

# A Control and Management Architecture Supporting Autonomic NFV Services

L. Velasco<sup>1\*</sup>, R. Casellas<sup>2</sup>, S. Llana<sup>1</sup>, Ll. Gifre<sup>3</sup>, R. Martínez<sup>2</sup>, R. Vilalta<sup>2</sup>, R. Muñoz<sup>2</sup>, and M. Ruiz<sup>1</sup>

<sup>1</sup> Optical Communications Group (GCO), Universitat Politècnica de Catalunya (UPC), Barcelona, Spain.

<sup>2</sup> Centre Tecnològic de Telecomunicacions de Catalunya (CTTC), Barcelona, Spain

<sup>3</sup> Universidad Autónoma de Madrid (UAM), Madrid, Spain

\*Corresponding author: [lvelasco@ac.upc.edu](mailto:lvelasco@ac.upc.edu)

*Abstract*—The proposed Control, Orchestration and Management (COM) architecture is presented from a high-level point of view; it enables the dynamic provisioning of services such as network data connectivity or generic network slicing instances based on Virtual Network Functions (VNF). The COM is based on Software Defined Networking (SDN) principles and is hierarchical, with a dedicated controller per technology domain. Along with the SDN control plane for the provisioning of connectivity, an ETSI NFV Management and Orchestration (MANO) system is responsible for the instantiation of Network Services, understood in this context as interconnected VNFs. A key, novel component of the COM architecture is the Monitoring and Data Analytics (MDA) system, able to collect monitoring data from the network, datacenters (DC) and applications which outputs can be used to proactively reconfigure resources thus adapting to future conditions, like load or degradations. To illustrate the COM architecture, a use case of a Content Delivery Network (CDN) network service taking advantage of the MDA ability to collect and deliver monitoring data is experimentally demonstrated.

**Keywords:** Control, Orchestration and Management. Disaggregated optical networks. Autonomic networking.

## 1 INTRODUCTION

The need to provision services within complex end-to-end scenarios, spanning multiple knowledge domains, technologies and administrative boundaries, while doing so dynamically and in a cost-effective way, has driven the design and refinement of functional and protocol architectures and frameworks for the operation of such telecommunication networks and infrastructures. Loosely speaking, the systems that implement such architectures are conceived following a layered approach, are conceptually placed in different planes of operation depending on the associated functions and are collectively referred to as control, orchestration and management. For example, it is commonly accepted that the control aspect (or plane) is the subsystem and set of functions especially dedicated to the dynamic and on-demand provisioning of network connectivity services between endpoints. It often relies on standard interfaces, operating across domains ensuring vendor inter-operability, and is responsible for configuring associated switching and forwarding state at the data plane level. On the other hand, the management plane focuses on aspects involving Fault management, Configuration, Accounting, Performance and Security (FCAPS) including, in particular, the configuration of the control plane. The targeted services to be supported are increasingly complex, ranging from simple voice and data connections within single administrative domains, to services involving highly dynamic usage patterns requiring the allocation of heterogeneous resources with complex placement constraints [1].

In this line, the overall objective of the EU H2020 Metro-Haul project [2] is to architect and design cost-effective, energy-efficient, agile and programmable metro networks that are scalable for 5G access and future requirements, encompassing the design of all-optical metro nodes (including full compute and storage capabilities), which interface effectively with both 5G access and multi-Tb/s elastic core networks.

Consequently, in view of the requirements associated to 5G networks that involve service provisioning, a key component of the targeted Metro-Haul network is the control, orchestration and management (COM) system that allows dynamic provisioning. A common trend is to design such service and resource orchestration systems by adopting, extending and building on top of frameworks that follow Software Defined Networking (SDN) and Network Function Virtualization (NFV) principles.

SDN can be defined as a centralized control model architecture and protocols, highlighting the control plane and data plane separation, and enabling an application layer. The simplest architectures encompass a single, logically centralized Controller on top of the data plane Network Equipment (NE) or devices (infrastructure or data plane

layer), with the control logic placed within the controller. The interface and (associated protocol) by which a controller communicates with devices is referred to as the South Bound Interface (SBI), while the set of Application Programming Interfaces (API) offered to applications is named the North Bound Interface (NBI).

NFV can be initially defined as an architecture and deployment model around the idea of replacing dedicated network appliances with software implementations running on common shared hardware (hosts), becoming VNFs. The benefits have been well established, including: *i*) lower costs, replacing dedicated appliances with shared servers, *ii*) use of capacity on demand and efficient resource usage, *iii*) reduction of operational costs with fewer appliances to deploy and maintain, *iv*) deliver on-demand and/or pay-as-you-go business models, and *v*) enable innovation by making it easier to develop, deploy and orchestrate network functions [3]. The ETSI NFV architecture defines the NFV Infrastructure (NFVI) deployed across multiple points of presence (NFVI-PoP) for supporting the instantiation of Virtual Machines (VM), along with the Management and Orchestration (MANO) subsystem, which deals with the orchestration of VNFs and how to deploy them as components of the so-called Network Services.

In order to attain true autonomic networking [4], SDN control and NFV management need to be augmented with instantaneous data-driven decision-making, using advanced monitoring concepts and machine learning (ML) tools [5], feeding management and control planes. This holistic platform not only caters for centralized and programmable control, but also makes ML-driven decisions to trigger actions, essentially connecting data-driven automation with policy-based orchestration and management. To this end, the architecture includes a monitoring and data analytics (MDA) framework that collates monitoring data records from network, datacenters (DC) and applications and contains ML algorithms. The data analytics capabilities enable to implement local *control loops*; monitoring data can be analyzed, and reconfiguration can be triggered to adapt resources to changing conditions. It is worth highlighting the importance of the control loops for autonomic networking, as it fundamentally changes the way networks are operated today -empowering truly dynamic and autonomous operation. Examples of control loops include soft-failure detection and localization [6], [7], and autonomic virtual network topology (VNT) management [8], [9].

The rest of the paper is organized as follows. Section 2 presents the Metro-Haul COM architecture, which is motivated by the kind of services to be provided, as well as by some of the decisions made regarding the infrastructure. Section 3 describes an illustrative use case where a virtualized Content Delivery Network (CDN) service autonomously adapts to the load by requesting the instantiation of new VMs in selected leaf cache nodes, as well as by incrementing the capacity of the network connecting users with the caches. Autonomous decisions are supported by the use of data analytics techniques on monitoring data. Next, Section 4 experimentally demonstrates the proposed CDN use case and validated the Metro-Haul COM architecture. Finally, Section 5 concludes the paper.

## 2 METRO-HAUL SERVICES, INFRASTRUCTURE, AND CONTROL

This section presents the Metro-Haul architecture. First of all, some targeted services are introduced. Second, the infrastructure, consisting of a set of interconnected Central Offices (COs) in a regional or metropolitan area, is described. Finally, the former aspects are put together in order to motivate the design of the COM architecture.

### 2.1 Service provisioning

As stated, the Metro-Haul COM is responsible for the dynamic provisioning of services; from a control and management point of view, we have identified a set of basic services to be developed and on top of which main use cases will be demonstrated.

- A. *Network connectivity services.* The core services considered include connectivity provisioning between MAC and IP addresses, optical layer ports, configuration of Passive Optical Networks (PON), and connectivity for Virtualized Network Functions (VNFs) paths.
- B. *Virtual Services.* Metro-Haul include cloud resources in the COs to support the instantiation of VMs and VNFs, being able to configure VMs and attach virtual NICs to soft switches. The COM system shall enable virtualized services, allowing maximum flexibility in the placement of the VMs, based on the actual services requirements (latency, bandwidth) that constraint the feasible geographic locations.
- C. *ETSI / Network Function Virtualization (NFV) Network Services and slicing.* A key service offered by the COM system is network slices. Within the scope of Metro-Haul, the main focus of a network slice is the ETSI NFV Network Services, that is, a set of interconnected VNFs, across the Metro-Haul infrastructure. It is worth

noting that Metro-Haul aims at extending the current state-of-the-art to include not only VNFs, but also Physical Network Functions (PNF) (e.g., PNFs are physically attached to packet switches in Metro-Haul nodes).

- D. *Monitoring Services.* Cognitive network architectures have been proven to self-adapt the network in a cost-effective manner (autonomic networking) [10]. Specifically, by applying data analytics to monitored data, control loops can be enabled, where analysis outcomes can be used to recommend network reconfiguration actions to the SDN controller. Autonomic networking entails thus the capability to do measurements on the data plane and generating data records that are collected and analyzed to discover patterns from data. Monitoring data need to be available for the different NFV network services, so control loops can be implemented at the application layer.

It is worth highlighting that the main challenge and novelty of the COM system is to offer the aforementioned services in a context characterized by having multiple, geographically dispersed sites or locations, interconnected by a multi-layer transport network, heavily relying on the optical technology to support the increasing bandwidth and latency requirements as we present next. This involves a better integration of existing NFV MANO systems with transport networks, which remain a challenge themselves due to heterogeneity of data, control and management planes [11].

## 2.2 Metro-Haul Infrastructure

The Metro-Haul infrastructure includes a set of COs together with connectivity among the COs. Fig. 1 shows the heterogeneous resources in each CO. The four main pillars of the Metro-Haul architecture are:

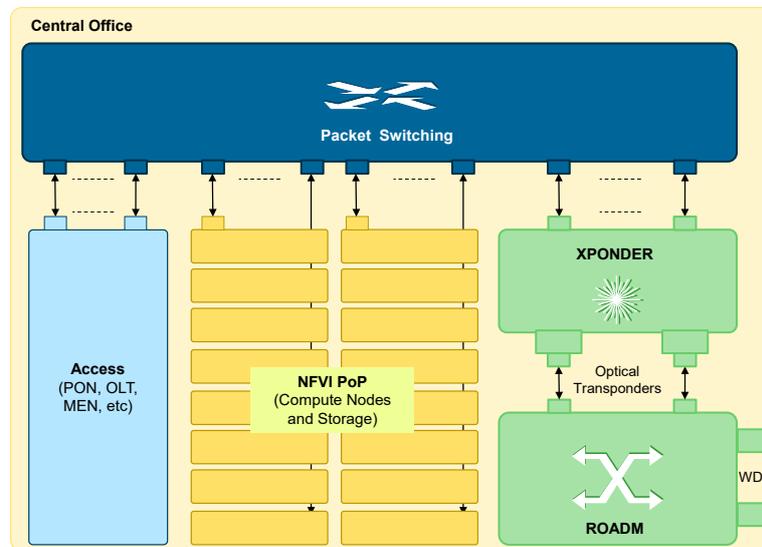


Fig. 1. Metro-Haul Central Office Infrastructure.

- A. The Optical disaggregated transport network, which provides high bandwidth, low latency connections between remote COs and constitutes the core part of the Metro-Haul infrastructure. The optical network is attained as a combination of Optical- Network Elements (O-NEs), where one or more O-NEs are physically located in each CO. Each O-NE includes a local controller with a direct NETCONF [12] connection to the SDN Controller based on a common YANG data model [13], as well as a connection to the monitoring and data analytics (MDA) system. Two types of O-NEs have already been identified, the ROADM, and the XPONDER.
- B. Computing and Storage Infrastructure. A variable number of computing, storage, and virtualization servers, are available on each CO, which constitute part of the NFVI of the Metro-Haul network. The set of servers in each CO are managed by a dedicated Virtual Infrastructure Manager (VIM), within a unique administrative domain that spans the whole Metro-Haul. Another term for the NFVI-PoP is local DC; it is assumed that compute nodes are attached to the packet switched networks, e.g., through Ethernet interfaces.
- C. Packet switched networks. Each CO includes a Layer 2/Layer 3 packet switched network that aggregates traffic coming from access and aggregation networks (such as PON) and which provides connectivity to functions and applications running in the local DC. The integration of PON is part of the Metro-Haul COM system,

where the network orchestrator coordinates the configuration of the PON during the provisioning of connectivity services.

- D. **Active Monitoring.** As part of the monitoring and telemetry system, active probes working at 10 and 100 Gb/s will be available [14]. Such probes will enable the characterization of network paths and the identification of capacity bottlenecks.

### 2.3 Control, Management and Orchestration System

With the above in mind, the COM architecture consists of four main building blocks for the control and orchestration of the network, compute and storage, NFV, and monitoring.

#### A. Network Control and Orchestration

Since transport networks are increasingly segmented in domains, to enhance scalability, due to confidentiality reasons or by virtue of having non-interoperable vendor islands, Metro-haul Network Orchestration relies on an over-arching control adopting hierarchical control architectures with a parent SDN controller abstracting the underlying complexity. Regarding the control of the optical domain, disaggregated optical networks are considered in contrast to traditional proprietary optical transport networks, so networks are deployed by composing and assembling open, available components, devices and sub-systems. There are new challenges in its control and management associated to disaggregated networks, and we adopt open interfaces exporting programmability along with unified and systematic information and data modelling. That said, optical systems remain hard to model due to the lack of agreed-upon hardware models and exhaustive vendor proprietary extensions, which remains critical for the development of an interoperable ecosystem around disaggregated hardware. Finally, the coordinated control of the packet layer and the optical layer is needed to provision end-to-end services and steering traffic coming from aggregation networks.

#### B. Compute and Storage Integration via SDN/NFV

Joint IT/Cloud and Network Orchestration is used to refer to the coordination of resources to deploy services and applications that require storage, computing and networking resources. The approach involves integrating the NFV MANO functional elements with hierarchical transport network control planes, which are abstracted as WAN infrastructure managers (WIM), as show in Fig. 2.

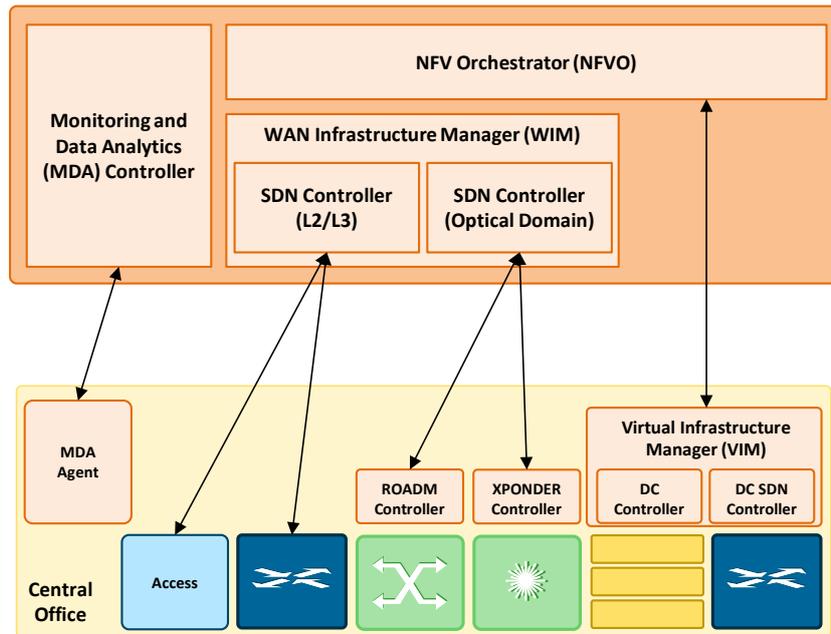


Fig. 2. Metro-Haul COM Architecture.

#### C. ETSI MANO / Slicing Integration

The ETSI NFV framework can be used as a starting point for a concrete implementation of a generic slicing architecture, in which network slice instances are NFV Network Services (NS), encompassing NS endpoints and one or more VNFs interconnected by logical links, forming VNF forwarding graphs (VNFFGs). Logical links are thus mapped to supporting network connectivity services which may, in turn, span multiple network segments.

#### D. Monitoring and Data Analytics

Autonomic networking entails the capability to do measurements on the data plane and generating data records that are collected and analyzed to discover patterns; known as knowledge discovery from data (KDD). Such knowledge can be used to issue re-configuration/re-optimization recommendations toward COM modules, such as an SDN controller or orchestrator.

In Metro-Haul, data analytics is distributed; a centralized *MDA controller* that contains a big data repository and data analytics capabilities is used to collate monitoring data records from COs. Data records and notifications are stored and processed in correlation with operational databases. In addition, every CO includes a *MDA agent* that collates monitoring data from the network, DCs and applications. The MDA agent includes a local module containing data analytics applications for handling and processing data records, and allows local control loops implementation, e.g., collecting monitoring data and applying re-configuration/re-tuning to devices in the CO.

### 3 USE CASE: CONTENT DELIVERY NETWORK (CDN)

To illustrate how the Metro-Haul COM architecture can support applications based on VNFFGs, in this section we target the adaptation of a typical operator-owned CDN (see e.g., [15]-[16]); in the proposed architecture, a CDN manager is in charge of the virtualized CDN function for live-TV and video on demand (VoD) distribution. The virtualized CDN collects monitoring data from the caches and dynamically reconfigures them to fulfil current and future demand. By virtualizing the CDN function, CDN costs can be minimized by dynamically adapting computing and network resources to current and future users' needs while ensuring the highest quality [17].

#### 3.1 CDN architecture

Many live-TV and VoD distribution systems are based on the MPEG Dynamic Adaptive Streaming over HTTP (MPEG-DASH) technique [18]. MPEG-DASH enables media content delivering over HTTP protocol using standard HTTP web servers' infrastructures to users' devices. In that regard, MPEG-DASH divides contents into a sequence of small file *segments*, each containing a short interval, e.g., 2 seconds, of the content and provides mechanisms to request the segmented contents. Every time a new content is generated, a Media Presentation Descriptor (MPD) file is constructed containing all the parameters related to segment sizes, representations, codecs, file URLs, etc., required to reproduce that content. In fact, distribution of segments can be load balanced, among others, to mitigate disruptions in the video distribution service during the reconfigurations and spikes in demand.

Fig. 3 presents the proposed architecture and the control loops that allow to dynamically adapt the CDN. A virtualized hierarchical CDN can be deployed on the Metro-Haul infrastructure with some few primary caches that are the entrance point for new multimedia contents and a number of *Leaf Cache Nodes* running in COs, close to end users: a centralized *CDN Admission and Control* module implements CDN user access policies and redirects users' requests, e.g., based on their geographical location, to the leaf cache node that will serve them. Leaf cache nodes distribute VoD contents that are stored locally based on its popularity [19], whereas live-TV content is locally prepared from a raw video stream [20].

A virtualized leaf cache node would consist of the following components running as software inside VMs deployed in the same CO. The *packager* is in charge of live-TV preparation, including segmentation and MPD file generation. The *HTTP server* component serves end users' segment requests. The *Cache Manager* is the entry point of the cache node; it receives users' requests, identifies which contents will be locally stored, and redirects users' requests to the appropriate HTTP server.

Each component usually consists in a pool of resources for load balancing and redundancy purposes. The amount of resources in every resource pool can be dynamically adapted in response to spikes in demand, e.g. during a sports event. Additionally, the Cache Manager exposes an NBI to the CDN manager to allow its remote reconfiguration.

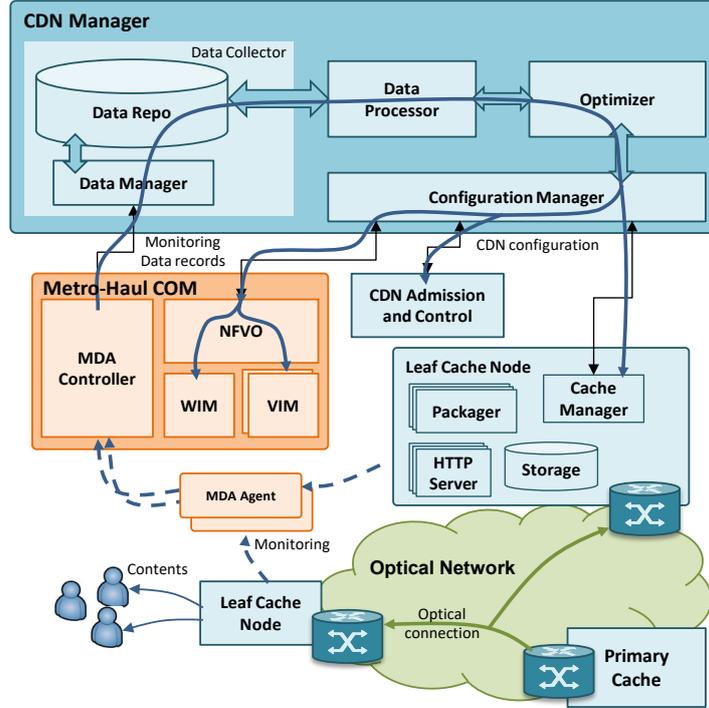


Fig. 3. Autonomous network service management use case

Finally, a *CDN Manager* is responsible for adapting the CDN to the current and future load. To that end, monitoring data is collected from the MDA controller and analyzed; specifically, the logs from the cache manager that contain useful information regarding user activity and contents access together with the activity of the packagers and HTTP servers are analyzed. Specifically, contents' popularity can be computed not only at the leaf cache level, but also at the CDN manager level. This enables reassigning users among leaf caches without affecting the structure of the cached contents. The analyzed performance and load of the CDN is used to elastically adapt its resources to current and near future service needs.

A configuration manager is in charge of interfacing the NFVO to request and release IT and networking resources and of properly configuring every component.

### 3.2 Dynamic CDN Adaptation based on Demand Prediction

To illustrate how the targeted elastic adaptation can be realized, in this section we state a possible optimization problem that a CDN operator might want to solve, so as to use resources as a function of the demand. Hence, the Dynamic CDN Adaptation problem can run periodically, e.g., every hour, focusing on minimizing the CDN cost by using more efficiently the resources and releasing unused resources, as well as on minimizing the impact of the reconfiguration on the users being served, while ensuring the committed Quality of Service (QoS). The problem can be formally stated as follows:

Given:

- A set  $LC$  of leaf cache locations; index  $l$ .
- A set  $A$  of metro areas with users consuming contents, index  $a$ .
- A set  $C$  of multimedia contents being consumed, e.g., an episode of a TV series; index  $c$ .
- A set  $F$  of content formats, e.g., high definition (HD), full HD, ultra HD. Every format  $f$  is defined as a tuple  $\langle m, d \rangle$ , where  $m$  is the maximum size of its segments (in MB) and  $d$  is the maximum delay to be ensured, measured from the time a segment is requested by a user to the time it is completely downloaded.
- The capacity  $c_l$  of every LC location  $l$  in terms of number of VMs that can be requested.
- The number of VMs  $v_l$  currently used in location  $l$ .
- The current assignments of users in every area  $a$  consuming content  $c$  with a specific format  $f$  and the leaf cache from where they are being served.
- The current connectivity between every area  $a$  and every location  $l$ , specified by the tuple  $\langle b_{al}, r_{al}, h_{al} \rangle$ ,

where  $b_{al}$  is the capacity of the connection,  $r_{al}$  the measured round-trip time, and  $h_{al}$  the measured throughput.

- The number of users (demand)  $u_{acf}$  in an area  $a$  consuming content  $c$  with a specific format  $f$  to be considered for the next period  $t+\Delta t$ .  $u_{acf}$  is actually the maximum between the current demand and the estimated demand for the next period  $t+\Delta t$ .

*Output:*

- Configuration of every location in terms of number of VMs.
- Assignment of every set of users in an area consuming a content in a given format to a given location.
- The capacity (in Mb/s) of every connection between areas and locations, where the capacity is limited to a given maximum. Note that multicast might be used to distribute live-TV contents from locations to areas.

*Objective:*

Minimize the CDN cost from serving the demand for the next period, while guaranteeing the committed QoS in terms of maximum delay and minimize the migration costs from the current configuration to the new one.

Although most of the input data (sets and parameters) of the problem can be precisely determined, it is worth highlighting that the number of users  $u_{acf}$  for which the CDN is optimized requires from some sort of prediction. Such demand has to be estimated for the next optimization period; ML algorithms can be used to produce a predictive model for every tuple area-content-format, based on monitoring data available in the Data Repository. In addition, ML-based models can be estimated to predict the evolution of the relative popularity of a group of contents, e.g., a TV series, with time; these models would require a longer period of time to be produced, e.g., months. In addition to these two groups of predictive models based on monitoring data, some other information, that might be represented as well as models, is the so-called contextual; it includes the contents plans, the competence counter-programming, special events, etc. With all such sources of predictions, the estimated number of users  $u_{acf}$  can be computed. Fig. 4 presents a possible approach for the CDN manager.

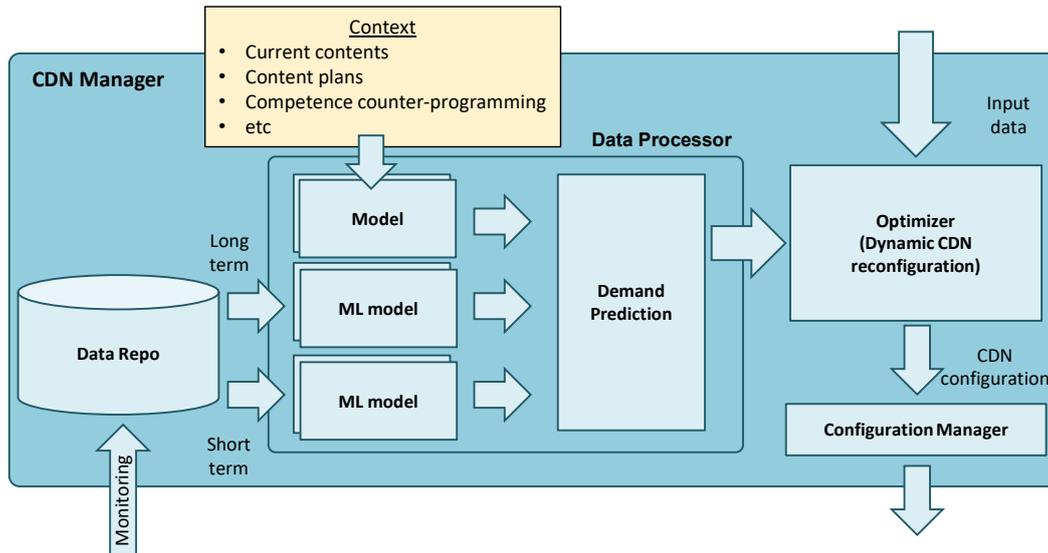


Fig. 4. Details of data generation and optimization in the CDN Manager

### 3.3 Proposed workflows

Specifically, in this paper we focus on two workflows: WF1 is related to CDN self-adaptation scaling up/down in the event of an increment/decrement in the amount of contents being served from a given leaf cache node; WF2 focuses on pushing new contents from primary caches toward the leaf cache nodes and it is supported by the dynamic set-up of optical connections (specifically, light-trees are considered to minimize the number of optical transponders needed [21], [22]). Fig. 5 details the proposed workflows.

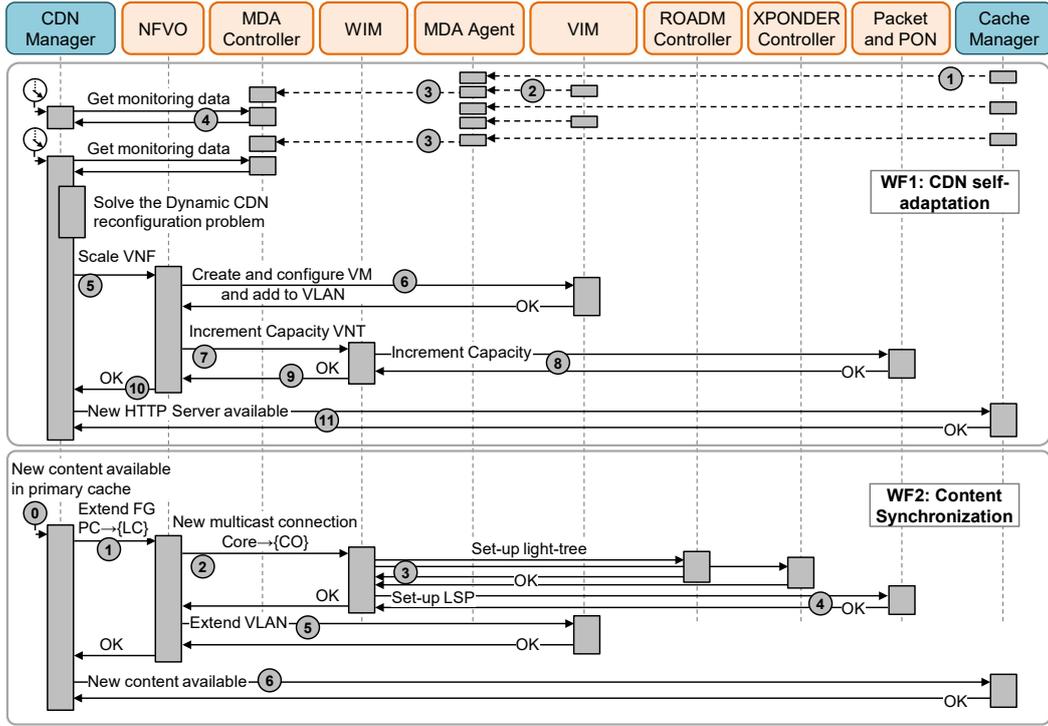


Fig. 5. CDN-driven workflows supported by the Metro-Haul COM system.

WF1 starts when new monitoring data is collected by the MDA agents. In particular, a set of observation points (OP) are defined for the CDN from which measurements are performed, including the logs from the cache managers, HTTP servers, and packagers, status of computing (e.g., CPU load, memory, etc.), storage resources (e.g., capacity used, number of files, etc.), and networking (e.g., throughput and latency). In the workflow, measurements are collected from both, the local cache manager and the VIM in a CO (messages 1 and 2 of WF1 in Fig. 5). The MDA agent can aggregate monitoring data to adapt the monitoring period to the one required by the MDA controller, aiming to control the amount of monitoring data that is conveyed [23]. MDA agents send aggregated monitoring data to the MDA controller (3), which becomes accessible for the applications. Periodically, the CDN manager interrogates the MDA controller regarding data availability for the defined OPs, and monitoring data is retrieved from the MDA monitoring data repository by the CDN manager (4).

Collected monitoring data are analyzed by the CDN manager; precisely, the *Data Processor* module is the placeholder for ML algorithms able to discover knowledge from data, generating predictive models for the load, content popularity, etc. Periodically, the dynamic CDN reconfiguration problem is solved taking as input the predicted load for the next period. In the case that the resources allocated in a leaf cache node become not enough in the near future to satisfy the required QoS for the users being served by a leaf cache node, the CDN controller might reconfigure such leaf cache in advance [17]. Specifically, we assume in this example that the pool of HTTP servers need to be increased and the capacity of the VNT connecting the leaf cache to the users incremented; we assume that such VNT includes capacity for live-TV and for VoD contents. Hence, the CDN controller request the NFVO to scale the VNF by increasing the number of HTTP servers and incrementing the capacity of the VNT toward the users (5-10). Finally, the availability of a new HTTP server is announced to the local cache manager, so that it can add that server to its local pool (11).

WF2 starts when new content is available in the primary caches (message 0 of WF2 in Fig. 5) and the CDN manager decides to make it available for the users. In such case, the CDN manager issues a request to the NFVO to extend the current forwarding graph to connect a primary cache (PC) with a set of leaf caches (LC), specifying the required capacity to be supported (1). The NFVO translates the request into a new request to set-up a multicast connection to the WIM (2), and depending on the capacity requirements, the connection is set-up at the optical layer or at the MPLS layer [24]. In the case of a high-capacity connection, the WIM decides to establish a light-tree (3) and extends the connectivity in the CO at the packet layer (4). When the connectivity is ready, the NFVO extends the VLAN associated to the VNFFG (5) and announces the cache manager that new content is available (6).

## 4 EXPERIMENTAL DEMONSTRATION

This section is devoted to experimentally demonstrate the support of the Metro-Haul architecture to the autonomic network service reconfiguration; specifically, the workflow WF1 defined in Section 3 is demonstrated for VoD contents.

### 4.1 Scenario

The experiments have been carried out on a distributed test-bed connecting CTTC and UPC premises in Barcelona area (Spain).

CTTC's WIM is implemented extending the ONOS controller framework. The WIM is the entry point for other entities to operate the transport network that interconnects the different datacenter in the COs; it is a dual layer network, encompassing a packet switched, OpenFlow controlled layer, as well as an optical layer implemented as a disaggregated optical network. The latter is controlled by a dedicated ONOS instance, using YANG/NETCONF interfaces. For this validation, a point to point link between two OpenFlow packet switches relies on an optical connection supported over 3 devices emulating an Open Line System (OLS) with a transceiver, a WSS and a receiver. Fig. 6 shows the testbed setup with the transport network segment, interconnecting the PON network (uuid=uuid1) with a remote datacenter CO (uuid=uuid2)

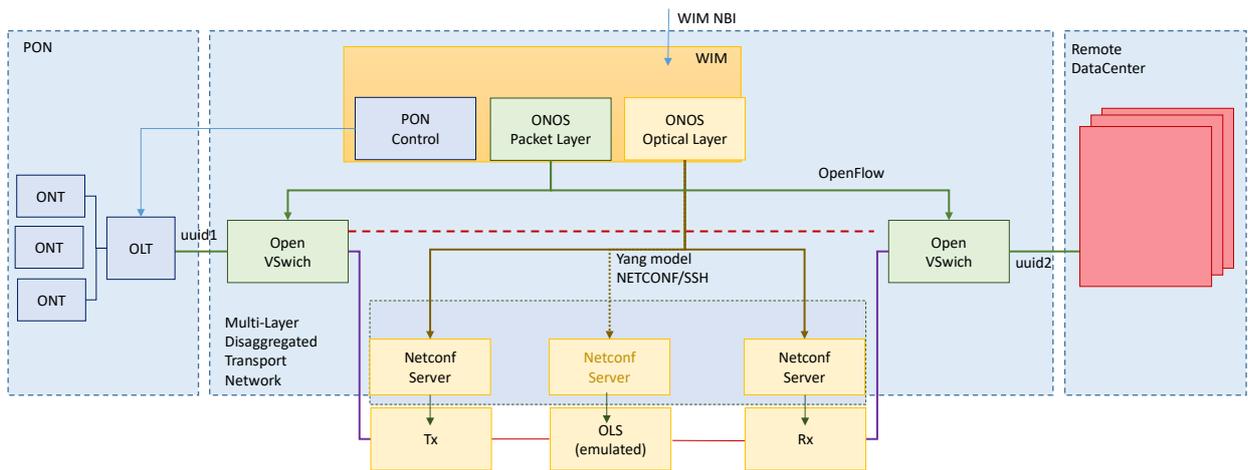


Fig. 6. Experimental testbed for the transport network segment.

From the point of view of the WIM NBI, a service YANG data model is registered into the ONOS YANG subsystem, so a RESTCONF based interface can automatically be used to trigger service provisioning. At this stage, the service provisioning is based on a simple YANG data model to request connectivity, although it is planned to evolve it to either the Transport API (TAPI) interfaces [25], IETF NBI models [26] or a (future) ETSI interface specification between a NFVO and a WIM. In any case, the WIM provides connectivity to different service endpoints. Upon request through its NBI, the WIM proceeds to configure the optical elements through edit-config NETCONF messages.

UPC's setup includes the CDN service, running in a OpenStack Pike [30] DC, the NFVO, and the MDA system based on CASTOR Monitoring and Data Analytics platform [27]-[29]. The CDN manager and cache manager were developed in Python. Communication between the CDN manager and the MDA controller and NFVO is performed through ad-hoc, bidirectional REST APIs. The Ceilometer module in OpenStack has been used to periodically send measurements such as CPU, disk, and network statistics toward the local MDA agent. In addition, Filebeat [31] is used to export applications' logs toward a centralized Logstash [32] instance, which is in charge of processing and forwarding them. Both, Ceilometer and Logstash use HTTP to export monitoring data. An IT controller has been developed in Python as an intermediate module between MDA agents and OpenStack and Logstash aiming at reducing the complexity of the MDA agent (see Fig. 7). The IT controller includes a Monitoring Manager in charge of the monitoring configuration; it receives samples and processed logs and forwards those that match the defined OPs. In addition, the IT controller contains a pool of Project Managers where each manager is in charge of resource tracking of individual OpenStack project. Finally, the NBI is based on gRPC [33].

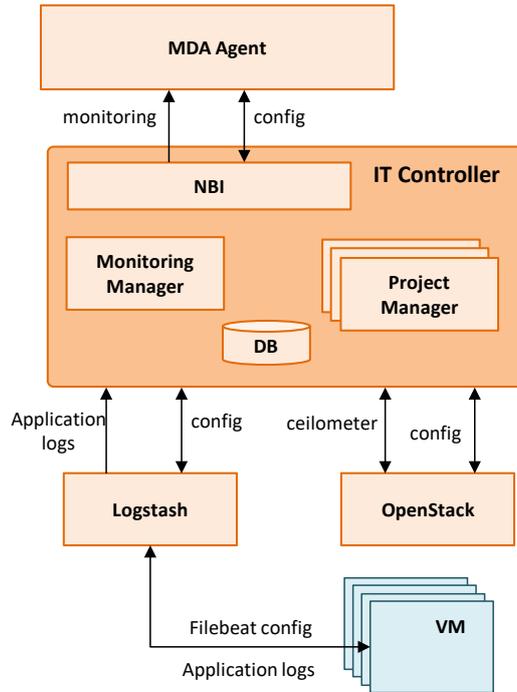


Fig. 7. Measurement collection from data centers and applications.

For demonstration purposes, a CDN service with two leaf caches serving contents was set up. Up to 1400 users were distributed in three different metro areas, where video segment requests (we defined video segments of 2 seconds) were emulated according to the number of active end users at every time instant. Finally, 10 different contents were released at different time instants according to a pre-defined schedule. In consequence, the load of the CDN was synthetically generated, based on the generic users' evolution and the popularity of contents (see Fig. 8), among others.

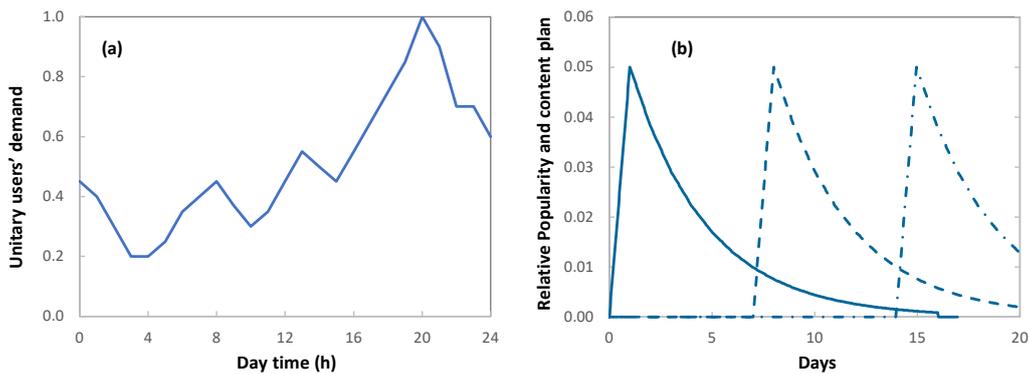


Fig. 8. CDN load characteristics: users daily profile (a) and contents release and popularity evolution (b)

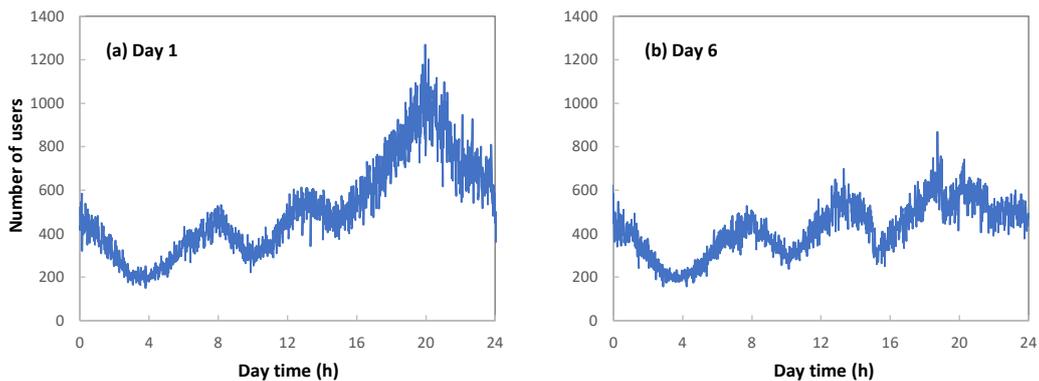


Fig. 9. Active user during day 1 (a) and day 6 (b)

To illustrate how the users' demand can largely change from one day to another as a function of contextual information, Fig. 9a-b reproduce the variation of the number of active users for two different days. It can be observed how the number of users in Fig. 9a follows the general users' daily profile in Fig. 8a, whereas the load in Fig. 9b does not. Such differences can be as a result of a sports event, or some sort of counter-programming that has attracted a large number of users.

#### 4.2 Demonstration

Fig. 10 presents the capture of messages exchanged for WF1; the same message numbering as in Fig. 5 has been used for the sake of clarity. A REST API has been used for the Cache Manager and the VIM to send monitoring data periodically to the local MDA agent (messages 1 and 2), whereas IPFIX [34] is used between MDA agents and the MDA controller (3). The CDN manager uses a REST API to retrieve monitoring data for a set of OPs from the MDA controller (4) and uses such measurements to decide scaling leaf caches and connectivity (5). In the demonstration, one HTTP server is added to one of the leaf caches and the connectivity capacity between users and that cache is increased (6-9).

	SrcIP	DstIP	Protocol	Info
①	CacheManager	MDAAgent	HTTP	POST /receiver/logs HTTP/1.1
	MDAAgent	CacheManager	HTTP	HTTP/1.1 200 OK (application/json)
②	VIM	MDAAgent	HTTP	POST /receiver/samples?batch=False HTTP/1.1
	MDAAgent	VIM	HTTP	HTTP/1.1 200 OK (application/json)
③	MDAAgent	MDAController	CFLOW	IPFIX flow ( 668 bytes) Obs-Domain-ID= 0
④	CDNManager	MDAController	HTTP	GET /getMonitoring HTTP/1.1
	MDAController	CDNManager	HTTP	HTTP/1.1 200 OK (application/json)
⑤	CDNManager	NFVO	HTTP	POST /scaleVNF HTTP/1.1
	NFVO	VIM	HTTP	POST /identity/v3/auth/tokens HTTP/1.1
⑥	VIM	NFVO	HTTP	HTTP/1.1 200 OK (application/json)
	NFVO	VIM	HTTP	POST /compute/v2.1/servers HTTP/1.1
	VIM	NFVO	HTTP	HTTP/1.1 200 OK (application/json)
⑦	NFVO	WIM	HTTP	POST /onos/restconf/data HTTP/1.1
	WIM	TxTPCtrl	NetConf	NetConf
⑧	TxTPCtrl	WIM	NetConf	NetConf
	WIM	RxTPCtrl	NetConf	NetConf
	RxTPCtrl	WIM	NetConf	NetConf
⑨	WIM	NFVO	HTTP	HTTP/1.1 200 OK (application/json)
⑩	NFVO	CDNManager	HTTP	HTTP/1.1 200 OK (application/json)
	CDNManager	CacheManager	HTTP	POST /publishHTTPServer HTTP/1.1
⑪	CacheManager	CDNManager	HTTP	HTTP/1.1 200 OK (application/json)

Fig. 10. Message list for CDN self-adaptation (WF1).

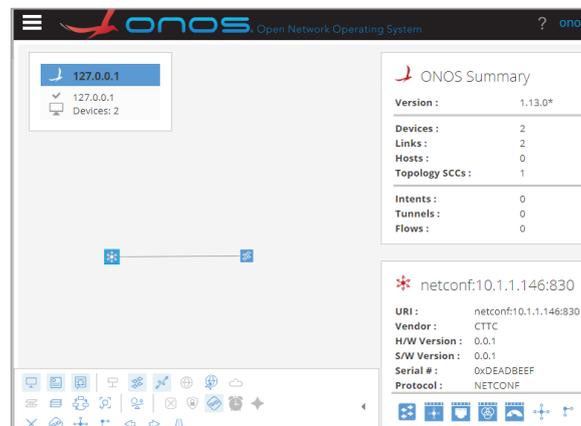


Fig. 11. ONOS SDN

```

{ "eu-metrohaul-service:vnts": {
  "vnt": [
    {
      "uuid": "46f11d97-055c-4613-9c94-da5d1095a675",
      "service-endpoint": [
        {"uuid": "b3cdd9c4-bf36-430c-8dd9-ba9b4f9d7c01",
         "name": "uuid1"
        }, {"uuid": "66d18f3d-36c5-4f4d-a097-21feb141ea42",
         "name": "uuid2"
        }
      ]
    }
  ]
}}

```

Fig. 12. Sample JSON

Fig. 11 shows the integration of NETCONF/YANG devices in the ONOS SDN controller Graphical User Interface. Before triggering the service, the controller is configured with the IP address, port and credentials to access both the Tx and Rx devices. At this point, ONOS establishes the NETCONF-over-SSH transport connection and executes the initial handshake and capability exchange. Fig. 11 shows the details of the Tx device after the handshake and other initial exchanges (port discovery) have been completed. With regards to the ONOS/WIM NBI, service provisioning is based on a RESTCONF API, based on a service YANG model that is used to trigger the dynamic provisioning. Fig. 12 shows a (partial) JSON encoding of the POST operation to configure the system (including the endpoints of the connection service that are later mapped to the NETCONF devices).

Finally, once the VNF has been scaled, the new HTTP servers are announced to the Cache Manager (11) by means of an ad-hoc REST API, so the server enters in the round-robin selection pool as soon as the VM starts.

Interestingly, the scale VNF request (messages 5-10) was completed in less than 1.5 seconds, where the REST API POST VM creation (messages 6) took 0.43 seconds (plus VM start-up time) and the RESTCONF POST operation for link provisioning (messages 8) was completed in 0.89 seconds (optical hardware configuration delay was not accounted).

## 5 CONCLUDING REMARKS

The Metro-Haul Control, Orchestration and Management Architecture has been presented and motivated by services to be provided, as well as for the objective of disaggregating the optical layer. An use case has been used to illustrate how the proposed architecture supports self-adaptive virtualized services; specifically a CDN has been proposed to show how the service autonomously adapts to the load by requesting the instantiation of new resources and increasing the capacity of existing ones. An MDA system supports autonomous decisions using data analytics techniques on monitoring data. The experimental demonstration has been carried out on a distributed test-bed connecting CTTC and UPC premises.

## ACKNOWLEDGEMENTS

The research leading to these results has received funding from the European Community's through the METRO-HAUL project (G.A. n° 761727), from the AEI/FEDER TWINS project (TEC2017-90097-R), from the MINECO project DESTELLO (TEC2015-69256-R) and from the Catalan Institution for Research and Advanced Studies (ICREA).

## REFERENCES

- [1] R. Casellas *et al.*, "Control, Management and Orchestration of Optical Networks: Evolution, Trends and Challenges," IEEE/OSA Journal of Lightwave Technology, vol. 36, pp. 1-13, 2018.
- [2] METRO High bandwidth, 5G Application-aware optical network, with edge storage, compUte and low Latency (Metro-Haul). [On-line] <https://metro-haul.eu>.
- [3] L. Velasco *et al.*, "A Service-Oriented Hybrid Access Network and Cloud Architecture," IEEE Communications Magazine, vol. 53, pp. 159-165, 2015.
- [4] M. Behringer, M. Pritikin, S. Bjarnason, A. Clemm, B. Carpenter, S. Jiang, L. Ciavaglia, "Autonomic Networking: Definitions and Design Goals," IETF RFC 7575, 2015.
- [5] D. Rafique and L. Velasco, "Machine Learning for Optical Network Automation: Overview, Architecture and Applications," (Invited Tutorial) IEEE/OSA Journal of Optical Communications and Networking (JOCN), vol. 10, 2018.
- [6] A. P. Vela *et al.*, "BER Degradation Detection and Failure Identification in Elastic Optical Networks," IEEE/OSA Journal of Lightwave Technology (JLT), vol. 35, pp. 4595-4604, 2017.
- [7] A. P. Vela *et al.*, "Soft Failure Localization during Commissioning Testing and Lightpath Operation [Invited]," IEEE/OSA Journal of Optical Communications and Networking (JOCN), vol. 10, pp. A27-A36, 2018.

- [8] F. Morales *et al.*, “Dynamic Core VNT Adaptability based on Predictive Metro-Flow Traffic Models,” *IEEE/OSA Journal of Optical Communications and Networking (JOCN)*, vol. 9, pp. 1202-1211, 2017.
- [9] F. Morales *et al.*, “Virtual Network Topology Adaptability based on Data Analytics for Traffic Prediction,” (Invited) *IEEE/OSA Journal of Optical Communications and Networking (JOCN)*, vol. 9, pp. A35-A45, 2017.
- [10] I. de Miguel *et al.*, “Cognitive dynamic optical networks [invited],” *IEEE/OSA Journal of Optical Communications and Networking*, vol. 5, pp. A107-A118, 2013.
- [11] R. Muñoz *et al.*, “Transport Network Orchestration for end-to-end Multi-layer Provisioning Across heterogeneous SDN/OpenFlow and GMPLS/PCE Control Domains,” *IEEE/OSA Journal of Lightwave Technology*, vol. 33, pp. 1540–1548, 2015.
- [12] R. Enns *et al.*, “Network Configuration Protocol (NETCONF),” *IETF RFC 6241*, 2011.
- [13] M. Bjorklund, “YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF),” *IETF RFC 6020*, 2010.
- [14] R. Leira, J. Aracil, J. López de Vergara, P. Roquero, I. González, “High-speed optical networks latency measurements in the microsecond timescale with software-based traffic injection,” *Elsevier Optical Switching and Networking*, vol. 29, pp. 39-45, 2018.
- [15] N. Kamiyama, *et al.* “Optimally designing ISP-operated CDN,” *IEICE Transactions on Communications*, vol. E96-B, pp. 790-801, 2013.
- [16] S. Spagna, *et al.* “Design Principles of an Operator-Owned Highly Distributed Content Delivery Network,” *IEEE Communications Magazine*, vol. 51, pp. 132-140, 2013.
- [17] M. Ruiz *et al.*, “Big Data-backed Video Distribution in the Telecom Cloud,” *Elsevier Computer Communications*, vol. 84, pp. 1-11, 2016.
- [18] ISO Standard, “Dynamic adaptive streaming over HTTP (DASH) - Part 1: Media presentation description and segment formats,” *ISO/IEC 23009-1*, 2014.
- [19] D. de Vleeschauwer and D. C. Robinson., “Optimum caching strategies for a telco CDN,” *Bell Labs Technical Journal*, vol. 16, 2011.
- [20] A. Asensio *et al.*, “Scalability of Telecom Cloud Architectures for Live-TV Distribution,” in *Proc. OFC 2015*.
- [21] M. Ruiz and L. Velasco, “Performance Evaluation of Light-tree Schemes in Flexgrid Optical Networks,” *IEEE Communications Letters*, vol. 18, pp. 1731-1734, 2014.
- [22] M. Ruiz and L. Velasco, “Serving Multicast Requests on Single Layer and Multilayer Flexgrid Networks,” *IEEE/OSA Journal of Optical Communications and Networking (JOCN)*, vol. 7, pp. 146-155, 2015.
- [23] A. P. Vela *et al.*, “Distributing Data Analytics for Efficient Multiple Traffic Anomalies Detection,” *Elsevier Computer Communications*, vol. 107, pp. 1-12, 2017.
- [24] Ll. Gifre *et al.*, “Experimental Assessment of ABNO-driven Multicast Connectivity in Flexgrid Networks [Invited],” *IEEE/OSA Journal of Lightwave Technology (JLT)*, vol. 33, pp. 1549-1556, 2015.
- [25] Open Networking Foundation, “Functional Requirements for Transport API,” *ONF TR-527*, 2016.
- [26] I. Busi, “Transport Northbound Interface Applicability Statement and Use Cases,” *draft IETF, work-in-progress*, October 2017.
- [27] L. Velasco *et al.*, “An Architecture to Support Autonomic Slice Networking [Invited],” *IEEE/OSA Journal of Lightwave Technology (JLT)*, 2018.
- [28] L. Velasco *et al.*, “Building Autonomic Optical Whitebox-based Networks,” *IEEE/OSA Journal of Lightwave Technology (JLT)*, vol. 36, pp. 3097-3104, 2018.
- [29] Ll. Gifre, J.-L. Izquierdo-Zaragoza, M. Ruiz, and L. Velasco, “Autonomic Disaggregated Multilayer Networking,” *IEEE/OSA Journal of Optical Communications and Networking (JOCN)*, vol. 10, pp. 482-492, 2018.
- [30] OpenStack Pike. [On-line] <https://www.openstack.org/software/pike/>
- [31] Filebeat. [On-line] <https://www.elastic.co/products/beats/filebeat>
- [32] Logstash. [On-line] <https://www.elastic.co/products/logstash>
- [33] gRPC: A high performance, open-source universal RPC framework. [On-line] <https://grpc.io/>
- [34] B. Claise *et al.*, “Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of Flow Information,” *IETF RFC 7011*, 2013.