Chapter 10
Application-Based Network Operations (ABNO)

Daniel King, Víctor Lopez, Oscar Gonzalez de Dios, Ramon Casellas, Nektarios Georgalas, and Adrian Farrel

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

D. King (✉)
Lancaster University, Lancaster, UK
e-mail: daniel@olddog.co.uk

V. Lopez • O.G. de Dios
Telefonica, Madrid, Spain

R. Casellas
CTTC, Castelldefels, Spain

N. Georgalas
British Telecom, London, UK

A. Farrel
Old Dog Consulting, Llangollen, UK

V. López, L. Velasco (eds.), *Elastic Optical Networks*, Optical Networks,
DOI 10.1007/978-3-319-30174-7_10

SDN	Software Defined Networking
TE	Traffic Engineering
TED	TE Database

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 multilayer 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

10.2 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.

10.2.1 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.).

10.2.2 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 multilayer 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 to 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 and 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 and discusses key control plane principles and architectures. It introduces the Application-Based Network Operations (ABNO) Framework [1], and how this framework and functional components are combined for Adaptive Network Manager (ANM) [2], used to address the requirements for operating Elastic Optical Networks (EONs) [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.

10.3 Network Control

A central principle of SDN is the separation of network forwarding and control planes (Fig. 10.1). By separating these functions, a set of specific advantages in terms of centralized or distributed programmatic control might arise. 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

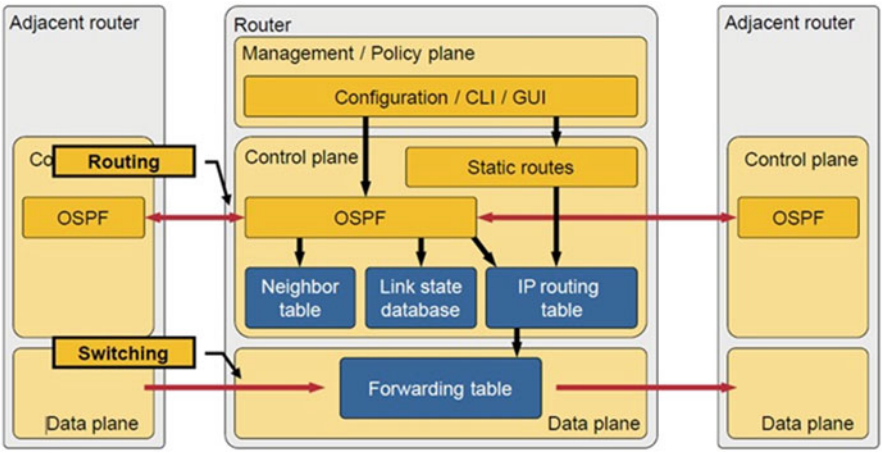

Fig. 10.1 Management, control, and forwarding example

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.

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 are exchanged, later results in data plane paths being programmed to realize those paths; however, paths are often suboptimal 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.

10.3.1 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. 10.2). 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 the IP forwarding elements.

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

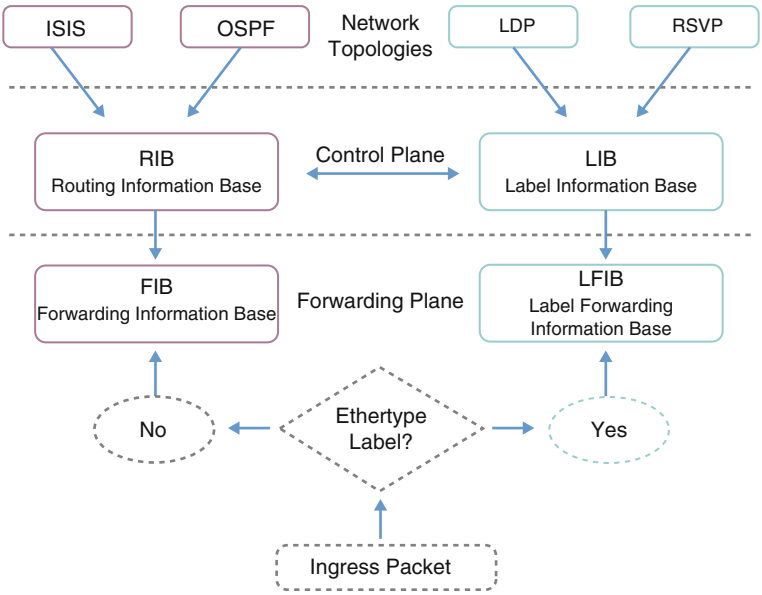

Fig. 10.2 Relationship of control and forwarding plane

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 and 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 setup/teardown
- Connection protection/restoration
- Path Computation and Traffic engineering

10.3.2 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, and 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 rerouting 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 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 methods include summarization, filtering, recursion, and segregation.

10.3.3 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, rerouting, 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 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 multipoint) 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.

10.4 Distributed and Centralized Control Planes

10.4.1 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.

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) [5–7] 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 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.

As seen in Fig. 10.3, 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, e.g., 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

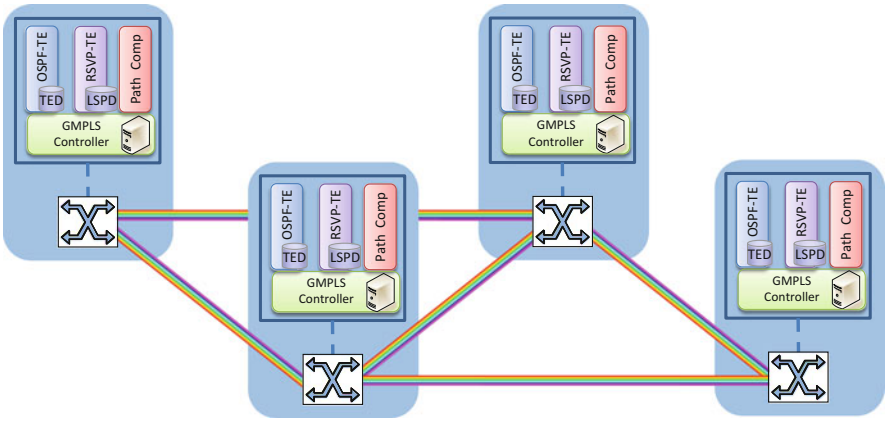

Fig. 10.3 Example of GMPLS-controlled optical network

mechanisms and procedures to address scaling of the control plane state (memory and CPU) in the network, both for physical and logical design. These 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) [9]. 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 [10] and Generalized Switch Management Protocol (GSMP) [11] 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 the 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 are exchanged that later result in data plane paths being programmed to realize those paths; however, paths are often suboptimal 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 as SDN controller, and this model 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 is less subject to latencies and out-of-date information due to the need for 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

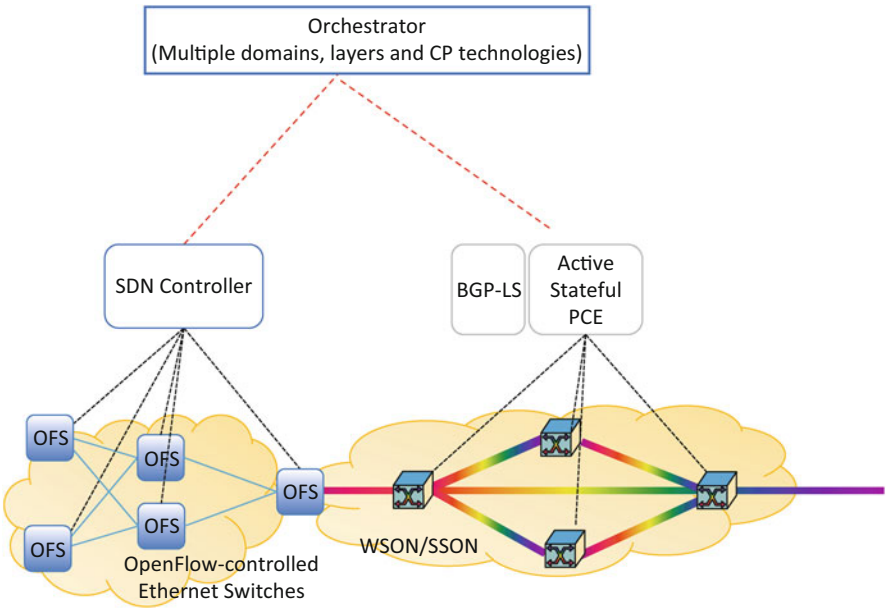

Fig. 10.4 The use of an orchestrator for the over-arching control of heterogeneous control technologies

be distributed among several children PCE in Hierarchical PCE (H-PCE) architecture [12] within a multi-domain scenario.

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. 10.4). 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 mention 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.

10.5 Framework for Application-Based Network Operations

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 actual 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.

Figure 10.5 presents an example of network architecture using ABNO.

10.5.1 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.

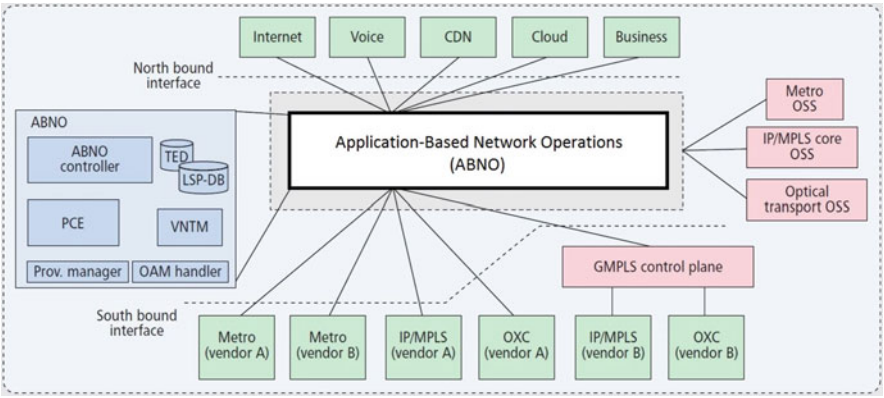

Fig. 10.5 ABNO architecture example

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) [13] 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 VPN. 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 Controlleris 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) [14], and the Interface to the Routing System (I2RS) Client will also be a PEP as noted in [15].

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

The Path Computation Element (PCE) is a functional component that services request 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 (VNTM), 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 [16]. 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 multilayer 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 [12], 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) is 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

A Virtual Network Topology (VNT) is defined as a set of one or more LSPs in one or more lower-layer networks that provide 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 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 nodes.

10.5.2 South Bound Interfaces

The network devices maybe configured or programmed directly from the NMS/OSS. Many protocols already exist to perform these functions, including the following:

- SNMP [17]
- The Network Configuration Protocol (NETCONF) [18, 19]
- RESTCONF [20]
- ForCES [10]
- OpenFlow [9]
- PCEP [21]

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.

10.6 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 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 also a wider scope, a multilayer IP/MPLS over optical Network.

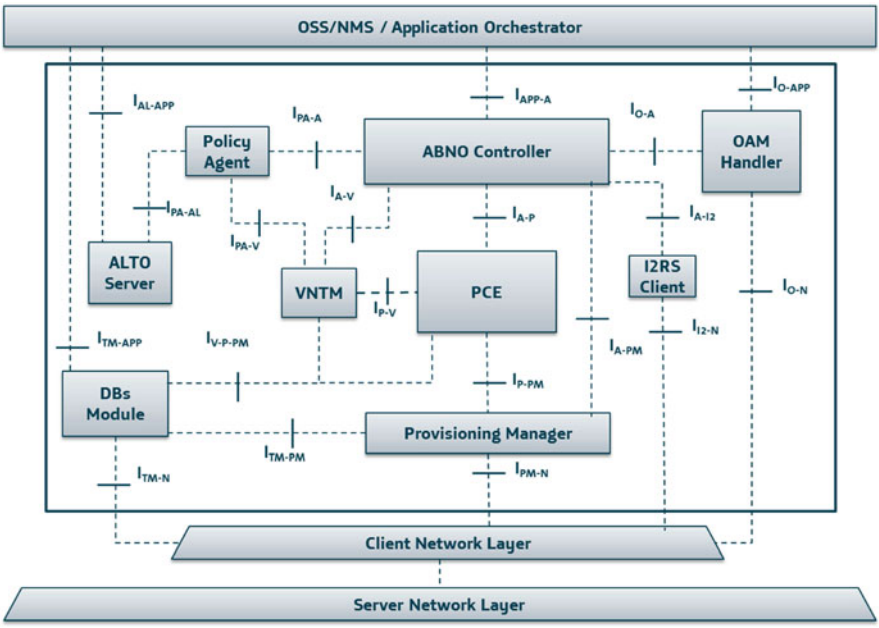

Fig. 10.6 Adaptive network manager functional components and interfaces

10.6.1 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. 10.6). 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.

- *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 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.

10.7 Adaptive Network Manager Use-Cases

10.7.1 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 large-scale 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 optimization. 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.

10.8 Next Steps for ABNO-Based Control and 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 South Bound Interface (SBI) protocol. These SBIs exist 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 of 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.

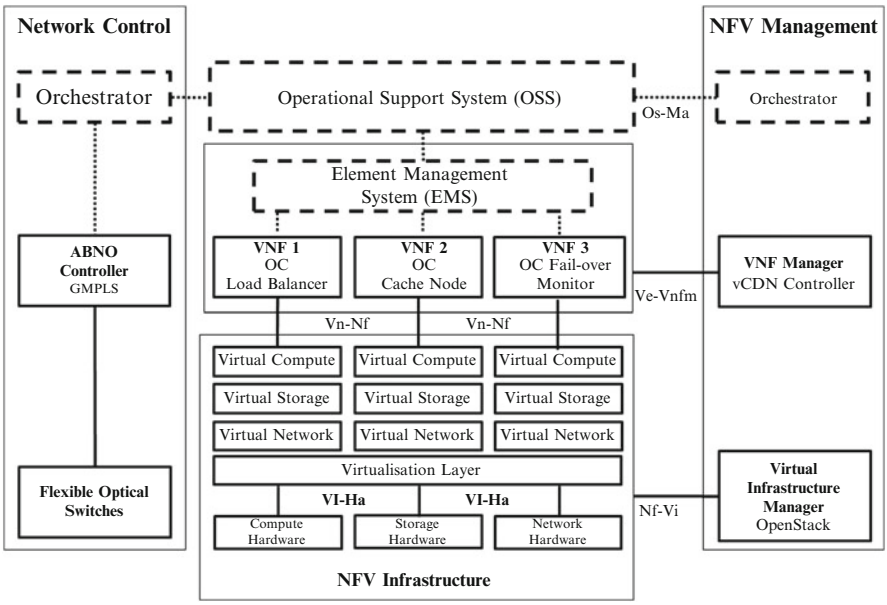

Fig. 10.7 Candidate SDN & NFV framework based on ETSI NFV ISG model

10.8.1 Control and Orchestration of Virtual Content Distribution Network

Virtualization 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 Virtual Content Distribution Network (vCDN) is reconfigurable bandwidth as content moved from HD content at 1080p to 4k streams 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.

Figure 10.7 describes how an ABO-enabled network controller would integrate with an NFV-based CDN.

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 EONs, simply not previously available in optical transport domain.

References

1. D. King, A. Farrel, *A PCE-Based Architecture for Application-Based Network Operations*, IETF Internet RFC 7491, (2015, March)
2. R. Muñoz, et al., IDEALIST control and service management solutions for dynamic and adaptive flexi-grid DWDM networks, in *Proceedings of Future Network and Mobile Summit*, Lisbon, 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, (2015, December)
4. N. Sprecher et al., *An Overview of Operations, Administration, and Maintenance (OAM) Tools*, RFC 7276, (2014, June)
5. ITU-T Recommendation G.8080/Y.1304, *Architecture for the Automatically Switched Optical Network (ASON) 02/2012*
6. ITU-T Recommendation G.872, *Architecture of Optical Transport Networks 10/2012*
7. ITU-T Recommendation G.709/Y.1331, *Interface for the Optical Transport Network* (OTN) 02/2012
8. E. Mannie (ed.), *Generalized Multi-Protocol Label Switching (GMPLS) Architecture*, IETF RFC 3945, (2004, October)
9. Open Networking Foundation, *OpenFlow Switch Specification Version 1.4.0 (Wire Protocol 0x05)*, (2013, October)
10. J. Halpern, J. Hadi Salim, *Forwarding and Control Element Separation (ForCES) Forwarding Element Model*, RFC 5812, (2010, March)
11. A. Doria, K. Sundell, F. Hellstrand, T. Worster, *General Switch Management Protocol (GSMP) V3*, RFC 3292, (2002, June)
12. D. King, A. Farrel (eds.), *The Application of the Path Computation Element Architecture to the Determination of a Sequence of Domains in MPLS and GMPLS*, RFC 6805, (2012, November)
13. O. Dugeon, et al., *Path Computation Element (PCE) Database Requirements*, IETF Internet Draft draft-dugeon-pce-ted-reqs-03, (2014, February)
14. I. Bryskin, et al., *Policy-Enabled Path Computation Framework*, RFC 5394, (2008, December)
15. A. Atlas, T. Nadeau, D. Ward (eds.), *Interface to the Routing System Problem Statement*, draft-ietf-i2rs-problem-statement (2015, March)
16. E. Crabbe, I. Minei, S. Sivabalan, R. Varga, *PCEP Extensions for PCE-initiated LSP Setup in a Stateful PCE Model*, draft-ietf-pce-pce-initiated-lsp, (2015, October)
17. J. Case, D. Harrington, R. Presuhn, B. Wijnen, *Message Processing and Dispatching for the Simple Network Management Protocol (SNMP)*, STD 62, RFC 3412, (2002, December)
18. R. Enns, et al., *Network Configuration Protocol (NETCONF)*, RFC 6241, (2011, June)
19. M. Bjorklund, (ed.), *YANG—A Data Modeling Language for the Network Configuration Protocol (NETCONF)*, IETF Request or Comments 6020, (2010, October)
20. A. Bierman, M. Bjorklund, K. Watsen, *RESTCONF Protocol*, draft-ietf-netconf- restconf, (2015, July)
21. J.P. Vasseur, J.L. Le Roux (eds.), *Path Computation Element (PCE) Communication Protocol (PCEP)*, RFC 5440 (2009, March)
22. A. Aguado, et al., ABNO: a Feasible SDN Approach for Multi-Vendor IP and Optical Networks, in *OFC Conference*, Th3I.5 (2014, March)
23. J. Seedorf, E. Burger, *Application-Layer Traffic Optimization (ALTO) Problem Statement*, RFC 5693, (2009, October)
24. L. Velasco, D. King, O. Gerstel, R. Casellas, A. Castro, V. López, In-operation network planning. IEEE Commun. Mag. **52**(1), 52–60 (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. M. Broadbent, D. King, S. Baildon, N. Georgalas, N. Race, OpenCache: a software-defined content caching platform, in *1st IEEE Conference on Network Softwarization (NetSoft)*, London (2015, April)