning, on-line network reoptimization in across the IP and optical transport network
    - These tools are called Adaptive Network Management (ANM)
    - They are based on the ABNO architecture
    - Implementations exist!

FP7 IDEALIST Findings - Articles & Input to SDOs

- Publications (just a few)
  - In-Operation Network Planning
    IEEE Communications Magazine
  - Experimental Demonstration of an Active Stateful PCE performing Elastic Operations and Hitless Defragmentation
    ECOC European Conference on Optical Communications
  - Planning Fixed to Flexgrid Gradual Migration: Drivers and Open Issues
    IEEE Communications Magazine
  - Dynamic Restoration in Multi-layer IP/MPLS-over-Flexgrid Networks
    IEEE Design of Reliable Communication Networks (DRCN)
  - A Traffic Intensity Model for Flexgrid Optical Network Planning under Dynamic Traffic Operation
    OSA Optical Fiber Communication (OFC)
  - Full list of IDEALIST publications: www.ict-idealist.eu/index.php/publications-standards
- Standards Input
  - Unanswered Questions in the Path Computation Element Architecture
    tools.ietf.org/html/draft-ietf-pce-questions
  - A PCE-based Architecture for Application-based Network Operations
    tools.ietf.org/html/draft-farrkingel-pce-abno-architecture

Other FP7 Projects with ABNO

- **FP7 OFERTIE** (www.ofertie.org) Enhances the OFELIA testbed facility to allow researchers to request, control and extend network resources dynamically
- **FP7 DISCUS** (www.discus-fp7.eu) Distributed Core for unlimited bandwidth supply for all Users and Services
- **FP7 CONTENT** (www.content-fp7.eu) Convergence of Wirele Optical Network and IT Resources in Support of Cloud Services
- **FI-PPP XIFI** (www.wiki.fi-xifi.eu) Creating a multi-DC commun cloud across Europe
    - Flexible User Interface
    - Federated Cloud and Service Management
    - Dynamic Network Management
    - Resource Monitoring



TOUCAN

- Towards Ultimate Convergence of All Networks (TOUCAN)
- A UK funded project for 5 years from August 2014
- Academic Leadership
  - Lancaster, Heriot Watt, Edinburgh, and Bristol Universities
- Technology Partners
  - BT, Plextek, NEC, Samsung, JANET, and Broadcom
- Technology agnostic architecture for convergence based on SDN principles
  - Facilitate optimal interconnection of any network technology domains, networked devices and data sets with high flexibility, resource, and energy efficiency
  - Widely diverse networking technologies
    - Fiber-optic core
    - DSL, GigE
    - GSM/LTE
    - WiFi
    - Sensors
  - Service driven control with on demand delivery across virtualised infrastructure
  - Optimization based on capacity, connectivity, spectrum utilization, resource allocation and energy efficiency
  - Commoditisation of network and IT hardware devices
  - Exploit notion of adaptivity and programmability for optimal IT resource and workload allocation
- Investigating ABNO architecture as a cornerstone

The PACE Project

- Next Steps in PAth Computation Element (PCE) Architectures
- FP7 Coordination and Support Action
- Education and dissemination of PCE concepts
  - Tutorials, papers, knowledge base, outreach
- Development and applicability of new uses of PCE
  - Including SDN and NFV through support of ABNO
- Consolidate and coordinate existing (OpenSource) PCE developments
- http://www.ict-pace.net/
  - Funding from the European Union's Seventh Framework Programme for research, technological development and demonstration through the PACE project under grant agreement number 619712

ABNO and Research - Next Steps

- The research community is already embracing ABNO
- That should lead to important feedback
    - What is not clear in the architecture?
    - What pieces are missing or wrong?
    - How well do implementations behave?
    - How is PCE integrated into the whole?
        - What new PCE algorithms are needed?
        - How does PCEP need to be enhanced?
    - What new network types can be managed?
    - How can NFV, SFC, and network slicing be operated?
    - What are the security, management, and economic implications?

ABNO and Industry / Standards

- draft-farrkingel-pce-abno-architecture will soon be published as an RFC
  - It is informational and not a mandatory standard
    - It leaves a number of interfaces unspecified
      - For example, service request interface
    - It presents too many choices
  - Next steps
    - Applicability statements to show how to profile ABNO for specific environments
      - A few are captured in the draft
      - More (such as SDN) could be documented
    - New requirements documents and protocol specifications to fill the gaps
- This work will be done in coordination with industry
  - What do people really want to build and deploy?

Assertions

- PCE is here to stay as a functional component of SDN
- Implementing PCE as a distinct unit enables
    - Scaling
    - Load-balancing
    - Rapid advancement of algorithms
- That means PCEP is a necessary protocol for accessing PCE
- PCEP can be used as a "provisioning protocol"
    - Most clear use in circuit-switched networks (MPLS-TE, GMPLS, …)
    - Jury is out on the use of PCEP as a per-node control protocol
- SDN should be seen as a critical part of a wider view of network operation
- Re-use of components and protocols makes sense
- The ABNO architecture embraces SDN and factors it into the bigger picture

References

- The PACE project “PCE Primer”
  http://www.ict-pace.net/files/3313/8929/2782/PCE_Primer.pdf
- Path Computation Element Tutorial
  http://www.olddog.co.uk/Farrel_PCE_Tutorial.ppt
- IETF's PCE Working Group
  https://datatracker.ietf.org/wg/pce/documents/
- RFC 4655, ”A Path Computation Element (PCE)-Based Architecture”
  https://www.rfc-editor.org/rfc/rfc4655.txt
- RFC 5440, ”Path Computation Element Communications Protocol“
  https://www.rfc-editor.org/rfc/rfc5440.txt
- RFC 6805, ”Hierarchical PCE“
  https://www.rfc-editor.org/rfc/rfc6805.txt
- draft-farrkingel-pce-abno-architecture, “A PCE-based Architecture for Application-based Network Operations”
  https://www.ietf.org/id/draft-farrkingel-pce-abno-architecture
- draft-ietf-pce-questions, “Unanswered Questions in the Path Computation Element Architecture“
  https://www.ietf.org/id/draft-ietf-pce-questions
- “PCE: What is It, How Does It Work and What are its Limitations?”
  Journal of Lightwave Technology, 2014.
- “In-Operation Network Planning”
  IEEE Communications Magazine, 2014.
- “Towards a carrier SDN: an example for elastic inter-datacenter connectivity”
  Optics Express, 2014.
- “PCEP - A Protocol for All Uses? How and when to extend an existing protocol”
  PACE Workshop, 2014.
- “A Survey on the Path Computation Element (PCE) Architecture”
  IEEE Communications Surveys and Tutorials, 2013.
- “Using the Path Computation Element to Enhance SDN for Elastic Optical Networks (EON)”
  iPOP Tokyo, 2013.

LANCASTER
UNIVERSITY

Old Dog Consulting

Questions?

Follow-up

adrian@olddog.co.uk

d.king@lancaster.ac.uk

EWSDN 2014
European Workshop on
Software Defined Networks

pace

SDN-based Elastic and Adaptive Optical Transport:
Findings and Future Research

WDM & Next Generation Optical Networking
Nice, France
Wednesday, 24 June, 2015

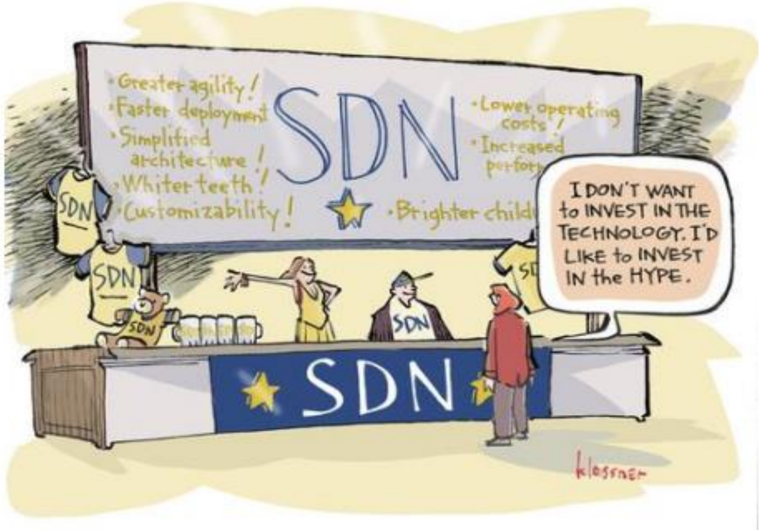
Daniel King
Senior Researcher, Lancaster University (BT & Intel, NFV Co-Lab)
Co-Chair of SDN Internet Research Task Force (IRTF)
Open Networking Foundation (ONF) Research Associate
d.king@lancaster.ac.uk
daniel.king@bt.com

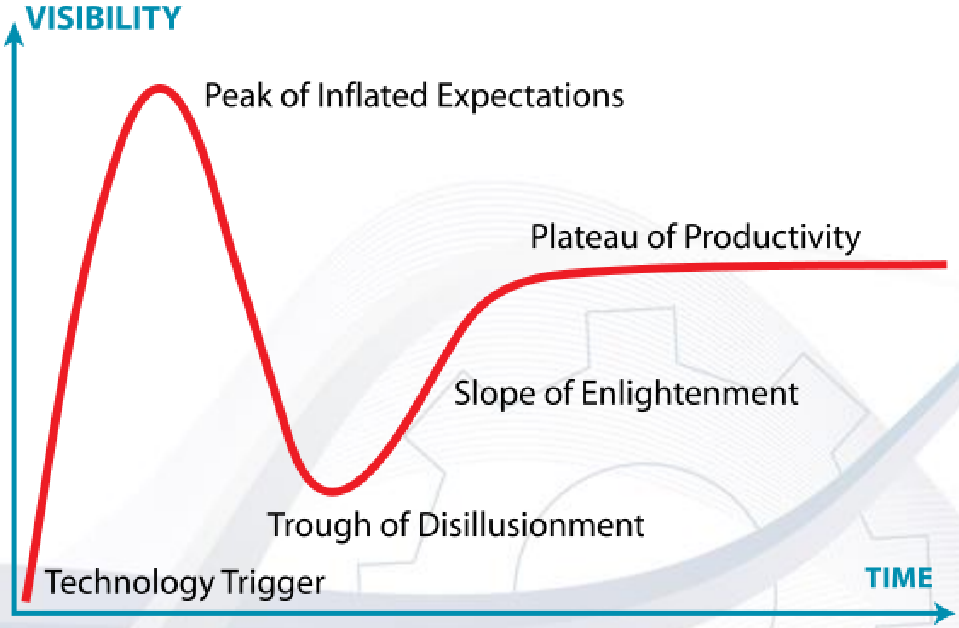
Andrew Lord
Head of Optical Research, British Telecom
andrew.lord@bt.com

Network Evolution

SDN, A reality check

- Why Software Defined Networking?
  - There's a hype in the industry! *(no, really?)*
  - Where are we on the hype cycle?
- Dispelling some myths
  - SDN is not just a provisioning system or configuration management tool
- Can you really buy "off the shelf" SDN
  - Which architectural approach?
  - What are SDN protocols these days?
- SDN for large, complex networks requires internal development
  - Do we have to mirror the Google and Facebook approach?





An opportunity for SDN & NFV
Variable bit-rate technology

- Flexible and Elastic Optical Networks
  - Photonic Integrated Circuit (PIC) technology
    - Paving the path towards cost effective transmission schemes beyond 100Gbps.
  - Digital Coherent and SuperChannel technology solutions
    - Variable >100Gbps client signals and cost effective >100Gbps transponders
    - Capable of long reach up to 400Gbps without regeneration
  - Cost effective and flexible transponders
    - The Sliceable-Bandwidth Variable Transponder (SBVT).
      - Reduce bandwidth to extend reach
      - More spectrum to extend reach
      - More bandwidth over short reach
- FlexGrid
  - A variable-sized optical frequency range
  - ITU-T Study Group 15 (www.itu.int/rec/T-REC-G.694.1)

Leveraging
FlexGrid with SDN & NFV

- The network architecture we developed is based on four technical cornerstones:
  1. An optical transport system enabling flexible transmission and switching up to, and beyond 400Gbps per channel.
  2. Hybrid control plane architecture for multi-layer and multi-domain optical transport network, extended for flexi-grid labels and variable bandwidth
  3. Dynamic network resources allocation at both IP and optical transport network. layer
  4. Leveraging Software Defined Networks and Network Functions Virtualisation paradigms

- Focus on standards-based development
  - Framework for GMPLS based control of Flexi-grid DWDM networks
  - Generalized Labels for the Flexi-Grid in LSC Label Switching Routers
  - GMPLS OSPF-TE Extensions in for Flexible Grid DWDM Networks
  - RSVP-TE Signaling Extensions in support of Flexible Grid
  - Extensions to PCEP for Hierarchical Path Computation Elements (H-PCE)
  - A YANG data model for FlexGrid Optical Networks

A Controller for
Optical Network Operations

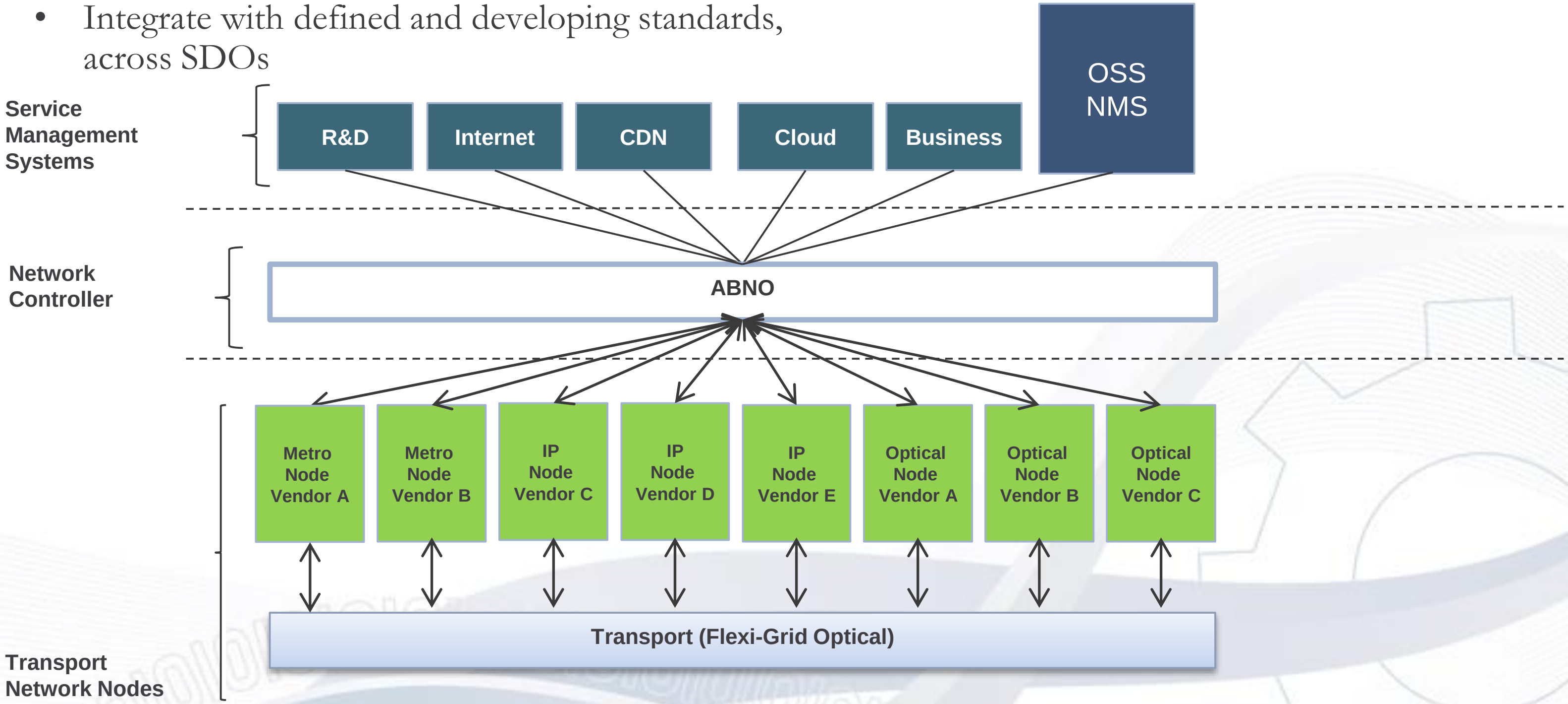
- "SDN Controller" is a contentious term, it can have many different meanings:
  - Historically the term was derived from the network domain, technology and protocol mechanism
- SDN Controller wars are ongoing:
  - Operators have an expectation of standards-based technologies for deploying and operating networks
  - SDN controller vendors rarely provide multivendor interoperability using open standards
  - Provisioning should be a compelling feature of SDN, however many SDN controllers use non-standardised APIs
  - Recent Open Source initiatives tend to be vendor led
- Typically SDN controllers have a very limited view of topology, multi-layer and multi-domain scenarios are slowly being added
- Flexibility has been notably absent from most controller architectures both in terms of southbound protocol support and northbound application requests

Decomposition of an Optical network controller

- Avoiding the mistake of a single "controller" architecture
  - As it encourages the expansion and use of specific protocols
- Discovery of network resources and topology management
- Network resource abstraction, and high-layer presentation
- Wavelength assignment and path computation
- Multi-layer coordination and interworking
  - Multi-domain & multi-vendor network resources provisioning through different control mechanisms (e.g., OpenFlow, ForCES)
- Policy Control
- OAM and Performance Monitoring
- Security & Resiliency
- A wide variety of southbound northbound protocol support
- Leveraging existing technologies
  - What is currently available?
  - Must integrate with existing and developing standards

What is an Application-Based Network Operation?

- Applications-Based Network Operations (ABNO - RFC7491)
  - A PCE-based Architecture for Application-based Network Operations https://tools.ietf.org/html/rfc7491
- Network Controller Framework
  - Avoiding single technology domain "controller" architecture
  - Reuse well-defined components
  - Support a variety of southbound protocols
    - Leveraging existing technologies, support new ones
- Integrate with defined and developing standards, across SDOs



Service Management Systems

R&D

Internet

CDN

Cloud

Business

OSS NMS

Network Controller

ABNO

Transport Network Nodes

Metro Node Vendor A

Metro Node Vendor B

IP Node Vendor C

IP Node Vendor D

IP Node Vendor E

Optical Node Vendor A

Optical Node Vendor B

Optical Node Vendor C

Transport (Flexi-Grid Optical)

ABNO for FlexGrid

Uses & Applications

- The network does not need to be seen any longer as a composition of individual elements
  - Applications need to be capable of interaction with the network.
- Support of the next generation of variable and dynamic optical transport characteristics
  - Multi-layer path provisioning
  - Network optimization after restoration
- Automated deployment and operation of services.
  - “Create a new transport connection for me”
  - “Reoptimize my network after restoration switching”
  - “Respond to how my network is being used”
  - “Schedule these services”
  - “Identify lease loaded links, and targets for future capacity planning”

ABNO
Functional Components

- "Standardized" components
- Policy Management
- Network Topology
  - LSP-DB
  - TED
  - Inventory Management
- Path Computation and Traffic Engineering
  - PCE, PCC
  - Stateful & Stateless
  - Online & Offline
  - P2P, P2MP, MP2MP
- Multi-layer Coordination
  - Virtual Network Topology Manager
- Network Signaling & Programming
  - RSVP-TE
  - ForCES
  - OpenFlow
  - Interface to the Routing System
  - Emerging technologies: Segment Routing & Service Function Chaining

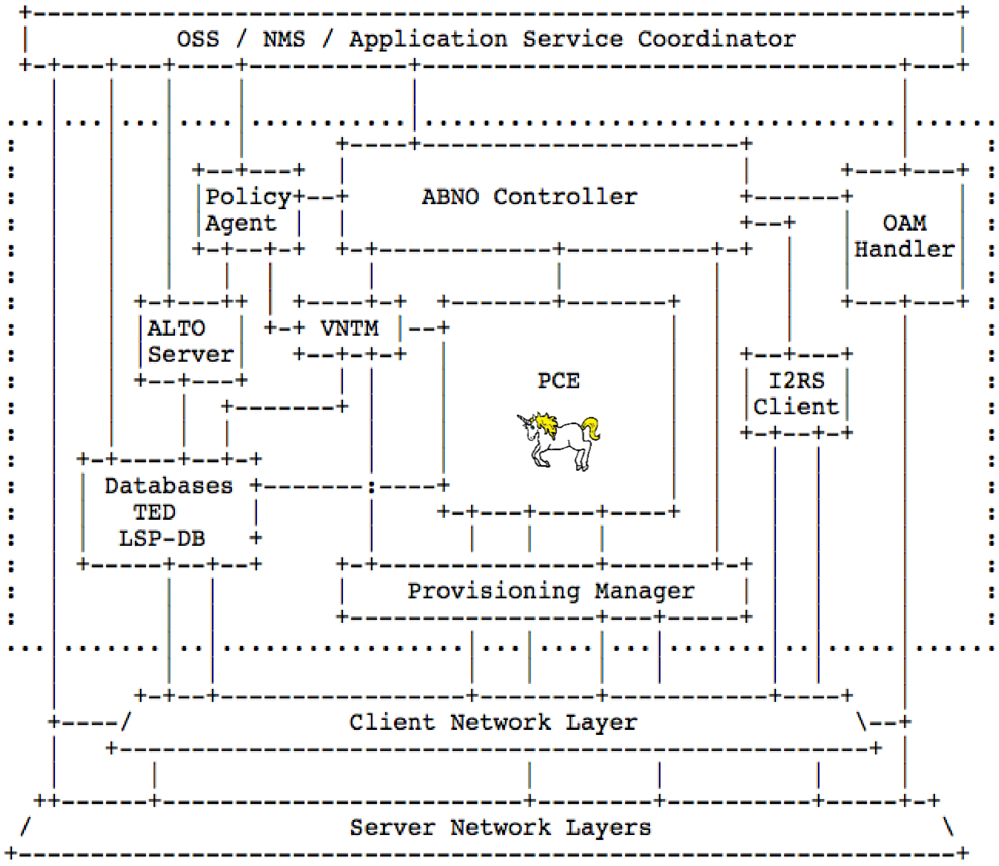
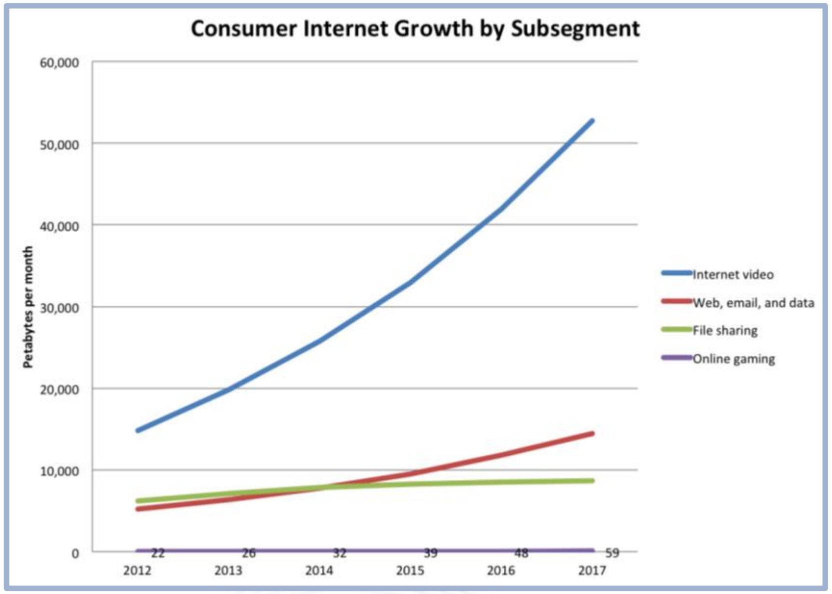


Figure 1: Generic ABNO Architecture

Is Content Delivery the “Killer Application” for SDN & NFV?

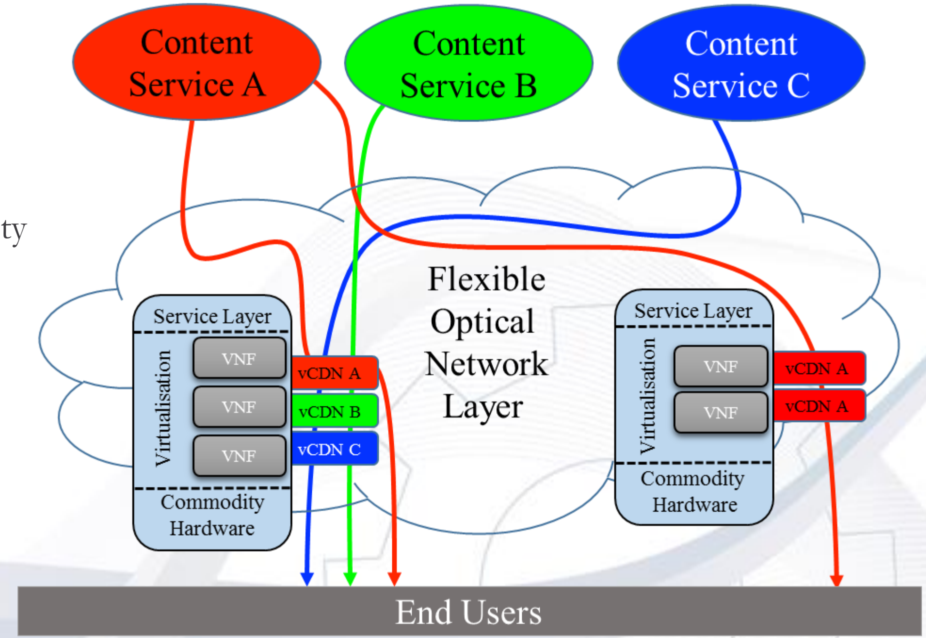
- Delivery of content, especially of video, is one of the major challenges of all operator networks due to massive growing amount of traffic.
- Complementary to the growth of today’s Video Traffic
  - On-demand Content Services to internet end-users, with similar quality constraints as for traditional TV Service of Network Operators
  - Delivery of terrestrial transmissions over IP/optical networks
- Distribution of terrestrial transmissions:
  - Uncompressed: Serial Digital Interface (SDI)
  - Compressed: Motion JPEG



Name	Video	Bitrate
SD-SDI	480i/576i	270 Mbit/s
HD-SDI	720p/1080i	1.5 Gbit/s
3G-SDI	1080p	3 Gbit/s
6G UHD-SDI	4K 30fps	6 Gbit/s
12G UHD-SDI	4K 60fps	12 Gbit/s
24G UHD-SDI	4K 120fps	24 Gbit/s

SDN & NFV "Killer Application"
Content Distribution Network (CDN)

- Design principles require an efficient, reliable and responsive CDN
  - Fault-tolerant network with appropriate load balancing
  - Performance of a CDN is typically characterized by the response time (i.e. latency) perceived by the end-users
  - Slow response time is the single greatest contributor to users abandoning content and web sites and processes
  - The performance of a CDN is affected by
    - Distributed content location
    - Switching mechanism
    - Data replication and caching strategies
    - Reliable functions and network connectivity



Blending SDN & NFV for the Virtualized CDN (vCDN)

- SDN Network Control
  - Centralized control
  - Dynamic connectivity
  - Elastic bandwidth
- NFV Flexibility, Performance & Predictability
  - Performance: Mean Response Time, Latency, Hit Ratio Percentage, Number of Completed Requests, Rejection rate and Mean CDN load
  - Dimensioning: remaining stable whatever the use of virtualized HW resources for CDN components
  - Resource management: allow the right balance of network i/o to CPU power to storage i/o performance (e.g., RAM and HDD)
- Efficient use of resources (storage)
  - Fulfil specific storage density requirements, e.g. to cache a large catalog of popular content
- Deployment & Operational tools
  - Compliance of cache nodes with main monitoring and reporting requirements (e.g., JSON, YANG, SNMP, syslog, etc.) so that operator is able to manage different types of cache nodes together for a Delivery Service
- Content Management
  - Ability to select specific cached content (e.g., video/HTTP) and replicate/duplicate these selected content items during delivery via virtual switching to a Quality of Experience (QoE) virtualized function without degrading the overall performance of the virtualized CDN function

Yes, but.
Is it actually being used/developed?

- EC-Funded Projects investigating, using and/or developing ABNO
  - FI-PPP XIFI
  - FP7 OFERTIE
  - FP7 DISCUS
  - FP7 CONTENT
  - FP7 PACE (ict-pace.net) - Next Steps for the Path Computation Element
- Deployments and Code Availability
  - iONE, Universitat Politècnica de Catalunya (UPC) (OpenSource)
  - ANM, Telefonica (OpenSource)
  - Infinera (Closed Proof of concept)
- Publications & Standards
  - A PCE-Based Architecture for Application-Based Network Operations, IETF RFC7491
  - Unanswered Questions in the Path Computation Element Architecture, IETF RFC7399
  - "In-Operation Network Planning", IEEE Communications Magazine
  - "Adaptive Network Manager: Coordinating Operations in Flex-grid Networks", ICTON (IEEE)
  - "Experimental Demonstration of an Active Stateful PCE performing Elastic Operations and Hitless Defragmentation", ECOC European Conference on Optical Communications
  - "Planning Fixed to Flexgrid Gradual Migration: Drivers and Open Issues", IEEE Communications Magazine
  - "Dynamic Restoration in Multi-layer IP/MPLS-over-Flexgrid Networks", IEEE Design of Reliable Communication Networks (DRCN)
  - "A Traffic Intensity Model for Flexgrid Optical Network Planning under Dynamic Traffic Operation", OSA Optical Fiber Communication (OFC)
  - **And many, many more…**

Future research and investigations
Where next?

- Commercializing ABNO for video distribution and storage
  - Ongoing re-use of components and protocols, and extending where necessary, makes sense
- Implementing a controller as a distinct unit with the SDN architecture provides a number of benefits:
  - Scaling
  - Load-balancing
  - Resilience
  - Multiple forwarding technology support (various optic flavours)
  - Rapid advancement of algorithms, the next disruptive wave of innovation will be Machine Learning (ML)
- Therefore, SDN should be seen as a critical part of a wider view of network operation
- However, gaps exist!

Assuming a basis for a controller

How to do we integrate into the orchestrator?

- Application specific orchestration layer needed
  - Can this ever be generalized to be application agnostic?
  - How might we define the service?
  - Are service information models are available?
- Optimization being performed in multiple layers
  - If using PCE, where should the PCE element(s) actually be located?
  - Is the PCE a candidate for Network Functions Virtualisation (NFV)?
  - How do we scale, load balance and ensure resilience for?
  - Speed at which path computations are provided
- Can paths be determined and provisioned any quicker?
- How can we combine offline (planning) and online (real-time) requests?
- Multi-layer support (packet layer over flexible optical networks)
  - Placement of video services at both the packet and optical layer (bandwidth dependent)
- Application of Policy/Intent when computing computation paths and configuring equipment

Standards, Open Source? Both?

- Standards, or Open Source?
  - A future of development only via Open API's risks user/operator ability to influence technology and specification progress, unless they are embedded in the project
- The role of Standards Development Organizations (SDO) is a question of relevant, not existence
  - It typically takes >2 years for SDOs to formalize a standard
- Open Source SDN has been incredibly successful
  - Network programmability
  - Management and operations (IT & NFV Orchestration)
  - However
    - Vendor bias
    - Small communities (underfunded monocultures)
    - Potential for security flaws
    - Fragmentation (many OSS projects that each solve 20% of a problem but cannot be used together)

Thank You!

Any comments or questions are welcome.

Daniel King
Senior Researcher, Lancaster University (BT & Intel, NFV Co-Lab)
Co-Chair of SDN Internet Research Task Force (IRTF)
Open Networking Foundation (ONF) Research Associate
d.king@lancaster.ac.uk
daniel.king@bt.com

Andrew Lord
Head of Optical Research, British Telecom
andrew.lord@bt.com

The Role of SDN and NFV for Flexible Optical Networks: Current Status, Challenges and Opportunities

Daniel King, A. Farrel, and Nektarios Georgalas
Lancaster University, United Kingdom, d.king@lancaster.ac.uk
Old Dog Consulting, United Kingdom, adrian@olddog.co.uk
British Telecom, United Kingdom, nektarios.georgalas@bt.com

ABSTRACT

Today's optical transport domains are typically built using fixed grid technology. They are statically configured and operationally intensive to manage, lacking the capability for dynamic services and elastic bandwidth. Recent research has established the benefits of flexible grid technologies for optical switching allowing dynamic and elastic management of the available bandwidth resources. Combined with Software Defined Networks (SDN) control principles and Network Functions Virtualization (NFV) infrastructure, we have the potential to fundamentally change the way we build, deploy and control network applications built on top of flexible optical networks.

This paper outlines the current Internet Engineering Task Force (IETF) developments for standardizing flexible grid optical technologies, and discusses how software-defined and function virtualisation principles have and will continue to provide the key capabilities to further enable flexible optical switching technologies to control and deliver NFV-based services and applications. It addition it describes the benefits for the virtual Content Distribution Network (vCDN) use case when combined with an IETF's SDN framework Application-Based Network Operations (ABNO). Finally, we highlight the research opportunities for furthering the application of SDN and NFV for control and orchestration of flexible optical networks using the IETF ABNO-based framework.

Keywords: flexible optical switching, software defined networks, SDN, network function virtualisation, NFV, application-based network operations, ABNO, network control, orchestration, flexi-grid, virtual content distribution network, vCDN.

1. INTRODUCTION

Optical transport networks are evolving rapidly from current static Dense Wavelength Division Multiplexing (DWDM) systems towards flexible and elastic optical switching, using flexible grid transmission schemes and dynamic switching technologies. In such an environment, a data plane connection is switched based on allocated, variable-sized frequency ranges within the optical spectrum creating what is known as a flexible grid (flexi-grid) [1]. This approach aims to utilise technology to increase both the scalability and agility of the optical network, allowing resource optimisation and scaling of bandwidth as demands change in bandwidth requirements.

The flexi-grid optical switching technology creates a need to develop innovative network control and orchestration mechanisms to reduce deployment and operational complexity, and maximize benefits of flexi-grid capabilities. While control plane approaches based on Generalized Multiprotocol Label Switching (GMPLS) are being developed [2], Network Management System (NMS) control remains popular within the transport network community. Traditional NMS platforms lack the flexibility to fully enable flexi-grid so we needed to look towards the architecture and principles defined by the Software Defined Networking (SDN) architecture developed and ratified by the Open Networking Foundation (ONF) [3]. These core SDN architectural principles offer a variety of possibilities when looking to plan, control, and manage flexible network resources both centrally and dynamically. Solutions exist that encompass direct control of switching resources from a central orchestrator, distributed control through a set of controllers, or devolved control through a hybrids with an active control plane.

The advent of Network Functions Virtualisation (NFV) [4] will provide the ability to deploy network functions on virtualised infrastructure hosted on commodity hardware, decoupling dedicated network function from proprietary hardware infrastructure. Consequently this allows network function to be instantiated from a common resource pool and to exploit performance predictability where dimensioning remains stable whatever the use of virtualised hardware resources. Emboldened with the suitable control and orchestration tools, these virtual and on-demand capabilities could have a significant impact on how telecom infrastructure is managed.

Most recently (March 2015), the Internet Engineering Task Force (IETF) published the Application-Based Network Operations (ABNO) framework as RFC7491 [5]. The ABNO framework provides a generic toolkit for a variety of network technologies and use cases. In its most basic form it describes how specific, well-defined functional components may be brought together within a single architecture to provide the capability to control a range of forwarding technologies in order to set-up and tear-down end-to-end services. However, it also provides a variety of deployment options to support a range of architectural principles including: programmatic control of optical and packet-optical transport elements; centralised or distributed deployment models; and northbound and southbound interfaces.

978-1-4673-7880-2/15/$31.00 ©2015 IEEE

When we combine the elements described previously (flexi-grid, SDN, and NFV), we are able to consider a range of capabilities and functions that may be achieved using a unified framework.

In this paper we discuss the key design objectives necessary to build a unified framework underpinned by the flexible optical network platform for providing NFV-based use cases, these are derived from state-of-the-art developments across Standard Development Organisations (SDOs) combined with considerations of emerging technologies. As a result of this analysis we are able to highlight gaps and discuss the current challenges and opportunities for using SDN (ABNO) and NFV, to further empower the development and deployment of flexible optical networks. We also demonstrate the applicability of this unified architecture based on a virtual Content Distribution Network (vCDN) use case. Finally, we describe the state of the art, open source contributions and framework research gaps.

2. NEXT GENERATION FLEXIBLE TRANSPORT NETWORKS

In the current etymology of transport networks, dynamic and flexible optical resources are increasingly seen as a method to provide bespoke bandwidth, scale, distribution, and flexibility to match user demands. However, these are rarely real-time capabilities as they require significant engineering resources, and often lack the flexibility for dynamic scenarios. With the combination of flexi-grid, SDN, NFV, and ABNO the capability to programmatically control resources and scale with given user bandwidth demand becomes feasible, providing resilient and elastic network capability in response to both real-time and predicted demands. This section outlines the core requirements, design principles, and enabling architecture for achieving use case requirements and design goals, and sets out the interfaces and protocols that facilitate deployment of infrastructure to meet these objectives.

2.1 Flexible Optical Switching

Flexible optical switching was defined by the International Telecommunications Union Telecommunications Standardization Sector (ITU-T) Study Group 15 [6] and refers to the updated set of nominal central frequencies (a frequency grid), channel spacing and optical spectrum management/allocation. A principle of flexi-grid is the "frequency slot"; a variable-sized optical frequency range that can be allocated to a data connection. Compared to a traditional fixed grid network, which uses fixed size optical spectrum frequency ranges or frequency slots with various channel separations, a flexible grid network can select its media channels with a more flexible choice of slot widths, allocating optical spectrum as required and available.

A flexible optical network will be constructed from DWDM subsystems that include links, tunable transmitters/receivers, and electro-optical network elements. It is assumed that, for our unified framework, we will require control of the media layer within the DWDM network, and of the adaptations at the signal layer: specifically defining the resource as a Spectrum-Switched Optical Network (SSON) and managing them using a distributed signaling mechanism from a centralized control architecture

2.1.1 Control Plane Resource Modeling

Flexible optical resources (transmitters and receivers) may have different tunability constraints, and media channel matrixes may have switching restrictions. A set of common constraints have been defined in [1], these are described below:

- Slot widths: The minimum and maximum slot width
- Granularity: The optical hardware may not be able to select parameters with the lowest granularity (e.g., 6.25 GHz for nominal central frequencies, or 12.5 GHz for slot width granularity);
- Available frequency ranges: The set or union of frequency ranges that have not been allocated (i.e., are available). The relative grouping and distribution of available frequency ranges in a fiber is usually referred to as "fragmentation";
- Available slot width ranges: The set or union of slot width ranges supported by media matrices. It includes the following information:
  - Slot width threshold: The minimum and maximum Slot Width supported by the media matrix. For example, the slot width could be from 50 GHz to 200 GHz;
  - Step granularity: The minimum step by which the optical filter bandwidth of the media matrix can be increased or decreased. This parameter is typically equal to slot width granularity (i.e., 12.5 GHz) or integer multiples of 12.5 GHz.

2.1.2 End to End Service

An "end-to-end service" may be characterized by one or a set of required effective frequency slot widths. This does not preclude that the request may add additional constraints such as imposing the nominal central frequency. A given effective frequency slot may be requested for the media channel in the control plane setup messages, and a specific frequency slot can be requested on any specific hop of the service setup. We will use the Label Switch Path (LSP) construct as the representation of a media channel and therefore “service”, and the LSP is assumed to be comprised of a nominal frequency and connects the endpoints (transceivers) including the cross-connects at the ingress and egress nodes.

2.2 Software Defined Networks

The key principles of Software Defined Networking (SDN) include:

- Programmatic and abstracted interaction with the network. These interactions include: control, provisioning, configuration, management, and monitoring;
- Use of an SDN Controller to exercise the aforementioned programmatic direct control of forwarding behavior.

Use for an SDN Controller and Programmable control facilitates network behaviour to be implemented and modified quickly and cohesively: automation techniques may be used to set up end-to-end services, with flexibility beyond the initial deployment, and with the capability to modify paths and network function nodes to be modified (torn down, resized, relocated) at any time particularly in response to rapid changes in the operational environment. This includes revised network conditions, fluctuations in the resource location or availability, and in the event of partial or catastrophic failure.

2.2.1 Application-Based Network Operations (ABNO)

The ABNO framework document [5] 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.

Within the framework resides the ABNO Controller which represents the main component of the architecture and is responsible for orchestrating the workflows and invokes the necessary components in the right order. ABNO is able store the workflows in a repository, and then execute the network operations, such as setting up or tearing down services, via the GMPLS provisioning plane for flexi-grid resources.

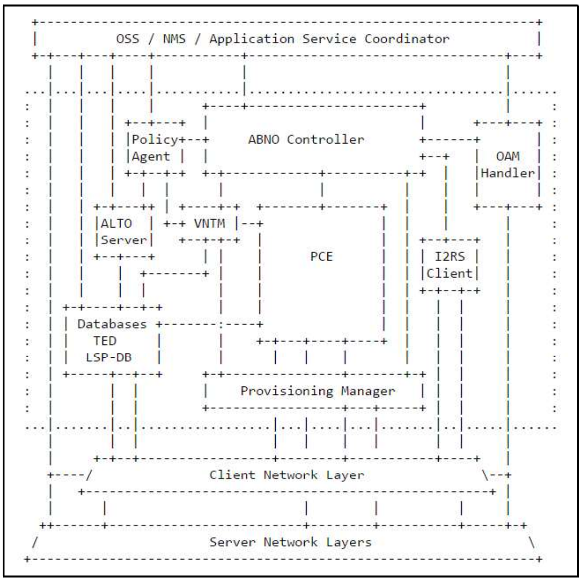


Figure 1. Generic ABNO architecture.

2.3 Network Functions Virtualisation

Network functions virtualisation (NFV) is used to leverage Information Technology (IT) virtualisation techniques to migrate entire classes of network functions typically hosted on proprietary hardware onto virtual platforms based on general compute and storage servers [7]. Each virtual function node is known as a Virtualised Network Function (VNF), which may run on a single or set of Virtual Machines (VMs), instead of having custom hardware appliances for the proposed network function.

Content delivery, especially of video, is one of the major challenges of all operator networks due to massive growing amount of traffic. Delivery of terrestrial transmissions over fixed networks is proving to be a huge consumer of bandwidth. The table 1 illustrates the Serial Digital Interface (SDI) bandwidth requirements for a variety of uncompressed interfaces and stream types.

Table 1. SDI bandwidth requirements.

Interface Type	Video Stream	Bitrate
SD-SDI	480i/576i	270 Mbit/s
HD-SDI	720p/1080i	1.5 Gbit/s
3G-SDI	1080p	3 Gbit/s
6G UHD-SDI	4K 30 fps	6 Gbit/s
12G UHD-SDI	4K 60 fps	12 Gbit/s
24G UHD-SDI	4K 12 0fps	24 Gbit/s

2.3.1 Content Delivery Requirements

A Content Delivery Network (CDN) is a generic term describing a set of common components, such as: Cache Controller, Cache Nodes, Surrogate Server, Load Balancer, Proxy, and Peering Gateway. Normally the Cache Controller will select a Cache Node (or a pool of Cache Nodes) for answering to the end-user request, and then redirect the end-user to the selected Cache Node. The Cache Node shall answer to the end-user request and deliver the requested content to the end user. The CDN Controller is a centralized component, and CDN Cache Nodes are distributed within the network or situated within a Data Centre [8].

For industry, core requirements when designing and deploying a CDN include: capital cost-efficiency, flexibility of content fulfilment, performance predictability, and bandwidth or latency guarantees. These requirements would be well serviced with a server-layer comprised of flexible optical network technology.

2.3.2 OpenCache: Content Caching Platform

OpenCache [9] has been identified as a candidate open source vCDN platform, leveraging existing SDN research and embracing industrial demand for virtualising network functions. These principles directly impacted the OpenCache architecture, and enabled its use and manipulation within virtualised environments. A key facet of this architecture is the API-based control of caching function (instantiation, resize, and tear-down).

Figure 2 represents a virtualised Contend Distribution Network (vCDN) running on commodity hardware over a flexible optical network.

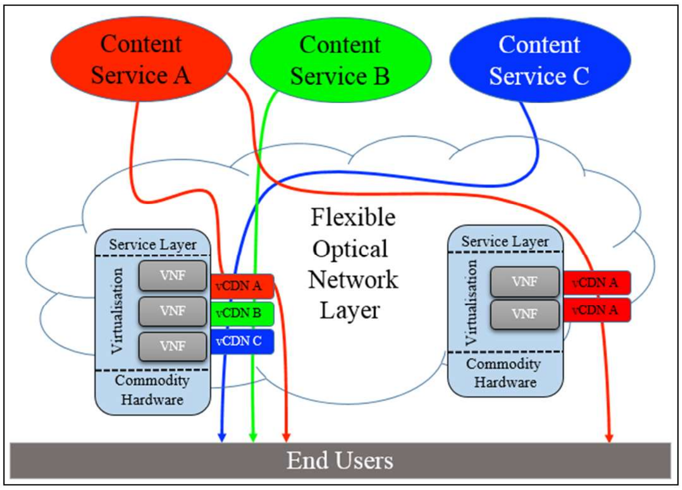


Figure 2. vCDN application running over flexible optical network.

Furthermore, this virtualisation allows multiple isolated VNFs or unused resources to be allocated to other VNF-based applications during weekdays and business hours, facilitating overall IT capacity to be shared by all content delivery components, or even other network function appliances. Industry, via the European Telecommunications Standards Institute (ETSI), has defined a suitable architectural framework [7], and has also documented a number resiliency requirements [10] and specific objectives for virtualised CDN infrastructure [11].

A final fundamental requirement is the need for the CDN to be resilient and reliable, beyond the capability to cope with a Distributed Denial of Service (DDOS) attack, the CDN must be capable of recovering from catastrophic failure that may affect the aforementioned CDN components [12].

Clearly, there is a need for an experimental platform to drive and develop the next-generation of CDN infrastructure for delivering future SDI steams up to 24 Gbit/s, led by both academia and industry, over a flexible optical network. The rest of this paper outlines a converged SDN and NFV architecture in support of programmable elastic optical networks to support NFV-based applications, based on the vCDN use case described previously.

3. CONVERGED SDN AND NFV ARCHITECTURE

Figure 3 (Blending Network Control & NFV Management based on ETSI NFV Reference Architectural Framework). It demonstrates a proposed converged SDN and NFV architecture facilitating programmable control of flexible optical network resources, for the NFV-based vCDN use case.

The combined SDN & NFV architecture is comprised of two elements: Network Control & NFV Management. The Network Control element is underpinned with the ABNO Controller for programmable control of the optical network. The NFV Management is split into VNF Manager (vCDN Controller) and Virtual Infrastructure Manager (OpenStack) providing the hypervisor and virtualisation layer.

The central component is the NFV Infrastructure itself and these functional components and functions are mapped into interfaces within the unified SDN and NFV framework:

- Os-Ma: interface to OSS and handles network service lifecycle management and other functions
- Vn-Nf: represents the execution environment provided by the Vim to a VNF (e.g. a single VNF could have multiple VMs)
- Nf-Vi: interface to the Vim and used for VM lifecycle management
- Ve-Vnfm: interface between VNF and Vnfm and handles VNF set-up and tear-down
- Vi-Ha: an interface between the virtualisation layer (e.g. hypervisor for hardware compute servers) and hardware resources

The dotted lines in Fig. 3 represent missing functional components and interfaces, and represent a research opportunity for orchestration between the SDN to NFV domains.

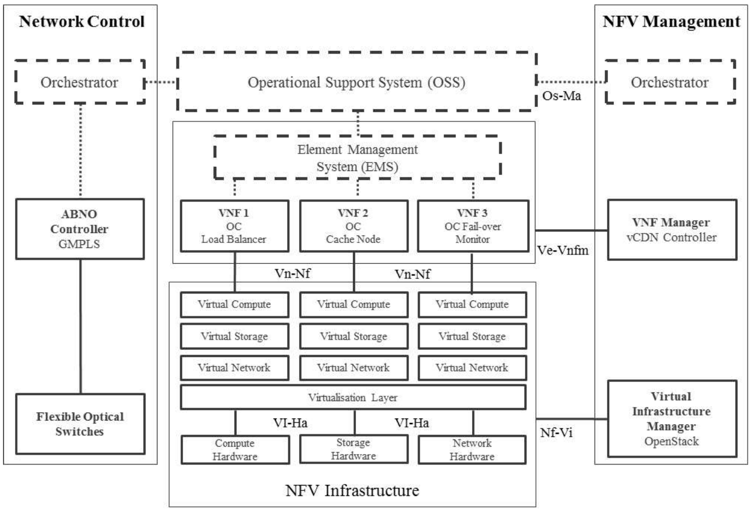


Figure 3. Blending network control & NFV management based on ETSI NFV reference architectural framework.

4. IN SUMMARY

The opportunity exists to combine SDN principles with an NFV-based architecture, providing the capability to deploy a vCDN and scale bandwidth for given user demand. Using an ABNO Controller to manage the flexible optical network, coupled with the required NFV infrastructure components, and OpenCache platform to deliver a resilient and elastic vCDN capability in response to high bandwidth real-time and predicted video stream demands for terrestrial TV services.

4.1 CURRENT STATUS

ABNO has been successfully demonstrated for a variety of flexi-grid network operations, including but not limited to:

- In-Operation Network Planning [13]
- ABNO: a feasible SDN approach for multi-vendor IP and optical networks [14]
- ABNO-based Network Orchestration of end-to-end Multi-layer (OPS/OCS) Provisioning across SDN/OpenFlow and GMPLS/PCE Control Domains [15]
- Adaptive Network Manager: Coordinating Operations in Flex-grid Networks [16]

4.1.1 ROLE OF STANDARDISATION

In order to facilitate industry adoption of the flexi-grid architecture and components outlined in this paper continued development of required flexi-grid standard proposal will be critical, these proposals include:

- Framework for GMPLS based control of Flexi-grid DWDM networks
- Generalized Labels for the Flexi-Grid in LSC Label Switching Routers
- GMPLS OSPF-TE Extensions in for Flexible Grid DWDM Networks
- RSVP-TE Signaling Extensions in support of Flexible Grid
- Extensions to PCEP for Hierarchical Path Computation Elements (H-PCE)
- A YANG data model for FlexGrid Optical Networks

4.1.2 AVAILIBILITY OF OPEN SOURCE

4.1.2.1 ABNO Interfaces and Controller

Where possible, the interfaces of the ABNO Framework and ABNO Controller itself, described in Fig. 1, have been implemented in Java and are available via the IDEALIST GitHub source code repository [17].

4.1.2.2 Experimental Caching Platform

An Open-Cache implementation is available at [9].

5. FUTURE WORK

Using a prototype implementation of the ABNO Controller with an NFV-based infrastructure, we plan to use OpenCache as the vCDN platform to prove the architecture described in Fig. 3 (Blending Network Control & NFV Management based on ETSI NFV Reference Architectural Framework). However, orchestration between the SDN and NFV domains remains an outstanding technical gap.

REFERENCES

[1] O. Gonzalez de Dios, R. Casellas, *et al.*: Framework and Requirements for GMPLS based control of flexi-grid DWDM networks, IETF Internet-Draft draft-ietf-ccamp-flexi-grid-fwk-03, 2015, work in progress.

[2] F. Zhang, X. Zhang, A. Farrel, *et al.*: RSVP-TE signaling extensions in support of flexible grid", IETF Internet-Draft, draft-ietf-ccamp-flexible-grid-rsvp-te-ext-01, 2015, work in progress.

[3] Software Defined Networking Architecture Overview, ONF TR-504, 2014.

[4] Network Functions Virtualisation – White Paper #3, ETSI, 2014.

[5] D. King and A. Farrel: A PCE-based Architecture for Application-based Network Operations, IETF RFC 7491, 2015.

[6] International Telecommunications Union, ITU-T Recommendation G.694.1 (revision 2_, "Spectral grids for WDM applications: DWDM frequency grid", Feb. 2012.

[7] Network Functions Virtualization (NFV); Architectural Framework, ETSI GS NFV 002, 2014.

[8] B.M. Moreno, C.P. Salvador, M.E. Domingo, I.A. Pena, and V.R. Extremera: On content delivery network implementation, *Computer Communications*, 29(12):2396-2412, 2006.

[9] M. Broadbent: OpenCache; An Experimental Caching Platform https://github.com/broadbent/opencache

[10] Network Functions Virtualization (NFV); Resiliency Requirements, ETSI GS NFV 001, 2015.

[11] Network Functions Virtualization (NFV); Use Cases, ETSI GS NFV 001, 2013.

[12] P. Aranda, D. King, and M. Fukushima: Virtualization of content distribution network use case, Internet-Draft draft-aranda-vnfpoolcdn-use-case-00, IETF Secretariat, October 2014.

[13] L. Velasco, D. King, O. Gerstel, R. Casellas, A. Castro, and V. López: In-operation network planning, *IEEE Communications Magazine*, 2014.

[14] A. Aguado, V. López J. Marhuenda, Ó. González de Dios, and J. P. Fernández-Palacios: ABNO: A feasible SDN approach for multi-vendor IP and optical networks, *Journal of Optical Communications and Networking*, Feb. 2015.

[15] R. Muñoz, R. Vilalta, R. Casellas, R. Martínez, F. Francois, M. Channegowda, A. Hammad, S. Peng, R. Nejabati, D. Simeonidou, N. Yoshikane, T. Tsuritani, V. López, and Achim Autenrieth: Experimental assessment of ABNO-based network orchestration of end-to-end multi-layer (OPS/OCS) provisioning across SDN/OpenFlow and GMPLS/PCE control domains, in *Proc. European Conference on Optical Communication (ECOC)*, Sep. 2014.

[16] V. Lopez, O. Gerstel, R. Casellas, A. Farrel, D. King, S. López-Buedo, and J. Fernandez-Palacios: Adaptive network manager: Coordinating operations in flex-grid networks, in *Proc. Transparent Optical Networks (ICTON)*, Jun. 2013.

[17] IDEALIST ABNO Repository available at https://github.com/telefonicaid/netphony-network-protocols.

Elastic Optical Networks Architectures, Technologies, and Control

Springer Publishing, 2016

Chapter 10. Application-Based Network Operations (ABNO)

Daniel King (Editor), Lancaster University
Victor Lopez, Telefonica
Oscar Gonzalez de Dios, Telefonica
Ramon Casellas, CTTC
Nektarios Georgalas, British Telecom
Adrian Farrel, Old Dog Consulting

Networks today integrate multiple technologies allowing network infrastructure to deliver a variety of services to support the different characteristics and dynamic demands of applications. There is an increasing goal to make the network responsive to service requests issued directly from the application layer and high-layer client interfaces. This differs from the established model where services in the network are instantiated in response to management commands driven by a human user using a wide variety of Operational Support Systems (OSS), and where networks are typically over-provisioned to ensure minimal traffic loss, even at peak traffic periods.

10.1 General Concepts

An idealized network resource controller would be based on an architecture that combines a number of technology components, mechanisms and procedures. These include:

- Policy control of entities and applications for managing requests for network resource information and connections
- Gathering information about the resources available in a network
- Consideration of multi-layer resources and how topologies map to underlying network resources
- Handling of path computation requests and responses
- Provisioning and reserving network resources
- Verification of connection and resource setup

10.1.1 Network Abstraction

A major purpose of Software Defined Networks (SDN) is to bury complexity and make service deployment and overall network operation simpler without invoking the management and provisioning software of the many manufacturers deployed in the network. Consequently, allowing higher-layer applications to automate requests and creation of services simpler and more direct.

10.1.2 Logically Centralized Control

We use the term “logical centralized” to signify that network control may appear focused in a single entity, independent of its possible implementation in distributed form. The centralized control principle states that resources can be used more efficiently when viewed from a global perspective.

A centralized SDN controller would be able to orchestrate resources that span a number of subordinate domains or in cooperation with other entities, and thereby offer resource efficiency when setting up services and overall operation of network resources. Other reasons for logically centralized control include scale, optimization of information exchange and minimization of propagation delay.

Given constraints of not being able to always deploy green field networks it is necessary that a controller co-exist with both native SDN forwarding technologies (OpenFlow) non-native SDN traffic engineered technology (MPLS, GMPLS, etc.).

10.1.3 Application Driven Use-Cases

Dynamic 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) and optical transport networks, to Software Defined Networks (SDN) forwarding technologies, application-driven use cases include:

- **Virtual Private Network (VPN) Planning** –Support and deployment of new VPN customers and resizing of existing customer connections across packet and optical networks
- **Optimization of Traffic Flows** – Applications with the capability to request and create overlay networks for communication connectivity between file sharing servers, data caching or mirroring, media streaming, or real-time communications
- **Interconnection of Content Delivery Networks (CDN) and Data Centers (DC)** – Establishment and resizing of connections across core networks and distribution networks
- **Automated Network Coordination** – Automate resource provisioning, facilitate grooming and regrooming, bandwidth scheduling, and concurrent resource optimization
- **Centralized Control** – Remote network components allowing coordinated programming of network resources through such techniques as Forwarding and Control Element Separation (ForCES) OpenFlow (OF)

An SDN Controller framework for network operator environments must combine a number of technology components, mechanisms and procedures, including:

- Policy control of entities and applications for managing requests for network resource information and connections
- Gathering information about the resources available in a network.
- Consideration of multi-layer resources, and how these topologies map to underlying network resources
- Handling of path computation requests and responses
- Provisioning and reserving network resources
- Verification of connection and resource setup

The overall objective is develop a control and management architecture of transport networks to allow network operators to manage their networks using the core principles of Software Defined Networks to allow high-layer applications and clients to request, reconfigure and re-optimize the network resources in near real time, and in response to fluid traffic changes and network failures.

This chapter outlines the core network control principles required for application-based network operations of transport networks, discusses key control plane principles and architectures. It introduces the Application-Based Network Operations (ABNO) Framework [1], and how this framework and functional components and how they are combined for Adaptive Network Manager (ANM) [2], used to address the requirements for operating Elastic Optical Networks (EON) [3]. Finally, the chapter provides a view of the research challenges and areas for investigation to continue development of Transport SDN, and control of EONs.

10.2 Network Control

A central principle of SDN is the separation of a network forwarding and control planes (Fig. 1). By separating these functions, a set of specific advantages in terms of centralized or distributed programmatic control. Firstly, there is a potential economic advantage by using commodity hardware rather than proprietary specific hardware. Secondly, remove the need for a fully distributed control plane with capability often requiring senior engineering experience to deploy and operate, with a wide range of features, which are very often underutilized. Thirdly, the ability to consolidate in one or a few places what is often a considerably complex piece of OSS software to configure and control network resources.

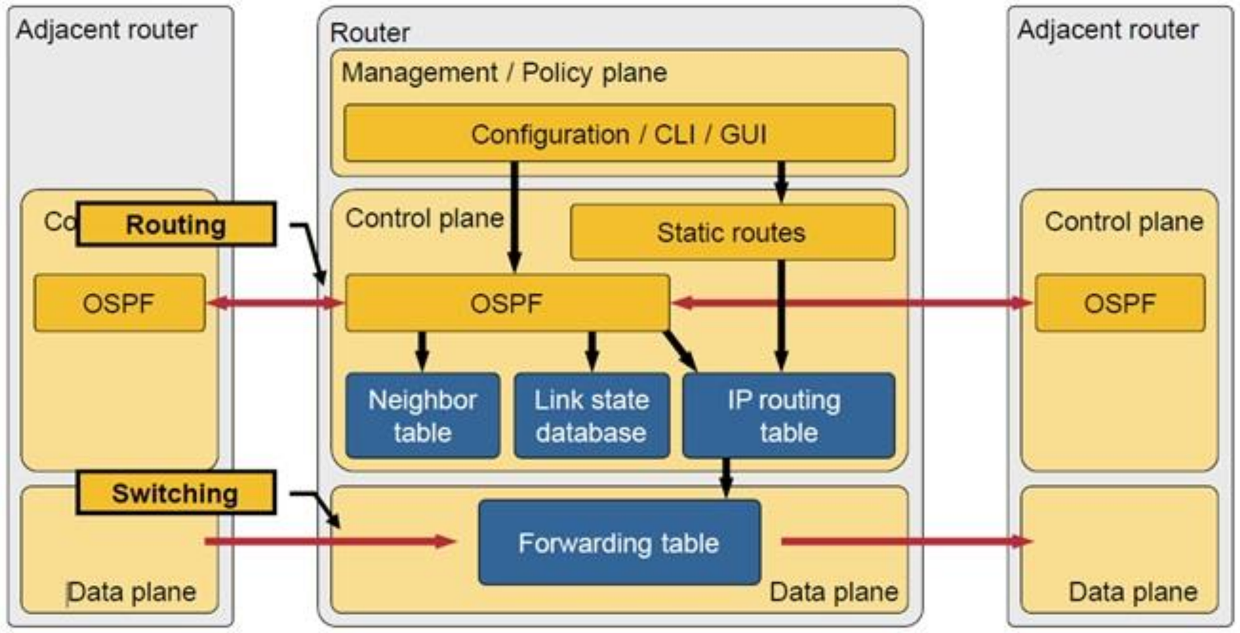


Fig. 1 Management, Control and Forwarding Example

Typically, the network operator has followed a prescribed path for hardware upgrade to circumnavigate the networking scaling issues. This requires the operator to consider the node forwarding performance versus price-to-performance numbers to pick just the right time to participate in an upgrade. Conversely, as network topologies increase the complexity of the control plane and scalability also need consideration.

The Internet represents an example of a significant scaling problem. Vast numbers of administrative regions loosely tied with the interconnections changing constantly as traffic patterns fluctuate and failures occur. Therefore, to address the control paradigm the Internet was designed accordingly. Its structure was federated, where individual nodes participate together to distribute reachability information in order to develop a localized view of a consistent, loop-free network using IP forwarding. The Internet forwarding paradigm, where routes and reachability information is exchanged that later results in data plane paths being programmed to realize those paths, however paths are often sub-optimal and prone to traffic congestion, so clearly this approach has weaknesses which might be addressed using a centralized approach.

As network technology evolved and the concepts of SDN were invented (centralized control, superstation of control and forwarding, and network programmability), the cycle of growth and scaling management and upgrade in the control plane to accommodate scale, was a clear objective. It is much easier to pursue solutions for a centralized management environment controlling distributed, but simple, forwarding elements.

Control Plane

The control plane is the part of the node architecture that is concerned with establishing the network map. Control plane functions, such as participating in routing protocols, are control elements. This establishes the local rule set used to create the forwarding table entries, interpreted by the data plane, to forward traffic between incoming and outgoing ports on a node (Fig. 2). The foundation of the current IP control plane model is to use an Interior Gateway Protocol (IGP). This normally is in the form of a link-state protocol such as Open Shortest Path First (OSPF) or Intermediate-System-to-Intermediate-System (ISIS). The IGP will establish layer 3 reachability between a connected, acyclic graph of IP forwarding elements.

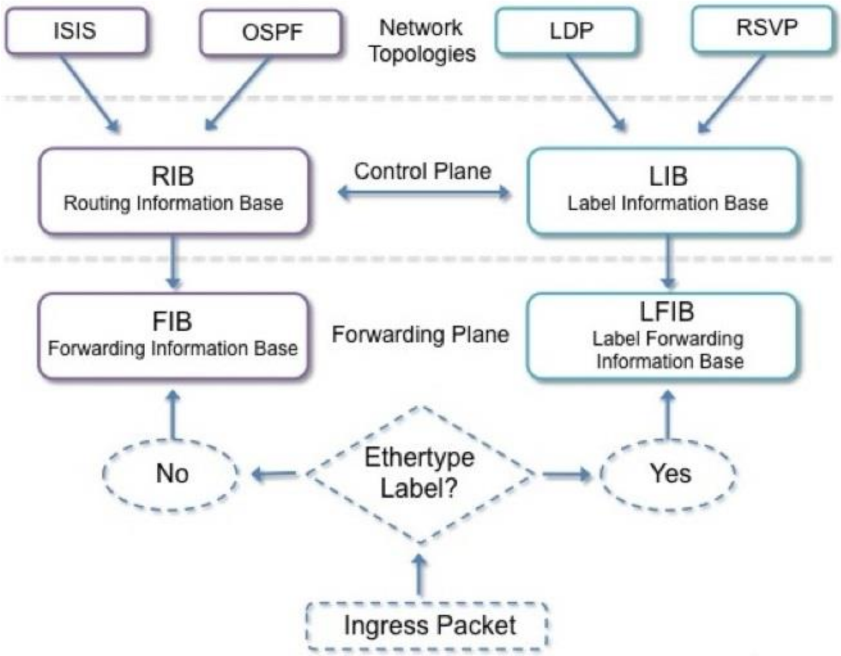


Fig. 2 Relationship of Control and Forwarding Plane

Layer 3 network reachability information primarily concerns itself with the reachability of a destination IP prefix. In all modern uses, layer 3 is used to segment or stitch together layer 2 domains in order to overcome layer 2 scaling problems. In most cases, the routing table contains a list of destination layer 3 addresses and the outgoing interface(s) associated with them. Control plane logic can define certain traffic rules, for priority treatment of specific traffic for which a high quality of service is defined known as differentiated services. Forwarding focuses on the reachability of network addresses.

The role of the control plane includes:

- Network topology discovery (resource discovery)
- Signaling, routing, address assignment
- Connection set-up/tear-down
- Connection protection/restoration
- Path Computation & Traffic engineering

Management Plane

The Management Plane is responsible for managing the control plane. It performs a number of responsibilities, including configuration management and applying policy. It also provides Fault Management, Performance Management, Accounting and Security Management functions.

In their early deployments, optical transport networks were inherently managed, deployed in a single administrative domain, and locked to a single vendor hardware solution (i.e., arranged into *vendor islands*). Such small and mid-sized networks, in terms of number of nodes, were relatively homogeneous, thus reducing interoperability issues. A single, vendor-specific Network Management System (NMS) was deployed, being responsible for the management of the optical network, tailored to the underlying hardware, and using proprietary interfaces and extensions.

Those systems were perceived as closed, bundled together as a whole, and with a limited set of functionalities that were dependent on a given release. The provisioning of a network connectivity service involved manual processes, where a service activation or modification could involve human intervention, with a user requesting the service provider, which was then manually planning and configuring the route and resources in the network to support the service.

Several challenges motivated the evolution towards the control plane. First, network operators continuously have specific requirements to reduce operational costs, while ensuring that the network still meets the requirements of the supported services. Second, the manual, long-lasting processes associated to NMS-based networks did not seem adapted for the dynamic provisioning of services with recovery and Quality of Service (QoS). In short, the introduction of a dynamic control plane was justified, from an operational perspective, for the automation of certain tasks, freeing the operator from the burden of manually managing and configuring individual nodes, leading to significant cost reductions.

In this context, the introduction of a control plane aims at fulfilling the requirements of fast and automatic end-to-end provisioning and re-routing of flexi-grid connections, while supporting different levels of quality of service. Regardless, of the actual technology, a control plane needs to address common functions like addressing, automatic topology discovery, network abstraction, path computation, and connection provisioning, as stated earlier in this chapter. From a high level perspective, and as any software system that automates tasks and processes, the functions of a control plane can, from a simplistic point of view, be distributed or centralized, although we will later see that this separation is becoming blurry. This dichotomy applies not only from a functional perspective, but also from a resource allocation perspective. Both models are viable; both have their own strengths and weaknesses, and both are being extended to address the new requirements associated to the aforementioned emerging optical technologies, such as flexible spectrum allocation, efficient co-routed connection setup and configuration of related optical parameters. Thus, the selection of a centralized or distributed control plane is conditioned by diverse aspects, such as the desired functions, flexibility and extensibility, availability, etc., as well as by more concrete aspects such as the inherent constraints of the optical technology (e.g., the need to account for physical impairments which are collected from monitoring systems and not standardized), already installed deployments, and actual network size and scalability.

The network elements participating in distributed control plane environment exchange the accumulated advertisements from other nodes in a state database (e.g., OSPF database) and run a Dijkstra (shortest path) algorithm to establish a reachability graph of best paths to destinations. This process uses a distributed flooding algorithm within the IGP protocol procedure to propagate attachment information, thus, all nodes speaking a particular IGP protocol in the domain remain connected to each other (directly or indirectly) and participate with timely reachability information and establish a network topology, that reports change in connectivity in the event of failure. A key aspect is thus convergence, which is the time it takes from when a network element introduces a change in reachability of a destination due to a network. A variety of methods exist in various IGP mechanisms and procedures to address scaling of the control plane state (memory and CPU) in the network, both for physical and logical design. These the tools include summarization, filtering, recursion and segregation.

10.2.1 Control Elements for Operating Optical Networks

Path Computation

Path computation manages aspects related to finding a physical route between two network nodes, commonly referred to as endpoints. Path computation is a functional component of a control plane, invoked for the purposes of (dynamic) provisioning, re-routing, restoration, as well as advanced use cases such as overall optimization, adaptive network planning or, in the particular case of DWDM flexi-grid networks, spectrum de-fragmentation.

Service Provisioning

This would include the node and interface configuration, specifically known as service provisioning. The set up and tear down of connections. The control element would automatically configure the required hops between the source and destination nodes required to create a connection between two (or point to multi-point) points in the network. The procedure and protocols used via the controller to configure different elements to set up a connection is known as either distribute via the signaling mechanisms available (such as RSVP-TE), or direct using a flow provision process (such as OpenFlow).

OAM and Performance Monitoring

Operations, Administration, and Maintenance (OAM) is often used as a general term to describe a collection of tools for fault detection and isolation, and for performance measurement. Many OAM tools and capabilities have been defined for various technology layers [4].

OAM tools may, and quite often do, work in conjunction with a control plane and management plane. OAM provides instrumentation tools for measuring and monitoring the data plane. OAM tools often use control-plane functions, e.g., to initialize OAM sessions and to exchange various parameters. The OAM tools communicate with the management plane to raise alarms, and often OAM tools may be activated by the management plane (as well as by the control plane), e.g., to locate and localize problems, and initiate performance measurement of an optical segment, or end-to-end service.

10.3 Distributed and Centralized Control Planes

10.3.1 Control Plane architecture evolution

In their early deployments, optical transport networks were inherently managed, deployed in a single administrative domain, and locked to a single vendor hardware solution (i.e., arranged into *vendor islands*). Such small and mid-sized networks, in terms of number of nodes, were relatively homogeneous, thus reducing interoperability issues. A single, vendor-specific Network Management System (NMS) was deployed, being responsible for the management of the optical network, tailored to the underlying hardware, and using proprietary interfaces and extensions.

Those systems were perceived as closed, bundled together as a whole, and with a limited set of functionalities that were dependent on a given release. The provisioning of a network connectivity service involved manual processes, where a service activation or modification could involve human intervention, with a user requesting the service provider, which was then manually planning and configuring the route and resources in the network to support the service.

Several challenges motivated the evolution towards the control plane. First, network operators continuously have specific requirements to reduce operational costs, while ensuring that the network still meets the requirements of the supported services. Second, the manual, long-lasting processes associated to NMS-based networks did not seem adapted for the dynamic provisioning of services with recovery and QoS. In short, the introduction of a dynamic control plane was justified, from an operational perspective, for the automation of certain tasks, freeing the operator from the burden of manually managing and configuring individual nodes, leading to significant cost reductions.

In this context, the introduction of a control plane aims at fulfilling the requirements of fast and automatic end-to-end provisioning and re-routing of flexi-grid connections, while supporting different levels of quality of service. Regardless, of the actual technology, a control plane needs to address common functions like addressing, automatic topology discovery, network abstraction, path computation, and connection provisioning, as stated earlier in this chapter. From a high level perspective, and as any software system that automates tasks and processes, the functions of a control plane can, from a simplistic point of view, be distributed or centralized, although we will later see that this separation is becoming blurry. This dichotomy applies not only from a functional perspective, but also from a resource allocation perspective. Both models are viable; both have their own strengths and weaknesses, and both are being extended to address the new requirements associated to the aforementioned emerging optical technologies, such as flexible spectrum allocation, efficient co-routed connection setup and configuration of related optical parameters. Thus, the selection of a centralized or distributed control plane is conditioned by diverse aspects, such as the desired functions, flexibility and extensibility, availability, etc., as well as by more concrete aspects such as the inherent constraints of the optical technology (e.g., the need to account for physical impairments which are collected from monitoring systems and not standardized), already installed deployments, and actual network size and scalability.

For example, the Internet represents an example of a significant scaling problem. Vast numbers of administrative regions are loosely tied with the interconnections changing constantly as traffic patterns fluctuate and failures occur. To address this, the Internet control paradigm was designed to be distributed.

On the other hand, SDH/Optical core transport networks, while geographically spanning national or continental regions, are still relatively small in size /number of elements when compared to IP networks, and are commonly under the control of a single entity or operator. Services offered were relatively stable, characterized by long holding times, coupled to slow traffic dynamics, and service provisioning delays of the order of days/ weeks was acceptable. Such deployments models were, arguably, best addressed with a centralized control paradigm.

While the need of a control plane does not seem to present significant opposition, the choice of the technology is still debatable. From a historical perspective, the evolution of the control plane for optical networks started augmenting NMS based networks with a distributed control plane, based on the ASON (Automatically Switched Optical Networks) [5] [6] [7] architecture with Generalized Multi-Protocol Label Switching GMPLS [8] suite of protocols, as detailed next. Recently, the application of Software Defined Networking (SDN) principles to the control of optical networks is presented as a means to enable the programmability of the underlying network (in any case, the formal separation of the data and control planes is a key concept in optical network control). To some extent, there is an analogy between a Transport SDN architecture and a centralized NMS, although the former insists on using modern system architectures, open and standard interfaces, and flexible and modular software development.

Distributed Control

In this setting, the control plane is implemented by a set of cooperating entities (control plane controllers) that execute processes that communicate. Control plane functions such as topology management, path computation or signaling are distributed (for the first one, each node disseminates the topological elements that are directly under its control, and the IGP routing protocol enables the construction of a unified view of the network topology. Path computation is carried out by the ingress node of the connection and signaling is distributed along the nodes involved in the path). The protocols ensure the coordination and synchronization functions, autonomously (although commonly, the provisioning of a new service is done upon request from a NMS).

The reference architecture is defined by the ITU-T, named ASON enabling dynamic control of an optical network, automating the resource and connection management. ASON relies of the GMPLS set of protocols defined by the IETF (with minor variations). In short, the ASON/GMPLS architecture defines the transport, control and management planes. In particular, the control plane is responsible for the actual resource and connection control, and consists of Optical Connection Controllers (OCC), interconnected via Network to Network Interfaces (NNIs) for network topology and resource discovery, routing, signaling, and connection setup and release (with recovery). The Management Plane is responsible for managing and configuring the control plane and fault management, performance management, accounting and security.

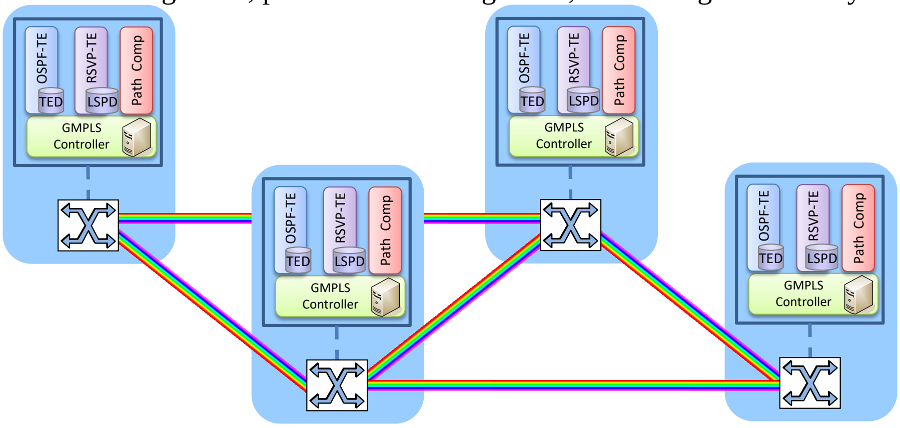


Fig. 3 Example of GMPLS-controlled optical network

As seen in Fig. 3, the main involved processes are the Connection Controller (CC) and the Routing Controller (RC), and optionally a path computation component. A data communication network, based on IP control channels (IPCC) to allow the exchange of control messages between GMPLS controllers, is also required, which can be deployed in-band or out-of-band (including, for example, a dedicated and separated physical network). A GMPLS-enabled node (both control and hardware) is named Label Switched Router (LSR). Each GMPLS controller manages the state of all the connections (i.e., Label Switched Path - LSPs) originated, terminated or passing-through a node, stored in the LSP Database (LSPDB), and maintains its own network state information (topology and resources), collected in a local Traffic Engineering Database (TED) repository.

The network elements participating in distributed control plane environment exchange the accumulated advertisements from other nodes in a state database (e.g. OSPF database) and run a Dijkstra (shortest path) algorithm to establish a reachability graph of best paths to destinations. This process uses a distributed flooding algorithm within the IGP protocol procedure to propagate attachment information, thus, all nodes speaking a particular IGP protocol in the domain remain connected to each other (directly or indirectly) and participate with timely reachability information and establish a network topology, that reports change in connectivity in the event of failure. A key aspect is thus convergence, which is the time it takes from when a network element introduces a change in reachability of a destination due to a network change, such as a failure. A variety of methods exist in various IGP mechanisms and procedures to address scaling of the control plane state (memory and CPU) in the network, both for physical and logical design. These the tools include summarization, filtering, recursion and segregation

Centralized Control

In a centralized control, a single entity, usually called controller, is responsible for the control plane functions, commonly using open and standard protocols, such as those defined by the SDN architectures and protocols e.g. OpenFlow protocol (OF/OFP) [19]. The controller performs path computation and service provisioning, and proceeds to configure the forwarding and switching behavior of the nodes. A centralized control plane provides a method for programmatic control of network resources and simplification of control plane process. Deployment and operation of connections requires an interaction with control points to establish the forwarding rules for specific traffic. These are not recent innovations, separation of the control and data planes occurred with the development of ForCES [9] and Generalized Switch Management Protocol (GSMP) [10] many years ago.

By deploying the control plane intelligence in the controller, resources allocated in hardware nodes for CP functions are reduced significantly. Moreover, such solutions involve deploying hardware (computational and storage) in a centralized location which is orders of magnitude more powerful than individual controllers are. Although a centralized controller does not seem significantly different from an NMS, it is worth noting aspects such as the automation of processes, and programmability, as well as the use of open interfaces and standard architectures, terminology, models and protocols. Note that a logically centralized controller may, itself, be implemented as a distributed system, while appearing, programmatically, as a single entity. Finally, SDN principles bring new opportunities such as joint allocation of IT and network resources, or the orchestration of heterogeneous control technologies, or the unified control of access and core network segments.

Comparison of Distributed versus Centralized

In a distributed control approach, individual nodes participate together to distribute reachability information in order to develop a localized view of a consistent, loop-free network. Routes and reachability information is exchanged that later results in data plane paths being programmed to realize those paths, however paths are often sub-optimal and prone to traffic congestion, so clearly this approach has weaknesses which might be addressed using a centralized approach. Mainly, a distributed control plane is affected by the latencies in the propagation and synchronization of data. Changes occurring at a given network element need to be propagated and the transitory may affect network performance.

On the other hand, in a distributed model, each node element is mainly self-sustained. There is no bottleneck or single point of failure, such a SDN controller, and is the model that seems most appropriate when there is no central authority and functional elements need to cooperate. Each node can survive failures at other nodes as long as the network remains connected.

The benefits of a centralized model are lower capital and operational cost, involving, in the case of a control plane, minimal control plane hardware and software at each node, while enabling computational scaling at the controller location. A centralized controller may be easier to implement, given the tight coupling of components, and the less stringent requirements of internal interfaces not subject to interoperability issues. It simplifies automation and management, enables network programmability and it is less subject to latencies and out-of-date information due to the need of synchronizing entities. It provides more flexibility, a single point of extension for operators' policies and customizations, and improved security. There is less control plane overhead, and arguably, network security is increased, with less complexity and greater control over potential risk areas. The downside is that centralized elements are always points of failure.

Hybrid Control plane models

In view of the current trends and evolutions of control plane architectures, it seems too simplistic to tag a control plane as distributed or centralized. Control plane architectures are evolving towards hybrid control-plane models, in which some elements may be centralized and some elements may be distributed, sometimes following the mantra "distribute when you can, centralize when you must". Even if a given control plane entity is centralized, it can be logically centralized, where a system is implemented in terms of the composition of functional components that appear as one. A given function can be centralized in a given domain (e.g. the path computation function can be centralized in a Path Computation Element (PCE) assuming a single PCE per domain deployment model, but the same function can be distributed amongst several children PCE in Hierarchical PCE (H-PCE) architecture [15] within a multi-domain scenario.

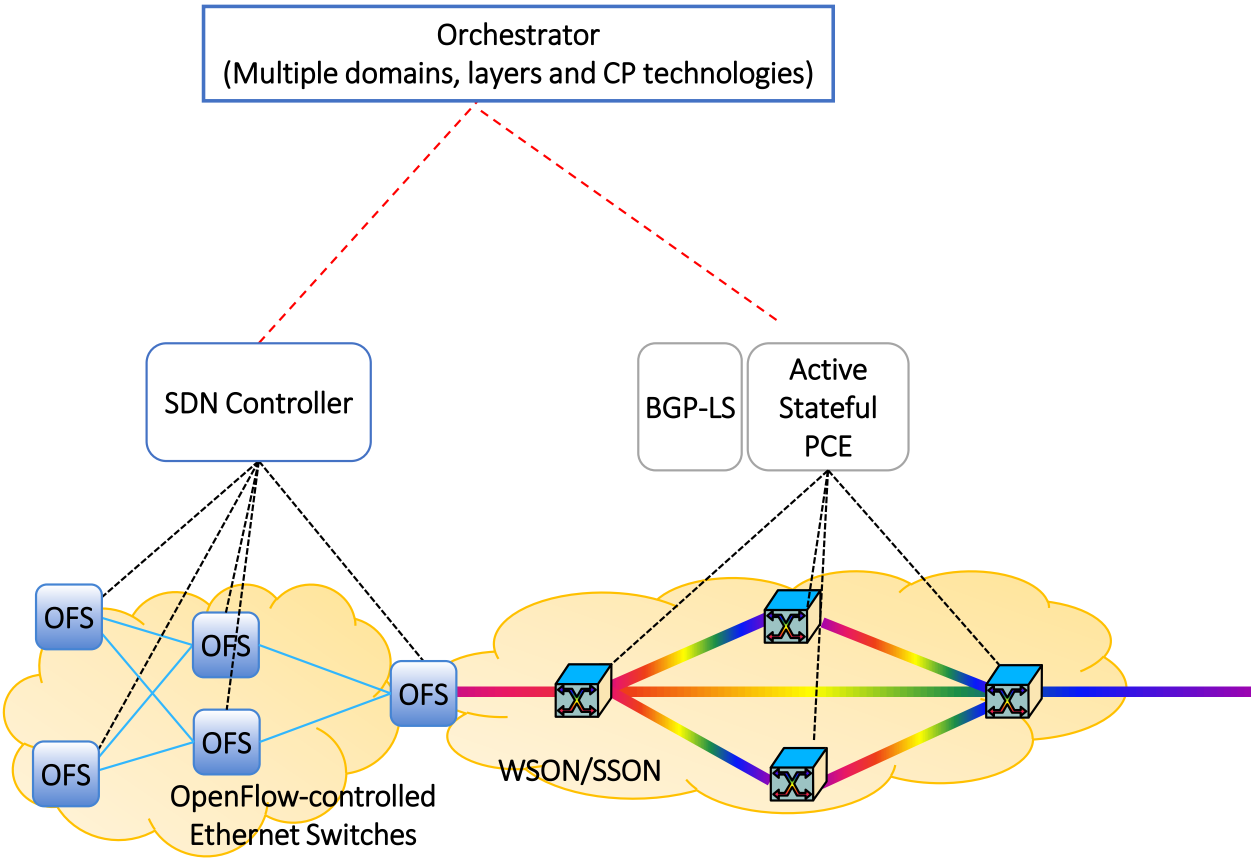


Fig. 4 The use of an orchestrator for the over-arching control of heterogeneous control technologies

New use cases, such as remote data center interconnection, highlight the need for multi-domain service provisioning and heterogeneous CP interworking, potentially requiring an overarching control (see Fig. 4). Additionally, network operators aim at addressing the joint control and allocation of network and IT resources (e.g. networking, computing and storage resources), or the joint optimization of different network segments, such as access, aggregation and core. Different alternatives, with varying degrees of integration and flexibility, are available: straightforward approaches characterized by the adaptation of one control model to the other or more advanced interworking requiring the definition of common models (e.g. a subset of attributes for network elements) and of coordination and orchestration functions. Such orchestrator may in turn, be (logically or physically) centralized while delegating specific functions, to subsystems that may be distributed (such as the provisioning of connectivity delegated to a GMPLS control plane) [8].

Finally, let us mentions that the adoption of new computing and interworking models, and concepts, such as those of server consolidation, host virtualization or Network Function Virtualization (NFV), are challenging common approaches and existing practice: for example a GMPLS control plane could be run as a Virtual Network Function running in a datacenter, for legacy purposes, in which a distributed system could run on a centralized physical infrastructure.

10.4 Framework for Application-Based Network Operations (ABNO)

The three tenants of SDN are programmability, the separation of the control and data planes, and the management of ephemeral network state in a centralized control model [1], regardless of the degree of centralization. In an ideal world, it should be possible to utilize a distributed control plane as well, providing the best practices of centralized control, and distributed control plane for ephemeral state management.

Application-Based Network Operations (ABNO) was designed using the following architectural principles:

1. Loose Coupling: For ease of implementation and fast development, we do not attempt to tightly integrate the functional components of the network controller. Instead, we use well-defined APIs and protocol mechanisms.
2. Low Overhead: The goal is to ensure that each management and control function is not duplicated, which reduces the overall platform overhead.
3. Modular: A modular design enables easier composition of existing features into new capabilities.
4. Intelligent: Designing the framework around the Path Computation Element and Traffic Engineered principles, provides significant benefits for controlling a range of network technologies and maximizing resource utilization.
5. Resource Management: The framework allows for various network and node state to be discovered and stored. This state information is collected using the protocol mechanisms provided by traditional and already existing network and service management tools.
6. Dynamic Management: A key goal of an SDN controller is actuate dynamic control based on application demands and other network events.
7. Policy Control: It is important to implement policy management to provide the mechanisms for specifying connection requirements (e.g., QoS, security) for various applications. It also allows network operators to associate different service levels.
8. Technology Agnostic: The ABNO framework communicates with the network nodes using a variety of Southbound APIs and protocols. Allowing for a wide variety of forwarding mechanisms to be managed using ABNO.

Fig. 5 presents an example of network architecture using ABNO.

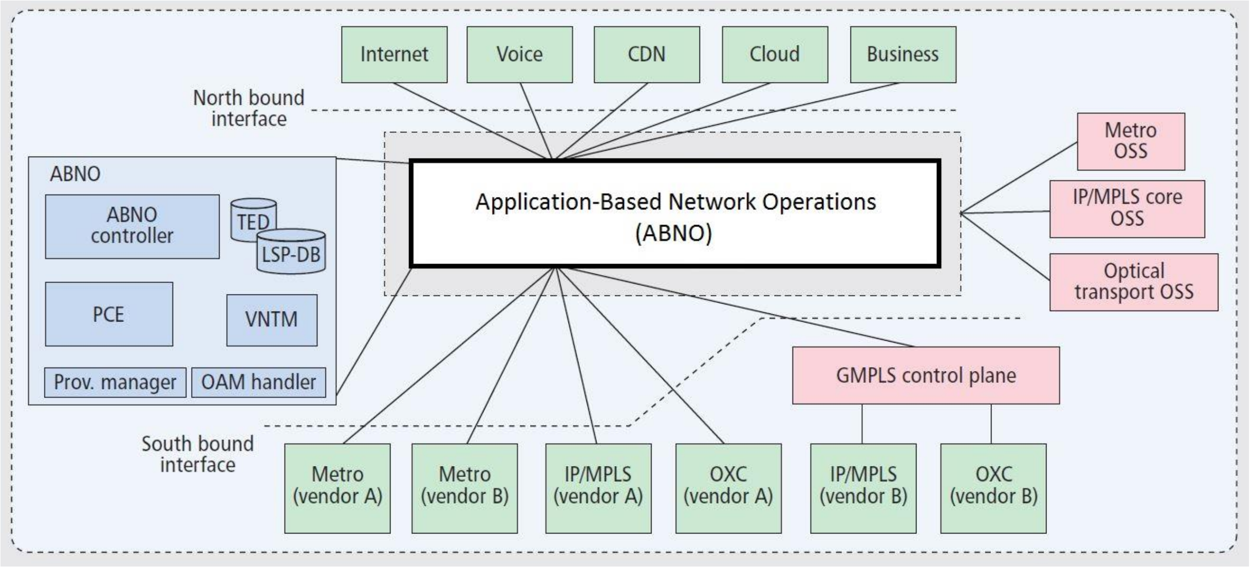


Fig. 5 ABNO Architecture Example

10.4.1 Functional Components

NMS and OSS

A Network Management System (NMS) or an Operations Support System (OSS) can be used to control, operate, and manage a network. Within the ABNO framework, 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) [11] 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 nodes within the network.

Application Service Coordinator

The Application Service Coordinator communicates with the ABNO Controller to request operations on the network. Requests may be initiated from entities such as 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 that a user uses to make requests to 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 ABNO architecture discussion, 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.

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.

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. The Policy Agent is responsible for propagating those policies into the other components of the system. Simplicity in this discussion necessitates leaving out many of the policy interactions that will take place. In our example, the Policy Agent is only discussed interacting with the ABNO Controller, in reality 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) [12], and the Interface to the Routing System (I2RS) Client will also be a PEP as noted in [13].

OAM Handler

Operations, Administration, and Maintenance (OAM) plays a critical role in understanding how a network is operating, detecting faults, sand 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 [4], such as monitoring and testing.

Path Computation Element (PCE)

The Path Computation Element (PCE) 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 a database (the LSP) containing information about the LSPs that are provisioned and operational within the network.

Additional functionality 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 [14]. This function may directly access the network elements or channelled through the Provisioning Manager. Coordination between multiple PCEs operating on different TEDs can prove useful for performing path computation in multi-domain or multi-layer networks. A domain in this case might be an Autonomous System (AS), thus enabling inter-AS path computation.

In the latter case, the ABNO controller will need to request an optimal path for the service. If the domains (ASes) require path setup to preserve confidentiality about their internal topologies and capabilities, they will not share a TED and subsequently each domain (AS) will operate its own PCE. In such a situation, the Hierarchical PCE (H-PCE) architecture, described in [15], is necessary.

Network Database

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

Typically the IGP (like OSPF-TE or IS-IS-TE) are responsible for generating and disseminating the TED within a domain. In multi-domain environments, it may be necessary to export the TED to another control element, such as a PCE, which can perform more complex path computation and optimization tasks.

Virtual Network Topology Manager (VNTM)

A Virtual Network Topology (VNT) is defined 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 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.

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 nodes.

10.4.2 South Bound Interfaces (SBI)

The network devices maybe configured or programmed directly from the NMS/OSS. Many protocols already exist to perform these functions, including the following:

- SNMP [16]
- The Network Configuration Protocol (NETCONF) [17], [21]
- RESTCONF [18]
- ForCES [9]
- OpenFlow [19]
- PCEP [20]

The role of the protocols described is to assign state to the forwarding element, either by programming each node individually or via a distributed signaling mechanism. Indeed the previous list is not an exhaustive representation of protocol methods and procedures available, and over time, new forwarding mechanisms will be developed. Therefore, the ABNO framework has been designed to be forwarding mechanism agnostic.

10.5 Adaptive Network Manager

The European Commission funded project “IDEALIST” identified the need for a control architecture to combine the best of distributed routing and signaling protocols, to provide real-time adaption and to survive against failures, and a centralized intelligence that, on the one hand, provides a point for optimization (e.g. interfacing with the planning tool), and also capable of interfacing with the higher-applications, including cloud platforms and data center (WAN) inter-connections.

The distributed functions are based on the well-known GMPLS architecture, while the centralized intelligence and interface with applications follows a SDN approach. Thus, the Adaptive Network Manager (ANM) is the IDEALIST network controller (based on the ABNO framework) [22], that considers not only the Flexi-grid Network (the main focus of IDEALIST), but a wider scope, a multi-layer IP/MPLS over optical Network.

10.5.1 Interfaces

As the ABNO architecture was generic in its intent, most of the interfaces are defined as concepts. In ANM architecture HTTP/JSON interfaces will be used in these interfaces not already defined (Fig. 6). There are two reasons: easy development and flexibility for the workflows definition. These interfaces will help to have a modular design, which can be adapted to the future requirements that may come during the project. If during the project, there are some other solutions in the standardization fora, this have been assessed and where applicable, included in the ANM architecture.

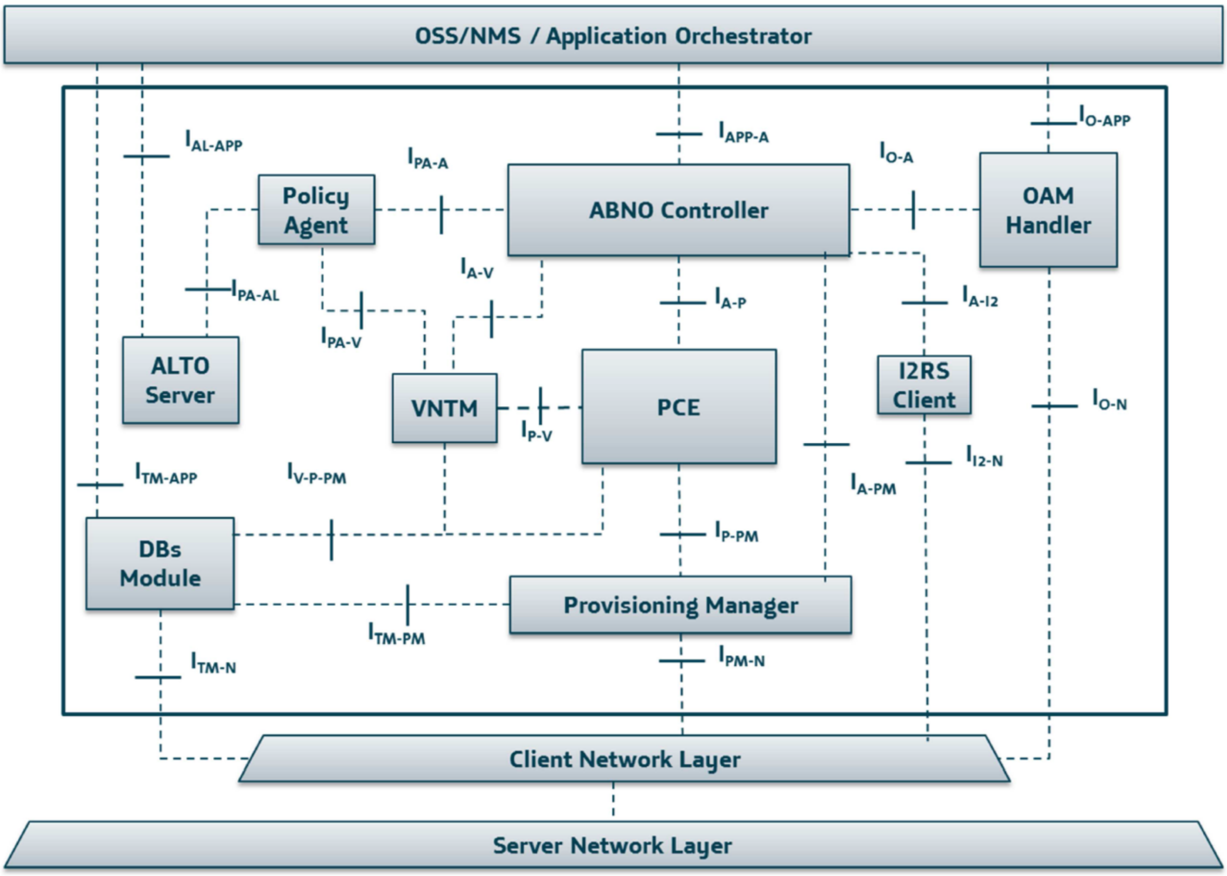


Fig. 6 Adaptive Network Manager Functional Components and Interfaces

- **IN-APP -** This is the interface between the application layer/NMS/OSS and the ABNO controller. Application layer makes requests to set up connections or to trigger any other workflow using HTTP/JSON. This interface is current under development in the Internet Engineering Task Force (IETF). The parameters of the request change depending on the workflow, but the operation type is always mandatory.
- **IAL-APP -** This is the interface between the ALTO Server and Application layer/NMS/OSS, where the Application layer acts as an ALTO Client. They communicate using the ALTO Protocol [23]. They communicate over HTTP/JSON. An information model has to be defined for this interface to support TED, LSPs and inventory requests.
- **IA-I2, II2-N -** The Interface to the Routing System (I2RS)
- **IPA-A, IPA-V, IPA-AL -** All the interfaces between the Policy Agent and the modules that request it for permission using a HTTP/JSON request.
- **IA-P -** This is the interface between the ABNO controller and the PCE. The ABNO controller queries the PCE using PCE, Stateless and Stateful PCEs may be used this interface will support requests for both PCEs.
- **IA-V -** This interface connects the ABNO controller and the VNTM. They communicate through PCEP.

10.6 Adaptive Network Manager (ANM) Use Cases

10.6.1 Catastrophic Network Failure

While most networks are designed to survive single failures without affecting customer service level agreements (SLAs), they are not designed to survive large-scale disasters, such as earthquakes, floods, wars, or terrorist acts, simply because of their low failure probability and the high cost of overprovisioning to address such events in today's network.

Since many systems might be affected, large network reconfigurations are necessary during large-scale disaster recovery. The disaster recovery process is similar to that of the virtual topology reconfiguration after a failure. However, multiple optical systems, IP links, and possible routers and OXCs (assuming central offices are affected) may be taken offline during the disaster. Several additional planning and operation requirements in response to largescale disasters are highlighted below:

- Consideration of potential IP layer traffic distribution changes, either using MPLS-TE tunnels or by modification of IP routing metrics, and evaluating benefits based on the candidate topology
- It may be impossible to reach the desired network end state with one-step optimizations. Therefore, two or more step optimizations may be necessary, for example, to reroute some other optical connections to make room for some new connections
- The system must verify that the intermediate configuration after each such step is robust and can support the current traffic and possibly withstand additional outages
- Based on preemption and traffic priorities, it might be desirable to disconnect some virtual links so as to reuse the resources for post-disaster priority connections and traffic

We have described the creation of one disaster recovery plan, but in a real network, there may be several possible plans, each with its pros and cons. The tool must present all these plans to the operator so that the operator can select the best plan, and possibly modify it and understand how it will be behave.

To summarize, the above process consists of several steps:

1. Immediate action by the network to recover some of the traffic
2. Dissemination of the new network state
3. Root cause analysis to understand what failed and why
4. An operator-assisted planning process to come up with a disaster recovery plan
5. Execution of the plan, possibly in multiple steps
6. Reconvergence of the network after each step and in its final state

This scenario for recovering from catastrophic network failures may also be known as "In-Operation Network Planning" [24]. The ANM platform and use cases are also discussed in-depth in the next chapter.

10.7 Next Steps for ABNO-based Control & Orchestration

We can assume that SDN is well-defined as a logically centralized control framework and architecture. It supports the programmability of network functions and protocols by decoupling the data plane from the control plane through a well-defined control SBI protocol. These SBI's existing in many forms, and assist in the hiding of technology or vendor specific forwarding mechanisms. As network evolution continues a new technology area known as "Network Functions Virtualization" (NFV) [25] is developing in parallel to SDN.

The development of NFV is to leverage Information Technology (IT) virtualization techniques to migrate entire classes of network functions typically hosted on proprietary hardware onto virtual platforms based on general compute and storage servers. Each virtual function node is known as a Virtualized Network Function (VNF), which may run on a single or set of Virtual Machines (VMs), instead of having custom hardware appliances for the proposed network function.

Furthermore, this virtualization allows multiple isolated VNFs or unused resources to be allocated to other VNF-based applications during weekdays and business hours, facilitating overall IT capacity to be shared by all content delivery components, or even other network function appliances. Industry, via the European Telecommunications Standards Institute (ETSI), has defined a suitable architectural framework [25], and has also documented a number resiliency requirements and specific objectives for virtualized media infrastructures.

Utilizing the benefits of enabling technologies (i.e. ABNO-based control principles and NFV-based infrastructure), we have the potential to fundamentally change the way we build, deploy and control broadcast services built on top of flexible optical networks allowing dynamic and elastic delivery and high-bandwidth broadcast and media resources.

10.7.1 Control & Orchestration of Virtual Content Distribution Network (vCDN)

Virtualisation of Content Distribution Networks (CDNs) components is a core design principle necessary to create a content network that can be deployed rapidly and in a scalable way. The first element to be virtualized is the cache node itself, and then required services such as content monitors and load balancers [26]. A key requirement of the vCDN is reconfigurable bandwidth as content moved from HD content at 1080p to 4k streams, and demands change based on time of day and week [27]. Deploying the various infrastructure elements of a CDN as a collection of virtual appliances (VNFs) and connecting content and access (user networks) with a flexible optical network infrastructure offers significant benefits.

Fig. 7 describes how an ABO-enabled network controller would integrate with an NFV-based CDN.

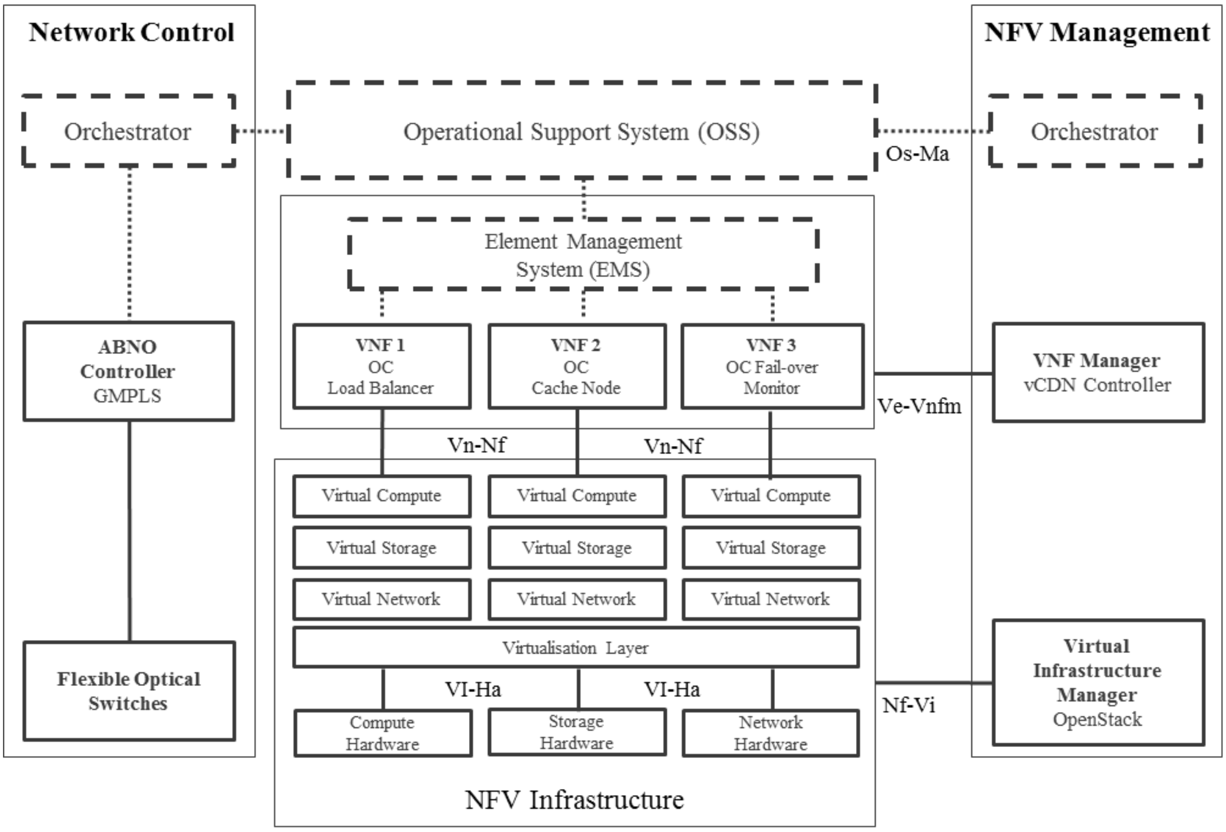


Fig. 7 Candidate SDN & NFV Framework based on ETSI NFV ISG Model

Using the ABNO-based controller in conjunction with the NFV Management and Infrastructure itself would provide the VNFs connectivity over a high-bitrate optical infrastructure, and similar flexibility that exists in the IP and Ethernet layer, which until recently and the advent of Elastic Optical Networks, simply not previously available in optical transport domain.

List of Acronyms

ABNO	Application-based Network Operations
ASON	Automatically Switched Optical Network
BGP-LS	Border Gateway Protocol Link State
GMPLS	Generalized Multi-protocol Label Switching
H-PCE	Hierarchical Path Computation Element
IP/MPLS	Multi-protocol Label Switching over Internet Protocol
LSP	Label-switched Path
LSP-DB	LSP Database
NFV	Network Function Virtualization
NMS	Network Management System
OF	Open Flow
OXC	Optical cross-connect
PCE	Path Computation Element
QoS	Quality of Service
SDN	Software Defined Networking
TE	Traffic Engineering
TED	TE Database

References

[1] D. King, A. Farrel, "A PCE-based Architecture for Application-based Network Operations," IETF Internet RFC 7491, March, 2015.

[2] R. Muñoz, et al., "IDEALIST control and service management solutions for dynamic and adaptive flexi-grid DWDM networks," in Proc. Future Network and Mobile Summit, 2013, 3-5 July, 2013.

[3] Ó. González de Dios, R. Casellas, "Framework and Requirements for GMPLS based control of Flexi-grid DWDM networks," RFC 7698, December, 2015.

[4] N. Sprecher et al., "An Overview of Operations, Administration, and Maintenance (OAM) Tools", RFC 7276, June, 2014.

[5] ITU-T Recommendation G.8080/Y.1304, Architecture for the automatically switched optical network (ASON)

[6] ITU-T Recommendation G.872, Architecture of optical transport networks

[7] ITU-T Recommendation G.709/Y.1331, Interface for the optical transport network (OTN)

[8] E. Mannie Ed., "Generalized Multi-Protocol Label Switching (GMPLS) Architecture", IETF RFC 3945, October 2004.

[9] Halpern, J. and J. Hadi Salim, "Forwarding and Control Element Separation (ForCES) Forwarding Element Model", RFC 5812, March 2010.

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

[11] O. Dugeon, et al., "Path Computation Element (PCE) Database Requirements," IETF Internet Draft draft-dugeon-pce-ted-reqs-03, February 2014, work in progress.

[12] I. Bryskin, et al. "Policy-Enabled Path Computation Framework", RFC 5394, December, 2008.

[13] Atlas, A., Ed., Nadeau, T., Ed., and D. Ward, "Interface to the Routing System Problem Statement", Work in Progress, draft-ietf-i2rs-problem-statement, March 2015.

[14] Crabbe, E., Minei, I., Sivabalan, S., and R. Varga, "PCEP Extensions for PCE-initiated LSP Setup in a Stateful PCE Model", Work in Progress, draft-ietf-pce-pce-initiated-lsp, October, 2015.

[15] King, D., Ed., and A. Farrel, Ed., "The Application of the Path Computation Element Architecture to the Determination of a Sequence of Domains in MPLS and GMPLS", RFC 6805, November 2012,

[16] Case, J., Harrington, D., Presuhn, R., and B. Wijnen, "Message Processing and Dispatching for the Simple Network Management Protocol (SNMP)", STD 62, RFC 3412, December 2002,

[17] R. Enns, et al., "Network Configuration Protocol (NETCONF)," RFC 6241, June, 2011.

[18] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", Work in Progress, draft-ietf-netconf-restconf, July, 2015.

[19] Open Networking Foundation, "OpenFlow Switch Specification Version 1.4.0 (Wire Protocol 0x05)", October 2013.

[20] Vasseur, JP., Ed., and JL. Le Roux, Ed., "Path Computation Element (PCE) Communication Protocol (PCEP)", RFC 5440, March, 2009.

[21] M. Bjorklund, Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)," IETF Request or Comments 6020, October, 2010.

[22] A. Aguado, et al., "ABNO: a feasible SDN approach for multi-vendor IP and optical networks," in OFC, Th3I.5, March, 2014.

[23] Seedorf, J. and E. Burger, "Application-Layer Traffic Optimization (ALTO) Problem Statement", RFC 5693, October, 2009.

[24] L. Velasco, D. King, O. Gerstel, R. Casellas, A. Castro, and V. López, In-Operation Network Planning, IEEE Communications Magazine, 2014.

[25] ETSI GS NFV 002. Network Functions Virtualization (NFV); Architectural Framework, 2014.

[26] ETSI GS NFV 001. Network Functions Virtualization (NFV); Use Cases, 2013.

[27] Broadbent, M.; King, D.; Baildon, S.; Georgalas, N.; Race, N., "OpenCache: A software-defined content caching platform," in Network Softwarization (NetSoft), April, 2015.

Internet Engineering Task Force (IETF)
Request for Comments: 7491
Category: Informational
ISSN: 2070-1721

D. King
Old Dog Consulting
A. Farrel
Juniper Networks

March 2015

A PCE-Based Architecture for Application-Based Network Operations

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 these types of requirements is said to have Application-Based Network Operations (ABNO). ABNO brings together many existing technologies and may be seen as the use of a toolbox of existing components enhanced with a few new elements.

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

Copyright Notice

Table of Contents

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 Operations (ABNO).

The Path Computation Element (PCE) [RFC4655] was developed to provide path computation services for GMPLS- and MPLS-controlled networks. The applicability of PCEs 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) [I2RS-Arch], or through the control plane coordinated through the PCE Communication Protocol (PCEP) [PCE-Init-LSP].

- 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, and to facilitate traffic grooming and regrooming, bandwidth scheduling, and Global Concurrent Optimization using PCEP [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 System (NMS) and Operations Support System (OSS) deployments as well as with more recent developments in 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 ongoing 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 Appendix A.

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 interprocess 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 Label Switched Path 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 Operations, Administration, and Maintenance (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 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

Figure 1 illustrates the ABNO architecture. The components and functional interfaces are discussed in Sections 2.3.1 and 2.3.2, respectively. The use cases described in Section 3 show how different components are used selectively to provide different services. It is important to understand that the relationships and interfaces shown between components in this figure are illustrative of some of the common or likely interactions; however, this figure does not preclude other interfaces and relationships as necessary to realize specific functionality.

```
    +----------------------------------------------------------------+
    |          OSS / NMS / Application Service Coordinator           |
    +-+---+---+----+-----------+---------------------------------+---+
      |   |   |    |           |                                 |
   ...|...|...|....|...........|.................................|......
   :  |   |   |    |      +----+----------------------+          |     :
   :  |   |   | +--+---+  |                           |      +---+---+ :
   :  |   |   | |Policy+--+     ABNO Controller       +------+       | :
   :  |   |   | |Agent |  |                           +--+   |  OAM  | :
   :  |   |   | +-+--+-+  +-+------------+----------+-+  |   |Handler| :
   :  |   |   |   |  |      |            |          |    |   |       | :
   :  |   | +-+---++ | +----+-+  +-------+-------+  |    |   +---+---+ :
   :  |   | |ALTO  | +-+ VNTM |--+               |  |    |       |     :
   :  |   | |Server|   +--+-+-+  |               |  | +--+---+   |     :
   :  |   | +--+---+      | |    |      PCE      |  | | I2RS |   |     :
   :  |   |    |  +-------+ |    |               |  | |Client|   |     :
   :  |   |    |  |         |    |               |  | +-+--+-+   |     :
   :  | +-+----+--+-+       |    |               |  |   |  |     |     :
   :  | | Databases +-------:----+               |  |   |  |     |     :
   :  | |   TED     |       |    +-+---+----+----+  |   |  |     |     :
   :  | |  LSP-DB   |       |      |   |    |       |   |  |     |     :
   :  | +-----+--+--+     +-+---------------+-------+-+ |  |     |     :
   :  |       |  |        |    Provisioning Manager   | |  |     |     :
   :  |       |  |        +-----------------+---+-----+ |  |     |     :
   ...|.......|..|.................|...|....|...|.......|..|.....|......
      |       |  |                 |   |    |   |       |  |     |
      |     +-+--+-----------------+--------+-----------+----+   |
      +----/               Client Network Layer               \--+
      |   +----------------------------------------------------+ |
      |      |                         |        |          |     |
     ++------+-------------------------+--------+----------+-----+-+
    /                      Server Network Layers                    \
   +-----------------------------------------------------------------+
```

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 System (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 that a user uses to make requests to 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 it 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 [I2RS-Arch].

2.3.1.5. Interface to the Routing System (I2RS) Client

The Interface to the Routing System (I2RS) is described in [I2RS-Arch]. 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 [I2RS-PS]. 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)

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 a database (the LSP-DB -- see Section 2.3.1.8.2) containing information about the LSPs that are provisioned and operational within the network as described in [RFC4655] and [Stateful-PCE].

Additional functionality 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 [Stateful-PCE] and [PCE-Init-LSP]. This function may directly access the network elements or may be channeled through the Provisioning Manager.

Coordination between multiple PCEs operating on different TEDs can prove useful for performing path computation in multi-domain or multi-layer networks. A domain in this case might be an Autonomous System (AS), thus enabling inter-AS path computation.

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

In the text that follows, specific key components that are consumers of the databases are highlighted. It should be noted that the databases are available for inspection by any of the ABNO components. Updates to the databases should be handled with some care, since allowing multiple components to write to a database can be the cause of a number of contention and sequencing problems.

2.3.1.8.1. Traffic Engineering Database (TED)

The 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-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. A Stateful PCE (see Section 2.3.1.7) is a primary consumer of the LSP-DB.

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 associate 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 cannot 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 that are referenced when operating the network. These databases might include information about, for example, traffic flows and demands, predicted or scheduled traffic demands, link 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 the 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 steer application-layer traffic. ALTO services enable service providers to share information about network locations and the costs of paths between them. The selection criteria to choose between two locations may depend on information such as maximum bandwidth, minimum cross-domain traffic, lower cost to the user, etc.

The ALTO Server generates ALTO views to share information with the Application Service Coordinator so that it can better select paths in the network to carry application-layer traffic. The ALTO views are computed based on information from the network databases, from policies configured by the Policy Agent, and through the algorithms used by the PCE.

Specifically, the base ALTO protocol [RFC7285] defines a single-node abstract view of a network to the Application Service Coordinator. Such a view consists of two maps: a network map and a cost map. A network map defines multiple Provider-defined Identifiers (PIDs), which represent entrance points to the network. Each node in the application layer is known as an End Point (EP), and each EP is assigned to a PID, because PIDs are the entry points of the application in the network. As defined in [RFC7285], a PID can denote a subnet, a set of subnets, a metropolitan area, a Point of Presence (PoP), etc. Each 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. A cost map provides costs between EPs and/or PIDs. The criteria that the Application Service Coordinator uses to choose application routes between two locations may depend on attributes such as maximum bandwidth, minimum cross-domain traffic, lower cost to the user, etc.

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 channeling 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.

As noted at the top of Section 2.3, it is important to understand that the relationships and interfaces shown between components in Figure 1 are illustrative of some of the common or likely interactions; however, this figure and the descriptions in the subsections below do not preclude other interfaces and relationships as necessary to realize specific functionality. Thus, some of the interfaces described below might not be visible as specific relationships in Figure 1, but they can nevertheless exist.

2.3.2.1. Configuration and Programmatic Interfaces

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

- SNMP [RFC3412]

- The Network Configuration Protocol (NETCONF) [RFC6241]

- RESTCONF [RESTCONF]

- The General Switch Management Protocol (GSMP) [RFC3292]

- ForCES [RFC5810]

- OpenFlow [ONF]

- PCEP [PCE-Init-LSP]

The TeleManagement Forum (TMF) Multi-Technology Operations Systems Interface (MTOSI) standard [TMF-MTOSI] was developed to facilitate application-to-application interworking and provides network-level management capabilities to discover, configure, and activate resources. Initially, the MTOSI information model was only capable of representing connection-oriented networks and resources. In later releases, support was added for connectionless networks. MTOSI is, from the NMS perspective, a north-bound interface and is based on SOAP web services.

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 IS-IS TE [RFC5305]). Alternatively, the TED may be fed using link-state distribution extensions to BGP [BGP-LS].

The ABNO system may maintain a single TED unified across multiple networks or may retain a separate TED 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) [I2RS-Arch] 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 combination of 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 [RFC7285] is designed to distribute the abstracted topology used by an ALTO Server and may prove useful for exporting the TED. The 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 [BGP-LS].
- The I2RS [I2RS-Arch] 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 modeling language such as YANG [RFC6020] so that an existing protocol such as NETCONF [RFC6241] or the Extensible Messaging and Presence Protocol (XMPP) [RFC6120] could be used.

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 PCEP [RFC5440]. This facilitates the network setting up LSPs in response to simple connectivity requests, and it allows the network to reoptimize 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 General 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 Transaction Language 1 (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; see [YANG-Rtg].
  - The [TMF-MTOSI] specification provides provisioning, activation, deactivation, and release of resources via the Service Activation Interface (SAI). The Common Communication Vehicle (CCV) is the middleware required to implement MTOSI. The CCV is then used to provide middleware abstraction in combination with the Web Services Description Language (WSDL) to allow MTOSIs to be bound to different middleware technologies as needed.

- 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 Communication Protocol (PCEP) [RFC5440] has been proposed as a suitable protocol for requesting the establishment of LSPs [PCE-Init-LSP]. This works well, because the protocol elements necessary are exactly the same as those 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 component responsible for requesting LSPs is the Provisioning Manager. Other controllers like 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. Similarly, when provisioned resources are modified or taken out of service, the changes in the network need to be understood by the ABNO system. 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 the LSP can be provided through extensions to PCEP as described in [Stateful-PCE] and [PCE-Init-LSP].

- OAM can be commissioned and the results inspected by the OAM Handler as described in Section 2.3.2.14.

- 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 will be based on a combination of NETCONF [RFC6241] and RESTCONF [RESTCONF] with some additional features.

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 standard PCEP 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 low on resources (for example, the capacity on some key link is nearly 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); alternatively, an extension to the PCEP Notify message (PCNtf) [RFC5440] could be made.

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 designed to satisfy I2RS will provide suitable features (see Section 2.3.2.8).

2.3.2.12. ABNO Provisioning Requests

Under some circumstances, the ABNO Controller may make requests directly to the Provisioning Manager. For example, if the Provisioning Manager is acting as an SDN Controller, then the ABNO Controller may use one of the APIs defined to allow requests to be made to the SDN Controller (such as the Floodlight REST API [Flood]). Alternatively, since the Provisioning Manager may also receive instructions from a Stateful PCE, the use of PCEP extensions might be appropriate in some cases [PCE-Init-LSP].

2.3.2.13. 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 functionality in the system might include the policy behavior of the routing and forwarding system, such as:

- ECMP behavior

- Classification of packets onto LSPs or QoS categories.

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. Work will also be needed 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.

2.3.2.14. OAM and Reporting

The OAM Handler must interact with the network 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. NETCONF notifications are described in [RFC5277], and OpenFlow supports a number of asynchronous event notifications [ONF]. Additionally, Syslog [RFC5424] is a protocol for reporting events from the network, and IP Flow Information Export (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 the mechanism used when events are reported from the network; no new protocol is needed, although new data models may be required for technology-independent OAM reporting.

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. The purpose of these examples is to give some concrete material to demonstrate the architecture so that it may be more easily comprehended, and to illustrate that the application of the architecture is achieved by "profiling" and by selecting only the relevant components and interfaces.

Similarly, it is not the intention that this section contain a complete list of all possible applications of ABNO. The examples are intended to broadly cover a number of applications that are commonly discussed, but this does not preclude other use cases.

The descriptions in this section are not fully detailed applicability statements for ABNO. It is anticipated that such applicability statements, for the use cases described and for other use cases, could be suitable material for separate documents.

3.1. Inter-AS Connectivity

The following use case describes how the ABNO framework can be used to 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 AS Border Routers (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.

```
       +--------------+ +-----------------+ +--------------+
       |ASa           | |       ASb       | |          ASc |
       |         +--+ | | +--+       +--+ | | +--+         |
       |         |a1|-|-|-|b1|       |b3|-|-|-|c1|         |
       | +-+     +--+ | | +--+       +--+ | | +--+     +-+ |
       | |s|          | |                 | |          |d| |
       | +-+     +--+ | | +--+       +--+ | | +--+     +-+ |
       |         |a2|-|-|-|b2|       |b4|-|-|-|c2|         |
       |         +--+ | | +--+       +--+ | | +--+         |
       |              | |                 | |              |
       +--------------+ +-----------------+ +--------------+
```

Figure 2: Inter-AS Domain Topology with Hierarchical 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.

```
                              +---------------------+
                              |       NMS/OSS       |
                              +----------+----------+
                                         |
                                         V
               +--------+    +-----------+-------------+
               | Policy +-->-+     ABNO Controller     |
               | Agent  |    |                         |
               +--------+    +-------------------------+
```

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, which 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 PCEs 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 types 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.

```
            +-----------------+
            | ABNO Controller |
            +----+-------+----+
                 |       A
                 V       |
              +--+-------+--+   +--------+
+--------+    |             |   |        |
| Policy +-->-+ Parent PCE  +---+ AS TED |
| Agent  |    |             |   |        |
+--------+    +-+----+----+-+   +--------+
               /     |     \
              /      |      \
       +-----+-+ +---+---+ +-+-----+
       |       | |       | |       |
       | PCE a | | PCE b | | PCE c |
       |       | |       | |       |
       +---+---+ +---+---+ +---+---+
           |         |         |
        +--+--+   +--+--+   +--+--+
        | TEDa|   | TEDb|   | TEDc|
        +-----+   +-----+   +-----+
```

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 [PCE-Init-LSP]. In either case, the provisioning request is sent to the head-end Label Switching Router (LSR), and that LSR signals in the control plane (using a protocol such as RSVP-TE [RFC3209]) to cause the LSP to be established.

```
                            +-----------------+
                            | ABNO Controller |
                            +--------+--------+
                                     |
                                     V
                              +------+-------+
                              | Provisioning |
                              | Manager      |
                              +------+-------+
                                     |
                                     V
                +--------------------+------------------------+
               /                  Network                      \
              +-------------------------------------------------+
```

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 that 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 be responsible for fanning out the requests to the individual network elements.

3c. Provisioning with an Active Parent PCE

The Active PCE is described in Section 2.3.1.7, based on the concepts expressed in [PCE-Init-LSP]. In this approach, the process described in option 3a is modified such that the PCE issues a direct PCEP command to the network, 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 option 3b.

```
                  +-----------------+
                  | ABNO Controller |
                  +----+------------+
                       |
                       V
                    +--+----------+         +--------------+
      +--------+    |             |         | Provisioning |
      | Policy +-->-+ Parent PCE  +---->----+ Manager      |
      | Agent  |    |             |         |              |
      +--------+    +-+----+----+-+         +-----+--------+
                     /     |     \                |
                    /      |      \               |
             +-----+-+ +---+---+ +-+-----+        V
             |       | |       | |       |        |
             | PCE a | | PCE b | | PCE c |        |
             |       | |       | |       |        |
             +-------+ +-------+ +-------+        |
                                                  |
                 +--------------------------------+------------+
                /                  Network                      \
               +-------------------------------------------------+
```

Figure 6: LSP Provisioning with an Active PCE

3d. Provisioning with Active Child PCEs and Segment Stitching

A mixture of the approaches described in options 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]. Philosophers may debate whether the parent PCE in this model is active (instructing the children to provision LSP segments) or passive (requesting path segments that the children provision).

```
                           +-----------------+
                           | ABNO Controller +--------->--------+
                           +----+-------+----+                 |
                                |       A                      |
                                V       |                      |
                             +--+-------+--+                   |
               +--------+    |             |                   |
               | Policy +-->-+ Parent PCE  |                   |
               | Agent  |    |             |                   |
               +--------+    ++-----+-----++                   |
                             /      |      \                   |
                            /       |       \                  |
                       +---+-+   +--+--+   +-+---+             |
                       |     |   |     |   |     |             |
                       |PCE a|   |PCE b|   |PCE c|             |
                       |     |   |     |   |     |             V
                       +--+--+   +--+--+   +---+-+             |
                          |         |          |               |
                          V         V          V               |
               +----------+-+ +------------+ +-+----------+    |
               |Provisioning| |Provisioning| |Provisioning|    |
               |Manager     | |Manager     | |Manager     |    |
               +-+----------+ +-----+------+ +-----+------+    |
                 |                  |              |           |
                 V                  V              V           |
              +--+-----+       +----+---+       +--+-----+     |
             /   AS a   \=====/   AS b   \=====/   AS c   \    |
            +------------+ A +------------+ A +------------+   |
                           |                |                  |
                     +-----+----------------+-----+            |
                     |    Provisioning Manager    +----<-------+
                     +----------------------------+
```

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 Section 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.

```
      +---+   +---+   +---+                   +---+   +---+   +---+
      |P1|---|P2|---|P3|                    |P4|---|P5|---|P6|
      +---+   +---+   +---+                   +---+   +---+   +---+
                        \                  /
                         \                /
                          +---+  +---+  +---+
                          |L1|--|L2|--|L3|
                          +---+  +---+  +---+
```

Figure 8: Multi-Layer Network

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.

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.

```
                          +-------------+-------------+
                          |    Application Service    |
                          |        Coordinator        |
                          +-------------+-------------+
                                        |
                                        V
                +------+   +------------+------------+
                |Policy+->-+     ABNO Controller     |
                |Agent |   |                         |
                +------+   +-------------------------+
```

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.

```
                             +-----------------+
                             | ABNO Controller |
                             +----+------------+
                                  |
                                  V
                +--------+     +--+-----------+   +--------+
                | Policy +-->--+ Packet-Layer +---+ Packet |
                | Agent  |     |      PCE     |   |   TED  |
                +--------+     +--------------+   +--------+
```

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 the 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. Assuming that the policy allows it, the VNTM asks the optical-layer PCE to 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.

```
                                 +------+
                  +--------+     |      |     +--------------+
                  | Policy +-->--+ VNTM +--<--+ Packet-Layer |
                  | Agent  |     |      |     |      PCE     |
                  +--------+     +---+--+     +--------------+
                                     |
                                     V
                               +---------------+   +---------+
                               | Optical-Layer +---+ Optical |
                               |      PCE      |   |   TED   |
                               +---------------+   +---------+
```

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 RSVP-TE [RFC3473]) can be used to provision the LSP. Alternatively, the network resources can be provisioned directly, 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 I2RS).

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.

```
                     +--------+
                     | Packet |
                     |   TED  |
                     +------+-+
                            A
                            |
                           +--+                    +--+
                           |P3|....................|P4|
                           +--+                    +--+
                               \                  /
                                \                /
                                 +--+  +--+  +--+
                                 |L1|--|L2|--|L3|
                                 +--+  +--+  +--+
```

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 Interconnection across Multi-Layer Networks

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 (DCs) 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 lambda switch capable (LSC), packet switch capable (PSC), or time division multiplexing (TDM) services, and the application and usage of the ABNO architecture and Controller enable the required dynamic end-to-end network service provisioning, regardless of underlying service and transport layers.

Consequently, the interconnection of DCs may involve the operation, control, and management of heterogeneous environments: 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-DC networks (packet-level control, lots of routing exceptions), a standardized GMPLS-based architecture would enable dynamic optical resource 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 from the Application Service Coordinator or NMS

   As shown in Figure 13, 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 end points, after considering any applicable policy as defined by the Policy Agent (see Figure 1).

```
                             +---------------------------+
                             |    Application Service    |
                             |     Coordinator or NMS    |
                             +-------------+-------------+
                                           |
                                           V
                   +------+   +------------+------------+
                   |Policy+->-+     ABNO Controller     |
                   |Agent |   |                         |
                   +------+   +-------------------------+
```

Figure 13: Application Service Coordinator Request Management

2. Address 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 end points 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 topology information is disseminated and gathered, there are two possible scenarios:

   2a. The Application Layer Knows the Client Network Layer

       Entities belonging to the application layer may have an interface with the TED or with an ALTO Server allowing those entities to map the high-level end points to network addresses. The mechanism used to enable this address correlation is out of scope for this document but relies on direct interfaces to other ABNO components in addition to the interface to the ABNO Controller.

In this scenario, the request from the NMS or Application Service Coordinator contains addresses in the client-layer network. Therefore, when the ABNO Controller requests the PCE to compute a path between two end points, the PCE is able to use the supplied addresses, compute the path, and continue the workflow in communication with the Provisioning Manager.

2b. The Application Layer Does Not Know the Client Network Layer

In this case, when the ABNO Controller receives a request from the NMS or Application Service Coordinator, the request contains only identifiers from the application-layer address space. In order for the PCE to compute an end-to-end path, these identifiers must be converted to addresses in the client-layer network. This translation can be performed by the ABNO Controller, which can access the TED and ALTO databases allowing the path computation request that it sends to the PCE to simply be contained within one network and TED. Alternatively, the computation request could use the application-layer identifiers, leaving the job of address mapping to the PCE.

Note that in order to avoid any confusion both approaches in this scenario require clear identification of the address spaces that are in use.

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 that issues 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 could use a number of mechanisms to program the network elements, as shown in Figure 14. It learns which technology is used for the actual provisioning at each segment by either manual configuration or discovery.

```
                                  +-----------------+
                                  | ABNO Controller |
                                  +-------+---------+
                                          |
                                          |
                                          V
                      +------+     +------+-------+
                      | VNTM +--<--+      PCE      |
                      +---+--+     +------+-------+
                          |               |
                          V               V
                    +-----+---------------+------------+
                    |       Provisioning Manager       |
                    +----------------------------------+
                      |       |       |       |       |
                      V       |       V       |       V
                    OpenFlow  V    ForCES     V      PCEP
                           NETCONF          SNMP
```

Figure 14: 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 with little or no 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 Reoptimization

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 reoptimization of in-service LSPs.

The process is simple, and the example shown in Figure 15 utilizes a Stateful PCE [Stateful-PCE] to monitor the network and take reoptimization 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 reoptimized under specific conditions according to policy. This allows the ABNO Controller to manage the sequence and prioritization of reoptimizing 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 reoptimized 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 being used correctly, the PCE tears down the old path and may report the reoptimization event to the ABNO Controller.

```
             +---------------------------------------------+
             | OSS / NMS / Application Service Coordinator |
             +----------------------+----------------------+
                                    |
                       +------------+------------+
                       |     ABNO Controller     |
                       +------------+------------+
                                    |
               +------+     +-------+-------+     +-----+
               |Policy+-----+      PCE      +-----+ TED |
               |Agent |     +-------+-------+     +-----+
               +------+             |
                                    |
             +----------------------+----------------------+
            /                    Network                    \
           +-------------------------------------------------+
```

Figure 15: 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 addresses a situation where the service is interrupted, but this interruption 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 leaves so that a make-before-break restoration approach will not cause disruption to the leaves that were not affected by the original failure.

Figure 16 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, reusing as much of the original path as possible, and the PCE sets up the new LSP.

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

```
             +---------------------------------------------+
             | OSS / NMS / Application Service Coordinator |
             +----------------------+----------------------+
                                    |
                       +------------+------------+   +-------+
                       |     ABNO Controller     +---+  OAM  |
                       +------------+------------+   |Handler|
                                    |                +---+---+
                            +-------+-------+            |
                            |      PCE      |            |
                            +-------+-------+            |
                                    |                    |
             +----------------------+--------------------+-+
            /                    Network                    \
           +-------------------------------------------------+
```

Figure 16: 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 17 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 path might be achieved by stitching together an MPLS segment, an Ethernet/VLAN segment, another IP segment, etc.

Otherwise, the process is similar to that for reoptimization as discussed in Section 3.3.1.

```
             +---------------------------------------------+
             | OSS / NMS / Application Service Coordinator |
             +----------------------+----------------------+
                                    |
            +------+   +------------+------------+    +-------+
            |Policy+---+     ABNO Controller     +----+  OAM  |
            |Agent |   |                         +--+ |Handler|
            +------+   +------------+------------+  | +---+---+
                                    |               |     |
                            +-------+-------+    +--+---+ |
                            |      PCE      |    | I2RS | |
                            +-------+-------+    |Client| |
                                    |            +--+---+ |
                                    |               |     |
            +-----------------------+---------------+-----+-+
           /                     Network                     \
          +---------------------------------------------------+
```

Figure 17: The Make-before-Break Path Test and Selection Process

The pseudocode 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 Handler, ABNO Controller, Policy Agent)
  keep sending traffic (Network)
  Test LSP (OAM Handler)
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:

- Routing of new services where the PCE should consider other services or network topology.
- A reoptimization of existing services due to fragmented network resources or suboptimized placement of sequentially computed services.
- 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 provide 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:

- Minimize aggregate Bandwidth Consumption (MBC).

- Minimize the load of the Most Loaded Link (MLL).

- Minimize Cumulative Cost of a set of paths (MCC).

This use case assumes that 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), as shown in Figure 18.

```
                                 +---------------------+
                                 |       NMS/OSS       |
                                 +----------+----------+
                                            |
                                            V
                  +--------+    +-----------+-------------+
                  | Policy +-->-+     ABNO Controller     |
                  | Agent  |    |                         |
                  +--------+    +-------------------------+
```

Figure 18: 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 GCO requests. The PCE uses the appropriate policy for each 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, as shown in Figure 19.

```
                               +-----------------+
                               | ABNO Controller |
                               +----+------------+
                                    |
                                    V
                  +--------+     +--+-----------+   +--------+
                  |        |     |              |   |        |
                  | Policy +-->--+ GCO-Capable  +---+ Packet |
                  | Agent  |     | Packet-Layer |   |  TED   |
                  |        |     |     PCE      |   |        |
                  +--------+     +--------------+   +--------+
```

Figure 19: 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. The response includes a fully computed explicit path for each service (TE LSP).

3. The concurrently computed solution received from the PCE is sent back to the NMS/OSS by the ABNO Controller as a PCEP response, as shown in Figure 20. 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.

```
                   +---------------------+
                   |       NMS/OSS       |
                   +----------+----------+
                              ^
                              |
                   +----------+----------+
                   |    ABNO Controller  |
                   |                     |
                   +---------------------+
```

Figure 20: ABNO Sends Solution to the NMS/OSS

3.5. Adaptive Network Management (ANM)

The ABNO architecture provides the capability for reactive network control of resources relying 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].

EON provides spectrum-efficient and scalable transport by introducing flexible granular traffic grooming in the optical frequency domain. This is achieved using arbitrary contiguous concatenation of the optical spectrum that allows the creation of custom-sized bandwidth. This bandwidth is defined in slots of 12.5 GHz.

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, optical modulation format, and associated reach, or following user requests, and can be achieved in a highly spectrum-efficient and scalable manner. Similarly, OTN provides for flexible and granular provisioning of bandwidth on top of Wavelength Switched Optical Networks (WSONs).

To efficiently use optical resources, a system is required that 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:

- 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 that 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.

- Human: Operators may request ABNO to run an adaptive network planning process via an NMS.

- Periodic: An adaptive network planning process can be run periodically to find an optimum configuration.

An ABNO Controller would receive a request from an 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.

This request for GCO is passed to the PCE, along the lines of the description in Section 3.4. The PCE can 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 the 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 setup/modification process.

This exchange is shown in Figure 21. Note that the figure does not show the interactions used by the OSS/NMS for establishing or modifying LSPs in the network.

```
                           +---------------------------+
                           |        OSS or NMS         |
                           +-----------+---+-----------+
                                       |   ^
                                       V   |
                 +------+   +----------+---+----------+
                 |Policy+->-+     ABNO Controller     |
                 |Agent |   |                         |
                 +------+   +----------+---+----------+
                                       |   ^
                                       V   |
                                 +-----+---+----+
                                 +      PCE     |
                                 +--------------+
```

Figure 21: Adaptive Network Management with Human Intervention

3.5.3. Automated Provisioning Process

Although most network operations are supervised by the operator, there are some actions that 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 process, where human intervention is not required, the PCE sends the Provisioning Manager a new configuration to configure the network elements, as shown in Figure 22.

```
                         +------------------------+
                         |       OSS or NMS       |
                         +-----------+------------+
                                     |
                                     V
               +------+   +----------+------------+
               |Policy+->-+     ABNO Controller   |
               |Agent |   |                       |
               +------+   +----------+------------+
                                     |
                                     V
                              +------+------+
                              +     PCE     |
                              +------+------+
                                     |
                                     V
                     +----------------------------------+
                     |       Provisioning Manager       |
                     +----------------------------------+
```

Figure 22: 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] describes the architecture for multi-segment pseudowires. An end-to-end service, as shown in Figure 23, can consist of a series of stitched segments shown in the figure as AC, PW1, PW2, PW3, and AC. Each pseudowire segment is stitched at a "stitching Provider Edge" (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 PE nodes may traverse different MPLS networks with the PEs as border nodes, or the PEs may lie within the network such that each LSP spans only part of the network.

```
              -----         -----         -----         -----
     ---     |T-PE1|  LSP1 |S-PE1|  LSP2 |S-PE3|  LSP3 |T-PE2|    +---+
    |   | AC |     |=======|     |=======|     |=======|     | AC |   |
    |CE1|----|........PW1........|..PW2........|..PW3........|----|CE2|
    |   |    |     |=======|     |=======|     |=======|     |    |   |
     ---     |     |       |     |       |     |       |     |    +---+
              -----         -----         -----         -----
```

Figure 23: Multi-Segment Pseudowire

While the topology shown in Figure 23 is easy to navigate, the reality of a deployed network can be considerably more complex. The topology in Figure 24 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 Customer Edge nodes (CEs) 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

```
                                   -----
                                  |S-PE5|
                                  /-----\
     ---      -----         -----/       \-----         -----      ---
    |CE1|----|T-PE1|-------|S-PE1|-------|S-PE3|-------|T-PE2|----|CE2|
     ---      -----\        -----\        -----        /-----      ---
                    \         |   -------   |         /
                     \      -----         \-----     /
                      -----|S-PE2|-------|S-PE4|-----
                            -----         -----
```

Figure 24: 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 25 (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.

```
                  -------                         -------
                 |  PE1  |          LSP1          |  PE2  |
            AC   |       |=======================|       |   AC
             ----...................PW1...................----
     --- -  /    |       |=======================|       |    \  -----
    |     |/     |       |                       |       |     \|     |
    | CE1 +      |       |      MPLS Network     |       |      + CE2 |
    |     |\     |       |                       |       |     /|     |
     --- -  \    |       |=======================|       |    /  -----
             ----...................PW2...................----
            AC   |       |=======================|       |   AC
                 |       |          LSP2          |       |
                  -------                         -------
```

Figure 25: Path-Diverse Pseudowires

The path-diverse pseudowire is developed in Figure 26 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 only one of the LSPs. Thus, some form of coordinated path computation approach is needed.

```
                  -------                         -------
                 |  PE1  |          LSP1         |  PE2  |
             AC  |       |=======================|       |  AC
              ---...................PW1...................---
             /   |       |=======================|       |   \
     -----  /    |       |                       |       |    \  -----
    |     |/      -------                         -------      \|     |
    | CE1 +                     MPLS Network                    + CE2 |
    |     |\      -------                         -------      /|     |
     -----  \    |  PE3  |                       |  PE4  |    /  -----
             \   |       |=======================|       |   /
              ---...................PW2...................---
             AC  |       |=======================|       |  AC
                 |       |          LSP2         |       |
                  -------                         -------
```

Figure 26: Path-Diverse Pseudowires with Disjoint PEs

3.6.3. Path-Diverse Multi-Segment Pseudowires

Figure 27 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.

```
                                   -----                -----
                                  |S-PE5|--------------|T-PE4|
                                  /-----\               ----- \
              -----         -----/       \-----         -----  \ ---
             |T-PE1|-------|S-PE1|-------|S-PE3|-------|T-PE2|--|CE2|
       ---  / -----\        -----\        -----        /-----    ---
      |CE1|<        -------   |   -------   |         /
       ---  \ -----         \-----        \-----      /
             |T-PE3|-------|S-PE2|-------|S-PE4|-----
              -----         -----         -----
```

Figure 27: 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 27, 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 be 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 enabled by the mechanism described in Section 3.6.3 can be achieved by protecting individual pseudowire segments or PEs. For example, in Figure 27, 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 28.

```
             -------              -------              -------
            | S-PE1 |    LSP1    | S-PE5 |    LSP3    | S-PE3 |
            |       |============|       |============|       |
            |   .........PW1..................PW3..........   | Outgoing
   Incoming |  :    |============|       |============|    :  | Segment
   Segment  |  :    |             -------             |    :..........
    ...........:    |                                 |    :  |
            |  :    |                                 |    :  |
            |  :    |=================================|    :  |
            |   .........PW2...............................   |
            |       |=================================|       |
            |       |    LSP2                         |       |
             -------                                   -------
```

Figure 28: 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 of 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. At the time of this writing, PCEP does not support path computation requests and responses concerning pseudowires, but the concepts are very similar to existing uses and the necessary extensions would be very small.

Once the paths have been computed, 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 strata.

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 suboptimal from the application’s point of view or considering network resource utilization and quality of service.

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 Virtual Machines (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, Synchronous Digital Hierarchy (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 not only offer 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-multipoint

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 into account 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 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, 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 load conditions within a data center

- The network load 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 29 shows a diagram of an example data center-based application. A carrier network provides access for an end-user through 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 desired services and converts this information to service requests that it passes to the ABNO Controller. The end-users may already know which data center they wish to use, or the Application Service Coordinator may be able to make this determination; otherwise, the task of selecting the data center must be performed by the ABNO Controller, and this may utilize a further database (see Section 2.3.1.8) to contain information about server loads and other data center parameters.

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.

```
   +----------+    +---------------------------------+
   | End-User |--->| Application Service Coordinator |
   +----------+    +---------------------------------+
         |                          |
         |                          v
         |                 +-----------------+
         |                 | ABNO Controller |
         |                 +-----------------+
         |                          |
         |                          v
         |               +---------------------+       +--------------+
         |               |Other ABNO Components|       | o o o   DC 1 |
         |               +---------------------+       |  \|/         |
         |                          |            ------|---O          |
         |                          v           |      |              |
         |            --------------------------|--    +--------------+
         |           / Carrier Network      PE1 |  \
         |          /      .....................O   \   +--------------+
         |         |      .                          |  | o o o   DC 2 |
         |         | PE4 .                      PE2  |  |  \|/         |
          ---------|----O........................O---|--|---O          |
                   |     .                           |  |              |
                   |      .                    PE3   |  +--------------+
                    \      .....................O   /
                     \                          |  /   +--------------+
                      --------------------------|--    | o o o   DC 3 |
                                                |      |  \|/         |
                                                 ------|---O          |
                                                       |              |
                                                       +--------------+
```

Figure 29: 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 location to 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 the end-user to the 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. A view of how the PCE can fit in with CSO is provided in [CSO-PCE], on which the content of Figure 29 is based.

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 the ability 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. ALTO Server

The ABNO architecture allows use cases with joint network and application-layer optimization. In such a use case, an application is presented with an abstract network topology containing only information relevant to the application. The application computes its application-layer routing according to its application objective. The application may interact with the ABNO Controller to set up explicit LSPs to support its application-layer routing.

The following steps are performed to illustrate such a use case.

1. Application Request of Application-Layer Topology

   Consider the network shown in Figure 30. The network consists of five nodes and six links.

   The application, which has end points hosted at N0, N1, and N2, requests network topology so that it can compute its application-layer routing, for example, to maximize the throughput of content replication among end points at the three sites.

```
                 +----+       L0 Wt=10 BW=50       +----+
                 | N0 |............................| N3 |
                 +----+                            +----+
                   |   \     L4                       |
                   |    \    Wt=7                     |
                   |     \   BW=40                    |
                   |      \                           |
             L1    |       +----+                     |
             Wt=10 |       | N4 |               L2    |
             BW=45 |       +----+               Wt=12 |
                   |      /                     BW=30 |
                   |     /   L5                       |
                   |    /    Wt=10                    |
                   |   /     BW=45                    |
                 +----+                            +----+
                 | N1 |............................| N2 |
                 +----+       L3 Wt=15 BW=35       +----+
```

Figure 30: Raw Network Topology

The request arrives at the ABNO Controller, which forwards the request to the ALTO Server component. The ALTO Server consults the Policy Agent, the TED, and the PCE to return an abstract, application-layer topology.

For example, the policy may specify that the bandwidth exposed to an application may not exceed 40 Mbps. The network has precomputed that the route from N0 to N2 should use the path N0->N3->N2, according to goals such as GCO (see Section 3.4). The ALTO Server can then produce a reduced topology for the application, such as the topology shown in Figure 31.

```
                      +----+
                      | N0 |............
                      +----+            \
                        |   \            \
                        |    \            \
                        |     \            \
                        |      |            \   AL0M2
                  L1    |      | AL4M5       \  Wt=22
                  Wt=10 |      | Wt=17        \ BW=30
                  BW=40 |      | BW=40         \
                        |      |                \
                        |     /                  \
                        |    /                    \
                        |   /                      \
                      +----+                        +----+
                      | N1 |........................| N2 |
                      +----+   L3 Wt=15 BW=35       +----+
```

Figure 31: Reduced Graph for a Particular Application

The ALTO Server uses the topology and existing routing to compute an abstract network map consisting of three PIDs. The pair-wise bandwidth as well as shared bottlenecks will be computed from the internal network topology and reflected in cost maps.

2. Application Computes Application Overlay

Using the abstract topology, the application computes an application-layer routing. For concreteness, the application may compute a spanning tree to maximize the total bandwidth from N0 to N2. Figure 32 shows an example of application-layer routing, using a route of N0->N1->N2 for 35 Mbps and N0->N2 for 30 Mbps, for a total of 65 Mbps.

```
               +----+
               | N0 |----------------------------------+
               +----+        AL0M2 BW=30               |
                 |                                     |
                 |                                     |
                 |                                     |
                 |                                     |
                 | L1                                  |
                 |                                     |
                 | BW=35                               |
                 |                                     |
                 |                                     |
                 |                                     |
                 V                                     V
               +----+        L3 BW=35                +----+
               | N1 |...............................>| N2 |
               +----+                                +----+
```

Figure 32: Application-Layer Spanning Tree

3. Application Path Set Up by the ABNO Controller

The application may submit its application routes to the ABNO Controller to set up explicit LSPs to support its operation. The ABNO Controller consults the ALTO maps to map the application-layer routing back to internal network topology and then instructs the Provisioning Manager to set up the paths. The ABNO Controller may re-trigger GCO to reoptimize network traffic engineering.

3.9. Other Potential Use Cases

This section serves as a placeholder for other potential use cases that might get documented in future documents.

3.9.1. Traffic 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

3.9.2. Bandwidth Scheduling

Bandwidth scheduling consists 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.

4. Survivability and Redundancy within the ABNO Architecture

The ABNO architecture described in this document is presented in terms of functional units. Each unit could be implemented separately or bundled with other units into single programs or products. Furthermore, each implemented unit or bundle could be deployed on a separate device (for example, a network server) or on a separate virtual machine (for example, in a data center), or groups of programs could be deployed on the same processor. From the point of view of the architectural model, these implementation and deployment choices are entirely unimportant.

Similarly, the realization of a functional component of the ABNO architecture could be supported by more than one instance of an implementation, or by different instances of different implementations that provide the same or similar function. For example, the PCE component might have multiple instantiations for sharing the processing load of a large number of computation requests, and different instances might have different algorithmic capabilities so that one instance might serve parallel computation requests for disjoint paths, while another instance might have the capability to compute optimal point-to-multipoint paths.

This ability to have multiple instances of ABNO components also enables resiliency within the model, since in the event of the failure of one instance of one component (because of software failure, hardware failure, or connectivity problems) other instances can take over. In some circumstances, synchronization between instances of components may be needed in order to facilitate seamless resiliency.

How these features are achieved in an ABNO implementation or deployment is outside the scope of this document.

5. Security Considerations

The ABNO architecture describes a network system, and security must play an important part.

The first consideration is that the external protocols (those shown as entering or leaving the big box in Figure 1) must be appropriately secured. This security will include authentication and authorization to control access to the different functions that the ABNO system can perform, to enable different policies based on identity, and to manage the control of the network devices.

Secondly, the internal protocols that are used between ABNO components must also have appropriate security, particularly when the components are implemented on separate network nodes.

Considering that the ABNO system contains a lot of data about the network, the services carried by the network, and the services delivered to customers, access to information held in the system must be carefully managed. Since such access will be largely through the external protocols, the policy-based controls enabled by authentication will be powerful. But it should also be noted that any data sent from the databases in the ABNO system can reveal details of the network and should, therefore, be considered as a candidate for encryption. Furthermore, since ABNO components can access the information stored in the database, care is required to ensure that all such components are genuine and to consider encrypting data that flows between components when they are implemented at remote nodes.

The conclusion is that all protocols used to realize the ABNO architecture should have rich security features.

6. Manageability Considerations

The whole of the ABNO architecture is essentially about managing the network. In this respect, there is very little extra to say. ABNO provides a mechanism to gather and collate information about the network, reporting it to management applications, storing it for future inspection, and triggering actions according to configured policies.

The ABNO system will, itself, need monitoring and management. This can be seen as falling into several categories:

- Management of external protocols

- Management of internal protocols

- Management and monitoring of ABNO components

- Configuration of policy to be applied across the ABNO system

7. Informative References

[BGP-LS] Gredler, H., Medved, J., Previdi, S., Farrel, A., and S. Ray, "North-Bound Distribution of Link-State and TE Information using BGP", Work in Progress, draft-ietf-idr-ls-distribution-10, January 2015.

[CSO-PCE] Dhody, D., Lee, Y., Contreras, LM., Gonzalez de Dios, O., and N. Ciulli, "Cross Stratum Optimization enabled Path Computation", Work in Progress, draft-dhody-pce-cso-enabled-path-computation-07, January 2015.

[EON] Gerstel, O., Jinno, M., Lord, A., and S.J.B. Yoo, "Elastic optical networking: a new dawn for the optical layer?", IEEE Communications Magazine, Volume 50, Issue 2, ISSN 0163-6804, February 2012.

[Flood] Project Floodlight, "Floodlight REST API", <http://www.projectfloodlight.org>.

[G.694.1] ITU-T Recommendation G.694.1, "Spectral grids for WDM applications: DWDM frequency grid", February 2012.

[G.709] ITU-T Recommendation G.709, "Interface for the optical transport network", February 2012.

[I2RS-Arch] Atlas, A., Halpern, J., Hares, S., Ward, D., and T. Nadeau, "An Architecture for the Interface to the Routing System", Work in Progress, draft-ietf-i2rs-architecture-09, March 2015.

[I2RS-PS] Atlas, A., Ed., Nadeau, T., Ed., and D. Ward, "Interface to the Routing System Problem Statement", Work in Progress, draft-ietf-i2rs-problem-statement-06, January 2015.

[ONF] Open Networking Foundation, "OpenFlow Switch Specification Version 1.4.0 (Wire Protocol 0x05)", October 2013.

[PCE-Init-LSP] Crabbe, E., Minei, I., Sivabalan, S., and R. Varga, "PCEP Extensions for PCE-initiated LSP Setup in a Stateful PCE Model", Work in Progress, draft-ietf-pce-pce-initiated-lsp-03, March 2015.

[RESTCONF] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", Work in Progress, draft-ietf-netconf-restconf-04, January 2015.

[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, <http://www.rfc-editor.org/info/rfc2748>.

[RFC2753] Yavatkar, R., Pendarakis, D., and R. Guerin, "A Framework for Policy-based Admission Control", RFC 2753, January 2000, <http://www.rfc-editor.org/info/rfc2753>.

[RFC3209] Awduche, D., Berger, L., Gan, D., Li, T., Srinivasan, V., and G. Swallow, "RSVP-TE: Extensions to RSVP for LSP Tunnels", RFC 3209, December 2001, <http://www.rfc-editor.org/info/rfc3209>.

[RFC3292] Doria, A., Hellstrand, F., Sundell, K., and T. Worster, "General Switch Management Protocol (GSMP) V3", RFC 3292, June 2002, <http://www.rfc-editor.org/info/rfc3292>.

[RFC3412] Case, J., Harrington, D., Presuhn, R., and B. Wijnen, "Message Processing and Dispatching for the Simple Network Management Protocol (SNMP)", STD 62, RFC 3412, December 2002, <http://www.rfc-editor.org/info/rfc3412>.

[RFC3473] Berger, L., Ed., "Generalized Multi-Protocol Label Switching (GMPLS) Signaling Resource ReserVation Protocol-Traffic Engineering (RSVP-TE) Extensions", RFC 3473, January 2003, <http://www.rfc-editor.org/info/rfc3473>.

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

[RFC3746] Yang, L., Dantu, R., Anderson, T., and R. Gopal, "Forwarding and Control Element Separation (ForCES) Framework", RFC 3746, April 2004, <http://www.rfc-editor.org/info/rfc3746>.

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

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

[RFC5150] Ayyangar, A., Kompella, K., Vasseur, JP., and A. Farrel, "Label Switched Path Stitching with Generalized Multiprotocol Label Switching Traffic Engineering (GMPLS TE)", RFC 5150, February 2008, <http://www.rfc-editor.org/info/rfc5150>.

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

[RFC5254] Bitar, N., Ed., Bocci, M., Ed., and L. Martini, Ed., "Requirements for Multi-Segment Pseudowire Emulation Edge-to-Edge (PWE3)", RFC 5254, October 2008, <http://www.rfc-editor.org/info/rfc5254>.

[RFC5277] Chisholm, S. and H. Trevino, "NETCONF Event Notifications", RFC 5277, July 2008, <http://www.rfc-editor.org/info/rfc5277>.

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

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

[RFC5424] Gerhards, R., "The Syslog Protocol", RFC 5424, March 2009, <http://www.rfc-editor.org/info/rfc5424>.

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

[RFC5520] Bradford, R., Ed., Vasseur, JP., and A. Farrel, "Preserving Topology Confidentiality in Inter-Domain Path Computation Using a Path-Key-Based Mechanism", RFC 5520, April 2009, <http://www.rfc-editor.org/info/rfc5520>.

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

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

[RFC5693] Seedorf, J. and E. Burger, "Application-Layer Traffic Optimization (ALTO) Problem Statement", RFC 5693, October 2009, <http://www.rfc-editor.org/info/rfc5693>.

[RFC5810] Doria, A., Ed., Hadi Salim, J., Ed., Haas, R., Ed., Khosravi, H., Ed., Wang, W., Ed., Dong, L., Gopal, R., and J. Halpern, "Forwarding and Control Element Separation (ForCES) Protocol Specification", RFC 5810, March 2010, <http://www.rfc-editor.org/info/rfc5810>.

[RFC6007] Nishioka, I. and D. King, "Use of the Synchronization VECtor (SVEC) List for Synchronized Dependent Path Computations", RFC 6007, September 2010, <http://www.rfc-editor.org/info/rfc6007>.

[RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, October 2010, <http://www.rfc-editor.org/info/rfc6020>.

[RFC6107] Shiomoto, K., Ed., and A. Farrel, Ed., "Procedures for Dynamically Signaled Hierarchical Label Switched Paths", RFC 6107, February 2011, <http://www.rfc-editor.org/info/rfc6107>.

[RFC6120] Saint-Andre, P., "Extensible Messaging and Presence Protocol (XMPP): Core", RFC 6120, March 2011, <http://www.rfc-editor.org/info/rfc6120>.

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

[RFC6707] Niven-Jenkins, B., Le Faucheur, F., and N. Bitar, "Content Distribution Network Interconnection (CDNI) Problem Statement", RFC 6707, September 2012, <http://www.rfc-editor.org/info/rfc6707>.

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

[RFC7011] Claise, B., Ed., Trammell, B., Ed., and P. Aitken, "Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of Flow Information", STD 77, RFC 7011, September 2013, <http://www.rfc-editor.org/info/rfc7011>.

[RFC7285] Alimi, R., Ed., Penno, R., Ed., Yang, Y., Ed., Kiesel, S., Previdi, S., Roome, W., Shalunov, S., and R. Woundy, "Application-Layer Traffic Optimization (ALTO) Protocol", RFC 7285, September 2014, <http://www.rfc-editor.org/info/rfc7285>.

[RFC7297] Boucadair, M., Jacquenet, C., and N. Wang, "IP Connectivity Provisioning Profile (CPP)", RFC 7297, July 2014, <http://www.rfc-editor.org/info/rfc7297>.

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

[TL1] Telcorida, "Operations Application Messages - Language For Operations Application Messages", GR-831, November 1996.

[TMF-MTOSI]
TeleManagement Forum, "Multi-Technology Operations Systems Interface (MTOSI)", <https://www.tmforum.org/MTOSI/2319/home.html>.

[YANG-Rtg] Lhotka, L. and A. Lindem, "A YANG Data Model for Routing Management", Work in Progress, draft-ietf-netmod-routing-cfg-17, March 2015.

Appendix A. Undefined Interfaces

This appendix provides a brief list of interfaces that are not yet defined at the time of this 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 [I2RS-Arch]
  - NETCONF [RFC6241]
- The protocol to be used by the Interface to the Routing System is described in Section 2.3.2.8. The I2RS working group has determined that this protocol will be based on a combination of NETCONF [RFC6241] and RESTCONF [RESTCONF] with further additions and modifications as deemed necessary to deliver the desired function. The details of the protocol are still to be determined.
- 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, or this could be achieved using extensions to the PCEP Notify message (PCNtf).
- 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 will address the requirement.
  - A potential approach for this type of interface is described in [RFC7297] for a simple use case.
- As noted in Section 2.3.2.14, there may be layer-independent data models for offering common interfaces to control, configure, and report OAM.

- As noted in Section 3.6, the ABNO model could be applied to placing multi-segment pseudowires in a network topology made up of S-PEs and MPLS tunnels. The current definition of PCEP [RFC5440] and associated extensions that are works in progress do not include all of the details to request such paths, so some work might be necessary, although the general concepts will be easily reusable. Indeed, such work may be necessary for the wider applicability of PCEs in many networking scenarios.

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, Cyril Margaria, Jonathan Hardwick, Nico Wauters, Tom Taylor, Qin Wu, and Luis Contreras. Thanks to George Swallow for suggesting the existence of the SRLG database. Tomonori Takeda and Julien Meuric provided valuable comments as part of their Routing Directorate reviews. Tina Tsou provided comments as part of her Operational Directorate review.

This work received funding from the European Union’s Seventh Framework Programme for research, technological development, and demonstration, through the PACE project under grant agreement number 619712 and through the IDEALIST project under grant agreement number 317999.

Contributors

Quintin Zhao
Huawei Technologies
125 Nagog Technology Park
Acton, MA 01719
United States
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

Young Lee
Huawei Technologies
EMail: leeyoung@huawei.com

Y. Richard Yang
Yale University
EMail: yry@cs.yale.edu

Authors’ Addresses

Daniel King
Old Dog Consulting

EMail: daniel@olddog.co.uk

Adrian Farrel
Juniper Networks

EMail: adrian@olddog.co.uk

OPTICAL COMMUNICATIONS

A Control Plane Architecture for Multi-Domain Elastic Optical Networks: The View of the IDEALIST Project

Ramon Casellas, Oscar González, Francesco Paolucci, Roberto Morro, Víctor López, Daniel King, Ricardo Martínez, Filippo Cugini, Ral Muñoz, Adrian Farrel, Ricard Vilalta, Juan-Pedro Fernández-Palacios

ABSTRACT

A key objective of the IDEALIST project included the design and implementation of a GMPLS and PCE-based control plane for multi-vendor and multi-domain flexi-grid EON, leveraging the project advances in optical switching and transmission technology, an enabling interoperable deployment. A control plane, relying on a set of entities, interfaces, and protocols, provides the automation of the provisioning, recovery, and monitoring of end-to-end optical connections. This article provides an overview of the implemented architecture. We present the macroscopic system along with the core functional blocks, control procedures, message flows, and protocol extensions. The implemented end-to-end architecture adopted active stateful hierarchical PCE, under the control and orchestration of an adaptive network manager, interacting with a parent PCE, which first coordinates the selection of domains and the end-to-end provisioning using an abstracted view of the topology, and second, delegates the actual computation and intra-domain provisioning to the corresponding children PCEs. End-to-end connectivity is obtained by either a single LSP, or by the concatenation of multiple LSP segments, which are set up independently by the underlying GMPLS control plane at each domain. The architecture and protocol extensions have been implemented by several partners, assessing interoperability in a multi-partner testbed and adoption by the relevant Internet SDO.

A key objective of the IDEALIST project included the design and implementation of a GMPLS and PCE-based control plane for multi-vendor and multi-domain flexi-grid EON, leveraging the project advances in the optical switching and transmission technology, an enabling interoperable deployment. A control plane, relying on a set of entities, interfaces and protocols, provides the automation of the provisioning, recovery and monitoring of end-to-end optical connections.

INTRODUCTION

FLEXI-GRID NETWORKS

Optical transport networks [1] (OTN) are composed of network elements connected by optical fibers allowing the transport, multiplexing, routing, management, supervision, and survivability of optical channels carrying client signals. Such channels were constrained by a DWDM fixed frequency grid, inefficient for low rate signals and not adequate for high rate signals. The term *"flexible grid or flexi-grid"* [2] relates to the updated set of nominal central frequencies (NCF), defined within an abstract grid anchored at 193.1 THz, a new channel spacing (6.25 GHz), and other optical spectrum management considerations covering the efficient and flexible allocation of optical spectral bandwidth. A *frequency slot* (i.e., a variable-sized optical frequency range) is thus characterized by its nominal central frequency and its width, expressed in multiples of a given width granularity (12.5 GHz), and can be allocated to a connection, based on the signal modulation format and data rate.

The functional architecture of an OTN is decomposed into independent layers [1] and, in our context, the media layer is the server layer of the optical signal layer, and the optical signal is guided to its destination by means of a network media channel where the switching is based on a frequency slot.

HARDWARE MODELS

An information model is an abstract description used to represent and manage objects (such as a network device) on a conceptual level, independent of any specific protocols used to transport data. A data model is protocol specific and includes many technology specific details. Using well-defined standards-based common information and data models, provides interoperable data exchange between different implementations.

Standardization, notably at the Internet Engineering Task Force (IETF), is often influenced by early implementations and cooperative development by vendors and open source projects. Particularly pertinent to this article is the fact that the data models that were used to represent and configure optical interfaces with flexi-grid capabilities, or to describe a network topology (nodes, links, and connectivity) enhanced with details of optical capabilities and available resources, enabling network optimization and dynamic and online path computation, were developed by the project members themselves and contributed to the IETF.

Ramon Casellas, Ricardo Martínez, Raül Muñoz, and Ricard Vilalta are with CTTC; Oscar González, Víctor López, and Juan-Pedro Fernández-Palacios are with Telefónica Research and Development (I+D); Francesco Paolucci and Filippo Cugini are with CNIT-Scuola Superiore Sant'Anna; Roberto Morro is with Telecom Italia; Daniel King and Adrian Farrel are with Old Dog Consulting.

DRIVERS AND MOTIVATIONS FOR AN "IDEALIST" CONTROL PLANE

Backbone networks are intended to transport the aggregated traffic from several metropolitan networks. However, existing transport networks are based on the assumption that the traffic demands are predictable, and are not adapted to varying traffic requirements. Therefore, current networks require multiple manual configurations in the metro and core network nodes.

Dynamic optical networks are possible thanks to a distributed generalized multi-protocol label switching (GMPLS) control plane. There is a need for an end-to-end architecture to reduce the provisioning process of legacy network management systems (NMS), using standard network configuration interfaces, which will trigger automated standard control plane for multi-domain/vendor/layer operation. The control plane allows the reconfiguration of the optical service, its protection and restoration capabilities, not only for a single domain, but also for multi-domain scenarios. The benefits of a standardized control plane extend beyond the absolute functions enabled by the control plane itself, because such a common approach also facilitates interoperability between equipment supplied by different vendors, and so enables a network operator to construct a heterogeneous network yet operate it in a homogeneous way.

The implemented control plane architecture covers the automated provisioning and recovery of network connectivity services in a multi-domain setting. Such developments are increasingly driven by use cases such as interconnecting distributed data-centers, associated traffic patterns, and dynamicity.

The implemented control plane architecture covers the automated provisioning and recovery of network connectivity services in a multi-domain setting. Such developments are increasingly driven by use cases such as interconnecting distributed data-centers, associated traffic patterns and dynamicity.

EXISTING CONTROL PLANE FRAMEWORK

There is extensive experience in the use of a dynamic distributed control plane. Standardization of this work has been conducted principally within the IETF, with some architectural and use-case documents developed within the ITU-T. The GMPLS architecture [3] comprises the following elements.

A link/neighbor discovery/verification protocol, such as the Link Management Protocol (LMP), that allows neighboring nodes part of the control plane adjacency to unambiguously associate data plane adjacencies (e.g., fiber links), correlate identifiers, and assure compatible capabilities.

A routing protocol. The Open Shortest Path First (OSPF) protocol describes the characteristics of nodes and links, so the state and capabilities of the resources are distributed and updated to all of the nodes, knowing which resources are in use, faulted/out of service, or available.

A signaling protocol. The ReSerVation Protocol with Traffic Engineering extensions (RSVP-TE) is used to set up label switched paths (LSPs). RSVP-TE messages specify the path of the LSP, request specific capacity on the path, and report back the exact allocated network resources to support the LSP.

A path computation service. A key aspect is determining which path an LSP should follow. This function can be performed externally (the path is supplied to the control plane), or delegated to the control plane. In either case, the computation can be complex. The path computation element (PCE) is a functional component that can be queried using the Path Computation Element communication Protocol (PCEP), recently extended to allow the network to delegate control of an LSP to a PCE, and to allow a PCE to direct the establishment of new LSPs (becoming an active PCE) [4].

A network state reporting mechanism. The Link State Border Gateway Protocol (BGP-LS) allows an entity to collect, synthesize, and report the full set of state and capability information from the network to an external consumer such as a management system [5].

A coherent view of these protocols in a managed or software defined networking (SDN) context is provided by the IETF through their application based network operation (ABNO) architecture [6].

CONTROL PLANE ARCHITECTURE

Our GMPLS/PCE control plane for multi-domain flexi-grid networks addresses the provisioning of either a network media channel or a constant bit rate service between optical transceivers, which can support multiple bit rates. A media channel is a media association representing the topology path and the allocated resource (i.e., the frequency slot). It is similar to the GMPLS concept of LSP where, from a data plane perspective, it is the path in the network resulting from reserving and configuring transmission and switching resources across TE links and nodes in a way that can transport client signals and data from its entry point or interface to the exit point or interface. It represents a (effective) frequency slot supported by a concatenation of media elements. GMPLS labels locally represent the media channel and its associated frequency slot, which is the switched resource. Network media channels are considered a particular case of media channels when the end points are transceivers, and transport a single optical tributary signal (OTS), as shown in Fig. 1. The control plane deals with the resource reservation and configuration of media layer matrixes that switch frequency slots and the configuration of the transceivers at the endpoints, with an agreed hardware model that, as of today, is not standard. No signal layer (e.g., OTS) switching is considered. Switching at the media layer is configured by configuring optical filters and configuring cross-connections.

From a bottom-top approach, each domain deploys its own GMPLS control plane instance. On top of it, each domain deploys an active stateful PCE (AS-PCE) for the purposes of both optimal path computation and service provisioning within its domain. Multi-domain path computation and provisioning is carried out by means of a hierarchical path computation element (H-PCE) [7], with the parent PCE (pPCE) coordinating the procedures between children PCEs (cPCE) and under the control and orchestration of an adaptive network manager (ANM). The macroscopic architecture is shown in Fig. 2.

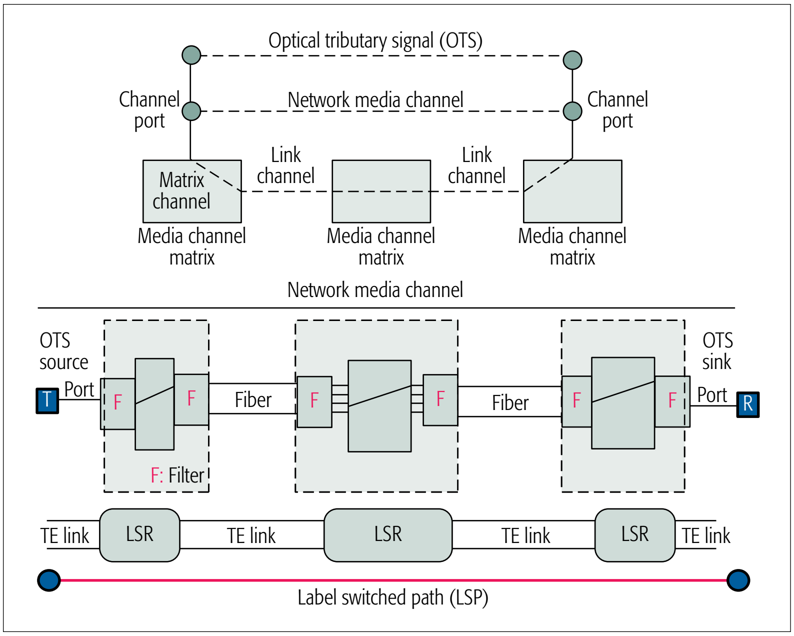


Figure 1. Relationship between optical tributary signal, network media channel, and media layer elements, and its view as a GMPLS LSP construct.

ADAPTIVE NETWORK MANAGER AND IN-OPERATION NETWORK PLANNING

The control plane has relied only on distributed functionalities, but the advent of PCE demonstrates that having a central entity can provide multiple benefits. The ANM was conceived with the idea of orchestrating network processes beyond the PCE capabilities. Its functionalities are to monitor network resources, and to decide the optimal network configuration based on the status, bandwidth availability, and user service. It does not replace the control plane, but extends and complements it (e.g., interacting with the client layer) and delegating specific functions (e.g., path computation) to it.

The ANM was implemented, utilizing the ABNO architecture, and relies on standards-based and open interfaces, providing the capability for application interaction via a north bound interface (NBI) and south bound interface to the data plane, either directly to each network element or via the control plane. The link between the ANM and the control plane is the parent PCE, which receives queries to carry out path computation and provision end-to-end connections.

The ANM platform allows automatic IP link provisioning, multi-layer restoration, dynamic bandwidth allocation based on traffic changes, periodic defragmentation, and network reoptimization after network failure recovery [8], so an operator planning tool has updated network information and maintains a provisioning interface with the network. This architecture benefits from the GMPLS/PCE control plane, reducing network CAPEX by minimizing the over-provisioning required in today's static environments.

HIERARCHICAL PATH COMPUTATION ELEMENT

A parent PCE (pPCE) is responsible for inter-domain path computation, while in each domain a local child PCE (cPCE) performs intra-domain computation. The pPCE resorts to the hierarchical traffic engineering database (H-TED) storing the list of the domains and inter-domain connectivity information, to determine the sequence of domains. Moreover, to perform effective inter-domain computation, the pPCE is allowed to ask cPCEs for the path computation of the several border-to-border LSP segments.

A number of innovative extensions have been implemented by IDEALIST. First, besides reachability information, abstract intra-domain TE information is announced to the pPCE (e.g., in the form of mesh of abstracted TE links between border nodes) with the aim of improving the effectiveness of the domain sequence computation. In particular, the north-bound distribution of link state and TE information using BGP-LS is utilized by domains' BGP speakers to populate the H-TED. Second, in order to enable advanced TE functionalities, e.g., elastic operations and re-optimizations [9, 10], the H-PCE architecture has been extended to support the active stateful PCE with instantiation capabilities.

In summary, the H-PCE achieves end-to-end path computation by performing domain sequence selection and segment expansion, based on spectrum availability information provided by BGP-LS and PCEP requests submitted to cPCEs. The same H-PCE deployment is used in some use cases to perform the provisioning, where the end-to-end path is split in segments, sent to the cPCE by means of instantiation messages, and each cPCE performs segment instantiation. The end-to-end LSP is set up in the form of a "stitching on the wire" of several segments.

GMPLS DISTRIBUTED CONTROL PLANE

Within each domain, there is an instance of a GMPLS control plane. GMPLS controllers execute several collaborative processes, and a data communication network based on IP control channels allows the exchange of control messages between controllers. Noteworthy processes are the connection controller, the routing controller, or the link resource manager. We assume that a GMPLS controller is associated with a single flexi-grid optical node.

Under distributed control, each GMPLS controller manages the state of the connections (i.e., LSPs) originating, terminating, or passing-through a node and maintains its own network state information (topology and resources), collected in a local TED and synchronized thanks to the routing and topology dissemination protocol. Controllers then appropriately configure the underlying hardware (filter, transceiver, or switch configuration) during the establishment of an LSP, as per the basic operation of a GMPLS control plane [3]. In the next section, we overview the main involved procedures focusing on the specific aspects of the optical technology (see [11] for a detailed view).

CONTROL PLANE PROCEDURES

INTRA-DOMAIN AND INTER-DOMAIN TOPOLOGY DISSEMINATION

Within a domain, each node routing controller is responsible for disseminating changes in the network state regarding the resources under its control (e.g., originating links) through OSPF-TE link state advertisements (LSA). Each LSA is sent to the neighboring nodes, which update their TED repositories and forward the LSA in turn. This mechanism allows synchronizing all the nodes' repositories within a given time, referred to as the routing convergence time. The basic procedures remain mostly unchanged, relying on extending the actual information objects within the LSAs.

OSPF-TE has been extended to support the dissemination of per-node and per-link TE attributes, reflecting device restrictions and overall optical spectrum availability. In particular, nodes may have asymmetric switching capabilities or different minimum slot size restrictions; optical transmitters/receivers may have different tunability constraints. Other extensions have been implemented for disseminating the capabilities of sliceable bandwidth variable transceivers (S-BVTs), including, for example, the number of available sub-transponders and their parameters. Let us note that in this approach, OSPF-TE is one of the methods by which a cPCE obtains the TED to perform constrained routing and spectrum assignment (RSA) and is the source of the (abstracted) information conveyed toward the pPCE.

BGP-LS has also been suitably extended to support specific information exchange, such as spectrum availability, transponders' physical parameters, and interoperability capabilities. BGP-LS is also used to report the relevant attributes of inter-domain links. Without disclosing the internal domain topology, this allows a pPCE to have, at least, a graph that represents inter-domain connectivity and to perform basic multi-domain path computation.

MULTI-DOMAIN PATH COMPUTATION

Following Fig. 3, when a service request, driven by an operator, is received by the ANM, the controller asks the pPCE for an inter-domain path (step 1). The pPCE, based on the (possibly abstracted and aggregated) information obtained from the cPCEs, computes the domain sequence (including each domain entry and exit nodes) and subsequently requests from the cPCEs the corresponding border-to-border expansion (also by means of PCEP PCReq messages, step 2). Once the pPCE receives the responses (PCRep, step 3), which include, among other objects, the segment spectrum availability, the pPCE performs a detailed end-to-end path computation including the routing, spectrum assignment, and transponder selection. Optical constraints are considered based, for instance, on node switching capabilities, optical reach, and transponder capabilities. For example, in the case of an end-to-end spectrum continuity constraint, the pPCE has to assign a frequency slot such that it is able to convey the requested bandwidth, it is available across all the end-to-end path, including inter-domain links, and it allows the selection of available end-point transponders. Then, the pPCE answers the ABNO controller via a PCEP Response message.

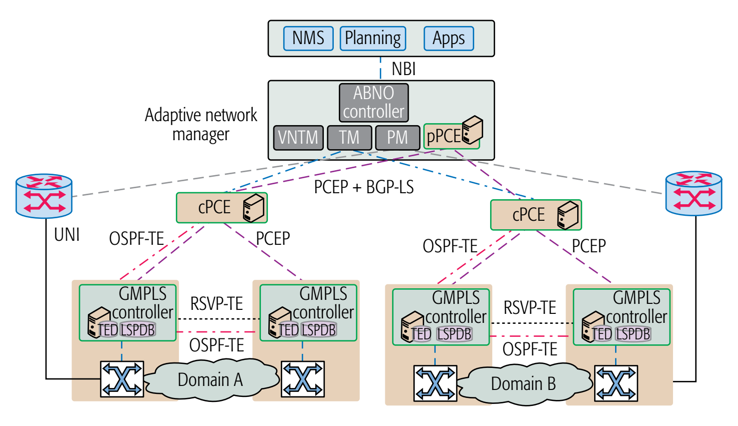


Figure 2. Control Plane architecture showing a multi-domain network with an AS-PCE per domain acting as a Child PCE, a Parent PCE and an ANM.

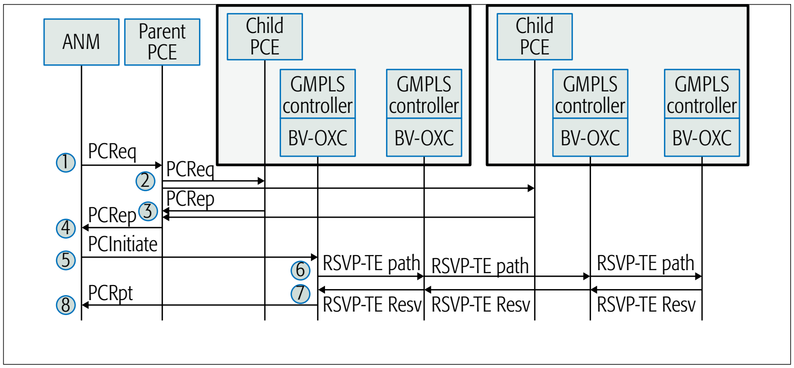


Figure 3. Single Session provisioning model, with stateless H-PCE.

INTER-DOMAIN SERVICE PROVISIONING VIA ANM WITH ACTIVE STATEFUL CAPABILITIES

Once the path is computed, the ABNO controller asks the pPCE to establish the path with a PCEP Initiate message. There are several provisioning models, with varying requirements of control plane interoperability. Here, we focus on the contiguous LSP with a single end-to-end RSVP-TE session, and the model relying on the stateful capabilities of the H-PCE structure with multiple (one per domain) RSVP-TE sessions.

In the single session case, the provisioning interface is a dedicated PCEP session with either the cPCE of the ingress domain or directly the ingress node, and there is a single RSVP-TE session from the source node within the source domain to the destination node. The multiple session case requires that all PCEs are stateful with instantiation capabilities. The connectivity at the data plane level is insured by concatenating compatible media channels at every domain, each set up by the local RSVP-TE session. Note that the first case implies interoperability at the control plane signaling level between different optical vendors' respective RSVP-TE implementations at the inter-domain boundaries, since there is a single end-to-end session that crosses the external network-to-network interfaces. On the contrary, for the second case, interoperability requirements are limited to PCEP, vertically, from the cPCEs to the pPCE, between each vendor and the provider of the pPCE. Both approaches can be seen in Figs. 3 and 4.

Specific extensions were defined for the RSA procedures in a hierarchical framework. Upon request from the pPCE, all cPCEs compute the path segment (sequence of nodes and links) inside their respective domain and reply this information to the pPCE, along with spectrum availability.

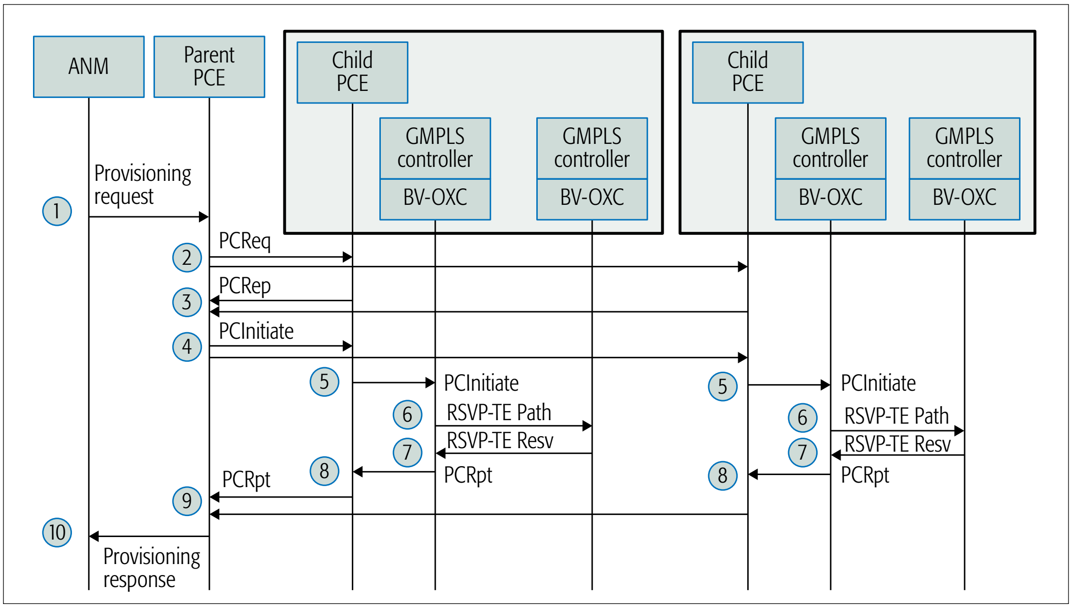


Figure 4. Stateful H-PCE with per-domain instantiation and local RSVP-TE session provisioning model.

In either case, once the end-to-end path or the specific segment is computed, the assigned slot is included in the explicit route objects (EROs) after each hop. In the first case (Fig. 3) the end-to-end ERO is sent to the ingress node in a PCInitiate message (step 5), triggering the signaling process (6, 7) and final report to the ANM (8). In the second case (Fig. 4), the obtained ERO per segment are enclosed in PCInitiate messages sent by pPCE to each involved cPCE (step 4). Once intra-domain provisioning is performed (step 5-8), PCE Report (PCRpt) messages are sent to pPCE to acknowledge the segments' status (step 9). Finally, the multi-domain LSP is stored in the H-TED and provisioning response is provided to the ANM (step 10). Similar procedures for inter-domain LSP update and LSP deletion are envisioned.

CONTROL PLANE PROTOCOL EXTENSIONS

Control plane extensions affect all the protocols of the GMPLS suite together with the those adopted as northbound interfaces (i.e., PCEP and BGP-LS).

PROVISIONING AND LSPDB SYNCHRONIZATION INTERFACE

The provisioning of LSPs relies on the use of the PCEP protocol, enhanced with stateful and instantiation extensions. Specific extensions to PCEP to cope with flexi-grid involve the BANDWIDTH object to convey the traffic descriptor that specifies the requested or allocated frequency slot width, and the ERO object with the resources to use along the path, which has been extended to carry the information describing the configuration of the optical transponders, such as the selected modulation format, baud rate, FEC, and so on. To this end, a new sub-object, called explicit transponder control (ETC), has been defined. It is formed by a variable list of sub-transponder TLVs, each of them describing one of the specific sub-carriers forming the super-channel LSP. To overcome scalability limitations, we enable the summarization of a set of parameters in a single parameter, the transceiver class, which considers the main parameters such as trunk mode and type, framing, channel band and grid, minimum and maximum chromatic dispersion, maximum polarization mode dispersion, differential group delay, and so on. A transceiver vendor is thus responsible for specifying the class contents and values. The vendor can publish the parameters of its classes or declare them to be compatible with already published classes.

INTRA-DOMAIN TOPOLOGY DISSEMINATION

The OSPF-TE protocol has been extended to convey, on a per link basis, the status of each possible central frequency or NCF (referred to as *NCF availability*) and the presence and attributes of transceivers. The former is done by means of a new object within the switching capability-specific information (SCSI) field. NCF availability is advertised using a bitmap format with bit position zero representing the lowest central frequency, each succeeding bit position representing the next central frequency; a bit set to 1 means the NCF is not in use.

MULTI-DOMAIN TOPOLOGY ABSTRACTION

BGP-LS extensions addressed both the propagation of the NCFs' availability and the announcement of an S-BVT transceiver's capabilities to the pPCE, in order to perform routing and spectrum assignment (RSA) for the multi-domain path. The first extension involves adding a new LINK_STATE attributes object TLV into the BGP-LS Update message, further characterizing a given optical link. The latter extension involves announcing the capabilities of an S-BVT attached to a given link using two new BGP-LS TLVs called "MF-OTP encoding" (for multi-flow optical transponder) and "transceiver class and application code", respectively. Both BGP-LS extensions reuse the same encoding as those proposed in OSPF-TE.

PATH COMPUTATION

Specific extensions were defined for the RSA procedures in a hierarchical framework. Upon request from the pPCE, all cPCEs compute the path segment (sequence of nodes and links) inside their respective domain and reply this information to the pPCE, along with spectrum availability. This is accomplished by sending a PCEP Reply (PCRep) message containing the ERO object and two new objects: a LABEL_SET object that encodes the free NCFs along the computed path, and a SUGGESTED_LABEL object, suggesting (but not mandating) the label (i.e., the specific frequency slot) to be used in that domain. The pPCE performs an end-to-end allocation with this information.

SIGNALING PROTOCOL

The extensions to the signaling protocol include:

- A new 64-bit label format, used in all the objects carrying a label (GENERALIZED_LABEL, SUGGESTED_LABEL, LABEL_SET, ERO, etc.) specifying frequency slot center and width in terms of two integer values, n and m, according to the following formulas: Center Frequency (THz) = 193.1 + n * 0.00625, slot width (GHz) = 12.5 * m.
- A new traffic descriptor type for the SENDER_TSPEC and FLOWSPEC objects to specify traffic parameters, carrying the slot width.

Note that the label value, used in GMPLS to define what is switched, indicates, in this case, the slot features and, in particular, its width, therefore also affecting the LSP bandwidth. The same ERO extensions already described apply to the ERO object contained in the signaling messages.

EXPERIMENTAL VALIDATION

The architecture and its integration with the underlying data plane has been demonstrated in several stages, starting from control plane testbeds and ultimately integrating both control and data planes. In [12] the optical channel provisioning was evaluated in a distributed multi-partner control plane testbed with locations in Madrid (Telefnica I+D), Barcelona (CTTC), Torino (Telecom Italia), and Pisa (CNIT). The testbed was connected at the control plane level by means of dedicated IPsec tunnel, emulating a multi-domain network (Fig. 5). On top of this connectivity, logical relationships between PCEs were established. We reported the details of the interoperability of routing (BGP-LS), path computation and instantiation (PCEP), and signaling (RSVP-TE) implementations [12]. In [13], a higher degree of interoperability was achieved, demonstrating the aforementioned different provisioning models. Experimental results showed all protocol interactions and LSP setup times. The adoption of BGP-LS extensions fully enabled multi-domain TE and was demonstrated in a limited number of domains. The system was integrated and demonstrated at both the control and data plane levels [14], where domains can have real hardware optical nodes that switch frequency slots, although by necessity inter-domain links between remote locations are emulated. The data plane included both real and emulated flexi-grid nodes and SBVTs. Two real S-BVT prototypes were provided by different IDEALIST vendors (e.g., CNIT/Ericsson and Coriant). These S-BVTs performed super-channel transmission with a configurable number of PM-16QAM Nyquist-shaped carriers, overall providing up to 1Tb/s. At the receiver, coherent strategy with off-line post-processing was adopted. The S-BVTs supported the configuration of the number of active carriers, their central frequencies, modulation format, symbol rate and FEC.

FUTURE CONSIDERATIONS

The evolution of transmission and data plane technologies, supporting rates at 1Tb/s and beyond, will reach its maximum potential when supported by automatic configuration procedures enabling the deployment of spectrally-efficient plug-and-play transponders. Control plane solutions will have to be improved to provide procedures for the commissioning and self-tuning of the transmission parameters (e.g., upon failure recovery) while aiming to optimize the use of network resources. Plug-and-play 1Tb/s transponders will also have to operate in interoperable multi-vendor environments.

While the control plane supports the dynamic configuration of transceivers, the full automation and self-tuning of parameters will rely on the integration with functional components related to cognitive and self-adaptive networks. The solutions require, for example, the deployment of passive and active monitoring and measurement systems beyond what currently exists, along with the adoption of formal languages and frameworks for the specification of rules and policies typical of expert systems.

Multi-vendor interoperability still remains a major issue to solve. While there are incentives (e.g., from operators or service providers trying to drive down costs), there is huge pressure for vendors to increase margins and differentiate from competing offers.

The decoupling of the data plane and the control plane is expected to also be applied in the context of optical core networks through the concept of transport SDN. A unified control plane architecture is expected to successfully orchestrate the core with metro and data center premises, enabling the challenging support for future front/back-haul networks and 5G applications. Once flexible and open frameworks and interfaces have been adopted for the control and orchestration of network connectivity services across, for example, multiple heterogeneous domains, extending the know-how and conceiving new architectures for the joint allocation of heterogeneous resources is the next logical step, and addresses uses cases that require the allocation of both computing and storage resources.

To achieve the goal of effective interoperability, two factors are also expected to play key roles in addition to standardization, i.e., the definition of common, standard data models, and the use of open source software, offering common core components and allowing "plug-ins" for different applications and vendor devices. Although some vendors may still include proprietary optimizations, a common basis is expected to improve interoperability performance.

Ongoing efforts at the SDOs regarding the definition of common information models (e.g.,

To achieve the goal of effective interoperability, two aspects are also expected to play key roles in addition to standardization, i.e. the definition of common, standard data models, and the use of open source software, offering common core components and allowing "plug-ins" for different applications and vendor devices.

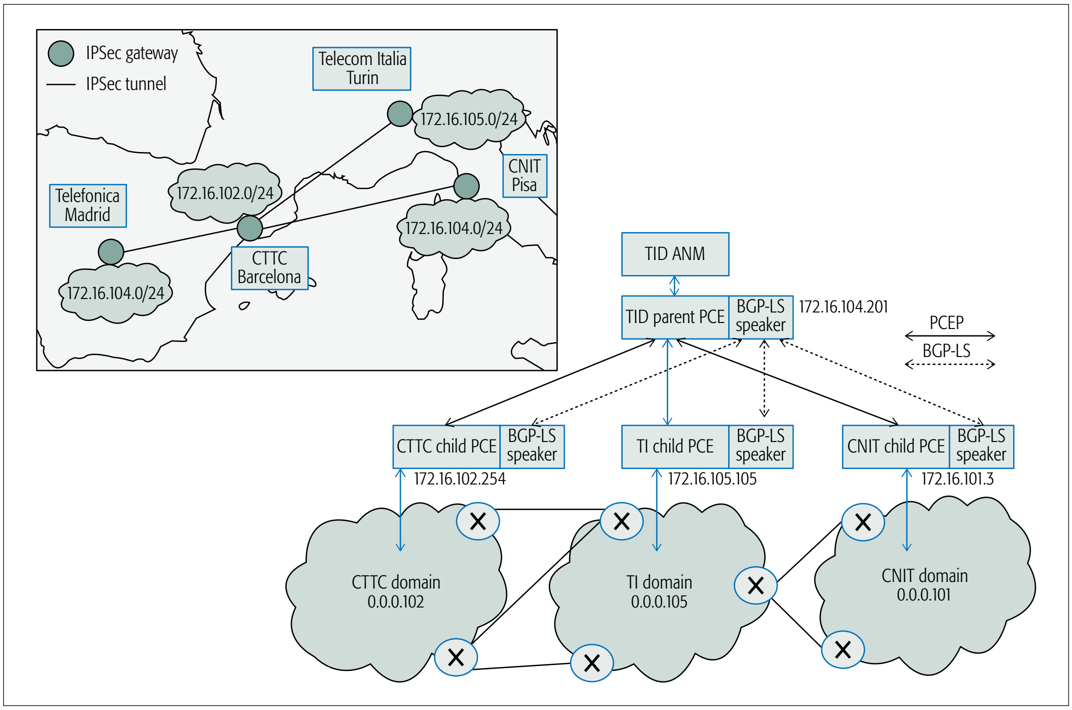


Figure 5. Multi-domain flexi-grid elastic optical network resulting from interconnecting partners' testbeds.

related to network topologies) are a step in this direction. Nevertheless, the goal of achieving total interoperability still remains a hard issue, even more difficult in the domain of optical transport networks.

CONCLUSIONS

A GMPLS/PCE control plane for flexi-grid networks orchestrated by the ANM requires an architecture and protocols fulfilling the initial requirements while ensuring robustness, security, and scalability. Although the framework is considered to be stable and quite mature, addressing the constraints associated with flexi-grid DWDM networks, variable bandwidth transceivers, and programmable devices is a complex problem. We have detailed the components of such a multi-domain control plane. The summarization of TE capabilities per domain, underlay network abstraction, and applicability of stateful PCE capabilities to end-to-end path computation across multi-domain networks, are part of the IDEALIST solutions based on a hierarchy of the PCEs, which have been implemented, demonstrated in a multi-vendor testbed, and reported for standardization.

While other standardization bodies are working on the specification of the architecture of an SDN-based solution for multi-domain transport networks, our original goal was to extend the GMPLS protocol suite. The final architecture shares several aspects with SDN, since each domain is scoped and encapsulated by an active stateful PCE, and architectural elements still apply even if the network is composed of heterogeneous control technologies, including, for example, SDN and Openflow [15].

The architecture is hybrid, combining distributed and centralized elements. An additional role of the ANM and PCE is to enable a progressive migration to a transport SDN, since the architecture fits in a wider SDN applicability context in which driving a GMPLS domain is one south-bound interface of an orchestrator. From the perspective of the ANM and the H-PCE, the main differences would be the mechanism to retrieve the topologies and the actual service provisioning, which would be either delegated (GMPLS) or using a dedicated protocol that directly configures the hardware (OpenFlow).

The architecture is hybrid, combining distributed and centralized elements. An additional role of the ANM and PCE is to enable a progressive migration to a transport SDN, since the architecture fits in a wider SDN applicability context in which driving a GMPLS domain is one south-bound interface of an orchestrator.

A standard control plane for a multi-domain/multi-vendor flexi-grid network can only be realistically designed, assuming a standard data plane, to a level of detail that does not currently exist. Current data plane standards imply that flexible network media channels are unlikely except in specially designed subnetworks, and while allowing mixed rate signals on the same fiber, standardized multi-vendor interoperability is not, as of today, covered. IDEALIST has addressed this by having (data and control plane) implementation agreements, but without further advances (including, e.g., S-BVTs) a control plane cannot fully exploit the theoretical advantages of flexi-grid in an interoperable scenario. Interoperability still remains a major issue unlikely to be solved in the short term. While there are drivers and incentives (e.g., from operators or service providers trying to drive down costs), there remains a huge pressure for vendors.

ACKNOWLEDGMENTS

This article was supported by the European Community's Seventh Framework Programme FP7/2007-2013 through the IDEALIST (Industry-Driven Elastic and Adaptive Lambda Infrastructure for Service and Transport Networks) project, grant agreement no. 317999 (http://www.ict-idealist.eu).

References

[1] ITU-T Recommendation G.872, "Architecture of optical transport networks," Oct. 2012

[2] ITU-T Recommendation G.694.1, "Spectral grids for WDM applications: DWDM frequency grid," Feb. 2012.

[3] E. Mannie ed., "Generalized Multi-Protocol Label Switching (GMPLS) Architecture," IETF RFC 3945, Oct. 2004.

[4] E. Crabbe *et al.*, "PCEP Extensions for PCE-initiated LSP Setup in a Stateful PCE Model," IETF Internet draft, work in progress, Oct. 2015

[5] H. Gredler *et al.*, "North-Bound Distribution of Link-State and TE Information using BGP," IETF Internet draft, work in progress, Oct. 2015.

[6] D. King and A. Farrel, "A PCE-Based Architecture for Application-Based Network Operations," IETF RFC 7491, Mar. 2015.

[7] D. King and A. Farrel, "The Application of the Path Computation Element Architecture to the Determination of a Sequence of Domains in MPLS and GMPLS," IETF RFC 6805, Nov. 2012.

[8] L. Velasco *et al.*, "In-Operation Network Planning," *IEEE Commun. Mag.*, vol. 52, no. 1, pp. 52-60, Jan. 2014.

[9] F. Paolucci *et al*., "Active PCE Demonstration Performing Elastic Operations and Hitless Defragmentation in Flexible Grid Optical Networks," *Photonic Network Commun.*, Springer, vol. 29, issue 1, 2014, pp 57–66.

[10] R. Munoz *et al*., Dynamic and Adaptive Control Plane Solutions for Flexi-Grid Optical Networks Based on Stateful PCE, *J. Lightwave Technol*., vol. 32, no. 16, Aug. 15, 2014, pp. 2703–15.

[11] O. Gonzlez de Dios, R. Casellas, eds. "Framework and Requirements for GMPLS-Based Control of Flexi-Grid Dense Wavelength Division Multiplexing (DWDM) Networks," IETF RFC 7698, Nov. 2015.

[12] O. Gonzlez de Dios *et al.*, "First Multi-Partner Demonstration of BGP-LS Enabled Inter-domain EON Control with H-PCE," *Proc. OFC2015*, 22–26 Mar. 2015, Los Angeles (USA).

[13] O. Gonzlez de Dios *et al*., "Multi-Partner Demonstration of BGPLS Enabled Multi-Domain EON Control and Instantiation with H-PCE," *J. Optical Communications and Networking*, vol. 7, no. 11.

[14] O. Gonzlez de Dios *et al.*, "First Demonstration of Multi-Vendor and Multi-Domain EON with S-BVT and Control Interoperability over Pan-European Testbed," *Proc. 41st ECOC2015*, PDP4.1, Sept. 2015, Valencia (Spain).

[15] R. Casellas *et al.*, "SDN Orchestration of OpenFlow and GMPLS Flexi-Grid Networks with a Stateful Hierarchical PCE [invited]," *IEEE/OSA J. Optical Commun. Netw*., vol. 7, no.1, A106-A117, Jan. 2015.

Biographies

Ramon Casellas [SM '12] graduated in telecommunications in 1999 from the UPC and ENST. He worked as an undergraduate researcher at FT R&D and BT Labs, completed a Ph.D. in 2002, and worked as an associate professor (ENST) until joining the CTTC in 2006. He is a senior research associate involved in R&D and technology transfer projects. His research interests include network control GMPLS/PCE and SDN/NFV. He has co-authored more than 150 papers and contributes to IETF.

Oscar González received his M.Sc. and Ph.D. from the University of Valladolid in 2000 and 2012, respectively. He has 14 years of experience at Telefonica I+D, involved in European R&D projects (STRONGEST, ONE, IDEALIST). He has co-authored more than 40 research papers and is active in IETF CCAMP and PCE WGs and ITU-T Study group 15. His main research interests include photonic networks, flexi-grid, inter-domain routing, PCE, OBS, automatic network configuration, end-to-end MPLS, TCP performance, and SDN.

Francesco Paolucci received the Laurea degree in telecommunications engineering in 2002 from the University of Pisa, Italy, and the Ph.D. in 2009 from Scuola Superiore Sant'Anna, Pisa, Italy. In 2008 he was granted a research merit scholarship at INRS, Montreal. Canada. Currently, he is an assistant professor at Scuola Superiore Sant'Anna. His main research interests are transport network control plane, including GMPLS, PCE, and SDN. He is a co-author of four patents and more than 100 international publications.

Roberto Morro received a Dr. Ing. degree in electronic engineering from the University of Genoa, Italy, in 1988. After six years with Marconi, he joined TIM (at that time CSELT) in 1995, where he is currently in the IP & transport innovation unit. He has been involved in European projects (LION, NOBEL, NOBEL-2, MUPBED, STRONGEST, IDEALIST) and technology transfer activities. His research interests include network control with focus on multi-layer, multi-domain and traffic engineering aspects.

Víctor López received the M.Sc. from the Universidad de Alcalá de Henares, Spain, in 2005, and the Ph.D. from UAM in 2009. In 2006 he joined the High-Performance Computing and Networking Research Group (UAM). He worked as an assistant professor at UAM. In 2011 he joined Telefonica I+D as a technology specialist. He has co-authored more than 100 publications and contributed to IETF drafts. His research interests include IP/MPLS, optical networks, and control plane (PCE, SDN, GMPLS).

Daniel King received an MBA, and is currently finishing his Ph.D., at the University of Lancaster. He is the co-chair of the Software Defined Networks (SDN) Research Group (SDNRG), and chair of the Simplified Use Policy Abstraction (SUPA) at the Internet Engineering Task Force (IETF). He is an Openetworking Foundation research associate, and an editor and author of numerous technology papers, journals, book chapters, ONF documents, and IETF RFCs.

Interoperability still remains a major issue unlikely to be solved in the short term. While there are drivers and incentives (e.g., from operators or service providers trying to drive down costs), there remains a huge pressure for vendors.

Ricardo Martínez [SM '14] graduated and received a Ph.D. in telecommunications engineering from UPC-BarcelonaTech University in 2002 and 2007, respectively. He has been actively involved in several public-funded (national and EU) R&D as well as industrial technology transfer projects. Since 2013 he has been a senior researcher in the Communication Networks Division (CND) at CTTC. His research interests include control and network management, protocols and traffic engineering mechanisms for packet and optical transport networks within aggregation and core segments.

Filippo Cugini is head of research at CNIT, Pisa, Italy. His main research interests include theoretical and experimental studies in the field of optical communications and networking. He is a co-author of 12 patents and more than 200 international publications.

Raül Muñoz [SM '12] graduated and received a Ph.D. in telecommunications engineering in 2001 and 2005, respectively, from UPC (Spain). He joined CTTC in 2002, where is a senior researcher and head of the Optical Networks and System Department. Since 2000 he has participated in several public-funded R&D and technology transfer projects. He has led several Spanish R&D projects and the EU-Japan STRAUSS project STRAUSS. His research interests include control and management architectures, protocols and traffic engineering algorithms for future transport networks.

Adrian Farrel graduated in mathematics from the University of Durham, UK. He worked for 18 years as a software engineer before becoming an independent consultant specializing in Internet routing and signaling protocols. He has authored or co-authored six books on Internet protocols and more than 60 RFCs. He leads innovation in the areas of path computation, traffic engineering, optical networking, and network operations (including SDN). He currently chairs the L3SM and I2NSF working groups in the IETF.

Ricard Vilalta (Girona, 1983) has a telecommunications engineering degree (2007) and a Ph.D. degree (2013) from UPC, Spain. Since 2010 he has been a researcher at CTTC, in the Communication Networks Division. He is currently a research associate at the Open Networking Foundation. His research is focused on SDN/NFV, network virtualization, and network orchestration. He has been involved in international, EU, national, and industrial research projects, and he has published more than 100 journals, conference papers, and invited talks.

Juan-Pedro Fernández-Palacios received the M.S. in telecommunications engineering from UPV in 2000. In Sept. of 2000 he joined Telefonica I+D, where he is currently leading the Core Network Evolution unit. He has been involved in several European projects such as NOBEL, NOBEL-2, STRONGEST, MAINS, and IDEALIST, as well as in the design of core network architectures in the Telefónica Group.

Computer Standards & Interfaces 54 (2017) 203–215

Contents lists available at ScienceDirect

Computer Standards & Interfaces

journal homepage: www.elsevier.com/locate/csi

Network service orchestration standardization: A technology survey

Charalampos Rotsos[a,*], Daniel King[a], Arsham Farshad[a], Jamie Bird[a], Lyndon Fawcett[a], Nektarios Georgalas[b], Matthias Gunkel[c], Kohei Shiomoto[d], Aijun Wang[e], Andreas Mauthe[a], Nicholas Race[a], David Hutchison[a]

[a] *Infolab21, School of Computing and Communications, Lancaster University, United Kingdom*
[b] *British Telecom, United Kingdom*
[c] *Deutsche Telekom, Germany*
[d] *NTT, Japan*
[e] *China Telecom, China*

ARTICLE INFO

Keywords:
Network Service
Orchestration
SDN
NFV
Standardization

ABSTRACT

Network services underpin operator revenues, and value-added services provide income beyond core (voice and data) infrastructure capability. Today, operators face multiple challenges: a need to innovate and offer a wider choice of value-added services, whilst increasing network scale, bandwidth and flexibility. They must also reduce operational costs, and deploy services far faster - in minutes rather than days or weeks.

In the recent years, the network community, motivated by the aforementioned challenges, has developed production network architectures and seeded technologies, like Software Defined Networking, Application-based Network Operations and Network Function Virtualization. These technologies enhance the highly desired properties for elasticity, agility and cost-effectiveness in the operator environment. A key requirement to fully exploit the benefits of these new architectures and technologies is a fundamental shift in management and control of resources, and the ability to orchestrate the network infrastructure: coordinate the instantiation of high-level network services across different technological domains and automate service deployment and re-optimization.

This paper surveys existing standardization efforts for the orchestration - automation, coordination, and management - of complex set of network and function resources (both physical and virtual), and highlights the various enabling technologies, strengths and weaknesses, adoption challenges for operators, and areas where further research is required.

1. Introduction

> Flexibility, agility and automation and a much faster time-to-market cycle, where the latter is something that we, as operators, lack today
> (Christos Kolias, Network Architect, Orange [1]).

Network services are the primary value-added products for Network Operators (operators), enabling them to monetize their infrastructure investments. Operator service portfolios cover a wide range of functionalities, spanning from basic Internet connectivity services, such as IPTV delivery, to highly-available and secure connectivity between business sites. This operator business model has been highly successful, their user base continuously expands [2], while new services are adopted by end-users.

As a direct consequence, network infrastructures have grown significantly in the recent years and operators face significant challenges maintaining high revenues, while supporting innovative new network services. On the one hand, traffic volumes increase exponentially [3] and forces operators to upgrade infrastructures frequently. Additionally, the established service management model relies extensively on manual device reconfiguration by the network engineers, coordinated through Operational Support Systems (OSS), while link over-provision is used to enforce SLAs. Effectively, the predominant service management model incurs significant capital (CAPEX) and operational expenditures (OPEX) for the operator [4]. On the other hand, network infrastructures employ a widening range of heterogeneous technologies to support the diverse characteristics and dynamic demands of residential and enterprise network services. Unfortunately,

---

[*] Corresponding author.
E-mail addresses: h.rotsos@gmail.com, cr409@cl.cam.ac.uk (C. Rotsos).

http://dx.doi.org/10.1016/j.csi.2016.12.006
Received 15 April 2016; Received in revised form 4 November 2016; Accepted 31 December 2016

the control and management interfaces of the relevant technologies do not keep abreast with the requirements of network applications for fluid and dynamic control. The different technological domains and layers exhibit significant interface proliferation, while vertical control integration in network devices impairs management flexibility and responsiveness. As a result, the futuristic vision of network operators to provide service-oriented control interfaces to end-user applications, still remains unfulfilled.

These limitations have motivated the network and systems community to develop new paradigms and architectures which improve network infrastructure flexibility, agility, programmability and elasticity and ensure low OPEX. Recent network paradigms, like Software Defined Networking (SDN) and Application-based Network Operations (ABNO), promote control convergence across network layers and logical centralization of network infrastructure management through the specification of common device control interfaces. In parallel, the Network Function Virtualization (NFV) paradigm promotes the ”*softwarization*” and virtualization of network functions, in order to enable data plane processing with similar elasticity, scalability and resilience available in cloud environments. Furthermore, new network architectures including Service Functions Chaining (SFC) and Segment Routing (SR), simplify service deployment and allow seamless integration of traffic-engineered (deterministic) network services and network policy.

To capitalize on the fluidity of these novel networking paradigms and architectures, operators a require new control and management system, capable to *orchestrate* the different technologies and resource types available in modern network infrastructures. These systems are responsible to converge control and management heterogeneity between technologies, in an effort to synthesize innovative service-oriented interfaces, and enable autonomous and automated service deployment and adaptation. The development of service orchestration architectures and interfaces has been accelerating, but since each vendor typically develops its own protocols and mechanisms, integration remains a challenge. Towards the goal for automated, flexible and cost-effective service orchestration, interoperability and standardization play a crucial role for its success.

This paper surveys standardization efforts towards enabling network service orchestration from an operator perspective. To elaborate on available interfaces, standards and recommendations we follow a top-down approach. We begin with a definition of the document terminology, and we elaborate on the network service orchestrator requirements and objectives from the perspective of four of the world's largest and complex network operators —British Telecom, Deutsche Telekom, NTT and China Telecom — (Section 2). Furthermore, to motivate our discussion on network services, we present the design and requirements of three popular network service use cases, namely Radio Access Network and Mobile Evolved Packet Core connectivity services and end-to-end content distribution service (Section 3). We then elaborate on the capabilities and interfaces of the predominant network (Section 4) and function (Section 5) management and control architectures. Finally, we discuss the future directions for network orchestration standardization efforts (Section 6) and conclude this paper (Section 7).

2. What is network service orchestration?

2.1. Terminology

A *network service* is a high-level network functionality that generates business value for customers and/or the operator. Network services are typically represented as directed graphs, where the nodes of the graph represent low-level network functions and the directed edges describe ordering and connectivity.

A *network or service function (NF)* is a specialized network element, designed to efficiently perform a restricted set of low-level operations on traffic. An NF can manipulate traffic at multiple layers of the protocol stack and it is common to manipulate packets traversing the network, as well as terminate network flows. Virtual software instances, such as a Broadband Network Gateway (vBNG) or IP Multimedia Subsystem (vIMS) running on a virtual machine, or specialized physical hosts, such as hardware load-balancers, are both common approaches to realize NFs. Furthermore, virtualization allows instantiation of multiple NFs on a single physical node, while a single physical node can potentially support the instantiation of multiple different NF types. Finally, NFs predominantly are designed to modify network traffic, but passive monitoring NFs are equally popular, such as intrusion detection systems.

A *Service Orchestrator* is a control system for the provision, management and re-optimization of network services. Effectively, a service orchestrator receives network service requests from individual applications, service consumers and the operator. Based on the received service requests, the available infrastructure resources and the topological properties of the underlying network, the orchestrator is responsible to define and execute a deployment plan that fulfills the NF and connectivity requirements of each service. In parallel, the service orchestrator monitors the performance of all services and dynamically adjusts the infrastructure configuration to continuously ensure the performance guarantees and cost goals.

Service Orchestration aims to support a wide range of infrastructure technologies and resource types and depends on technical standards to broaden its applicability. A technical standard reflects an established set of requirements or norms to precise technical systems. They are typically formal documents that establish uniform engineering or technical criteria, procedures, protocols and practices. This survey paper investigates the myriad of SDN and NFV standards (both formal and de-facto) across a range of Standards Development Organizations (SDO), and rapidly expanding environment of Open Source software projects. Typically, the impedance mismatch between SDOs and Open Source is at least 2:1 (two years to a paper standard versus one year to a product that creates a de-facto standard) [5].

2.2. Requirements

A Service orchestration is a complex high-level control system and relevant research efforts have proposed a wide range of goals for a service orchestrator. We identify the following functional properties:

Coordination: Operator infrastructures comprise of a wide range of network and computation systems providing a diverse set of resources, including network bandwidth, CPU and storage. Effective deployment of a network service depends on their coordinated configuration. The network manager must provision network resources and modify the forwarding policy of the network, to ensure ordering and connectivity between the service NFs. This process becomes complex when considering the different control capabilities and interfaces across network technologies found in the metropolitan, access and wide area layers of the operator network. Furthermore, the network manager must configure the devices that will host the service NFs, either in software or hardware. The service orchestrator is responsible for abstracting the management and configuration heterogeneity of the different technologies and administrative domains [6,7].

Automation: Existing infrastructures incur significant operational workload for the configuration, troubleshooting and management of network services. Network technologies typically provide different configuration interfaces in each network layer and require manual and repetitive configuration by network managers to deploy a network service [8]. In addition, vertical integration of network devices requires extensive human intervention to deploy and manage a network service in a multi-vendor and multi-technology environment. A key goal for service orchestration is to minimize human intervention during the deployment and management of network services. Efforts in programmable network and NFV control, like SDN, ABNO and ETSI NFV

MANO, provide low-level automation capabilities, which can be exploited by the service orchestrator to synthesize high-level automation service deployment and management mechanisms [9].

Resource provision and monitor: The specification of network services contain complex SLA guarantees, which perplex network management. For example, allocating resources, which meet service delivery guarantees, is an NP-hard problem from the perspective of the operator and the re-optimization of a large network can take days. In parallel, existing service deployment approaches rely on static resource allocations and require resource provision for the worst-case service load scenarios. A key goal for service orchestration is to enable dynamic and flexible resource control and monitoring mechanisms, which converge resource control across the underlying technologies and abstract their heterogeneity [10,11].

Efforts towards service orchestration are still limited. Relevant architecture and interface specifications define mechanisms for effective automation and programmability of individual resource types, like the SDN and ABNO paradigms for network resources and the NFV MANO for compute and storage resources. Nonetheless, these architectures remain low-level and provide partial control over the infrastructure towards service orchestration. Service orchestration initiatives from network operators and vendors [12,13] propose the development of a new orchestration layer above and beyond the existing individual control mechanisms which will capitalize on their low-level automation and flexibility capabilities to support a service-oriented control abstraction exposed to the OSS/BSS, as depicted in Fig. 1. In terms of network control, the service orchestrator can access low-level forwarding interfaces, as well as high high-level control interfaces implementing standardized forwarding control mechanisms, like Segment Routing and Service Function Chain, through the network controller. In parallel, NF management across the operator datacenters can be achieved through a dual-layer control and management stack, as suggested by relevant NF management architectures. The lower layer contains the Virtual Infrastructure Manager (VIM), which manages and configures the virtualization policy of compute and storage resources. The top layer contains the VNF Manager (VNFM) responsible for the configuration, control and monitor of individual NFs. The service orchestrator will operate on top of these two management services (network and IT, see Fig. 1) and will be responsible for exploiting their functionality to provide network service delivery, given the policy of the operator, channeled through the OSS. The effectiveness of the service orchestrator highly depends on the granularity and flexibility of the underlying control interfaces. This paper surveys standardization efforts for infrastructure control in an effort to discuss the existing opportunities and challenges towards service orchestration.

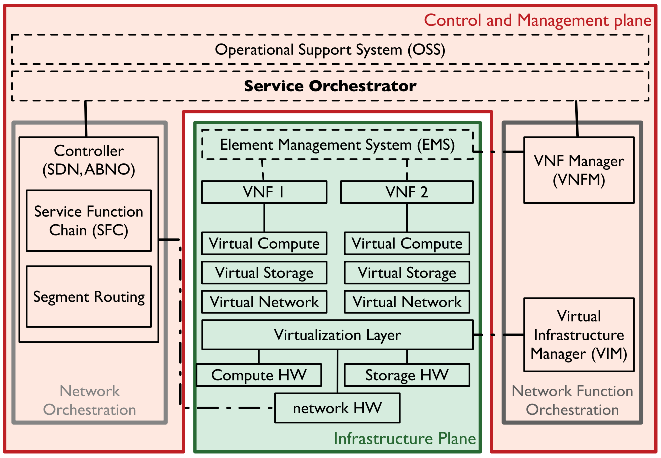


Fig. 1. An architectural model for service orchestration in operator infrastructure. The orchestrator uses the interfaces exported by the network controller and the VNF Manager to control the deployment, management, configuration and troubleshooting of network services.

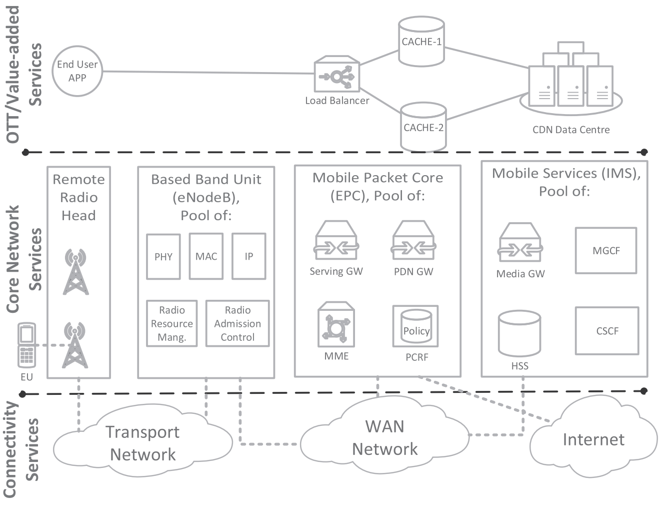


Fig. 2. An aggregate view of the functional blocks which deliver CDN and other value-added services to a mobile network.

3. Network services

Network services enable a wide range of value-added functionalities for operators and users across all layers of the infrastructure. This section presents three popular network services to identify control requirements for a service orchestrator. Specifically, we elaborate on the architecture of mobile radio access and core networks, followed by a discussion on CDN services as an example of a value-added service.

Fig. 2 depicts the abstract view of the service chain of the discussed services, along with their functional block. The figure illustrates three layers of network services: connectivity services provided by the network infrastructure; core network services that provide communication and value-added services to end-users of the network; and a top application layer, which delivers an application service to the end-user.

3.1. Radio access network (RAN)

The 3G standards split the mobile RAN in two functional blocks: the *Remote Radio Head (RRH)*, which receives and transmit the wireless signal and applies the appropriate signal transformations and amplification, and the *Base Band Unit (BBU)*, which runs the MAC protocol and coordinates neighboring cells. The channel between these two entities has high bandwidth and ultra-low latency requirements and the two systems are typically co-located in production deployments. Nonetheless, this design choice increases the operator cost to deploy and operate its RAN. BBUs are expensive components which increase the overall acquisition cost of a base station, while the BBU cooling requirements makes the RAN a significant contributor to the aggregate power consumption of the operator [14].

Recent trends in RAN design separate the two components, by moving the BBU to the central office of the operator; an architectural paradigm commonly termed Cloud-RAN (C-RAN). C-RAN significantly reduces deployment and operational costs and improves elasticity and resilience of the RAN. In parallel, the centralization of multiple RRHs under the control of a single BBU improves resource utilization and cell handovers, and minimizes cell-interference. Currently multiple interfaces, architectures and testbeds provide the technological capabilities to run and test C-RAN systems [15,16], while vendors currently provide production-ready virtualized BBU appliances [17]. In addition, novel control abstractions can converge RAN control with underlying transport technologies and enable flexible deployment strategies [18].

A challenge for C-RAN architectures is the high multi-Gb bandwidth requirements and strict sub-milliseconds latency and jitter demands for the links between the RRH and the datacenter [19]. These connectivity guarantees exhibit significant variability (from a few Mb to 30 Gb) within the course of a day, reflecting the varying loads of mobile cell, as well as the signal modulation and channel configuration. To provide flexible and on-demand front-haul connectivity with strong latency guarantees, operators require novel orchestration mechanisms supporting dynamic and multi-technology resource management. In addition, effective RAN virtualization requires a framework for the management and monitoring of BBU instances to provide service resiliency. The service orchestrator can monitor the performance of the BBU VNF instances and adjust the compute resource allocation, the VNF replication degree and the load distribution policy. In parallel, the orchestrator can improve front-haul efficiency by mapping the connectivity requirements between the BBU and the RRH in network resource allocation policy.

The 3rd Generation Partnership Project (3GPP) is actively exploring the applicability of NFV technologies on a range of mobile network use-cases, like fault-management and performance monitoring, and has defined a set of management requirements in the RAN, the Mobile Core Network and the IP Multimedia Subsystem (IMS) [20]. In parallel, the 5G Public Private Partnership (5G PPP), within its effort to standardize the technologies and protocols for the next generation communication network defines end-to-end network service orchestration as a core design goal [21].

3.2. Evolved packet core (EPC)

Evolved Packet Core (EPC) is a network architecture for the core network of mobile operators, introduced in the 4G standards. It converges voice and data traffic in a single IP-based infrastructure. EPC comprises of different functional elements providing the core mobile network services. The EPCs main functional blocks are presented in Fig. 2. The Service Gateway (SGW) is the gateway terminating the interface toward the RAN. Packet Data Network Gateway (PGW) is the gateway to Packet Delivery Network (PDN) and enforces per-user packet filtering, policing/shaping rate and traffic accounting. The Mobility Management Entity (MME) and Policy and Charging Rules Functions (PCRF) are acting as controllers for mobility and billing functions. Furthermore, the IMS provides signaling for the establishment and termination of end-to-end packet-based multimedia services, like Voice over LTE (VoLTE). These functions are currently delivered using expensive integrated network devices, which provide limited modularity and interoperability between vendors. Thus, ensuring EPC service delivery guarantees during peak times, can be achieved only during the network planning phase through network and function overprovision. Furthermore, running multiple logical networks, each providing different performance guarantees and functionalities, over a single physical infrastructure, a key functionality for 5G technologies termed *network slicing*, will require extensive virtualization of the key EPC functions [22].

Multiple studies have argued for the softwarization of the key EPC functional blocks and the introduction of programmability in the EPC network control. SoftAir [23] is a software-defined architecture for next generation mobile networks using network and function virtualization paradigms for both the EPC and the RAN. Open5GCore [24] is another effort toward the cloudification of the EPC. Effectively, the framework provides an LTE protocol stack and supports uniform and distributed control plane. Furthermore, carrier-grade IMS VNF products are readily available from different vendors [25]. Finally, both IMS and EPC services are primary use cases for the European Telecommunications Standards Institute (ETSI) NFV Industrial Specification Group (ISG) [26].

3.3. Content delivery network (CDN)

CDN services provide efficient distribution of static content on behalf of third-party Internet applications [27]. They rely on a well-provisioned and highly-available network of cache servers and allow end-users to retrieve static content with low latency by automatically redirecting them to an appropriate cache server, based on the user location, the caching policy and cache load. CDN traffic currently constitutes a large portion of the operator traffic volumes and providers, like Akamai, serve 15–30% of the global Internet traffic [28].

The CDN service chain is simple and consists of a load-balancing function and a cache function, as depicted in Fig. 2. The greatest challenge in the deployment of such a service is the aggregate network data volumes of the service and the large number of network endpoints. As a result, temporal variations in CDN traffic patterns can have a dramatic effect on the traffic matrix of the operator and affect Internet service delivery. In parallel, CDN-ISP integration lacks support for dynamical resource provision, in order to gracefully manage the dynamic traffic patterns. Connectivity relies on fixed-capacity peering relationships through popular IXPs or CDN-operated peering locations [29], which must be provisioned for the worst-case scenario.

The current design of CDN services introduces an interesting joint optimization problem between operators and CDN service providers. A CDN service bring content closer to the user and enable dynamic deployment of caching NFs in the central offices of the operator and enforce network resource guarantees. The service can provide sufficient elasticity for the CDN caching layer, while the ISP can reduce core network load. Similar approaches have been proposed in the context of mobile operators, mobile CDN emerged to faster access to mobile apps, facilitate mobile video streaming and supporting dynamic contents [30,31]. In parallel, new network control architectures based on SDN and NFV principles enable CDN services to localize users and offload the redirection task in the network forwarding policy [32,33]. These approaches provide an innovative environment to improve CDN functionality, but require a flexible control mechanism to integrate CDN services and infrastructures. A service orchestrator can autonomously adapt the CDN service deployment plan to the CDN load characteristics, using a policy specification from the CDN provider. In parallel, the orchestrator can monitor traffic volumes to infer content locality and hotspot development and deploy NF caches close to the end-user to improve latency and network efficiency.

4. Network orchestration standardization

Modern operator infrastructures contain a wide range of technologies across all network layers. Typically, the network of an operator is separated into multiple control domains (access, metropolitan and core), each using different network technologies, control interfaces and implementing forwarding policy with diverse goals [34]. Management, configuration and troubleshooting processes rely extensively on human intervention, to translate high-level connectivity goals into individual device configurations, while service deployment is designed in paper by network managers. As a result, service lead-times for new services can take up to a few months [35], with the majority of this time spent in the design and configuration of network infrastructures.

The inflexibility and limited automation in the network infrastructure has motivated the development of new control and management architectures and protocols. An important design goal for these new networking paradigms is standardization and openness of interfaces, in order to overcome the existing inter-operability limitations created by the vertical integration of network devices. In this section, we elaborate on two recent and highly successful control architectures; SDN (Section 4.1) and ABNO (Section 4.2). Such paradigms provide the required low-level control interfaces to effectively deploy services across an operator network and to control network resources. Our presentation focuses on the architecture of the respective paradigms and elaborates on the standardization efforts for the interfaces exposed to the service orchestrator.

4.1. Software defined networking (SDN)

SDN [36] is a recent network paradigm aiming for automated, flexible and user-controlled network forwarding and management. SDN is motivated by earlier network programmability efforts, including Active Networks [37], ForCES [38], RCP [39] and Tempest [40]. Unlike most earlier network programmability architectures, which explored clean-slate design of data plane protocols, SDN maintains backwards compatibility with existing network technologies. SDN design is driven by four major design goals: (i) network control and data plane separation; (ii) logical control centralization; (iii) open and flexible interfaces between control layers; and iv) network programmability.

SDN standardization efforts are primarily driven by the Open Network Foundation (ONF), while the IRTF SDNRG WG [41] explores complementary standards for the higher control layers. Similar standardization activities take place within various SDOs, namely the Broadband Forum (broadband network applications) and the International Telecommunication Union (ITU) study groups (SG) 11 (SDN signaling), SG 13 (SDN applications in future networks), SG 15 (transport network applications of SDN) and SG 17 (applications of SDN for secure services), but efforts in these SDOs are currently in early stages and provide initial problem statements and requirement analysis.

Fig. 3 presents an architectural model of an SDN control stack. The architecture separates the control functionalities into three distinct layers. The *data plane* is the bottom layer and contains all the network devices of the infrastructure. Data plane devices are designed to efficiently perform a restricted set of low-level traffic monitoring and packet manipulation functions and have limited control intelligence. Each devices implements one or more southbound Interfaces (SBIs) which enable control of the forwarding and resource allocation policy from external entities. SBIs can be categorized into control interfaces like OpenFlow [42] and PCE [43], designed to manipulate the device forwarding policy, and management interfaces, like NETCONF [44] and OF-CONFIG [45], designed to provide remote device configuration, monitoring and fault management. SDN functionality is not limited to networks supporting new clean-slate programmable interfaces and includes SBIs based on existing control protocol, like routing protocols.

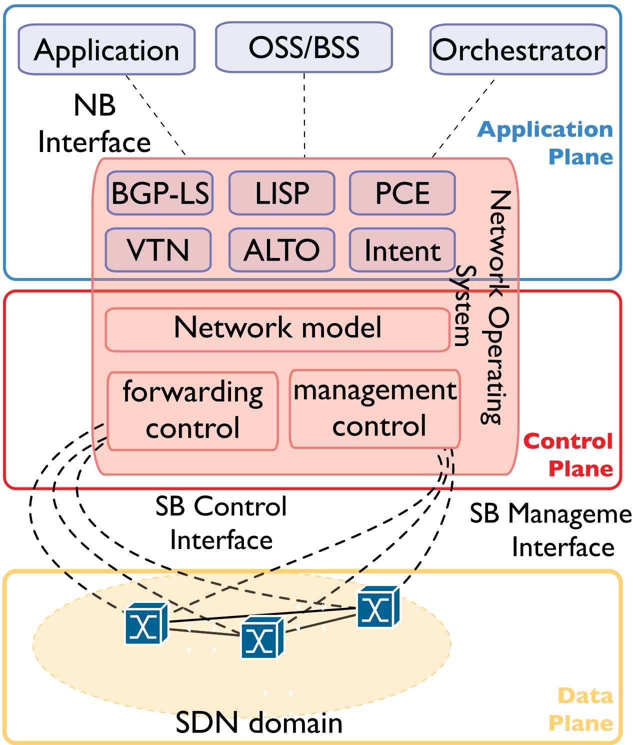


Fig. 3. The SDN architecture model can be separated in three layers: the data, control and application planes.

The *control plane* is the middle layer of the architecture and contains the Network Operating System (NOS), a focal point of the architecture. A NOS aggregates and centralizes control of multiple data plane devices and synthesize new high-level Northbound Interfaces (NBIs) for management applications. For example, existing NOS implementations provide topology monitoring and resource virtualization services and enable high-level policy specification languages, among other functionalities. Furthermore, a NOS aggregates control policy requirements from management applications and provides them accurate network state information. The NOS is responsible to analyze policy requests from individual management applications, ensure conformance with the administrative domain policy, detect and mitigate policy conflicts between management applications and translate these requests into appropriate data plane device configurations. A key element for the scalability of the architecture is logical centralization of network control; a control plane can consist of multiple NOS instances, each controlling an overlapping network segment, and use synchronization mechanisms, typically termed as eastbound and westbound interfaces, to converge in a common network-wide view of the network state and policy between NOS instances. This way, an SDN control domain can recover from multiple NOS instance failures and the control load can be distributed across the remaining instances. Finally, the *application plane* is the top layer of the architecture and contains specialized applications that use NBIs to implement high-level NFs, like load balancing and resource management.

Detailed presentation of the standardization, research and implementation efforts in the SDN community are presented in [46]. For the rest of this section we focus on NBI standardization efforts. NBIs are crucial for service orchestration, since they enable control and monitoring of service connectivity and network resource utilization and flexible fault-management. Nonetheless, NBI standardization is limited and existing control interface and mechanism design is driven by NOS development efforts.

NBIs can be organized in two broad categories. The first category contains low-level information modeling NBIs. Information models converge the state representation of data plane devices and abstract the heterogeneity of SBIs. Network information models have been developed before the introduction of the SDN paradigm by multiple SDOs, like the ITU [47,48] and the Distributed Management Task Force (DMTF) [49]. Relevant to the SDN paradigm is the ONF information modeling working group (WG), which develops the Common Information Model (CoreModel) [50] specifications. The CoreModel is hierarchical and includes a core model, which provides a basic abstraction for data plane forwarding elements, and a technology forwarding and an application-specific model, which evolve the core model abstraction. CoreModel specifications exploit object inheritance and allow control applications to acquire abstract network connectivity information and, in parallel, access technology-specific attributes of individual network devices. The CoreModel adoption is limited and existing NOSes employ custom information models.

The second NBI category contains high-level and innovative control abstractions, exploring interfaces beyond the typical match-action-forward model. These interfaces are typically implemented as NOS management applications, use the information model to implement their control logic and are consumed by external entities, like the Operation Support System (OSS), the service orchestrator and other control applications. Effectively, these interfaces manifest the reference points between the Network and Service Orchestrator components (Fig. 1). For the rest of this section we elaborate on NBI formal specifications, as well as NBI designs developed in production NOSes.

We elaborate on legacy control interfaces implemented in SDN environment, as well as interfaces supported by the ONOS [51] and OpenDayLight (ODL) [52] projects, the most popular and mature open-source NOS implementations.

Path Computation. Path Computation Element (PCE) is a control technology which addresses resource and forwarding control limitations in label-switched technologies. Generalized Multi-Protocol Label Switching (GMPLS) and Multi-Protocol Label Switching (MPLS) technologies follow a distributed approach for path establishment. Switches use traffic engineering extensions to routing protocols, like OSPF-TE [53], to collect network resource and topology information. Path requests trigger a label switch to compute an end-to-end path to the destination network using its topology information and provisions the path using signaling protocols, like RSVP-TE [54]. A significant limitation in MPLS path computation is the increased computational requirements for the co-processor of edge label switches in large networks, while limited visibility between network layers or across administrative domains can lead to sub-optimal path selections. PCE proposes a centralized path computation architecture and defines a protocol which allows the network controller to receive path requests from the NMS and to configure paths across individual network forwarding elements. PCE control can be used by the service orchestrator to provision connectivity between the NF nodes.

The ONOS PCEP project[1] enables ONOS to serve Path Computation Client (PCC) requests and to manage label switched paths (LSP) and MPLS-TE tunnels. In addition, the PCEP project develops a path computation mechanism for the ONOS tunneling subsystem and provides tunnels as a system resource. Tunnel establishment support, both as L2 and L3 VPNs, is available to application through a RESTful NBI and applications are distinguished between tunnel providers and tunnel consumers.

LSP computation relies on network topology information, stored in a traffic engineering database (TED) and populated by an Interior Gateway Protocol (IGP). This information remains local within an Autonomous System (AS), limiting Path Computation in a single administrative domain. The IETF Inter-Domain Routing WG defines a mechanism to share link-state information across domains using the Network Layer Reachability Information (NLRI) field of the BGP protocol, standardized in the BGP-LS protocol extensions [55]. The ONOS BGP-LS project introduces support for the BGP-LS protocol (peering and link state information support) as SBI to complement the ONOS PCEP project 1.

The BGP-LS/PCEP module[2] of the ODL project implements support for the aforementioned protocols as a control application. Furthermore, the module supports additional PCE extensions, like stateful-PCE [56], PCEP for segment routing (Section 5.4), and secure transport for PCEP (PCEPS) [57]. Stateful-PCE introduces time, sequence and resource usage synchronization within and across PCEP sessions, allowing dynamic LSP management. Furthermore, PCEPS adds security extension to the control channel of the PCE protocol.

ALTO. The Application Layer Traffic Optimization [58] is an IETF WG developing specifications that allow end-user applications to access accurate network performance information. Distributed network applications, like peer-to-peer and content distribution, can improve their peer-selection logic using network path information towards alternative service end-points. This better-than-random decision improves the performance of bandwidth-intensive or latency-sensitive applications, while the network provider can improve link utilization across its network. The ALTO protocol enables a service orchestrator to monitor the network of the operator and make informed service deployment decisions. ODL provides an ALTO server module[2] with a RESTful ALTO NBI.

Virtual Tenant Networks. Virtual Tenant Networks (VTNs) [59] is a network virtualization architecture, developed by NEC. VTN develops an abstraction that logically disassociates the specification of virtual overlay networks from the topology of the underlying network infrastructure. Effectively, users can define any network topology and the VTN management system will map the requested topology over the physical topology. VTN enables seamless service deployment for the service orchestrator, by decoupling the deployment plan from the underlying infrastructure. The VTN abstraction is extensively supported by the ODL project.[2]

Locator/ID Separation. The IETF Locator/ID separation protocol (LISP) [60] is a network architecture addressing the scalability problems of routing systems at Internet-scale. LISP proposes a dual addressing mechanism, which decouples the location of a host from its unique identifier. LISP-aware end-hosts require only a unique destination end-point identifier (EID) to transmit a packet, while intermediate routing nodes use a distributed mapping service to translate EIDs to Routing Locations (RLOCs), an identifier of the network of the destination host. A packet is send to an Edge LISP router in the EID domain, where a LISP header with the RLOC address of the destination network is added. The packet is then routed across the underlay network to the destination EID domain. The LISP architecture provides a scalable mechanism for NFs connectivity and mobility.

ODL provides a LISP flow mapping module.[2] The module uses an SBI to acquire RLOC and EID information from the underlying network and exposes this information through a RESTCONF NBI. In addition, the NBI allows applications, like load balancers, to create custom overlay networks. The module is currently compatible with the Service Function Chain (SFC) (Section 5.3) functionality and holds future integration plans with group-based policy mechanisms.

Real time media. The ONF has currently a dedicated WG exploring standardization requirements for SDN NBIs. At the time of writing, the group has released an NBI specifications for a Real Time Media [61] control protocol, in collaboration with the International Multimedia Telecommunication Consortium (IMTC). The protocol allows end-user applications to communicate with the local network controller, discover available resources and assign individual flows to specific quality of experience (QoE) classes, through a RESTful API. ONF is currently developing a proof-of-concept implementation of the API as part of the ASPEN project [62].

Intent-based networking. Intent-based networking is a popular SDN NBI exploring the applicability of declarative policy languages in network management. Unlike traditional imperative policy language, Intent-based policies describe to the NOS the set of acceptable network states and leave low-level network configuration and adaptation to the NOS. As a result, Intents are invariant to network parameters like link outages and vendor variance, because they lack any implementation details. In addition, intents are portable across controllers, thus simplifying application integration and run-time complexity, but requires a common NBI across platforms, which is currently an active goal for multiple SDOs WG.

The IETF has adopted the NEMO specifications [63], an Intent-based networking policy language. NEMO is a Domain Specific Language (DSL), following the declarative programming paradigm. NEMO applications do not define the underlying mechanisms for data storage and manipulation, but rather describe their goals. The language defines three major abstractions: an `end-point`, describes a network end-point, a `connection`, describes connectivity requirements between network end-points, and an `operation`, describes packet operations. Huawei is currently leading an implementation initiative, based on ODL and the OPNFV project [64].

In parallel, the ONF has recently organized a WG to standardize a common Intent model. The group aims to fulfill two objectives: i) describe the architecture and requirements of Intent implementations across controllers and define portable intent expressions, and ii)

---

[1] https://wiki.onosproject.org/display/ONOS/Feature+Proposals

[2] https://wiki.opendaylight.org/view/Project_list

develop a community-approved information model which unifies Intent interfaces across controllers. The respective standard is coupled with the development of the Boulder framework [65], an open-source and portable Intent framework which can integrate with all major SDN NOSes. Boulder organizes intents through a grammar which consists of subjects, predicates and targets. The language can be extended to include constraints and conditions. The reference Boulder implementation has established compatibility with ODL through the Network Intent Composition (NIC) project, while ONOS support is currently under development.

Group-Based Policy (GBP) is an alternative Intent-based networking paradigm, developed by the ODL project. Based upon promise-theory [66], GBP separates application concerns and simplifies dependency mapping, thus allowing greater automation during the consolidation and deployment of multiple policy specifications. The GBP abstraction models policy using the notions of end-point and end-point groups and provides language primitives to control the communication between them. Developers can specify through GBP their application requirements and the relationship between different tiers of their application, while remaining opaque towards the topology and capabilities of the underlying network. The ODL GBD module provides an NBI[2] which leverages the low-level control of several network virtualization technologies, like OpenStack Neutron [67] and SFC (Section 5.3).

4.2. Application-based network operations (ABNO)

The evolution of the SDN paradigm has highlighted that clean-slate design approaches are prone to protocol and interface proliferation which can limit the evolvability and interoperability of a deployment. ABNO [68] s an alternative modular control architecture standard, published as an Area Director sponsored RFC document, and it reuse existing standards to provide connectivity services. ABNO by-design provides network orchestration capabilities for multi-technology and multi-domain environments, since it relies on production protocols developed and adopted to fulfill these requirements. The architecture enables network applications to automatically provision network paths and access network state information, controlled by an operator-defined network policy.

ABNO consists of eight functional blocks, presented in Fig. 4 along with their interfaces, but production deployments do not require to implement all the components. A core element of the architecture is the *ABNO controller*. The controller allows applications and NMS/OSS to specify end-to-end path requirements and access path state information. A path request triggers the controller to inspect the current network connectivity and resource allocations, and to provision a path which fulfills the resource requirements and does not violate the network policy. In addition, the controller is responsible to re-optimize paths at run-time, taking under consideration other path requests, routing state and network errors. The architecture contains an *OAM handler* to collect network error from all network layers. The OAM handler monitors the network and collects error notifications from network devices, using interfaces like IPFIX and NETCONF, which are correlated in order to synthesize high-level error reports for the ABNO controller and the NMS. In addition, the ABNO architecture integrates with the network routing policy through an *Interface to the Routing System (I2RS) client*. I2RS [69] is an IETF WG that develops an architecture for real-time and event-based application interaction with the routing system of network devices. Furthermore, the WG has developed a detailed information model [70] that allows external applications to monitor the RIB of a forwarding device. As a result, the I2RS client of the ABNO architecture aggregates information from network routers in order to adapt its routing policy, while it can by modify routing tables the routing policy to reflect path availability.

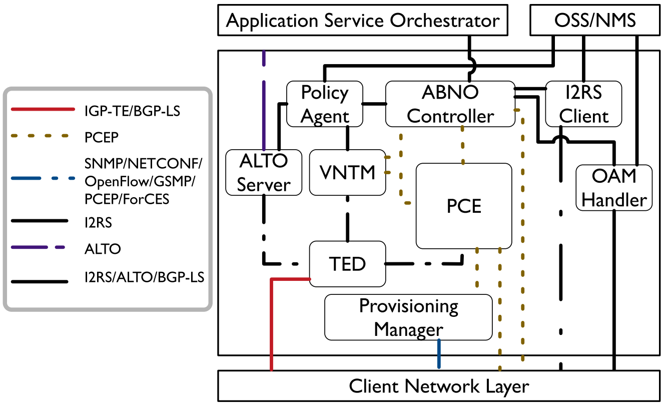


Fig. 4. The functional blocks of an ABNO architecture. Interface between functional block can re-use existing protocol standards.

Path selection is provided by a *PCE controller*, while a *provisioning manager* is responsible for path deployment and configuration using existing control plane protocols, like OpenFlow and NETCONF. It is important to highlight that these functional blocks may be omitted in a production deployment and the architecture proposes multiple overlapping control channels. In addition, the architecture contains an optional *Virtual Network Topology Manager (VNTM)*, which can provision connectivity in the network physical layer, like configuring virtual links in WDM networks.

Topology discovery is a key requirement for the path selection algorithm of the PCE controller and the ABNO architecture uses multiple databases to store relevant information. The *Traffic-Engineering Database (TED)* is a required database for any ABNO architecture and contains the network topology along with link resource and capability information. The database is populated with information through traffic engineering extensions in the routing protocol. Optionally, the architecture suggests support for an *LSP* database, which stores information for established paths, and a database to store associative information between physical links and network paths, for link capacity prediction during virtual link provision over optical technologies.

A critical element for production deployment is the ability of the ABNO architecture to employ a common policy for all path selection decisions. The ABNO architecture incorporates a *Policy Agent* which is controlled by the NMS/OSS. The policy agent authenticates requests, maintains accounting information and reflects policy restrictions for the path selection algorithm. The policy agent is a focal point in the architecture and any decision by the ABNO controller, the PCE controller and the ALTO server requires a check with the active network policy.

In addition to the ABNO control interfaces, the architecture provides additional application interfaces which expose network state information through an *ALTO server*. The server uses the ALTO protocol to provide accurate path capacity and load information to applications and assist the application server selection process and performance monitoring.

A number of ABNO-based implementations exist detailing how the architecture was used to orchestrate resources in complex network environments, including: iONE [71] for content distribution in the telecom Cloud [72], and Adaptive Network Manager [73] for co-ordinating operations in flex-grid optical and packet networks [74]. The large telecom vendor Infinera and network operator Telefonica, also provided a joint demonstration to orchestrate and provision bandwidth services in real-time (”Network as a Service - NaaS”) across a multi-vendor IP/MPLS and optical transport network, using a variety of APIs [75].

5. Function orchestration standardization

Along with the ability to control end-to-end connectivity, service

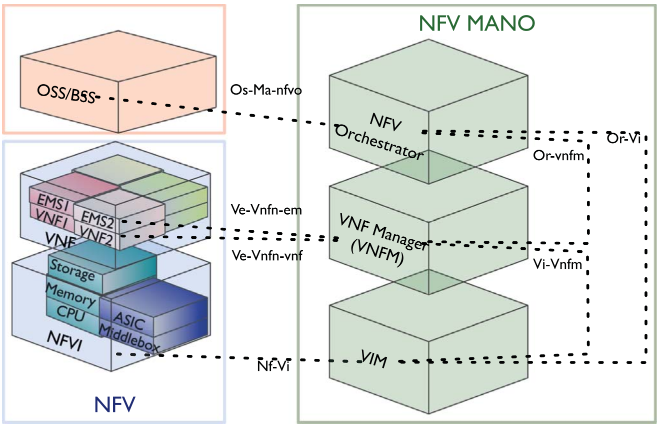


Fig. 5. ETSI NFV management and orchestration architecture.

orchestration requires support for automated control, management and configuration of NFs. Currently, NFs appear as a bump on the wire. In addition, NF implementations rely on specialized devices, while their control and management interfaces exhibit significant proliferation and heterogeneity and are not integrated with the network control plane. As a result, service deployment requires extensive human intervention to populate the network forwarding policy with static configurations that steer traffic to the desired NFs, resulting in limited service agility constrained by the underlying network topology. These limitations convolute the management of network services and increase service lead-times, especially for highly available services. Service management is further convoluted by the introduction of virtualized and software-based NFs (VNFs). Although VNFs provide service flexibility and elasticity, they introduce new functional properties, like lower performance predictability and reliability. Mixing VNF with traditional single-purpose NFs, must take under consideration these characteristics and requires fine-grain dynamic traffic steering mechanisms to ensure service liveness.

To address challenges towards flexible and agile services, multiple standardization bodies have proposed architectures, protocols, and control interfaces which enable seamless and dynamic function management. This section presents some popular NFV standardization efforts, namely the ETSI NFV Management and Orchestration (MANO) specifications (Section 5.1), the Metro Ethernet Forum (MEF) Lifecycle Service Orchestration (LSO) (Section 5.2) architecture, exploring the management organization of NFV solutions, and the IETF *Service Function Chain (SFC)* (Section 5.3) and *Segment Routing (SR)* (Section 5.4), designed to simplify the translation of service connectivity requirements into network policy.

5.1. NFV management and orchestration (NFV MANO)

The ETSI is the first SDOs to explore the applicability of the NFV paradigm in operator infrastructures [26] and to develop Proof of Concept [76] NFV implementations. Furthermore, ETSI leads the design of the popular NFV MANO architecture [77]. NFV standardization is not limited to ETSI, and other standardization bodies, like the IETF NFVRG charter [78], the Open Platform for NFV (OPNFV) industrial forum [64] and the TM Forum's ZOOM,[3] develop MANO reference implementations and propose extensions to the MANO architecture.

The MANO specifications abstract the control of virtualized infrastructures and VNF instances to external entities, like the OSS/BSS and the service orchestrator of an operator. It is currently the most popular NFV management framework, with numerous open-source and commercial implementations. Operators explore the adoption of MANO-compatible managements systems for various compounding reasons. Firstly, NFV MANO is a flexible component-based architecture which re-uses existing infrastructure management frameworks, like SDN NOSes and the OpenStack framework. Therefore, existing components can be extended by vendors, simplifying the development of NFV platforms. Secondly, the maturity and relatively detailed specification of the MANO components enable seamless interoperability between implementations from different vendors. Thirdly, the architecture provides by-design multiple carrier-grade features, like scalable hierarchical control, billing, and flexible service and function lifecycle specification.

Integration between the different functional components of the ETSI architecture is achieved through reference points, a distributed information plane which models state updates and control operations. The root element of the information plane is the Network Service (NS), which represents the service chain of a service. A NS consists of one or more *Virtual Network Functions (VNF)*, like firewalls or load balancers, connected using *Virtual Links*, while a *VNF Forwarding Graph (VNFFG)* defines VNF ordering. Furthermore, a NS may include *Physical Network Functions (PNF)*, available in the underlying network infrastructure. Finally, the MANO information model defines data repositories of NS templates, VNF catalogues, and NFVI resources, which simplify the specification and deployment of a NS.

For the rest of this section, we elaborate on the design of the MANO architecture and identify some design limitations. Fig. 5 depicts a diagram of the MANO components with the left-hand side representing the infrastructure and the right-hand side representing the management of the infrastructure. The architecture separates VNF management into three distinct layers, in an effort to support by-design clean control separation between the hosting infrastructures and the NFV managers.

Virtualized Infrastructure Manager (VIM). The VIM provides direct control and monitoring capabilities for a single NFV Infrastructure (NFVI) domain to the upper layers of the MANO architecture. VIM responsibilities include the management of the compute, network, and storage resources of a datacenter and it exposes interfaces for resource control and VNF image management. Current implementations re-use existing Cloud Management Systems (CMS), like the popular and open-source OpenStack, to realize the VIM layer. Nonetheless, the design goals of existing CMSs cannot accommodate some VIM requirements, like carrier-grade support, high-performance I/O and fine-grain and timely resource control [79,80]. Currently, OPNFV, in collaboration with ETSI, designs and develops new open-source VIM and infrastructure virtualization platforms, that bridge this requirement gap.

Virtual Network Function Manager (VNFM). The VNFM sits between the NFVO and the VIM systems and is responsible for the lifecycle management of individual VNF instances, including VNF configuration, monitoring, termination, and scaling. VNF management is typically realized using an *Element Manager (EMS)* which monitors and reports the state of each VNF to the VNFM and is capable to modify the configuration of the VNF. The deployment of an NFVM is not mandatory according to the MANO specifications and the functionality of this layer can be implemented by the NFV orchestrator. Current MANO frameworks either lack an NFVM or develop a very thin adaptation layer between the NFV orchestrator and the VIM, responsible to propagate VNF image deployment requests. Nonetheless, a VNFM can enable seamless interoperability between VNF implementations from different vendors and across cloud infrastructures [81].

Network Functions Virtualization Orchestrator (NFVO). The NFVO is responsible for the deployment and dynamic re-optimization of network services. Effectively, the NFVO receives NS requests from external entities, like the OSS and the service orchestrator, and coordinates the deployment and configuration of VNF instances across the NFVI domains. In parallel, the NFVO monitor the service perfor-

---

[3] https://www.tmforum.org/zoom/

mance and dynamically re-optimizes the deployment of VNF instance to meet the NS requirements. When creating a new NS, the NFVO optimizes placement of VNFs whilst ensuring sufficient resources and connectivity are available. Current NFVO implementations provide a thin layer capable to launch and destroy VNF chains across the NFVI domains of the operator and provide limited support for dynamic re-optimization of the service deployment.

5.2. MEF lifecycle service orchestration (LSO)

The MEF is an industrial forum, responsible for the standardization of Carrier Ethernet (CE) technologies. Furthermore, it steers the standardization efforts for the MEF LSO [82], an architecture aiming to improve automation in network service management. MEF extends the MANO architecture and introduces support for end-to-end network infrastructure management, capitalizing on the flexible control of CE technologies. LSO targets challenges of delivering Network as a Service (NaaS) functionalities in the operator infrastructure, such as on-demand, agility, and heterogeneity of virtual and physical NFs. LSO refines the service lifecycle model of the MANO standards and introduce new lifecycle capabilities, including mechanisms to automate network service request *fulfillment*, *control* of service resource and scaling, enhanced *performance* monitor and guarantees and *assurances* for service survivability. LSO aims to improve the time to establish and modify services for their future Internet vision [82]. The development of the LSO standards is still in early stages and it currently focuses on service requirement specification in order to drive the architecture design.

5.3. Service function chain (SFC)

SFC is a recently formed IETF WG which aims to define the architectural principles and protocols for the deployment and management of NF forwarding graphs. An SFC deployment operates as a network overlay, logically separating the control plane of the service from the control of the underlying network. The overlay functionality is implemented by specialized forwarding elements, using a new network header. Fig. 6 presents an example deployment scenario of an SFC domain.

An administrative network domain can contain one or more *SFC domains*. An SFC domain is a set of SFC-enabled network devices sharing a common information context. The information context contains state regarding the deployed service graphs, the available paths for each service graph and classification information mapping incoming traffic to a service path. An SFC-specific header is appended on all packets on the edges of the SFC domain by an *SFC-Classifier*. The SFC-Classifier assigns incoming traffic to a service path by appending an appropriate SFC header to each packet. For outgoing traffic, the SFC-Classifier is responsible to remove any SFC headers and forward each packet appropriately. Once the packet is within the SFC domain, it is forwarded by the classifier to an *SF Forwarder (SFF)*, an element responsible to forward traffic to an SF according to the service function ordering. Finally, the architecture is designed to accommodate both SFC-aware and legacy NFs. The main difference between them is that the SFC-aware NFs can parse and manipulate SFC headers. For legacy NFs, the architecture defines a specialized element to manipulate SFC headers on behalf of the service function, the *SFC-Proxy*. The network overlay of the SFC architecture is realized through a new protocol layer, the Network Service Header (NSH) [83]. NSH contains information which define the position of a packet in the service path, using a service path and path index identifiers, and carry metadata between service functions regarding policy and post-service delivery.

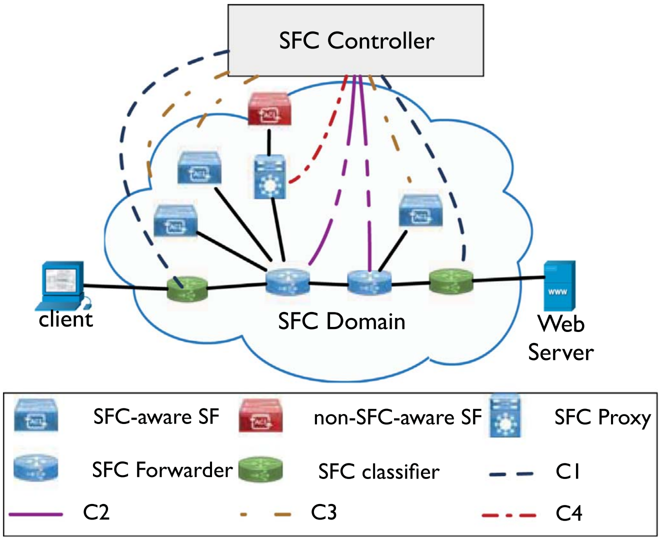


Fig. 6. IETF SFC architecture.

Highly relevant for service orchestration is the control and management interfaces of the SFC architecture. At the time of writing, the SFC WG currently explores the SFC control channel requirements and initial design goals [84] define four main control interfaces. *C1* is the control channel of the SFC-Classifier and allows manipulation of the classification policy which assigns incoming traffic to specific service paths. This control interface can be used to load balance traffic between service paths and optimize resource utilization. *C2* is a control channel of the SFF forwarding policy and exposes monitoring information, like latency and load. *C3* is the control protocol used to aggregate status, liveness and performance information from each NF-aware service function. Finally, the controller can use the *C4* protocol to configure SFC-Proxies with respect to NSH header manipulation before and after a packet traverses an SFC-unaware NF. In parallel, the WG has proposed a set of YANG models to implement the proposed control interfaces [85]. Furthermore, the WG has also specified a set of YANG models for the management interface of an SFC controller [84]. This interface provides information about the liveness of individual SFC paths, topological information for the underlying SFC infrastructure, performance counters and control of the fault and error management strategies. In addition, the management interface allows external applications to re-optimize service paths and control load balancing policy.

At the time of writing, multiple open-source platforms introduce SFC support. The Open vSwitch soft-switch has introduced SFC support both in the data and the control (OpenFlow extensions) plane. The OpenStack cloud management platform exploits the Open vSwitch SFC support and implements a high-level SFC control interface [86]. Furthermore, the ONOS controller currently supports SFC functionality using VTN overlays, while ODL implements SFC support using LISP tunnels. In addition, ONF has released recommendations for an L4-L7 SFC architecture [87] which uses OpenFlow as the SBI of the SFC controller and explores the applicability and required extension to the OpenFlow abstraction to improve support for SFF elements.

5.4. Segment routing (SR)

Segment Routing (SR) [88] is an architecture for the instantiation of service graphs over a network infrastructure using source routing mechanisms, standardized by the IETF Source Packet Routing in Networking (SPRING) WG [89].

SR is a data plane technology and uses existing protocols to store instructions (segments) for the packet path in its header. SR segments can have local or global semantics, and the architecture defines three segments types: a node segment forwards a packet over the shortest path towards a network node, an adjacency segment forwards the packet through a specific router port and a service segment introduces service differentiation on a service path. Currently, the SR architecture has defined a set of extensions for the IPv6 [90] and the MPLS [91] protocols, which define protocol-compliant mechanisms to store the segment stack and the active segment pointer in the protocol header. In addition, to enable dynamic adaptation of the forwarding policy, the architecture defines a set of control operations for forwarding elements to manipulate the packet segment list and to update established paths dynamically.

The selection of the packet path is implemented on the edge routers of the SR domain. The architecture specifies multiple path selection mechanisms, including static configurations, distributed shortest-path selection algorithms and programmatic control of segment path using SDN SBIs. The network IGP protocol can be used to provide segment visibility between routers and a YANG management interface is defined for SR segment information retrieval and SR routing entry control.

SR provides a readily-available framework to instantiate service forwarding graphs. A forwarding graph can be implemented as a segment stack and existing VNFs can be integrated with the architecture by introducing appropriate support for MPLS and IPv6 SR extensions. In comparison to the SFC architecture, SR provides a simpler architecture which does not require deployment of new network elements. Nonetheless, SFC provides wider protocol support and the architecture is designed to support different data plane technologies, while SR is closely aligned with MPLS technologies.

SR support is currently introduced in both major SDN NOSes. The ONOS project has introduced support for SR to implement CORD, a flexible central office architecture designed to simplify network service management [92]. Similarly, ODL supports SR functionality using MPLS labels and the PCE SBI module. In parallel, CISCO has introduced SR support in recent XR IOS versions [93].

6. Challenges and future directions

A variety of industry challenges remain for the standardization of key orchestration technologies. Some of the protocol solutions discussed in this paper are immature and will require further investigation and development before they can be operationalized and used by operators. In some cases, new forwarding mechanisms lack sufficient security and operational considerations required for complex and large-scale environments. The rest of this section outlines areas of new research and standardization efforts and their importance for network service orchestration.

6.1. In-operation analysis and network telemetry

he increasing demand for dynamic resource, function and connectivity provision in an orchestrated infrastructure can increase network incidents and unregulated network changes. The success of a service orchestrator depends on its ability to measure the network performance, to assess service quality using a small set of metrics and to provide network diagnosis and root cause analysis during service disruptions. In parallel, the orchestrator must support network resource scheduling which can adapt to near real-time service demands ("in-operation") [94].

To investigate network problems or identify the severity of major network events or interruptions, a network health index or network key performance index (KPI) or key quality index (KQI) is required. Generating the KPI or KQI would require data collection from various data sources using a set of automated communication processes and transmit them to one or more data aggregation services. This process is known as *network telemetry*.

The data collected from data sources include network performance data, network logging data, network warning and defects data, network statistics and state data, and network resource operation data (*e.g.*, operations on RIBs and FIBs). The process and ability to normalize the data to derive several end-to-end network composite metrics that reflect the network performance and quality from different perspectives, like network diagnosis, network performance, network QoS, network security. These end-to-end metrics can then be used for in-operation planning.

6.2. Orchestrator scalability

The size and scale of service orchestration interfaces manifest a complex distributed computing system. Operator infrastructures contain multiple computational resources (*i.e.*, CPU, memory, storage, and function) that are connected via the network and together they perform a task. Logical centralization for the infrastructure control and management systems, where a group of control elements exposes a unified and centralized abstraction to the layer above, has become a key design goal.

The CAP theorem [95] identifies three characteristics that are universally desirable, but cannot be met concurrently by any distributed system: *Consistency*, describes the ability of the system to respond identically to a request no matter which element receives the request; *Availability*, describes the ability of the system to always respond to a request; and *Partition Tolerance*, describes the ability of the system to function uninterrupted when nodes or communications links fail.

An orchestrator will act on request and connect to the various control elements. Tolerance to loss of connectivity from the orchestrator and various controllers is typically not discussed by most of the technologies discussed in this survey paper. The consistency, availability and partitioning issues may be solved by clustering critical components and duplicating databases, but large-scale resource pooling and state synchronization challenges will need to be addressed in the protocol and architecture design phase. It is critical for SDO to understand the consistency, performance and resilience requirements of each orchestration interface and define operational semantics for control operation.

6.3. Security and trust

The traditional attack vectors on traffic flows, switches, and functions, and recovery and fault diagnosis, have resulted in new security issues that are specific to SDN and NFV [96,97]. The features, capabilities and services outlined in our survey will introduce faults and risks that expose network infrastructure to threats that did not previously exist, or were ring-fenced by single OSS platforms, and are significantly more serious, with a greater potential for harm. Furthermore, security flaws can result when an open source project has a weak security focus (often the result of critical technology with too few reviewers and maintainers). This result has manifested recently in OpenSSL (HeartBleed), and is now being addressed through the Linux Foundation critical infrastructure project (for OpenSSL, OpenSSH and NTPd).

In co-operative controller environments or orchestrators that are capable of directly accessing and manipulating another technology or administrative domain controller, the risks associated with one compromised entity are now compounded, as attackers are able to attack a single resource control point. This is distinct from a larger number of autonomous assets in a completely distributed control architecture. Automation via orchestration is a double-edged sword; it offers flexibility to implement new, innovative and market-driven applications but it also opens the door to malicious and vulnerable applications. A sufficient *Trust Model* must be developed for SDN-based and NFV-based infrastructures, implementing robust authentication and enforcing different authorization levels during application registration to the orchestrator, in order to limit the exposure to misconfiguration, and malicious intent.

6.4. Service modeling

An important step towards effective network services orchestration is the development of models which capture the resource requirements, configuration parameters, performance metrics and fault management of network services. These models can drive the development of the interfaces between applications, service consumers and the service orchestrator. Standardizing a common set of service models can enable orchestrator-application interoperability between operators and address limitations arising in the deployment of services that span across multiple administrative domains.

Efforts towards service modeling are fairly recent and their outcomes are still limited. We identify two relevant SDO efforts: the Topology and Orchestration Specification for Cloud Applications (TOSCA) from the Organization for the Advancement of Structured Information Standards (OASIS) and the IETF NETCONF Data Modeling Language (NETMOD) WG. The TOSCA technical committee (TC) recently expanded its scope with a new goal to model VNF network services. At the time of writing, the TC has released a draft model [98], closely aligned with the information points in the ETSI MANO architecture. The IETF NETMOD WG provides a richer portfolio of model specifications, developed using the YANG [99] data modeling language. The respective models can be classified in two broad categories: network element models and network service models [100]. Relevant to network service modeling are the latter models, but the scope of these models remains limited and primarily focuses on connectivity services.

One of the key challenges towards network service modeling, is the definition of unified configuration and management VNF interfaces. Effectively, the interface between the VNF EMS layer and the VNFM service currently lacks standardization. VNF appliances comes in many different shapes and sizes and operate across all network layer. The high dimensionality of VNF interfaces can significantly impair automation in service orchestration. Relevant efforts in cloud computing have deliver frameworks, like Ansible [101] and Chef [102], which simplify the deployment of web services for large scale systems using configuration template. These systems provide *cookbooks* containing service *recipes* which abstract and automate web service and VM configuration. These approaches should be revisited and adapted in the context of network service deployment and configuration practices.

7. Summary

Operators currently face significant challenges to maintain profitability over their infrastructures and, in parallel, support network service innovation. Modern network infrastructures are complex systems, comprising of heterogeneous technologies, each with different proprietary configuration and management interfaces. Given the relatively long deployment times and static nature of existing customer services, the network service deployment and management is achieved using limited cross function collaboration, system focused and top-down command and control.

A key goal for operators is the development of new network service orchestration mechanisms which provide convergence between network technologies, automation in the deployment and management of network service and flexible and cross-layer resource control and provision. Towards this goal, new technological paradigms, including SDN and NFV, and new network architectures, such as SFC and SR, provide the opportunity to augment elasticity, programmability, interoperability and agility in the control and management of operator infrastructures and reduce CAPEX and OPEX.

This paper surveyed the standardization activities carried out in the recent years in the context of network service orchestration, in an effort to aid researchers and practitioners to understand the capabilities of the relevant technologies. We presented a simple architectural model for network service orchestration and we identified two principal elements in the management and control of operator infrastructures: network and NF orchestration. For each element, we presented the predominant architectural specifications and elaborated on the interfaces that each technology provides. Finally, we examined a number of future directions for the relevant SDO.

Acknowledgements

The authors are grateful to the UK Engineering and Physical Sciences Research Council (EPSRC) for funding the TOUCAN project (EP/L020009/1), which supported much of the work presented in this paper; and to the EPSRC , UK and BT for funding Daniel King's Industrial CASE Ph.D. research.

References

[1] R. Chua, SDXcentral: Christos Kolias, Orange Kicks off NFV Interview Series, 2013 URL ⟨https://www.sdxcentral.com/articles/interview/christos-kolias-sdncentrals-nfv-interview-series/2013/07/⟩.

[2] ITU, ICT Facts and Figures: The World in 2015, May 2015 ⟨http://www.itu.int/en/ITU-D/Statistics/Documents/facts/ICTFactsFigures2015.pdf⟩.

[3] Cisco, Cisco Visual Networking Index: Forecast and Methodology, 2014–2019 White Paper.

[4] Alcatel-Lucent, The Declining Profitability Trend of Mobile Data: What Can be Done? ⟨http://www3.alcatel-lucent.com/belllabs/advisory-services/documents/Declining_Profitability_Trend_of_Mobile_Data_EN_Market_Analysis.pdf⟩.

[5] D. Ward, Open standards, open source, open loop, IETF J. 10 (2015).

[6] A. Aguado, M. Davis, S. Peng, M.V. Ãlvarez, V. LÃpez, T. Szyrkowiec, A. Autenrieth, R. Vilalta, A. Mayoral, R. Muoz, R. Casellas, R. Martnez, N. Yoshikane, T. Tsuritani, R. Nejabati, D. Simeonidou, Dynamic virtual network reconfiguration over sdn orchestrated multitechnology optical transport domains, J. Lightwave Technol. 34 (8) (2016).

[7] A. Csaszar, W. John, M. Kind, C. Meirosu, G. Pongracz, D. Staessens, A. Takacs, F. J. Westphal, Unifying cloud and carrier network: EU FP7 Project UNIFY, in: IEEE/ACM UCC, Dec 2013.

[8] T. Benson, A. Akella, D. Maltz, Unraveling the complexity of network management, in: NSDI. USENIX, 2009.

[9] L. Lei, SDN orchestration for dynamic end-to-end control of data center multi-domain optical networking, China Commun. 12 (8) (2015).

[10] Y. Yoshida, A. Maruta, K. Kitayama, M. Nishihara, T. Tanaka, T. Takahara, J.C. Rasmussen, N. Yoshikane, T. Tsuritani, I. Morita, S. Yan, Y. Shu, M. Channegowda, Y. Yan, B.R. Rofoee, E. Hugues-Salas, G. Saridis, G. Zervas, R. Nejabati, D. Simeonidou, R. Vilalta, R. Munoz, R. Casellas, R. Martinez, M. Svaluto, J.M. Fabrega, A. Aguado, V. Lopez, J. Marhuenda, O.G. de Dios, J.P. Fernandez-Palacios, First international SDN-based network orchestration of variable-capacity OPS over programmable flexi-grid EON, in: OFC, March 2014.

[11] S. Sahhaf, W. Tavernier, J. Czentye, B. Sonkoly, P. Skoldstrom, D. Jocha, J. Garay, Scalable architecture for service function chain orchestration, in: EWSDN, Sept 2015.

[12] J. Ellerton, A. Lord, P. Gunning, K. Farrow, P. Wright, D. King, D. Hutchison, Prospects for software defined networking and network function virtualization in media and broadcast, in: SMPTE, Oct 2015.

[13] B. Ethirajulu, Above & Beyond Mano,⟨https://www.ericsson.com/spotlight/cloud/blog/2015/12/02/beyond-mano/⟩.

[14] China Mobile, C-RAN: The Road Towards Green RAN, 2011.

[15] S. Bhaumik, S.P. Chandrabose, M.K. Jataprolu, G. Kumar, A. Muralidhar, P. Polakos, V. Srinivasan, T. Woo, CloudIQ: A framework for processing base stations in a data center, in: Mobicom. ACM, 2012.

[16] N. Nikaein, M.K. Marina, S. Manickam, A. Dawson, R. Knopp, C. Bonnet, OpenAirInterface: a flexible platform for 5G research, SIGCOMM Comput. Commun. Rev. 44 (5) (. 2014).

[17] Alcater-Lucent, vRAN, URL ⟨https://www.alcatel-lucent.com/solutions/vran⟩.

[18] R. Riggio, K. Gomez, L. Goratti, R. Fedrizzi, T. Rasheed, V-Cell: Going beyond the cell abstraction in 5G mobile networks, in: IEEE NOMS, May 2014.

[19] Common Public Radio Interface (CPRI); Interface Specification V6.0 (2013-08-30), 2013 URL ⟨http://www.cpri.info/downloads/CPRI_v_6_0_2013-08-30.pdf⟩.

[20] 3G PPP, TR 32.842: Telecommunication management; Study on network management of virtualized networks, 2015 URL ⟨https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?SpecificationId=2248⟩.

[21] 5G PPP, 5G Vision: The 5G Infrastructure Public Private Partnership: The Next Generation of Communication Networks and Services, 2015 URL ⟨https://5g-ppp.eu/wp-content/uploads/2015/02/5G-Vision-Brochure-v1.pdf⟩.

[22] Ericsson, 5G systems: Enabling The Transformation of Industry And Society, 2017. ⟨https://www.ericsson.com/assets/local/publications/white-papers/wp-5g-systems.pdf⟩.

[23] I.F. Akyildiz, P. Wang, S.-C. Lin, SoftAir: a software defined networking architecture for 5G wireless systems, Comput. Netw. 85 (2015).

[24] Open5GCore: The Next Mobile Core Network Testbed Platform, ⟨http://www.open5gcore.org⟩.

[25] Alcatel-Lucent, Rapport cloud communications, ⟨https://www.alcatel-lucent.com/solutions/cloud-communications⟩.

[26] ETSI, Network Functions Virtualisation (NFV); Use Cases, 2013.

[27] B. Krishnamurthy, C. Wills, Y. Zhang, On the use and performance of content distribution networks, in: IMW. ACM, 2001.

[28] Akamai, Akamai Facts and Figures, 2015 ⟨https://www.akamai.com/us/en/about/facts-figures.jsp⟩.

[29] Netflix, Netflix Peering Locations, 2006 ⟨https://openconnect.netflix.com/en/peering-locations/⟩.

[30] X. Wang, M. Chen, T. Taleb, A. Ksentini, V.C.M. Leung, Cache in the air: exploiting content caching and delivery techniques for 5G systems, IEEE Commun. Mag. 52 (2) (2014).

[31] X. Li, K. Kanonakis, N. Cvijetic, A. Tanaka, C. Qiao, T. Wang, Joint bandwidth provisioning and cache management for video distribution in software-defined passive optical networks, in: OFC. OSA, 2014.

[32] P. Georgopoulos, M. Broadbent, A. Farshad, B. Plattner, N. Race, Using software defined networking to enhance the delivery of video-on-demand, Comput. Commun. 69 (2015) 9.

[33] B. Frank, I. Poese, Y. Lin, G. Smaragdakis, A. Feldmann, B. Maggs, J. Rake, S. Uhlig, R. Weber, Pushing CDN-ISP collaboration to the limit, SIGCOMM Comput. Commun. Rev. 43 (3) (2013).

[34] R.D. Doverspike, K.K. Ramakrishnan, and C. Chase, Guide to reliable internet services and applications. London: Springer, 2010, ch. Structural Overview of ISP Networks, pp. 19–93.

[35] British Telecom, BTnet: Market Leading Leased Line Internet, Aug. 2015. ⟨http://business.bt.com/assets/pdf/broadband-and-internet/datasheet/BTnet_target_availability.pdf⟩.

[36] Open Network Foundation, Software-Defined Networking: The New Norm for Networks, 2012 ⟨https://www.opennetworking.org/images/stories/downloads/sdn-resources/white-papers/wp-sdn-newnorm.pdf⟩.

[37] D.L. Tennenhouse, J.M. Smith, W.D. Sincoskie, D.J. Wetherall, G.J. Minden, A survey of active network research, IEEE Commun. Mag. 35 (1) (1997).

[38] A. Doria, J.H. Salim, R. Haas, H. Khosravi, W. Wang, L. Dong, R. Gopal, J. Halpern, Forwarding and control element separation (ForCES) protocol specification, Internet RFC, RFC 5810 (2010).

[39] M. Caesar, D. Caldwell, N. Feamster, J. Rexford, A. Shaikh, J. van der Merwe, Design and implementation of a routing control platform, in: NSDI. USENIX, 2005.

[40] J.E. van der Merwe, S. Rooney, L. Leslie, S. Crosby, The Tempest-a practical framework for network programmability, IEEE Network 12 (May (3)) (1998).

[41] IRTF, IRTF Software-Defined Networking Research Group, 2015 URL ⟨https://irtf.org/sdnrg⟩.

[42] Open Network Foundation, OpenFlow Switch Specifications 1.5.0, 2015 URL ⟨https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow/openflow-switch-v1.5.1.pdf⟩.

[43] J. Vasseur, J.L. Roux, Path computation element (PCE) communication protocol (PCEP), Internet RFC, RFC 5440, 2009.

[44] R. Enns M. Bjorklund J. Schoenwaelder A. Bierman, Network configuration protocol (NETCONF), Internet RFC, RFC 6241, 2011.

[45] Open Network Foundation, OF-CONFIG 1.2: OpenFlow Management and Configuration Protocol, 2014 URL ⟨https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow-config/of-config-1.2.pdf⟩.

[46] D. Kreutz, F.M.V. Ramos, P.E. Veríssimo, C.E. Rothenberg, S. Azodolmolky, S. Uhlig, Software-defined networking: a comprehensive survey, Proc. IEEE 103 (1) (2015).

[47] ITU, ITU-T Recommendation M.3100: Generic Network Information Model, 2005 URL ⟨https://www.itu.int/rec/T-REC-M.3100/en⟩.

[48] ITU, ITU-T Recommendation M.3102: Unified Generic Management Information Model for Connection-oriented and Connectionless Networks,⟨https://www.itu.int/rec/T-REC-M.3102/en⟩, 2011.

[49] DMTF, Dmtf Common Information Model,⟨http://www.dmtf.org/standards/cim⟩, 2016.

[50] Open Network Foundation, Core Information Model (CoreModel),⟨https://www.opennetworking.org/images/stories/downloads/sdn-resources/technical-reports/ONF-CIM_Core_Model_base_document_1.1.pdf⟩, 2015.

[51] P. Berde, M. Gerola, J. Hart, Y. Higuchi, M. Kobayashi, T. Koide, B. Lantz, B. O'Connor, P. Radoslavov, W. Snow, G. Parulkar, ONOS: Towards an Open, Distributed SDN OS, in: HotSDN. ACM, 2014.

[52] The OpenDaylight Platform, ⟨https://www.opendaylight.org/⟩.

[53] D. Katz K. Kompella D. Yeung, Traffic engineering (TE) extensions to OSPF version 2, Internet RFC, RFC 3630, September 2003.

[54] D. Awduche, L. Berger, D. Gan, T. Li, V. Srinivasan, G. Swallow, RSVP-TE: extensions to RSVP for LSP tunnels, Internet RFC, RFC 3209, December 2001.

[55] J. Medved, S. Previdi, S. Ray, H. Gredler, A. Farrel, North-Bound Distribution of Link-State and Traffic Engineering (TE) Information Using BGP, RFC 7752, Mar. 2016. [Online]. Available:⟨https://rfc-editor.org/rfc/rfc7752.txt⟩.

[56] J. Medved, I. Minei, E. Crabbe, R. Varga, PCEP Extensions for Stateful PCE, Internet Engineering Task Force, Internet-Draft draft-ietf-pce-stateful-pce-14, Mar. 2016, work in Progress. [Online]. Available: ⟨https://tools.ietf.org/html/draft-ietf-pce-stateful-pce-14⟩.

[57] D. Lopez, O.G. de Dios, Q. Wu, D. Dhody, Secure Transport for PCEP, Internet Engineering Task Force, Internet-Draft draft-ietf-pce-pceps-10, Jul. 2016, work in Progress. [Online]. Available: ⟨https://tools.ietf.org/html/draft-ietf-pce-pceps-10⟩.

[58] J. Seedorf, E. Burger, Application-Layer Traffic Optimization (ALTO) Problem Statement, RFC 5693, Oct. 2015.

[59] NEC, ProgrammableFlow: Redefining Cloud Network Virtualization With OpenFlow, 2011.

[60] D. Farinacci, V. Fuller, D. Meyer, D. Lewis, The Locator/ID Separation Protocol (LISP), RFC 6830, Oct. 2015. [Online]. Available:⟨https://rfc-editor.org/rfc/rfc6830.txt⟩.

[61] Open Network Foundation, ONF TR-517: Real Time Media NBI REST Specification, ⟨http://www.opennetworking.org/images/stories/downloads/sdn-resources/technical-reports/Real_Time_Media_NBI_REST_Specification.pdf⟩, 2015.

[62] Open Networking Foundation, Project ASPEN: real time media interface specification, ⟨http://opensourcesdn.org/projects/project-aspen-real-time-media-interface-specification/⟩.

[63] S. Hares, Intent-Based Nemo Overview, Internet Draft, RFC, Oct. 2015. [Online]. Available:⟨https://tools.ietf.org/pdf/draft-hares-ibnemo-overview-01.pdf⟩.

[64] C. Price, S. Rivera, OPNFV: An open platform to accelerate NFV,⟨https://networkbuilders.intel.com/docs/OPNFV_WhitePaper_Final.pdf⟩, 2012.

[65] Open Networking Foundation, Project BOULDER: intent northbound interface (NBI),⟨http://opensourcesdn.org/projects/project-boulder-intent-northbound-interface-nbi/⟩.

[66] P. Borril, M. Burgess, T. Craw, M. Dvorkin, A promise theory perspective on data networks, arXiv preprint arXiv:1405.2627, 2014.

[67] OpenStack, Neutron Developer Documentation,⟨http://docs.openstack.org/developer/neutron/⟩.

[68] D. King, A. Farrel, A PCE-Based architecture for application-based network operations, Internet RFC, RFC 7491, March 2015.

[69] A. Atlas, T. Nadeau, D. Ward, Problem statement for the interface to the routing system, Internet RFC, RFC 7920, June 2016.

[70] J. Clarke, G. Salgueiro, C. Pignataro, Interface to the Routing System (I2RS), Traceability: Framework and Information Model Internet RFC, RFC 7922, June 2016.

[71] L. Gifre, N. Navarro, A. Asensio, M. Ruiz, L. Velasco, iONE: An environment for experimentally assessing in-operation network planning algorithms, in: ICTON, July 2015.

[72] LluÃs Gifre, Massimo Tornatore, Luis M. Contreras, Biswanath Mukherjee, Luis Velasco, ABNO-driven content distribution in the telecom cloud, Opt. Switch. Netw. (2015).

[73] A. Aguado, V. Lopez, J. Marhuenda, O.G. de Dios, J.P. Fernandez-palacios, ABNO: a feasible SDN approach for multivendor IP and optical networks [Invited], IEEE/OSA J. Opt. Commun. Netw. 7 (2) (2015).

[74] V. Lopez, O. Gerstel, R. Casellas, A. Farrel, D. King, S. Lopez-Buedo, A. Cimmino, R. Morro, J. Fernandez-Palacios, Adaptive network manager: Coordinating operations in flex-grid networks, in: ICTON, June 2013.

[75] Telefonica, Telefonica & Infinera Network as a Service (naas) Demonstration Using Software Defined Networks,⟨https://www.infinera.com/wp-content/uploads/2015/08/Infinera-Telefonica_SDN_Demo.pdf⟩.

[76] ETSIs, NFV Proofs of Concept,⟨http://www.etsi.org/technologies-clusters/technologies/nfv/nfv-poc⟩.

[77] ETSI, Network Functions Virtualisation (NFV); Management and Orchestration, ⟨http://www.etsi.org/deliver/etsi_gs/NFV-MAN/001_099/001/01.01.01_60/gs_nfv-man001v010101p.pdf⟩, 2014.

[78] IETF, Network Function Virtualization Research Group,⟨https://irtf.org/nfvrg⟩, 2016.

[79] J. Soares, C. Goncalves, B. Parreira, P. Tavares, J. Carapinha, J.P. Barraca, R.L. Aguiar, S. Sargento, Toward a telco cloud environment for service functions, IEEE Commun. Mag. 53 (2) (2015).

[80] F.Z. Yousaf, T. Taleb, Fine-grained resource-aware virtual network function management for 5 g carrier cloud, IEEE Netw. 30 (2) (2016).

[81] Y. Chen, Y. Qin, M. Lambe, W. Chu, Realizing network function virtualization management and orchestration with model based open architecture, in: CNSM. IEEE, 2015.

[82] MEF, The third network: Lifecycle service orchestration vision,⟨https://www.mef.net/Assets/White_Papers/MEF_Third_Network_LSO_Vision_FINAL.pdf⟩, Nov. 2015.

[83] P. Quinn, U. Elzur, A Border Gateway Protocol 4 (BGP-4), Internet Draft, RFC, Jan. 2016. [Online]. Available:⟨https://tools.ietf.org/pdf/draft-ietf-sfc-nsh-02.pdf⟩.

[84] H. Li, Q. Wu, O. Huang, M. Boucadair, C. Jacquenet, W. Haeffner, S. Lee, R. Parker, L. Dunbar, A. Malis, J. Halpern, T. Reddy, P. Patil, Service Function Chaining (SFC) Control Plane Components and Requirements, Internet-Draft, Informational, Jan. 2016.

[85] R. Penno, P. Quinn, D. Zhou, J. Li, Yang Data Model for Service Function Chaining, Internet-Draft, Informational, Jan. 2016.

[86] OpenStack, Service Function Chaining Extension for Openstack Networking, ⟨http://docs.openstack.org/developer/networking-sfc/⟩.

[87] Open Network Foundation, ONF TS-027: L4-L7 Service Function Chaining Solution Architecture,⟨https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/L4-L7_Service_Function_Chaining_Solution_Architecture.pdf⟩, 2015.

[88] C. Filsfils, N.K. Nainar, C. Pignataro, J.C. Cardona, P. Francois, The Segment Routing Architecture, in: IEEE GLOBECOM, Dec 2015.

[89] C. Filsfils, S. Previdi, B. Decraene, S. Litkowski, R. Shakir, Segment Routing Architecture, Internet-Draft, Tech. Rep., Jun. 2016.

[90] S. Previdi, C. Filsfils, B. Field, I. Leung, J. Linkova, E. Aries, T. Kosugi, E. Vyncke, D. Lebrun, IPv6 Segment Routing Header (SRH), Internet-Draft, Tech. Rep., Mar. 2016.

[91] C. Filsfils, S. Previdi, A. Bashandy, B. Decraene, S. Litkowski, M. Horneffer, R. Shakir, J. Tantsura, E. Crabbe, Segment Routing with MPLS data plane, Internet-Draft, Tech. Rep., Mar. 2016.

[92] Open Network Operating System, CORD: reinventing central offices for efficiency and agility, ⟨http://opencord.org/⟩.

[93] CISCO, Cisco IOS XR Routing Configuration Guide - Segment Routing,⟨http://www.cisco.com/c/en/us/td/docs/routers/crs/software/crs_r5-2/routing/configuration/guide/b_routing_cg52xcrs/b_routing_cg52xcrs_chapter_0110.html#concept_360441CD7F564CF99C0E1765E15EB47F⟩.

[94] L. Velasco, A. Castro, D. King, O. Gerstel, R. Casellas, V. Lopez, In-operation network planning, IEEE Commun. Mag. 52 (1) (2014).

[95] E. A. Brewer, Towards robust distributed systems (abstract), in: PODC. ACM, 2000.

[96] S. Scott-Hayward, G. O'Callaghan, S. Sezer, SDN Security: A Survey, in: IEEE SDN4FNS, Nov 2013.

[97] ETSI, Network Functions Virtualisation (NFV); NFV Security; Privacy and Regulation; Report on Lawful Interception Implications,⟨http://www.etsi.org/deliver/etsi_gs/NFV-SEC/001_099/002/01.01.01_60/gs_NFV-SEC002v010101p.pdf⟩.

[98] OASIS TOSCA, TOSCA Simple Profile for Network Functions Virtualization (NFV) Version 1.0,⟨http://docs.oasis-open.org/tosca/tosca-nfv/v1.0/csd02/tosca-nfv-v1.0-csd02.pdf⟩, 2015.

[99] Bjorklund, YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF), RFC 6020, Oct. 2015. [Online]. Available:⟨https://rfc-editor.org/rfc/rfc6020.txt⟩.

[100] B. Claise, C. Moberg, YANG Model Classification, Internet Draft, RFC, Apr. 2016.

[101] Ansible, ⟨https://www.ansible.com⟩.

[102] Chef, ⟨https://www.chef.io⟩.

SMPTE Meeting Presentation

Prospects for Software Defined Networking and Network Function Virtualization in Media and Broadcast

John Ellerton

Andrew Lord

Paul Gunning

Kristan Farrow

Paul Wright

British Telecom, United Kingdom

Daniel King

David Hutchison

Lancaster University, United Kingdom

Written for presentation at the

SMPTE 2015 Annual Technical Conference & Exhibition

Abstract *Software Defined Networking (SDN) and Network Function Virtualization (NFV) provide an alluring vision of how to transform broadcast, contribution and content distribution networks. In our laboratory we assembled a multi-vendor, multi-layer media network environment that used SDN controllers and NFV-based applications to schedule, coordinate, and control media flows across broadcast and contribution network infrastructure.*

This paper will share our experiences of investigating, designing and experimenting in order to build the next generation broadcast and contribution network. We will describe our experience of dynamic workflow automation of high-bandwidth broadcast and media services across multi-layered optical network environment using SDN-based technologies for programmatic forwarding plane control and orchestration of key network functions hosted on virtual machines. Finally, we will outline the prospects for the future of how packet and optical technologies might continue to scale to support the transport of increasingly growing broadcast media.

Keywords *Software Defined Networks, Network Function Virtualization, Ethernet, Broadcast, Contribution, Optical Switching.*

(The SMPTE disclaimer is on a footer on this page, and will show in Print Preview or Page Layout view.)

1. Introduction

The broadcast industry is in a time of significant change which is impacting the way we design, deploy and operate broadcast and contribution network infrastructure.

As the number of consumer media consumption devices continues to increase exponentially, whether to watch live television or on-demand content, the pressure on the broadcast network operator to deliver fast, secure, and reliable connective capacity across the contribution and distribution infrastructure increases.

Although the contribution and distribution network share common technology requirements, distinct objectives must still be defined. Contribution networks need to support seamless, resilient, uncompressed and real-time transmission of multi-format production content. Distribution networks must also scale, but to support a wide variety of low bit-rate streams, as consumer electronics manufacturers push 4K Smart TVs into the home, and sell High Dynamic Range-equipped TVs, creating consumer demand for Ultra High Definition (UHD) content to view on Internet connected TVs.

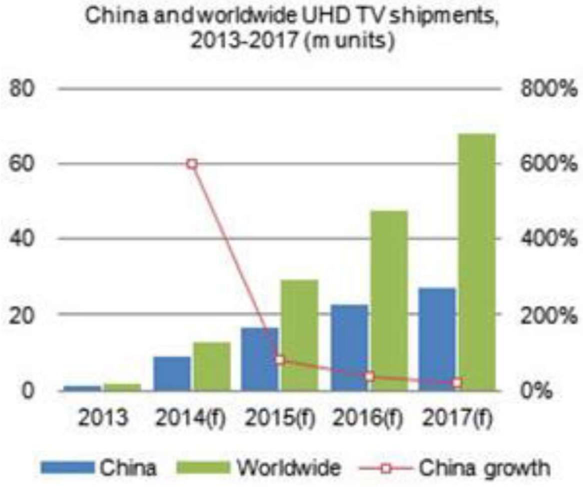


Figure 1: UHD Shipments from DIGITIMES Research 2014

> *"The primary problem we have is that our customer traffic continues to grow exponentially and the revenue we receive to carry the traffic is not growing at the same rate"* ***(Principal Member of the Technical Staff, Verizon)****.*

This paper provides a view into the British Telecom's media and broadcast, contribution and distribution laboratory efforts. We outline how the technology and economics of "Software Defined Networking" and "Network Functions Virtualization", both buzzwords of broadcast shows and conferences, are already impacting the way we consider requirements, design and deploy network infrastructure. We conclude with our future research objectives for continued development of media and broadcast network infrastructure.

2. Media & Broadcast Goals, Stakeholders and Infrastructure

The reader of this paper is no doubt familiar with broadcast, contribution and distribution network types. This paper may interchange the terms occasionally as key infrastructure components are shared across multiple network types.

As British Telecom operates multiple network types, we are subject to a variety of market forces that we must balance. These may be categorized into the stakeholders (i.e., producers, broadcasters, content and distribution operators, and consumers). These perspectives and requirements are sometimes shared, where unique to a specific stakeholder we will endeavor to underline the fact.

This paper will refer to network and infrastructure, i.e., the hardware and software resources enabling network connectivity, communication, operations and management of broadcast services. Deployment scenarios include in-facility (studios, production sites and broadcast plants) and Wide-Area Networks interconnecting media locations.

2.1 Leveraging an Economy of (Network) Scale

Broadcasters are challenged with increasing capacity demand, reducing service setup times and competitive pressures. The need for innovation is focused on finding more cost efficient ways of moving high volumes of data, and in particular the need to address the current dependence on expensive, dedicated hardware and processors.

> *"The biggest problem is that we're so used to using legacy equipment, where you've got dedicated equipment that do very specific functions" (**Senior R&D Engineer at BBC Research & Development**).*

A leading organization in this search for solutions based on cheaper, generic Ethernet and IT hardware has been British Telecom, working independently initially but then with a growing group of other operators from around the world.

> *"I had various discussions with colleagues going back over many years about the potential for generic processors to shift packets and got into various discussions as to what sort of packets; you know packet performance was obviously the main parameter of interest. We then got into more detailed discussion with Intel [about five] years ago and initiated a study for them which they grew into a wider set of partners" (**Chief Data Networks Strategist, British Telecom**).*

The development of these exploratory collaborations between operators and vendors was a significant precursor to the current move towards commodity-based Ethernet switching. In these early stages the main focus was on finding innovative ways to use cheaper, generic Ethernet hardware using a centralized controller as an alternative to the more costly dedicated network hardware, running proprietary chips and proprietary software. These current provisions were costly in part because the vendors could lock-in operators through the lack of interoperability of their hardware and software solutions with others on the market.

> *"It's about reducing, well, the direct hour costs, if you are buying normal standard switches and servers it's much cheaper than buying expensive dedicated boxes. One of the things that organizations like mine really hate is; you're always talking about vendor lock-in, you don't want to be caught by a single vendor" (**Head of Technology Exploration, Telefonica**).*

This lock-in effect is a legacy of the layering that evolved since privatizations took place and the vendors took an increasingly important role in R&D. The rapid improvements in generic switching and processors and their proven, cost effective use in large data centers makes them an attractive alternative, provided that their performance is satisfactory.

> *"Thanks to Moore's Law with respect to processor speed, and power and storage costs coming down, being able to take advantage of that, which you can do much more in a data centre environment." (**Principal Member of the Technical Staff, Verizon**).*

If infrastructure begin to look more like data centers, with commodity hardware managing the networks in place of distributed, specialist hardware, the costs of operating such networks will tumble as they have done with Cloud platforms.

Although the focus appears to be on commissioning of new hardware, the rapid obsolescence of existing specialist hardware is another important issue:

> *"[It's] as much about decommissioning as commissioning savings. We [currently] simply leave equipment at customer sites, it's cheaper than collecting and disposing" (**Chief Network Services Architect, British Telecom**).*

With commodity-based Ethernet and virtualized functions the full-life cost of hardware drops significantly, and costs savings may be passed onto the content consumer.

2.2 Ensuring Infrastructure Flexibility

In addition to hardware cost considerations, there are long term broadcast service implications that the new approaches must allow. As well as shifting the primary technological core of network infrastructures, there must be a shift towards the use of software-based network functions, in place of hardware reliant functions.

> *"Since it is software only, the composition or decomposition of functions allows us to be more flexible in responding to the market place" (**Distinguished Network Architect, AT&T**).*

The importance of deployment speed is emphasized within BT, an important internal driver for change by providing a clear indication of just how much faster and more responsive they want services to be:

> *"One of the tag lines we've used was 'from 90 days to 90 seconds' that our lead time to deploy a box to wherever in the world the customer premises happens to be" (**Chief Data Networks Strategist, British Telecom**).*

In addition to this aspect of flexibility, we also see real benefits to both operators and customers of being able to delay purchasing decisions.

> *"There's a real option which is being able to defer a decision on what you deployed because the hardware is exactly as you say, generic, so you've not committed to the particular functionality at the time you deployed the hardware" (**Chief Data Networks Strategist, British Telecom**).*

We will have the ability to select and install "applications" (software-based network functions) at the time and place they are most needed, without having to try and predict what might be needed ahead of time. In addition functionality can be scaled up, scaled down or repurposed in the event of changing demand without the need to redeploy engineers into the field, or incur both the economic and environmental cost of hardware removal.

The long term flexibility goals stated include a desire to create a true software infrastructure for broadcast networks, both wide-area and in-facility. The separation of hardware and software supply chains eliminates the de-facto lock-in associated with proprietary hardware, and at the same time it creates a potentially much more capable and competitive software-based broadcast network. It will encourage new entrants and start-ups to enter the marketplace with innovative products tailored to the needs of broadcasters but with their roots firmly in the IT and data services industries.

> *"[SDN], is a disruptive technology, and it requires new switches and a new way of working, and there are issues around bringing all the different application interfaces, software and hardware vendors together, to have a completely functioning system that replaces the original network." (**Senior R&D Engineer at BBC Research & Development**)*

Therefore we anticipate a shift in the skillsets required to design, develop, operate broadcast networks away from highly skilled broadcast specialists to more broadly skilled personnel with background in software engineering and information technology disciplines. This will include a greater emphasis on automation and a shift to the development and operations (DevOps) model.

3. Media and Broadcast Network Requirements

Multiple use cases exist depending on the type and scale of media and broadcast application, each with a specific set of requirements and capabilities depending on the type of media network. We may summarize core requirements across most use cases:

- Aggregation of multiple flows and formats across studio infrastructure
- Broadcast industry native interface support
- High-bandwidth connections

Each broadcast or contribution flows have their own formats, underpinned with the use of Serial Digital Interfaces (SDI). There is Standard Definition (SD), High Definition (HD), and Ultra High Definition (UltraHD, also known as 4K). Each of these formats is typically based on a well-defined protocol based on published standards. HD-SDI can be multiple format streams, i.e., 1080p, 1080i or 720p. The format type specifies vertical and horizontal resolution, aspect ratio, pixel aspect ratio, scanning and frame rate of the content.

The increasing use of 4K as UltraHD translates into a considerable increase in bandwidth consumption. As the trend to continue with yet further growth in frame rates, color depth, and number and quality of sound channels, only compounds the need to provide scalable high-capacity bit-rate services.

Additional application requirements are outlined in the following sub-sections.

3.1 Content Capture and Encoding

In some situations SDI must be encoded to a broad spectrum of formats for live or production content. One of the primary considerations with respect to selecting a format is its intended use or delivery platform. Once content is captured it may be encoded and forwarded across the network via a router, production switcher, or directly to a production server. Typically, this decision is handled by a Media Manager. In some cases the higher resolution content may use multiple outputs at the camera and need to be recompiled and synchronized at the router, production switcher, and encoder.

3.2 Content Transport

In addition to encoding, media will be ingested directly from other sources as files or flows and as mentioned may require encoding to traverse IP infrastructure. There are a number of well-defined standards and protocols allow media to be encapsulated and transported across network infrastructure, including:

- SD-SDI – SMPTE 259M
- HD-SDI – SMPTE 292M
- ETSI – ASI- TR 101 891
- MPEG2 – ISO/IEC 13818
- MPEGTS – ISO/IEC 13818-1
- MPEG4 – ISO/IEC 14496
- MPEG4 H.264 – ISO/IEC 14496-10

- JPEG2000 – ISO/IEC 15444-12

3.3 Bandwidth, Compute & Storage

Studio environments typical contain nodes with HD-SDI interfaces and 10Gb/s network cards, allowing for receive, transmit, encode and decode services, with centralized management.

Multicast may be used to distribute UHD (4K) compressed video at 2160p 59.94fps, using H.264 encoding this would require between 800Mb/s to 1.2Gb/s per service.

Demands by content consumers for increased video resolution, frame rate, color depth & sound channels, all add to bandwidth consumption for services. As indicated by the British Broadcasting Corporation (BBC), contribution network uses are requesting a move to near lossless or uncompressed video streams, these equate to:

- HD 1080p 8bit 4:2:2 50fps uncompressed bit rate @ 3Gb/s
- 4K UHD 2160p 12bit 4:2:2 50fps uncompressed bit rate @ 10Gb/s
- 8K SHV 4320p 12bit 4:2:2 50fps uncompressed bit rate @ 48Gb/s

3.4 Studio Media IP Evolution

Our ultimate objective is to facilitate end-to-end IP media production. This would require a mass migration from dedicated synchronous interfaces to generic IP networks. The rationale for migration to an all IP network, running over a high-capacity optical infrastructure, is compelling:

- Leverage the flexibility and operational experience of IP networks
- Deliver video, audio and data from a variety of sources and formats over IP infrastructure, low latency, and minimal jitter
- Efficiently utilize network resources, resource sharing where applicable
- Elastic control of the network, setting up and tearing down occasional-use services, links for optimal cost-effectiveness

If the studio production is live or recorded, it may have a slightly varying set of requirements. Typically content encoding and format decisions have already been made. When media is delivered from the field as SDI, it arrives in the facility and is encoded to a file in the house format and bitrate. If it’s an IP stream it will be encoded in the field, streamed to the broadcast centre, and captured to a file as it’s received.

During production workflow, media files may need to be accessible to various production applications and processes and possibly need to move between storage locations. Normally the applications (hardware or software) for production workflow are dedicated and/or fixed, and may only be used part-time. If functions were entirely software based and could be efficiently deployed in a “just in time” manner and scaled accordingly, it would provide significant cost savings and flexibility. However, different layers of automation to manage these applications and processes, with the capability to handle the file movement would also be required.

3.4.1 Linear Contribution and Content Transport

Our initial use cases for the lab were based on a linear contribution service, a typical requirement for broadcast networks. This type of service has the following key requirements:

- Automation: request, setup, teardown of the end-to-end service

- Initial support for HD and 4K contributions, but capable of scaling up to 8K
- Integrate encoding functions, scale-out storage, durability, adaptive performance, self-healing capabilities
- Supports high frame rates and other developing formats that exceed client expectations and requirements

The media flows are expected to be IP-based and support both live, linear TV programs and transport of media content files for production.

Today's commercially available broadcast video contribution links are typically based on data connections via Ethernet or SDH, with variable data rates up to 200Mb/s compressed, or 3Gb/s uncompressed. We therefore designed our infrastructure to support anything from a few 100Mb/s to 10Gb/s, based on a control architecture capable of evolving beyond 100Gb/s.

4. Applied SDN and NFV for Converged Architecture

Current networks consist of switches and routers using traditional distributed control planes and a data plane technologies. Ensuring network efficiency is limited in such networks as intelligence is distributed across many switches or routers and often involves complex protocols and procedures. By contrast, in an SDN network, with or without OpenFlow, we tend to use a centralized control plane (or Controller). This entity is directly responsible for establishing the paths or flows directly, and the data planes perform simple packet matches, forwarding, replication or dropping actions.

A Controller, per domain (administrative or technology) discovers, organizes and layers multiple services across infrastructure. Programmable control facilitates network behavior to be implemented and modified quickly and cohesively: automation techniques may be used to set up end-to-end services, with flexibility beyond the initial deployment, and with the capability to modify paths and network function nodes to be modified (torn down, resized, relocated) at any time particularly in response to rapid changes in the operational environment. This includes revised network conditions, fluctuations in the resource location or availability, and in the event of partial or catastrophic failure.

The advent of NFV is used to leverage Information Technology (IT) virtualization techniques to migrate entire classes of network functions typically hosted on proprietary hardware onto virtual platforms based on general compute and storage servers. Each virtual function node is known as a Virtualized Network Function (VNF), which may run on a single or set of Virtual Machines (VMs), instead of having custom hardware appliances for the proposed network function.

Furthermore, this virtualization allows multiple isolated VNFs or unused resources to be allocated to other VNF-based applications during weekdays and business hours, facilitating overall IT capacity to be shared by all content delivery components, or even other network function appliances. Industry, via the European Telecommunications Standards Institute (ETSI), has defined a suitable architectural framework, and has also documented a number resiliency requirements and specific objectives for virtualized media infrastructures.

Utilizing the benefits of enabling technologies, i.e. SDN control principles and NFV-based infrastructure, we have the potential to fundamentally change the way we build, deploy and control broadcast services built on top of flexible optical networks allowing dynamic and elastic delivery and high-bandwidth broadcast and media resources.

5. British Telecom Media and Broadcast Laboratory

BT has built a research laboratory to explore the potential impact of SDN & NFV on networks required to carry high bandwidth broadcast video traffic. The lay-out is depicted in the figure below which shows our intentions to do research on the various aspects of building end-to-end video contribution networks. Video creation at HD and UHD rates produces multi-Gb/s SDI formats that require (optional) compression and conversion into Ethernet before progressing into the network. From here we have the options of using labelled or white-box switches, both effectively setting up high bandwidth Ethernet circuits across a core network. There is also an option to include IP routers in the network – used to handle compressed video flows with lower bandwidths.

Traditional Network Management System (NMS) platforms lack the flexibility to fully enable our test infrastructure so we needed to look towards the architecture and principles defined by the Software Defined Networking (SDN) architecture developed and ratified by the Open Networking Foundation (ONF). These core SDN architectural principles offer a variety of possibilities when looking to plan, control, and manage flexible network resources both centrally and dynamically. Solutions exist that encompass direct control of switching resources from a central orchestrator, distributed control through a set of controllers, or devolved control through a hybrid with an active control plane.

The advent of Network Functions Virtualization (NFV) has also provided the ability to deploy network functions on virtualized infrastructure hosted on commodity hardware, decoupling dedicated network function from proprietary hardware infrastructure. Consequently this allows network function to be instantiated from a common resource pool and to exploit performance predictability where dimensioning remains stable whatever the use of virtualized hardware resources. Emboldened with the suitable control and orchestration tools, these virtual and on-demand capabilities could have a significant impact on how broadcast infrastructure is managed.

The optical cloud comprises a combination of optical switches, amplifiers and fiber. The switches here are Reconfigurable Optical Add-Drop Multiplexers (ROADM) which have at their heart Wavelength Selective Switch (WSS) technology. These route wavelength channels from any input to any output fibre and can be switched in just a few seconds.

Sitting above the hardware are a range of controllers, able to control each of the network elements – for example there is a controller whose job is to interface to the optical cloud. These controllers provide inputs to an orchestrator which has now a centralised view of all the network resources. Applications can take advantage of this SDN-based network orchestration and we have demonstrated a Scheduler application that can request on-demand large bandwidth pipes set up at specific times and durations.

The figure below presents our initial view of this idealised architecture.

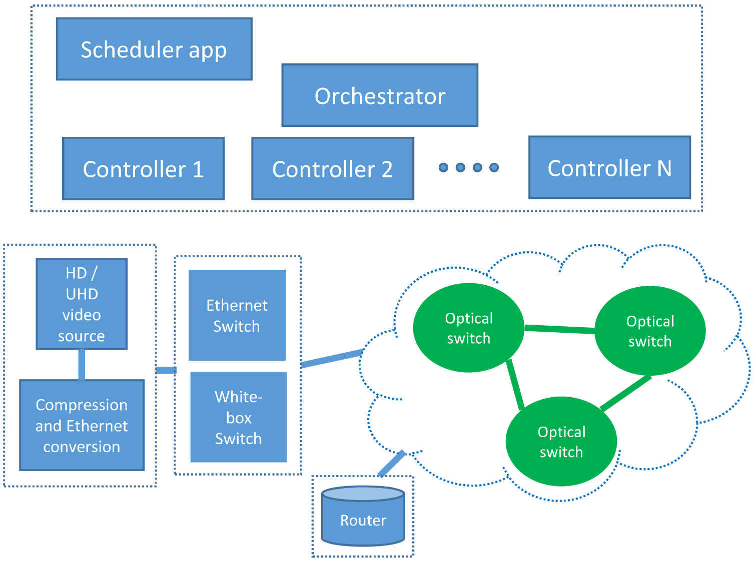


Figure 2: British Telecom Media & Broadcast Idealised View

One key purpose of the laboratory is to compare proprietary and more open methods to control networks like this. In the extreme case, assuming all the equipment provides OpenFlow means of control, open source software such as Open Daylight may be used to create complex behaviours, interlinking optical and electrical switches from multiple vendors.

The laboratory has had a great deal of use assessing the potential of the various SDN approaches available. It is absolutely essential to try out these concepts in a laboratory, as this is the only way to discover the potential issues involved when trying to do complex network coordination.

5.1 BT M&B Lab Architecture

Typically the purpose of a functional architecture is to decompose a problem space and separate distinct and discrete functions into capabilities so we could identify the components required and the functional interactions between components. We must consider the core requirements that are shared across contribution and distribution networks, as well as the specific capabilities of each environment.

It should be noted:

- An architecture is not a blue-print for implementation
- Each component is an abstract functional unit
- Functions can be realized as separate software blobs on different processors
- Depending on resiliency requirements, functions may be replicated and distributed, or centralized
- A protocol provides a realization of the interaction between two functional components

There have been a few useful attempts to document SDN and NFV network architecture, but very limited research has been published on said technologies for broadcast and media infrastructure.

Therefore:

- Our work has tried to present a blueprint for combining emerging technologies to solve commercial and technology requirements, we embrace SDN and NFV without becoming focused or obsessed with them
- We address a range of broadcast and media network operation and management scenarios
- We encompass (without changing) existing broadcast and media services
- We highlight available existing protocols and components that may be uses for solution development

Our architecture is designed and built around core SDN & NFV capabilities and their subsequent applicability to the broadcast contribution network and media distribution network. An idealized view of this model is presented below:

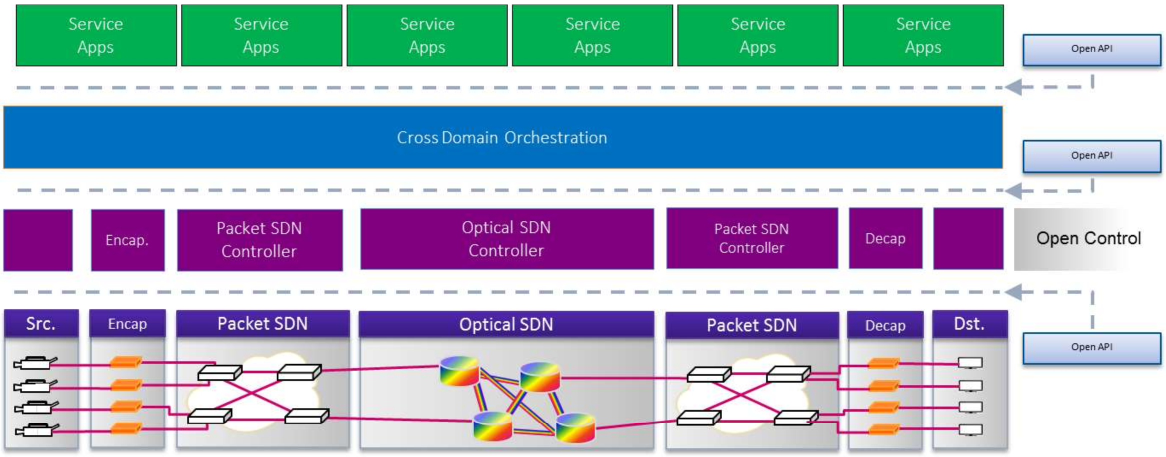


Figure 3: BT M&B Layered View

5.1.1 Design Considerations

Merchant silicon

A key principle for the lab network was to avoid complex IP switches and routers targeting small-volume, large feature sets, and high reliability. We identified general-purpose commodity off-the-shelf Ethernet platforms with merchant silicon switching ASICs.

Centralised Control

Control and management becomes substantially complex and expensive with distributed control planes. Existing routing and management protocols were not well-suited to our initial designs.

Reduce Network Complexity

Overall, our software architecture more closely resembles control in large-scale storage and compute platforms than traditional networking protocols. Network protocols typically use distributed soft state message exchange, emphasizing local autonomy. We were looking to use the distinguishing characteristics of distributed control planes via a centralized controller.

Optical Transport

The optical transport layer provides the high capacity underlay fabric. The flexible optical network concept is attracting a lot of attention from network infrastructure providers, with the purpose of offering their Infrastructure as a Service (IaaS) to variety of broadcast and contribution consumers.

In the future optical network virtualization technologies might allow the partitioning/aggregation of the network infrastructure into independent virtual resources, where each virtual resource has the same functionality as the physical resource, but it can be apportioned by the broadcast media user. Facilitating users to dynamically request, on a per need basis, a dedicated packet slice for each media interface when required.

Open Application Program Interfaces

An Open Application Program Interfaces (APIs) are important architectural components of our design goal. We need the capability to push or pull configuration or information directly to each layer of the network. This will facilitate applications being capable of interacting directly with the infrastructure itself.

5.2 Functional Components

A short description of each component, its function and the vendor or open source platform tested.

Applications

- Video Service Scheduler

Controller

The Controller is implemented strictly in software and is contained within its own Java Virtual Machine (JVM). As such, it can be deployed on any hardware and operating system platform that supports Java.

- Packet Controller (Open Daylight)

This Open Daylight project is a collaborative open source project hosted by The Linux Foundation. The goal of the project is to accelerate the adoption of SDN and create a solid foundation for NFV-based applications. The platform is an open source project with a modular, pluggable, and flexible SDN controller platform at its core.

Optical Controller

Optical Network Hypervisor is a multi-tenant capable application that creates and exposes abstract representations of the underlying transport network and exports that abstracted network to client SDN controllers. An abstracted network can be exposed as a single node or multiple nodes with abstract links.

From the perspective of the exposed SDN interface the Network Hypervisor acts as one or more (virtual) nodes.

Gateways

- Media Gateways

  - Physical solution
  - Virtual solution

The media gateway must be capable of encoding and decoding a variety of broadcast formats.

Optical Switching

- Optical ROADMs

5.3 Deployment Phases & Capabilities

5.3.1 Phase 1

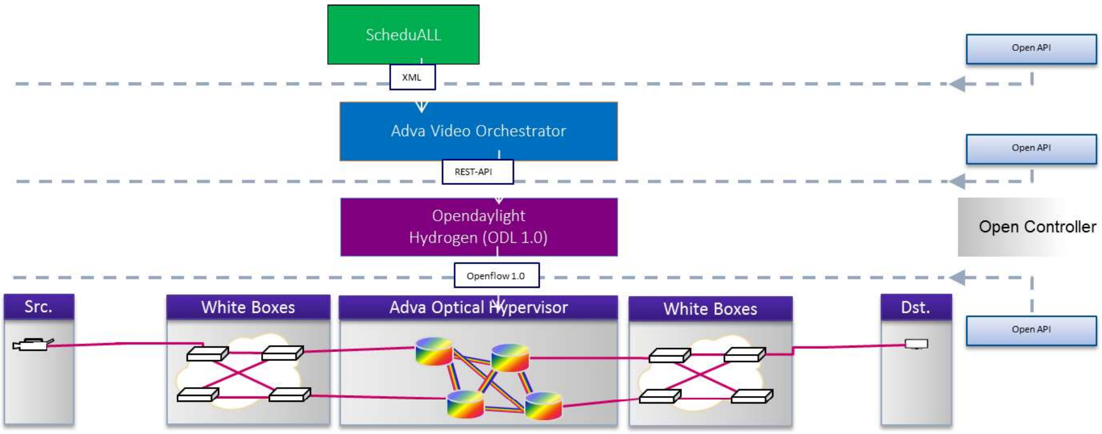


Figure 4: Phase 1 Architecture

Design and build out of the Phase 1 architecture started prior to 2013. Testing began in 2014 and by October 2014 we were able to demonstrate automated scheduling, setup and teardown of broadcast services across multi-layer (IP, over Open Flow, over optical)

The initial architecture used Open Daylight 1.0 (Hydrogen) and Open Flow 1.0 interacting with a limited number of whitebox switches.

Major issues were identified at this early stage of development, issues included:

Whitebox Software

Equipment was plagued with incompatibility problems, requiring numerous software upgrades and working around bugs.

Resource discovery and inventory management

The controller of nodes and elements in its domain needs to know about the devices, their capabilities, reachability, etc. Automated discovery of Open Flow switches was limited, and each switch would need to be configured with Controller location. Capability exchange and negotiation was also non-existent.

Limited Open Flow functions (using version 1.0)

We would have preferred to use Open Flow 1.3 but were limited to a version that was supported by the widest number of switches.

Hardware-based video encoding

General hardware-based video encoding provide cost and performance benefits but it also meant we needed to select specific sites to place the encoders and add new sites or moving locations meant the equipment also had to move.

Optical transport layer abstraction

Due to a limited API, we had minimal control automation between packet and service layers, to the optical transport domain. It then required manual intervention to setup or tear down new optical connections. Abstraction of the optical layer was nothing more representative than a switch.

5.3.2 Phase 2

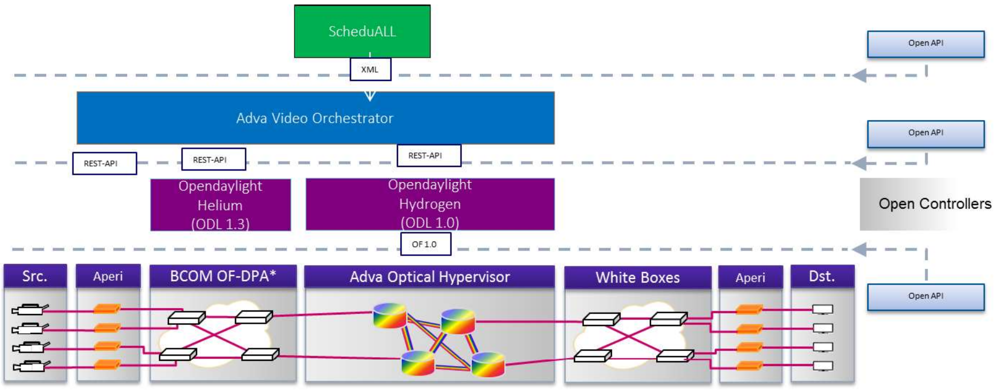


Figure 5: Phase 2 Architecture

Phase 2 saw a number of upgrades and enhancements to the network, these included:

Open Daylight Upgrade

Migration to the “Hydrogen” release of Open Daylight. Hydrogen Virtualization Edition for data centers includes all the components of Base plus functionality for creating and managing Virtual Tenant Networks (VTN) and virtual overlays, key goals for separating different types of broadcast and media content and users. The second release of Open Daylight also provided OpenFlow1.3 protocol library support, and Open vSwitch Database (OVSDB) configuration and management protocol support, a key requirement for commodity switching platforms.

‘Media Functions Virtualization’

We also added the product of another vendor: Aperi to our network. Aperi provided reprogrammable FPGA based cards capable of being dynamically transformed to perform different functions. Those included: JPEG 2K encode/decode, uncompressed to IP encapsulation, hitless switching and packet generation and analysis.

Consideration of Service and Network Resiliency

The testing program on the Phase 2 network underlined the need for hitless switching, again a key requirement for media and broadcast services. A large number (but not majority) of critical functional components could either be failed over/switched without interrupting existing services. However, the setup or teardown of services was impacted in the event of single failures of key components (either internal or external to the Controller). Therefore, resiliency continues to be an area of research and challenges for us.

Improvement of Maintenance and Stability of Whitebox Switches

A notably issue we saw was the time it took to load new firmware onto line interface cards. This could vary from a few seconds to several minutes.

5.3.3 Phase 3

A number of capability requirements have been identified as we move into the third phase, these include:

Optical Domain Flexibility

As our investigations and experiments continue we want to ensure the same flexibility that exists in the IP and Ethernet layer is available in the optical transport domain. This is non-trivial problem, if we pursue an open Controller architecture. Paths through an optical network are tricky as we consider non-linearity effects wavelength continuity, paths are often blocked and end-to-end optical connections be optimized in many different ways.

Increased Bandwidth

Bandwidth must continue to increase, but provide the flexibility requirement described previously. We have identified that Elastic Optical Networks (EON) may provide significant bandwidth flexibility by utilizing recent ITU-T flexi-grid (flexible bit rates to beyond 100Gb/s).

Virtual Network Function (VNF) Infrastructure Management (VIM)

Open Stack provides the tools required for managing application, compute and storage. In our lab this will equate to virtual media encoders, caching nodes and file storage. Initial testing has found that Open Stack does not currently meet important SDN & NFV requirements, such as distribution, networking, operational optimization, and data plane optimization. However, OpenStack is still under heavy development in many areas. As the platform matures, we anticipate that more stable and richer in functionality, allowing it to better meet SDN & NFV requirements.

Architecture, Interfaces (API's) and Models

The following figure utilizes the ETSI NFV Reference Architectural Framework, and demonstrates a proposed converged SDN and NFV candidate architecture for Phase 3 testing. It identifies the functional components and interfaces that were established for both SDN and NFV vendors to develop solutions and ensure interoperability:

1. Os-Ma: an interface to OSS and handles network service lifecycle management and other functions
2. Vn-Nf: represents the execution environment provided by the Vim to a VNF (e.g. a single VNF could have multiple VMs)

3. Nf-Vi: interface to the Vim and used for VM lifecycle management
4. Ve-Vnfm: interface between VNF and Vnfm and handles VNF set-up and tear-down
5. Vi-Ha: an interface between the virtualization layer (e.g. hypervisor for hardware compute servers) and hardware resources

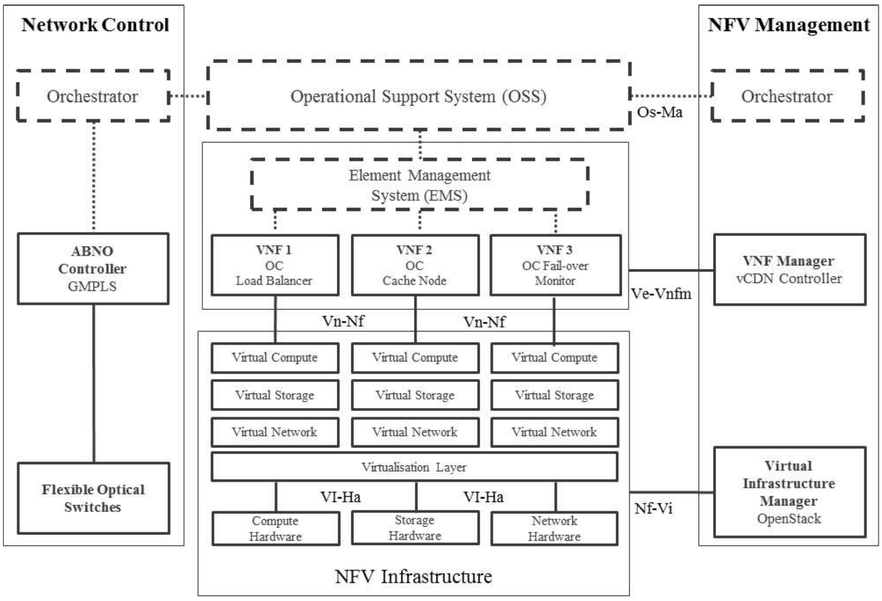


Figure 5: Candidate SDN & NFV Framework based on ETSI NFV ISG Model

5.4 Wider Challenges and Open Questions

5.4.1 Viability of OpenFlow for Optical Networks

We have found OpenFlow to be very efficient for our Ethernet layer but concerns remain for its optical technology viability. A new set of port properties add support for Optical ports was introduced in OpenFlow version 1.4, they include fields to configure and monitor the transmit and receive frequency of a laser, as well as its power. Those new properties can be used to configure and monitor either Ethernet optical ports or optical ports on circuit switches.

There is also motivation to provide additional optical transport extensions to future versions of OpenFlow: "Optical Transport Protocol Extensions".

5.4.2 Underlay Network Abstraction

Abstracted representation of each server (optical and Ethernet) layer and client layer (IP), is an important goal. We would like to leave each vendor to control their equipment and balance the decades of knowledge about how to manage complex optical parameters, engineering rules and non-linearity effects, whilst providing an open interface for a high-layer application to request a new service, resize an existing service or perform a network wide optimization.

Generating a well-defined and understood information model for multiple forwarding technologies remains an elusive goal. We recognize that different organizations are working toward a solution but we wonder if these models will be consistent with each other.

5.4.3 Role of Standards and Open Source

Our engagement and participation of Standards Development Organizations is limited. It is often a complex and costly affair. Open Source communities are much easier for us to engage with, we have immediate access to software platforms and an active and willing support community. Unfortunately, we also have to build interoperable networks so well defined interfaces, via formal standards, is sometimes a safer option over “de facto standards”.

The larger SDO’s should provide greater opportunities for their standard proposals to be implemented in Open Source and tested by a willing community of users, creating a feedback loop back into the SDO to improve the developing standard.

5.4.4 Integration of Whitebox Switching into Legacy OSS/BSS

Initial excitement for whitebox switching was motivated by a desire for significant capex reductions, thus forcing the consideration of SDN. In a large complex environment like ours, and especially with the interworking of our OSS/BSS layers, we have yet to see viable management platforms for very large number of whitebox switches that would also allow integration with existing OSS and BSS platforms.

6. Findings and Conclusions

Our efforts to design and build broadcast and contribution infrastructure based on the principles on SDN, NFV and related technologies are yielding exciting results. These benefits are manifesting as new service capabilities and flexibility, while reducing costs across multiple layers for the transport of media and broadcast services.

We are able to setup and tear down end-to-end connections, via a centralized controller, significantly faster and with less protocol complexity compared to existing IP/MPLS broadcast and contribution networks. Furthermore, using OpenFlow and commodity Ethernet switches, we have demonstrated rapid video path switching, and ‘clean’ switching by utilizing make-before-break mechanisms.

Emerging optical technologies are providing a compelling answer for exponential bandwidth consumption, but this must not come at any economic cost. Furthermore, current optical networks lack elasticity and operational complexity and costs increase as they scale. We have identified that Elastic Optical Networks (EON) and the flexi-grid (flexible bit rate) technology offers important benefits and capabilities, including wavelength slicing from 100Mb/s up to 200Gb/s, and beyond. Thus our Phase 3 testing will include components of the ITU-T and IETF Flexi-grid forwarding technology and Application-Based Network Operations (ABNO) controller framework and functional components.

Other challenges still remain, as highlighted in section 5.4 “Wider Challenges and Open Questions”. However, we are confident that by close cooperation with industry partners, Open Source communities, and Standards development organizations, solutions will be found.

Acknowledgements

We thank our hardware and software partners for their ongoing support, including: ScheduALL, Nevion, Aperi, ADVA Optical Networking, and Intel Labs.

Special thanks to Telefonica, Verizon, Orange and the BBC who were willing to share their thoughts and ideas with us.

Finally, we would like to acknowledge and thank our University partners and the EPSRC-funded project “Towards Ultimate Convergence of All Networks” (TOUCAN).

References

1. Software Defined Networking Architecture Overview, ONF TR-504, 2014.
2. Network Functions Virtualisation – White Paper #3, ETSI, 2014.
3. Planning and Designing The IP Broadcast Facility, Gary Olson, 2014
4. Network Functions Virtualization (NFV); Architectural Framework, ETSI GS NFV 002, 2014.
5. B. Molina Moreno, C. Palau Salvador, M. Esteve Domingo, I. Alonso Pe˜na, and V. Ruiz Extremera. On Content Delivery Network Implementation. Computer Communications, 29(12):2396–2412, 2006.
6. Network Functions Virtualization (NFV); Use Cases, ETSI GS NFV 001, 2013.
7. L. Velasco, D. King, O. Gerstel, R. Casellas, A. Castro, and V. López, In-Operation Network Planning, IEEE Communications Magazine, 2014.
8. Aguado, V. López J. Marhuenda, Ó. González de Dios and J. P. Fernández-Palacios: ABNO: a feasible SDN approach for multi-vendor IP and optical networks , in Journal of Optical Communications and Networking, February 2015.
9. R. Muñoz, R. Vilalta, R. Casellas, R. Martínez, F. Francois, M. Channegowda, A. Hammad, S. Peng, R. Nejabati, D. Simeonidou, N. Yoshikane, T. Tsuritani, V. López and Achim Autenrieth: Experimental Assessment of ABNO-based Network Orchestration of end-to-end Multi-layer (OPS/OCS) Provisioning across SDN/OpenFlow and GMPLS/PCE Control Domains, in European Conference on Optical Communication (ECOC), Sep 2014.
10. King, D., and Farrel, A., “A PCE-based Architecture for Application-based Network Operations”, IETF RFC 7491, 2015.
11. Open Networking Foundation, Optical Transport Protocol Extensions, Version1.0 March 15, 2015 ONF TS-022.
12. Ó. Gonzalez de Dios, R. Casellas, "Framework and Requirements for GMPLS based control of Flexigrid DWDM networks", IETF Internet Draft draft-ietf-ccamp-flexi-grid-fwk, 2015.

Transport Northbound Interface: The Need for Specification and Standards Coordination

D. King, C. Rotsos, University of Lancaster
I. Busi, F. Zhang, Huawei Technologies
N. Georgalas, British Telecom

***Abstract*—Next generation optical transport networks have high benchmarks for flexibility, reliability, and operational simplicity. These requirements underline a common, technology-independent orchestration paradigm that can be extended to represent and configure specific optical technology attributes. Although, orchestration is an ongoing aspect of the current optical transport network evolution, the meaning and scope of orchestration is often only implied, and various Specification and Standards communities cannot always agree the requirements and objectives.**

This paper describes the high-level requirements facing optical transport networks to provide well-defined Transport Northbound Interface (T-NBI) for optical resource programmability, control, and management automation. It explores the overall functionality that must be provided, whether encompassed in a single large-scale orchestration wrapper or partitioned into several sub-functions, of which only one component is designated as a transport orchestrator. It highlights the early efforts for optical transport resource modeling across Specification and Standardisation organisations.

The paper will report on recent Internet Engineering Task Force (IETF) Transport NBI Team Design Team efforts to collaborate across Standards Development Organisations (SDOs) to unify transport interface requirements and objectives. Finally, the paper will highlight use cases and applicability examples, and outline research gaps and challenges, opportunities for researchers, and areas for further collaboration between academia and industry.

***Index Terms*— Optical Modeling, Transport Northbound Interface (T-NBI), Transport Application Programming Interface (T-API).**

1 INTRODUCTION

TRANSPORT Operator (Operator) infrastructure is comprised of multiple technologies across network layers (traffic engineered optical and packet). Typically, these resources are separated into multiple transport domains, each using different network technologies, control interfaces and implementing forwarding policy with diverse goals.

Management, configuration, and troubleshooting processes rely extensively on human intervention, using Element Management Systems (EMS) and Network Management Systems (NMS) to translate high-level connectivity goals into individual device configurations, while service deployment is designed using whiteboards by the network planners [1]. Correspondingly, transport service delivery times for new connections may take many months, with significant portions of this time spent in the design and configuration phase of the deployment life-cycle.

The inflexibility, and limited automation of Transport Networks, led to the development of new control and management architectures and protocols. We often to refer to this technology as Transport Software Defined Networking (T-SDN): logically centralized control, separation of control and forwarding, open Application Programming Interface (API), and automation.

Existing optical transport networks often have separation of data plane and control elements; therefore, these are not new concepts, however establishing an open and well defined method for exposing transport capability via a Transport Northbound Interface (T-NBI), is now critical.

Potential success of Transport SDN in commercial environments is largely dependent on the success in specifying, documenting and standardising open transport interfaces between the Transport Orchestrator (T-O), Transport Controller (T-C) (Northbound Interface – NBI) and between TCs (East-West Interface).

A common open interface to each boundary is pre-requisite for network operators to control multi-vendor and multi-domain networks also enable service provisioning coordination/automation. This must be achieved by using standardised models, used together with an appropriate messaging protocol (interface).

Several popular optical and transport SDN architectures and interfaces are being developed, including:

1. Generic functional architecture of transport networks [1], developed by the ITU Telecommunication Standardization Sector (ITU-T);
2. Transport-Application Programming Interface (T-API)

The authors would also like to thank all members of the IETF Transport NBI Design Team involved in the definition of use cases, gap analysis and guidelines for using the IETF YANG models at the Northbound Interface (NBI) of a Transport SDN Controller.

Furthermore, we would like to thank our colleagues at the ONF, MEF and ITU-T, and IETF for their ongoing support and cooperation of the transport NBI.

The corresponding author (Daniel King) is a Senior Research at the University of Lancaster – d.king@lancaster.ac.uk.

Requirements [2] and Architecture [3], developed by the Open Networking Foundation (ONF);

3. Transport Northbound Interface Use Cases [3], Abstraction and Control of Traffic-Engineered Networks (ACTN) Framework [4], Traffic Engineering (TE) Topology [5] and TE Tunnel [6] YANG models defined by the Internet Engineering Task Force (IETF).

This document highlights the key components of control, interaction and naming of transport SDN functions, important use cases and requirements, and the type and scope of information that must be exchanged over the key interfaces.

2 TRANSPORT SDN

Transport network domains, including Optical Transport Network (OTN) and Wavelength Division Multiplexing (WDM) networks, are typically deployed based on a single vendor or technology platforms. They are often managed using proprietary interfaces to dedicated Element Management Systems (EMS), Network Management Systems (NMS) and increasingly Software Defined Network (SDN) controllers.

A well-defined open interface to each domain management system or controller is required for network operators to facilitate control automation and orchestrate end-to-end services across multi-domain networks. These functions may be enabled using standardized data models (e.g., YANG [7]), and appropriate messaging protocol (e.g., NETCONF (8)) or RESTCONF [9]) and encoding mechanisms.

2.1 Transport Service Perspectives

The following examples provide different use case perspectives for commercial transport SDN deployments.

1. **End-to-End Service Management**: Automated service creation covering Layer-0 to Layer-3.
2. **Elastic Bandwidth Provisioning**: Creation of elastic services with automatic or "on demand" changes in bandwidth.
3. **Dynamic Datacenter Interconnections**: Automatic load dependent fast service creation.
4. **Transport as a Service (TaaS)**: Fully automate service requests including network planning and node configuration.
5. **Multi-layer Network Operation**: Multilayer optimized Layer-0 to Layer-3 networking with automatic setup and teardown.
6. **Vendor Agnostic Transport Networking**: Standardised transport SDN control interfaces for automated integration and deployment of services across multi-vendor equipment.

2.2 Transport SDN Architecture

The architecture of SDN is specified in the ONF SDN architecture document [3], which identifies core principles of SDN and applies them to transport networks.

The ACTN Framework [4] describes a control hierarchy and interfaces that would enable deployment of multi-domain Transport SDN networks.

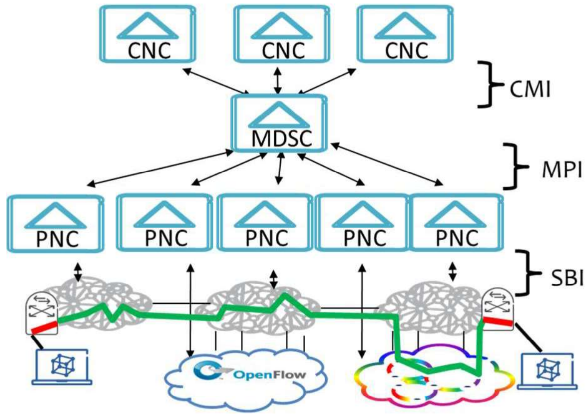


Fig. 1. IETF ACTN Control Hierarchy

The T-API Requirements [2] describes a functional architecture which has been used for the development of T-API requirements [2] and ongoing development of open source YANG modules.

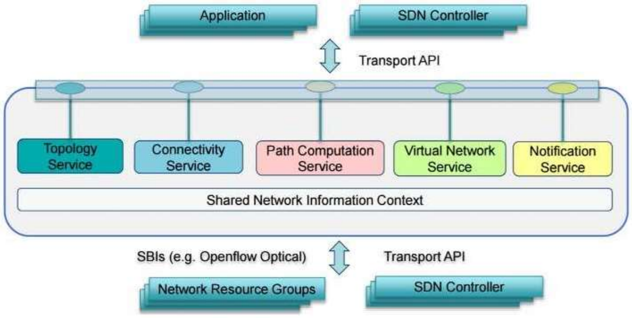


Fig. 2. ONF Transport API Functional Architecture

The underlying principles of these two reference architectures are very similar, but differences do exist.

An important design goal for application of these SDN principles to transport networks, is to be based SDN for transport on standardized and open interfaces at the northbound interface of the Transport SDN controller, to overcome the existing inter-operability limitations created by the lack of integration and interoperability of transport network devices.

Essentially, there is a clear need for a well-defined transport NBI and corresponding resource models. Combined, they are crucial for transport service orchestration, since they enable control and monitoring of service connectivity and network resource utilization and definition of custom fault management processes.

There are different opinions about whether this work would lead to interoperable and open resource models for the SBI; nevertheless, this work is complementary to the NBI definition, which would still be needed and it will also enable the integration of current deployments as well as a smooth migration of the transport network toward an open SBI paradigm, if it ever materializes.

2.3 *Transport Service Orchestration*

Orchestration is a hot topic of current industry conversation dealing with network evolution. However, formal definitions do not exist, it remains an area where the meaning and scope of orchestration is often only implied.

Current understandings of orchestration include "the idea of automatically selecting resources to satisfy client demands" [10], which also defines orchestration as "The ongoing selection and use of resources by a provider to satisfy client demands according to optimization criteria." This definition is intended to encompass all the necessary aspects of a solution, while not compelling any subdivision of functionality, e.g., into intent or policy or network analytics (telemetry) which may be discussed separately.

Prior to SDN, transport devices supported many of Southbound Interfaces (SBI) protocols like Path Computation Element Protocol (PCEP), GMPLS, TL1, SNMP, CLI, XML, et al, which had been standardized but multi-vendor interoperable. With the advent of SDN and the use of centralized controllers to interface with transport devices, new transport devices are supporting also new protocols the SBI like NETCONF, OpenFlow, et al, making the southbound even more fragmented and still not multi-vendor interoperable. However, the application of SDN allows domain controllers to abstract the fragmented southbound view for its northbound clients by normalizing the NBI across various technologies, protocols, and vendors.

NBIs Would allow the transport domain controller to communicate with the orchestrator via the normalized NBI to automate and programmed end-to-end transport resources, leveraging the transport infrastructure in an optimized way, across single or multi-domain technologies, and multiple SBIs.

2.4 *Transport Northbound Interface*

Firstly, Northbound interfaces (NBIs) can be organized in two broad categories.

The first category contains low-level information modeling NBIs. The primary role of an information model is to converge state representation of data plane devices and abstract the heterogeneity of forwarding technology. Network information models have been developed before the introduction of the SDN paradigm by multiple formal and information SDOs, including the IETF and ITU-T.

Relevant to the SDN paradigm is the ONF information modeling working group (WG), which develops the Common Information Model (Core Model) specifications for a variety of interfaces, and not only the Transport NBI.

The Core Model is hierarchical and includes a central model, which provides a basic abstraction for data plane forwarding elements, and a technology forwarding and an application specific model, which evolve the core model abstraction. Core Model specifications exploit object inheritance and allow control applications to acquire abstract network connectivity information and, in parallel, access technology-specific attributes of network elements.

The second NBI category contains high-level and innovative control abstractions of the service request. These interfaces are typically implemented as SDN management applications, use the information model to implement their control logic and are consumed by external entities, like the OSS, the service orchestrator and other control applications.

Effectively, both interface types manifest themselves between the functional interfaces between the Network and Service Orchestrator components.

Any NBI will require resource models these are being developed in formal and informal SDOs, including: IETF, ONF and MEF; which can be used on the interfaces of a domain controller and an orchestrator. Each domain controller and orchestrator can use models developed by different SDOs. Therefore, it is important to ensure that all models support deployment use cases and related functionalities to allow a seamless translation and mediation between systems using different models.

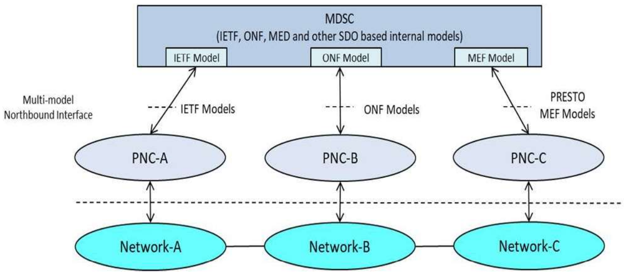


Fig. 3. IETF ACTN Applied YANG Models from multiple SDOs

2.5 *Defining the Transport Northbound Interface*

A transport network is a server-layer network designed to provide connectivity services, or more advanced services like Virtual Private Networks (VPN) for a client-layer network to carry the client traffic opaquely across the server-layer network resources. It acts as a pipe provider for upper-layer networks, such as IP network and mobile networks.

Transport networks, such as Synchronous Optical Networking (SONET) and Synchronous Digital Hierarchy (SDH), Optical Transport Network (OTN), Wavelength Division Multiplexing (WDM), and flexi-grid networks, are often built using equipment from a single vendor and are managed using private interfaces to dedicated Element Management Systems (EMS) and Network Management Systems (NMS). All transport networks have high benchmarks for reliability and operational simplicity. This suggests a common, technology-independent management and control paradigm that is extended to represent and configure specific technology attributes.

The need for operators to manage multi-vendor and multi-domain transport networks (where each domain is an island of equipment from a single supplier) has been further stressed by the expansion in network size. At the same time, applications such as data center interconnection require larger and more dynamic connectivity matrices. Therefore, transport networks face new challenges going beyond automatic provisioning of tunnel setup enabled by GMPLS (Generalized Multi-Protocol Label Switching) protocols to achieve automatic service provisioning, as well as address opportunities enabled by partitioning the network through the process of resource slicing. With lower operational expenditure (OPEX) and capital expenditure (CAPEX) as the usual objectives, open interfaces to transport networks to meet these requirements. Again, the concept of SDN mentioned earlier leverages these ideas.

The YANG modeling language is the data modeling language of choice within the IETF and has been adopted by several industry-wide open management and control initiatives. YANG may be used to model both configuration and operational states; it is vendor-neutral and supports extensible APIs for control and management of elements.

There are several scenarios where an open interface to access transport network resources would be useful. For the data centre operator, assuming the objective is to trigger the transport network to provide connectivity on demand, the following capabilities, would typically be required for any "open" interface between multiple controllers:

- Acquisition of the topology, be it physical or logical, of the transport infrastructure resources;
- The ability to obtain information about a set of access points of the transport network facing the client side, including information such as access point identifiers, capabilities, location, and environment types (Data Centers, Storage, et al.);
- The capability to send a request for a service using the access point information, as well as the ability to retrieve a list of service requests and status: source nodes, destination nodes, and current bandwidth and service attributes;
- Telemetry and monitoring of network performance information for real-time monitoring and optimization.

Each of these capabilities will require management and control via open interfaces for multi-domain networks with homogeneous technologies (such as OTN), but it can be extended further to multi-domain networks with heterogeneous technologies with higher complexity.

2.6 Core Requirements for the Transport Northbound Interface

2.6.1 Generic Requirements

User Intent: Transport models should maintain separation between high-level user intent and the operational state of the network. For e.g., maintain separation between user service request, including all constraints, and the actual service and connection state in the network transport network.

State Management: Network and service objects should support the following states: administrative state, operational state, and lifecycle state. Administrative state and operational states are well understood. Lifecycle state is defined in the ONF to model the following entity lifecycle states: planned state, potential state, installed state, in conflict state, and pending removal state.

Identifiers: Network and service objects and would include a unique entity ID provided by the controller. The identifier would be chosen such that the same entity in a real network topology will always be identified through the same ID, even if the model is instantiated in separate data stores. Controllers may choose to capture semantics in the identifier, for example to indicate the type of entity and/or the type of the parent identity.

2.6.2 Topology Requirements

The model should support the following topological link and node definitions:

- Link Requirements
    - Abstract Links
    - Compound Link which are internally aggregated lower level links
    - Access Links which connect the router port to the client port of the transport system
- Node Requirements
    - Physical Node
    - Abstract Node
    - Chassis / Forwarding Domain

The Link should support various link related attributes including cost, latency, capacity, risk characteristics (including Shared Risk Link Groups - SRLGs). The model should provide clear association between Link and its topology (including virtual topology), nodes and termination points.

In cases of multi-layer networks, the model should be capable to provide information about the adaptation capability between layers within a network element. The model should also provide association between the Link and any underlay circuit or service supporting the Link.

2.6.3 Telemetry Requirements

Topology service clients (which in the Transport-SDN context could be various: applications, orchestrators, controllers, big data collectors, analytics processors, network planners, etc.) require accurate real time network state information (this is known as network telemetry).

Telemetry information will be instrumental for maintaining network efficiency and optimal control under failure conditions. Network telemetry streams would provide resource failure prediction across network resources and provide knowledge to route the provided transport connectivity services away from predicted failure areas; identify and predict points of congestion and eliminate and/or mitigate the congestion by deploying extra network capacity in a timely manner. Clearly network telemetry is a valuable source of information useful for network planning, troubleshooting and resource optimization, and will require suitable models, such as "YANG models for ACTN TE Performance Monitoring Telemetry and Network Autonomics" [11].

3 TRANSPORT SERVICES

Transport networks are generally designed to deal with "connections" or "services", which are entities that encompass multiple related optical forwarding technologies.

The transport orchestrator needs to be capable to request service connectivity from the transport controller to support application and/or IP routers connectivity. The type of services could depend of the type of physical links (e.g., OTN link, WSON link, ETH link or SDH link) between the routers and transport network.

4 Applicability of YANG to Transport Northbound Interface

The transport NBI data models will required for representation of objects that can be configured or monitored within the transport system. Within the IETF, YANG [10] is the language of choice for documenting data models, and YANG models have been produced to allow configuration or modelling of a variety of network devices, protocol instances, and network services. YANG data models have been classified in [12] and for services in [13].

5 Conclusion

A variety of industry challenges remain for the development of standardised transport NBI. Emerging protocol and model solutions, as discussed in this paper, are immature and will require further investigation and development before they can be operationalised and used by operators.

The enabling SDOs for transport SDN need to work cooperatively, coordinated activities should include:

- Continued development of use cases and gap analysis [14], to identify a set of technology use cases and providing a gap analysis against existing transport models;
- Identify missing models: requirements for new models or where possible, augmentation of existing models;
- Providing guidelines, in terms of how all the related models, even when developed by different SDOs, may be used in a step-wise manner, these should be applied to network provider agreed transport network use cases;
- Finally, further research and investigation for network provider domain security and policy application and control, especially considering the inter-functional automation, should also be pursued.

Acknowledgment

The authors would also like to thank all members of the IETF Transport NBI Design Team involved in the definition of use cases, gap analysis and guidelines for using the IETF YANG models at the Northbound Interface (NBI) of a Transport SDN Controller.

Furthermore, we would like to thank our colleagues at the ONF, MEF and ITU-T for the ongoing support and cooperation of the transport NBI.

The authors are also grateful to the UK Engineering and Physical Sciences Research Council (EPSRC) for funding the TOUCAN project (EP/L020009/1), which supported much of the work presented in this paper.

References

[1] International Telecommunications Union, "Generic functional architecture of transport networks", ITU-T Recommendation G.805, March 2000.

[2] ONF Technical Recommendation TR-527, "Functional Requirements for Transport API", June 2016.

[3] ONF Technical Recommendation TR-522 “SDN Architecture for Transport Networks”, March 2016.

[4] D. Ceccarelli, Y. Lee, L. Fang, D. Lopez, S. Belotti, D. Dhody, D. King, “Framework for Abstraction and Control of Transport Networks", draft-ietf-teas-actn-framework, IETF Internet Draft, February 2017.

[5] Liu, X., Bryskin, I., Beeram, V., Saad, T., Shah, H., and O. Dios, "YANG Data Model for TE Topologies", draft-ietf-teas-yang-te-topo, IETF Internet Draft, October 2016.

[6] Saad, T., Gandhi, R., Liu, X., Beeram, V., Shah, H., Bryskin, I., Chen, X., Jones, R., and B. Wen, "A YANG Data Model for Traffic Engineering Tunnels and Interfaces", draft-ietf-teas-yang-te, IETF Internet Draft, October 2016.

[7] Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language", IETF RFC 7950, August 2016.

[8] Bjorklund, M., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", IETF RFC 6020, October 2010.

[9] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", IETF RFC 8040, January 2017.

[10] ONF Technical Recommendation TR-540 “Functions of Orchestration”, January 2017.

[11] Y. Lee, D. Dhody, S. Karunanithi, R. Vilalta, D. King, D. Ceccarelli, “YANG models for ACTN TE Performance Monitoring Telemetry and Network Autonomics”, draft-lee-teas-actn-pm-telemetry-autonomics, IETF Internet Draft, March 2017.

[12] Bogdanovic, D., Claise, B., and C. Moberg, "YANG Module Classification", draft-ietf-netmod-yang-model-classification, IETF Internet Draft, October 2016.

[13] Zhang, X. and J. Ryoo, "A Service YANG Model for Connection-oriented Transport Networks", draft-zhang-teas-transport-service-model, IETF Internet Draft, July 2016.

[14] I. Busi & D. King, et al, “Transport Northbound Interface Use Cases”, draft-tnbidt-ccamp-transport-nbi-use-case, IETF Internet Draft, March 2017.