IET Networks

Special Issue on Software Defined Networking

Journals
The Institution of Engineering and Technology

Trends on virtualisation with software defined networking and network function virtualisation

ISSN 2047-4954
Received on 31st October 2014
Revised on 6th February 2015
Accepted on 2nd March 2015
doi: 10.1049/iet-net.2014.0117
www.ietdl.org

Lorena Isabel Barona López[1], Ángel Leonardo Valdivieso Caraguay[1], Luis Javier García Villalba[1] ✉, Diego López[2]

[1]Group of Analysis, Security and Systems (GASS), Department of Software Engineering and Artificial Intelligence (DISIA), Faculty of Computer Science, Office 431, Universidad Complutense de Madrid (UCM), Calle Profesor José García Santesmases 9, Ciudad Universitaria, 28040 Madrid, Spain
[2]Telefónica I+D, c/Don Ramón de la Cruz, 82-84, Madrid 28006, Spain
✉ *E-mail: javierv@ucm.es*

Abstract: Software defined networking (SDN) and network function virtualisation (NFV) have become hot topics in recent years. On one hand, SDN decouples the control plane from the data plane allowing the rapid innovation and the introduction of new services in an easy way. On the other hand, currently proprietary appliances such as load balancers and firewalls are implemented in hardware, NFV aims to change these network functions to an open software environment using virtualisation and cloud technologies. This means a reduction of spends in the provisioning and management of telecom services. SDN and NFV are two different concepts but these can coexist and help each other. In this study, the authors present a survey of SDN and NFV focusing in virtualisation projects and the use cases where a synergy between these technologies is possible. This study includes the basic concepts of network virtualisation, NFV and SDN, current research and the relation between both technologies.

1 Introduction

Since Internet appeared in the 1970s, not only hardware equipment has been updated but also the appearance of new services. On one hand, the current services cover a broad range of topics such as cloud computing, real-time applications, video streaming, big data and social networks, among others. All of these services have triggered the exponential growth of network data traffic. However, the current architectures are not enough to cover these increased traffics in an efficient way and the standardisation process of new technologies takes a long time. On the other hand, there are many different proprietary devices, which are close and difficult to manage. When a service provider (SP) launches a different service it often needs the deployment of a new appliance, which increase both the operational (OpEx) and capital costs (CapEx). These issues have resulted in new concepts, such as software defined networking (SDN) and network function virtualisation (NFV), which are changing the way of managing and controlling networks.

SDN separates the data and the control plane in network devices in order to improve the programmability of the network [1, 2]. The SDN concept is not completely new, but rather, it takes the advances of three phases in network history [3]. First, active networks allow programmability capacities through the implementation of open interfaces in each network node. Second, the separation of the control and the data plane and finally the introduction of OpenFlow protocol [4] and network operative system (NOS) [5]. OpenFlow takes the common characteristic of traditional switches to create an open interface without exposing the internal structure of them.

Concerning hardware components, each time a new function is required, these are implemented in proprietary hardware. In this context, NFV is a novel approach based on virtualised network functions (VNFs), which aim to reduce the expenditures related to the deployment and management of new network applications. NFV increases the network elasticity because of that typical hardware appliances are changed for software functions that can run on homogeneous environment, breaking the constraints of proprietary hardware. These functions can be instantiated in different locations and can be deployed in the same standardised hardware [6].

SDN and NFV are two independent concepts but these are complementary between them. On one hand, SDN provides a full control of the network and automation capacities, which may eventually allow better performance of the virtual functions (VFs). Both technologies have common objectives such as faster time to market (TTM) for the deployment of new services and create independence from the hardware vendor. SDN/NFV could also enhance the current NFs providing better elasticity, scalability and programmability.

This paper presents a review of these technologies, their current development and the current challenges. This piece of work is organised as follows: Section 2 briefly gives an overview of network virtualisation history. In Section 3, we introduce the term NFV. Then, SDN approach is discussed in Section 4. In Section 5, we explain the relation between both technologies. Finally, in Section 6 we open a discussion and conclusion related with the topic.

2 Virtualisation network: a view in the history

Virtualisation is not a new concept; the first approach was the virtualisation of the computer's memory. Nowadays, it is possible to virtualise operative systems, computer hardware platforms, storage capacities and networks. Essentially, virtualisation concept refers to the abstraction of the logical resources from the physical resources, creating multiple logical instances over the same physical infrastructure [7]. In this sense, network virtualisation allows simultaneously multiple virtual networks (VNs) over a physical network as shown in Fig. 1.

There are some technologies that use the virtualisation principle within networks, such as virtual local area network (VLAN). VLAN is a logical network formed by a group of hosts in a single broadcast domain, where the logical topology is usually different from the physical network. VLANs introduce facilities for

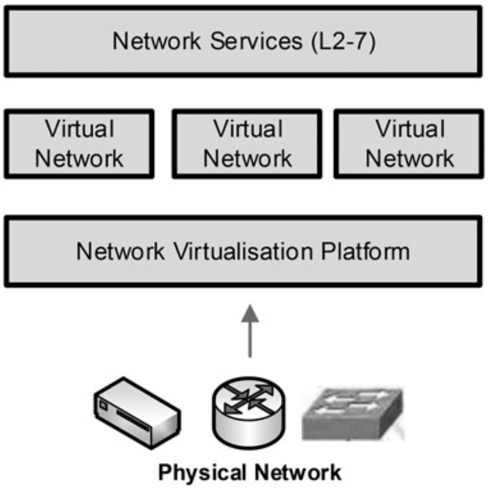

Fig. 1 *Network virtualisation*

management, reconfiguration and isolation of the network; however, the creation of 4094 VLANs is only possible over the same network. Virtual private network (VPN) is a VN that communicates multiple sites through a secured tunnel. These are geographically separated and the tunnel is carried over a public network [8]. There are different types of VPNs: layer 1 VPN (circuit switching domain), layer 2 VPN (L2 protocols such as Ethernet), layer 3 VPN (L3 protocols such as IP) and VPN over multi protocol label switching (MPLS). There are other approaches such as network interface card (NIC) virtualisation or Open vSwitch (OVS) [9]. OVS is an open source tool that allows the creation of switches in virtualisation environments. These switches allow the communication of virtual machines (VMs), providing better performance than the traditional bridge.

Other concepts such as active networks and overlay networks are also related to network virtualisation. Active networks control the network via an interface that exposes the nodes resources, allowing the automation and programmability of network devices. For its part, an overlay network allows the deployment of a logical network over the top of one or more physical infrastructures. The research community has generated many virtualisation projects in order to test with new technologies or applications such as global environment for network innovations (GENIs) or future Internet research for sustainable testbed (FIRST). GENI [8] provides a flexible platform in order to testing services and prototypes in computer networking and distributed systems. FIRST [10] is an experimental project in South Korea which promotes the research on future Internet architecture.

Recently, the virtualisation of the network components has given rise to novel concepts, such as the case of NFV. NFV applies virtualisation and cloud computing concepts in order to enhance the services provided by telecommunication SPs. NFV aims to design a standard networking device where any NF can be implemented, as is explained below.

3 NF virtualisation

NFV [6] is an initiative of main telecommunication SPs and European Telecommunications Standards Institute (ETSI), which made its first appearance in 2012. NFs industry specification group (NFV-ISG) includes more than 28 network operators and 150 enterprises from telecommunication industry. Nowadays, one of the main problems of telecom providers is the need for additional appliance when a new service is required. Often, these appliances are proprietary hardware, and therefore cannot be used by other providers. The investment in new hardware does not represent enough revenues if we take into account some factors such as the short life cycle of appliances, lack of space and complexity of the systems. NFV could help in these issues through the virtualisation of the NFs; these functions can run on standard switches, storage or high-volume servers. NFV may help in the following topics: accelerate the deployment of new services and NFs, faster TTM, reduce the energy consumption, capacity for multi-tenancy and multi-version, openness environment and reduce both CapEx and OpEx in the deployment and management of infrastructures and network services. At the same time, there are some challenges to overcome, these are: portability and mobility between vendors, high performance in virtualised appliances, security, elasticity, redundancy and compatibility with legacy appliances. NFV proposes the implementation of VNFs in software compatible with other platforms, allowing the rapid innovation and easy instantiation in several places. Fig. 2 describes the central idea of NFV.

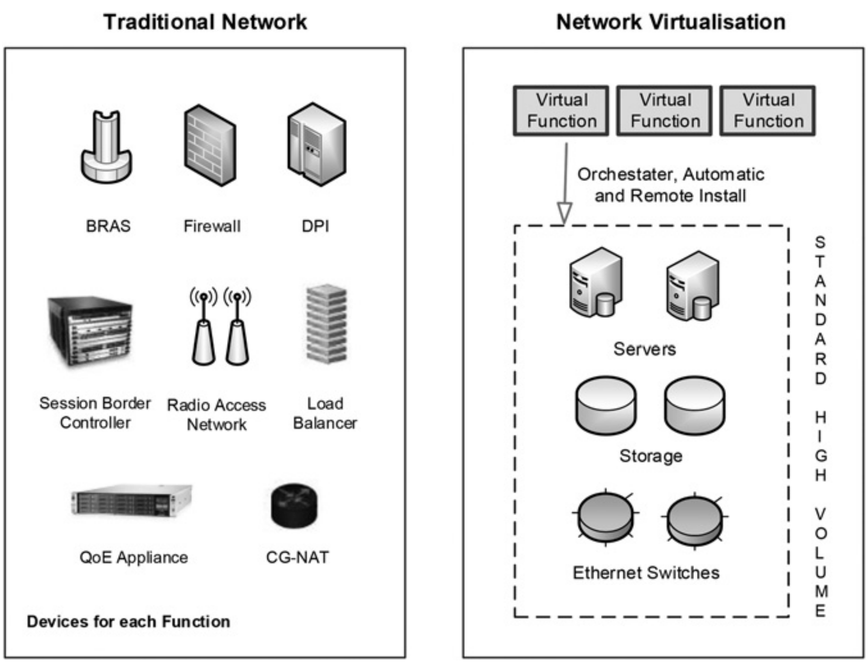

Fig. 2 *NFV approach*

IET Netw., 2015, Vol. 4, Iss. 5, pp. 255–263

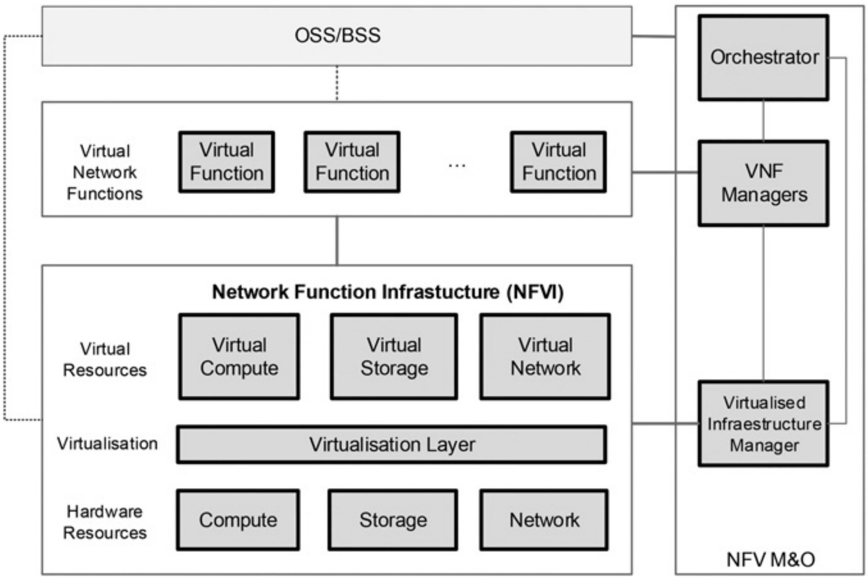

Fig. 3 *NFV architecture [11]*

NFV application fields mainly include switching elements, mobile network nodes, traffic analysis, service assurance, content distribution networks (CDNs), security functions and session border controller (SBC), among others. NFV uses the virtualisation concept of three hardware resources: computing, storage and network hardware. NFV architecture [11] is depicted in Fig. 3.

NFV identifies three main modules, these are:

- *NFV infrastructure (NFVI):* Represents the infrastructure of the system.

It includes the hardware resources (computing, storage and network), the virtualisation layer and the software instances of the hardware resources (virtual compute, virtual storage and VN). The virtualisation layer abstracts the hardware resources and ensures independency from different vendors and the deployment in different sites.

- *VNF:* Represents the implementation of an NF that runs over the NFVI. This module also contains the element management system, which manages the VNFs.
- *NFV M&O:* This component orchestrates and manages VNFs and NFVI. It has three elements, an orchestrator, a VNF manager and a virtualised infrastructure manager. Virtualised infrastructure manager has a resource inventory (computing, storage and network) in order to control the NFVI. Additionally, this module can also allow the integration with external business support system and operation support system (OSS) in order to bill the services rendered. Besides, this module guarantees the provisioning of services, no matter what the underlying technology.

There are a variety of NFV use cases defined by NFV-ISG [12]. Some of these cases are similar to services models of cloud computing (SaaS, infrastructure as a service and platform as a service). These cases are described below:

- *NFs virtualisation infrastructure as a service (NFVIaaS):* In this case, SPs can deploy its NFs in the infrastructure of another SP.
- VNF as a service (VNFaaS) allows the lease of VNFs. An enterprise can use a VNF of another SP. This case is similar to SaaS.
- In VN platform as a service (VNPaaS), an SP leases a set of applications and infrastructure (similar to platform as a service). The customer can introduce its own VNF.
- VNF forwarding graph (VNF-FG) creates a logical path with the hops (NFs) to deliver a service.

This case is useful to provide service chaining.

- Another use cases leverage the introduction of NFV in order to consolidate different types of network appliances to standard equipment, such as: virtualisation on mobile cloud network (virtualisation elements of evolved packet core), virtualisation of mobile base station (radio access network resources, eNodeB, worldwide interoperability for microwave access (WIMAX)), virtualisation of the home environment (virtualisation of customer premises equipment (CPE)) and the virtualisation of content delivery networks (CDNs).

These cases can be implemented with traditional mechanism, NFV, SDN or inclusive a combination of NFV with SDN. SDN can contribute to the network programmability and automation capacities in order to control VNs and functions, as is explained below.

4 Software defined networking

In recent years, we have seen a dramatic change in networks because of the exponential growth of network data traffic and the introduction of new technologies such as cloud computing or big data. To cover new requirements, protocols and standards must be created and tested before they are integrated in hardware. In conventional architectures, new protocols are included in a new software release or in dedicated hardware devices. Usually, this process takes a long time because all stakeholders must agree on the basic aspects related with the implementation and the deployment of new service. Once the standard is implemented, the network administrator must configure each device or change it; sometimes this process can take hours, days or even weeks, depending on the size of the network.

SDN takes into account the most outstanding contributions of the networking industry developed in the past 20 years. First, active networks introduce the programmability and automation of the network nodes (early 2000). Active networks allow the creation of customised services; however, this concept did not have a widespread use. There were performance problems and there was not a clear path for the deployment of these kinds of networks. Second, SDN applies the separating of control and data plane concept from projects such as ForCES (2001–2007) and finally it introduced two APIs (southbound and northbound) and NOSs (2007–present) [3].

In essence, SDN proposes a centralised control of the data plane, where the network intelligent resides in the controller (control plane). In this way, the manager can develop high-level applications to improve the network performance [4]. SDN architecture defines three main layers: application, control and data layers, as shown in Fig. 4.

SDN also defines two main APIs in order to connect these layers. Southbound API communicates hardware devices (data plane) with the controller (control plane) that, in turn, is connected to application layer through northbound API. Subsequently, SDN introduced East–West APIs for the communication between controllers in the same or a different domain.

Data layer is composed of the physical devices (switches and routers). The most well-known southbound interface is OpenFlow [4], that is, promoted by open networking foundation (ONF). ONF aims to accelerate the adoption of SDN and OpenFlow. It is comprised of over 100 members such as network operators, SPs and big companies such as Google.

OpenFlow switch (version 1.3.4) is based on the structure of traditional Ethernet switch. It takes the common characteristics from different vendors in order to programme the tables of the switch and handle packets based on a variety of packets header fields of different layers. The tables are divided in flow, group and meter tables [13]. The controller can add, delete or update flow entries in the flow tables. Each flow entry has associated match fields (for matching packets), counters (tracking packets), priority, timeouts and instructions (actions to be applied) in order to process the incoming packets. When a packet arrives, it is matched against flow entries of the first flow table (0) searching the highest priority. If a matching entry is found, the instructions are executed or may continue with the next tables, depend on the result of the match in the table. If the packet does not match with a flow entry in any table, the outcome depends on the configuration of the table miss flow entry (default rule).

For its part, the controller performs the rules that are applied on the switches. Nowadays, the NOS most widely used are: POX, Floodlight and the OpenDaylight (ODL) [2, 14] and all of them are open source projects.

ODL is an open source project promoted by Linux foundation which main objective is to foster the development and adoption of SDN applications. ODL introduces a plugin-based architecture that facilitates the centralised control and management of network in a flexible and modular way. ODL is implemented with Java and Python code and its structure is based on SDN architecture. ODL defines three layers: network applications and orchestration, the controller platform and the physical and VN devices. ODL has multiple southbound APIs for different protocols such as OpenFlow or Netconf and a bidirectional API based on representational state transfer (REST) (northbound API).

In some aspects, these NOSs are difficult to use, especially with the programming of complex functions. In this sense, northbound APIs such as Procera [15] and Frenetic [16] facilitate the creation of business applications or high-level programmes that are required by application layer.

There are several fields (wireless and mobile networks, security, quality of services (QoS), management, data centres and virtualisation) where SDN can be an enabler technology [17]. In particular, there are some successful migration cases from traditional network to an SDN such as Google Inter Datacenter-WAN and Stanford Campus Network, NTT Edge Case [18]. Indeed, network virtualisation has been a key to the success of these deployments, especially in data centres and cloud computing environments [19]. Currently, cloud computing infrastructure and large datacentres are composed of many racks, core and aggregation network devices and thousands of VMs. This paper is focused in the virtualisation concept and its applicability in SDN and NFV. Some SDN virtualisation projects are described below.

One of the first SDN approaches within network virtualisation is FlowVisor [20]. It allows the switch virtualisation to share the same network infrastructure with multiple tenants. It is based on the concept of computer virtualisation, that is, FlowVisor places a layer between data and control planes; in this way the network is divided in slices and each one can perform different tests without interference among them. FlowVisor uses the SDN concept to control each slice in order to test new research with production traffic. The slicing process takes into account four dimensions: the topology, the bandwidth, the device central processing unit (CPU) and the forwarding tables. To define the slice characteristics, the tenant uses a slice policy language to determine the network resources (fraction bandwidth and CPU), flowspace and the OpenFlow controller. Each slice has a text file composed of a list of tuples, and these in turn have specific actions (allow, read-only and deny).

FlowVisor rewrites the messages (from switch to controller and vice versa) to ensure the transparency between slices. Fig. 5 shows the FlowVisor architecture. FlowVisor is the base of many research and SDN deployments such as GENI and OFELIA [8, 18].

ADvanced FlowVisor (ADVisor) [21] is an enhanced version of FlowVisor that lets the creation of arbitrary topologies and it allows the sharing of the same flowspace between slices, which are the main limitations of FlowVisor. ADVisor is located between the hardware equipment and the controller and it directly sends the traffic to the respective slice, which are defined for the combinations of bits in the L2 field. For this purpose, ADVisor introduces two additional functions, virtual links management

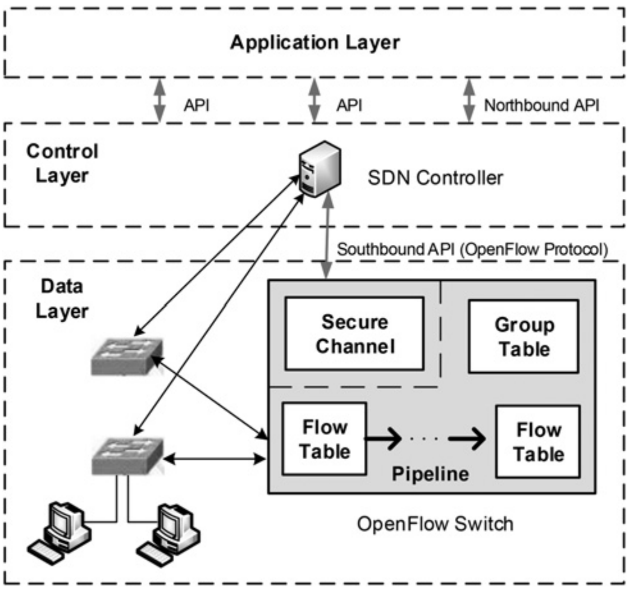

Fig. 4 *SDN architecture*

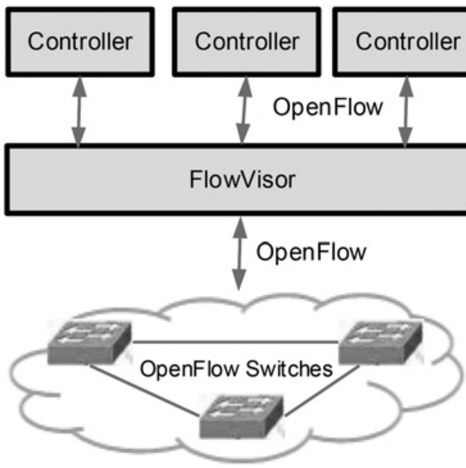

Fig. 5 *FlowVisor architecture*

(links between switches) and virtual ports management (connect virtual links with switches). In [22], a virtualisation framework is presented that allows simultaneous slices and different OpenFlow versions (OF 1.0–OF 1.1). Besides, it proposes enhanced functionalities related with QoS, management of the controller and the creation of SDN applications.

CloudNaaS [23] is a novel framework for cloud environments that allow the use of NFs such as isolation, QoS, custom addressing and transversal middleboxes. CloudNaaS lets fine-grained control over the network. It defines two main components, first the cloud controller which controls the virtual resources and the physical hosts. The second component is the network controller that manages the network devices. CloudNaaS uses four operations, first the customers specify their requirements through a simple policy language developed for this purpose. The second operation is to convert these policies in a communication matrix that is used to find out the best place for VMs. Then the matrix is converted to a network language which contains the rules to deploy the VMs. Finally, the rules are installed and configured in the network. Besides, CloudNaaS uses bin-packing for the placement of VM and allows the reuse of existing addresses.

Open application delivery networking platform (OpenADN) [24] is a design that combines the features of application SPs (ASPs) with the SDN benefits in order to allow the quick setup of application services in a distributed environment, looking such as a single data centre for each ASP. OpenADN is focused on cloud environment and uses current technologies such as OpenFlow, MPLS, session slicing and cross-layer communication. The project introduces the following elements: a virtualisation layer to slice the network, some NOSs, the network level control that invokes Internet service providers (ISPs) services and finally OpenADN creates and controls the applications for each ASP, as shown in Fig. 6. Besides, OpenADN introduces a label switching mechanism based on the composition of several sub-labels to allow the implementation of specific functions, for example, application label switching (Layer 3.5).

FlowN [25] is a system that allows network virtualisation providing an exclusive user space, arbitrary topologies and full control over the path for each tenant. FlowN is an extension of NOX controller and it uses advances in MySQL database to enhance the efficiency of the mapping process between the network and physical resources. FlowN uses an event handler to identify the packets that belong to each customer and it introduces the concept of container-based virtualisation to have multiple user-space containers independent of each other. The address space is defined by a set of fields of the packet headers and encapsulates incoming packets with an additional header (VLAN).

For its part, the relational database saves some aspects of the network: the physical topology, the virtual topology (nodes, interfaces and links) and the virtual–physical mapping. The virtual nodes can be a VM or an SDN-based switch and these are connected through some virtual interfaces and links. FlowN is able to establish some parameters such as the maximum number of flow table entries, the number of cores of the servers, bandwidth and latency for virtual links.

EHU OpenFlow enabled facility (EHU-OEF) [26] proposes a novel network virtualisation approach that enhances the flexibility and isolation between the slices. It is deployed in a real infrastructure, sharing resources between the production and the experimental traffic. EHU-OEF presents a modification of FlowVisor, enforcing slice isolation based on the MAC address settings, this method is known as Layer 2 prefix-based network virtualisation (L2PNV). EHU-OEF uses a modified NOX and allows a variety of headers such as VLANs. It also introduces an authentication and authorisation module.

VMware NSX [27] provides network virtualisation services (compute and storage) and security for data centres, where the user is able to deploy a VN as fast as a VM. NSX not only allows the deployment of L4–L7 network services but also the integration of additional capacities from third-party appliances such as specific load balancers, firewalls and so forth. NSX is based on the design of Nicira network virtualisation platform (NVP), which has an SDN-based architecture allowing the programmability of the VNs. NVP introduces a layer between the network and the final hosts. NSX components are depicted in Fig. 7.

Data plane contains an NSX vSwitch (vSphere distributed switch or OVS) which abstracts the physical network and allows communication with the hypervisor. Additionally, this plane uses a gateway between logical and physical networks (NSX Edge). The control plane has an SDN controller (NSX controller) and the management plane allows the configuration of vSphere environment. Finally, the consumption platform includes a cloud migration portal, which aids in the migration and the management in virtualisation and cloud environments. Another similar approach was developed by NEC, which is known as programmable flow (PFlow) [28]. PFlow is an SDN virtualisation solution that allows the deployment of multiple tenants in a secure environment. This paper was the first in introducing OpenFlow 1.3 and it supports OpenFlow 1.0.

A summary of current SDN virtualisation projects is presented in Table 1. It shows the applicability fields and the main characteristics of them.

As shown in Table 1, the target groups involve data centres, cloud environments, ISP, ASP, academy and research community. All of these projects ensure a degree of isolation between VNs and include OpenFlow 1.0. For its part, flexibility is related to some factors such as scalability, dependence of specific technologies and deployment facilities. High scalability requires more resources in order to support more tenants or virtual applications; this is limited by the memory size, the number of forwarding flows and bandwidth, among others. Moreover, projects carried out by enterprises [27, 28] provide modules to guarantee QoS, CloudNaaS [23] guarantees bandwidth for each VN and Advisor [21] guarantees address space.

The network virtualisation approaches are based on three methods: first, improving the functionalities of the controller [23, 25], second using an enhanced version of FlowVisor [21, 22, 26]

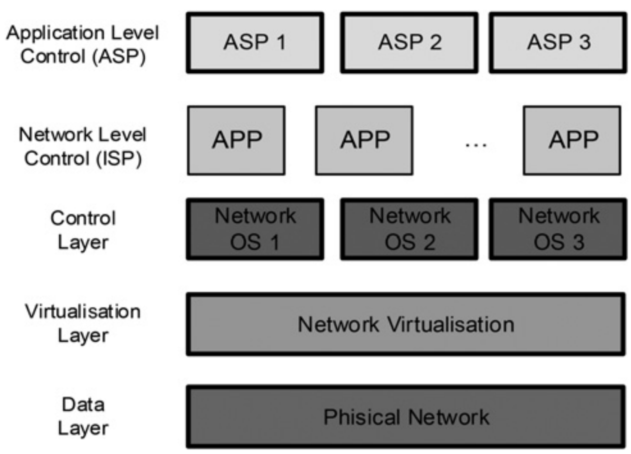

Fig. 6 *OpenADN architecture [24]*

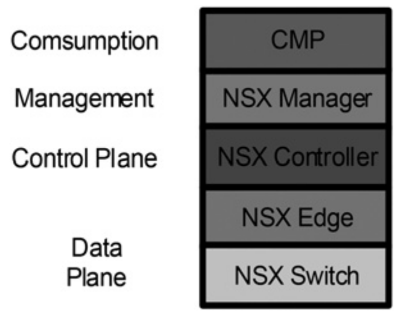

Fig. 7 *VMware NSX components*

Table 1 Characteristics of SDN virtualisation projects

Project	References	Domain	Features	Resource virtualised	ISO[a]	Flex[b]	QoS
FlowVisor	[20]	network testbed and simulation, academy research	line rate deployment, OVS	bandwidth, topology, traffic device CPU, forwarding tables	yes	not	not
ADVisor	[21]	enhanced FlowVisor, academy research	MPLS, NOX controller	traffic, address space, topology, bandwidth	yes	not	yes
CloudNaaS	[23]	network as a service, cloud providers	OVS, NOX controller	compute resources, storage resources	yes	not	yes
OpenADN	[24]	data centres, ASPs cloud providers, ISPs	MPLS, label switching 4–7 middleboxes, cross-layer communication	compute resources	yes	yes	not
FlowN	[25]	data centres, cloud providers	container-based virtualisation based on NOX controller	topology, bandwidth device, CPU (cores)	yes	yes	not
EHU-OEF	[26]	real testbeds, academy and research	L2PNV (layer 2 prefix-based network virtualisation), NOX controller	traffic, forwarding tables memory, interfaces	yes	yes	not
VMware NSX	[27]	datacentres, overlay networks ISPs public and private enterprises	OVS, vSphere distributed switch OVS, neutron	compute resources, storage resources	yes	yes	yes

a Isolation guarantee.
b Flexibility (deployment facilities and scalability).

and third creating a new component [26–28]. SDN can be benefited by the contribution of new concepts; it is the case of NFV.

5 Relation between SDN and NFV

As previously mentioned, NFV and SDN aim to break the innovation barrier between proprietary appliances and thus accelerate the introduction of new services. It is important to note that SDN is focused on maximising the network resources and NFV the storage and server resources. Fig. 8 shows the relation between SDN and NFV presented by ONF [29].

SDN maintains its structure based on three layers with OpenFlow protocol as the southbound API. The controller proposed is ODL, which supports not only OpenFlow but also another different southbound APIs. This overview presents open northbound APIs based on OpenStack [30] and NFV. Both technologies could control the compute, storage and networking resources in an efficient way. Finally, the application layer proposes the use of NFs in order to create network applications or new services.

The Network Virtualisation Report of 2014 [31] presents the main trends in this field and the SDN/NFV ecosystems created by the main

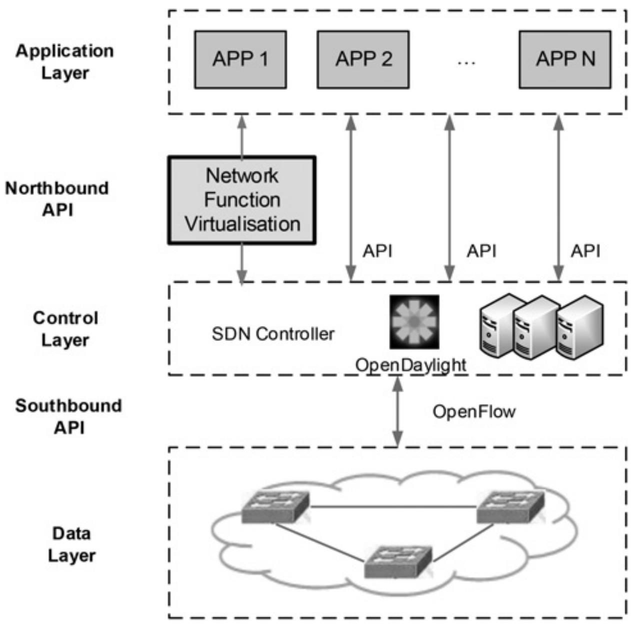

Fig. 8 *SDN/NFV*

vendors, such as Avaya, Big Switch Networks, NEC, Cisco Systems and ConteXtream, among others. For instance, ConteXtream Corporation realises SDN/NFV demonstrations related with virtual evolved packet core (EPC) and subscriber –aware service function chaining, load balancing and monitoring functionalities. In the same way, Huawei and Cisco have shared the latest progress in SDN/NFV deployments.

In such a manner, at the end of the year 2014, ETSI announced the launch the open platform for NFV project (OPNFV) [32], which is based on referential architecture defined by ETSI NFV-ISG [6]. OPNFV will aid to accelerate and promote NFV concept in a new platform based on open source tools (KVM and OVS), cloud technologies (OpenStack and CloudStack) and SDN projects (ODL controller). The research community may develop, deploy and test their own NFs in order to enhance the availability, scalability and reliability of services provided by telecom providers. This initiative is the first open source project within NFV and its first software release is planned for 2015.

The applicability of SDN has gained the attention of industry and research community not only in wired networks but also in mobile networks (software defined wireless network and cellular networks) [2]. In [33] are illustrated the challenges and the potential advantages of SDN in this area such as the virtualisation or the mobility control. In the wireless field, there are real testbeds such as OpenRoad [34] (wireless fidelity (Wifi) and WIMAX nodes). Other approaches allow the deployment of radio base stations in the cloud [35] or the improvement of the core of cellular networks such as the CellSDN project [36].

The challenges and requirements to integrate NFV in mobile networks are presented in [37]. In particular, the case of virtualisation of the EPC (vEPC). It analyses and divides the vEPC components in order to achieve better control and less congestion in data plane. This approach takes into account four segments, the first mobility management entity (MME) with home subscriber server (HSS) (enhance authentication and authorisation process), the second service GPRS support node (SGSN) with home location register (HLR) (support combined systems), the third segment consists of packet data network gateway (PGW), policy and charging enforcement function (PCEF) and serving gateway (SGW) (centralised processing in the data plane) and finally user data respository (UDR), policy and charging rules function (PCRF), online charging system (OCS) and off line charging system (OFCS) (Unify user database, less fragmentation).

The industry and research community go a step further in this direction and propose the combination of SDN and NFV [38]. This combination might provide a standardised environment, where the introduction of new services might be done in less time, with the lowest investment and allowing the easy integration of old and new techniques and technologies. However, there are some concerns related to the widespread use of SDN/NFV in mobile networks. First, there are not consensus in the way that SDN concept is applied in these kinds of networks, if is compared with the OpenFlow wire deployments. The other important concern is that NFV is still in early stage. Although both technologies will not fully integrate in the short term, it is clear that an architecture based on both technologies could be a referential point to telecom operators, as evidenced by ETSI NFV report [6]. This report presents some use cases that combine SDN and NFV [29]. First, VNF-FG applies the SDN concept to control the chaining process of virtual appliances in a dynamic way (add, delete and update), reducing the deployment times from weeks to minutes. Chaining also organises multiple VNFs in sequence in order to deliver a service.

In NFIaaS case, an SP ‘A’ can offer a specific service to the customers in sites where it does not have geographic coverage but the SP ‘B’ does. With a centralised control, the monetisation and the management process is more efficient and both SPs obtain revenues, the first with the service and the second with the infrastructure lease. In the case of the virtualisation of the CPE, the SP has the remote control of the devices and SDN can let a reliable connection. Traditional CPE may be replaced by a VF; in consequence the OpEx is reduced. It will not be necessary that an operator goes to the physical site in where CPE is placed. For its part, some functions such as deep packet inspection (DPI) may be benefited by NFV/SDN. Nowadays, DPI is used in a variety of applications such as gateways, load balancers, policy control and so on. A DPI virtualised function may provide a standard appliance in order to supervise the network and it reduces the capital investment.

Some projects combine both technologies, for instance, EmPOWER [39] shows a testbed composed of 30 nodes in the University of Trento. This facilitates the deployment of SDN/NFV experiments for Wifi networks and it also provides monitoring tools in order to control the energy consumption. EmPOWER can support high-level programming primitives, a set of primitives related with the network status and an interface for the service instantiation such as a web-based control framework or a command line interface. EmPOWER architecture consists of a single master agent (implemented within Floodlight controller) and some agents for access points (APs). The network services run on top of the controller and can use a Floodlight REST interface or an interpreter such as Pyretic. The main objective of EmPOWER is to test energy aware mobility management schemes in real infrastructure. It is also able to shut on/down the APs, depends on the capacity utilisation.

In the same way, Batallé *et al.* [40] designs a routing NF based on OpenFlow protocol and NFV. This paper shows some use cases: migration of IPv4 to IPv6 and inter domain routing. This NF is formed by three components. First, the controller module that analyses and controls the data traffic. The second module contains the routing function that receives instructions from controller. Finally, the third component is in charge of the communication of aforementioned modules. The proposal is based on a modified version of Floodlight controller (NFV module) and the development of the routing function as an OpenNaaS resource.

For its part, Italtel R&D in [41] presents the advances in NFV/SDN specifically the SBC. SBC was fully virtualised in order to control the interconnection between two networks. Virtualised SBC was deployed over the carrier grade Linux operating system of Italtel and it includes proprietary software application. The hypervisor is based on VMware vSphere Hypervisor 5.1. The testbed includes two Cisco servers, in which three VMs run, and each of one with a three different functions.

Cannistra *et al.* [42] present an SDN/NFV testbed that consists of three data centres in a ring topology. It uses OpenFlow protocol, Floodlight controller and distributed overlay virtual Ethernet (DOVE). Some applications were created such as a graphical user interface (Avior), which controls the OpenFlow appliances (monitor statistics and firewall) with mobile devices such as smart phones or tablets. The testbed probes a VM migration with a 75% of server utilisation. The applications continue without interruptions during the migration; however, there were problems with dynamic workloads. The VM migration takes into account some parameters: VM memory size, page dirty rate per second and network bandwidth.

Network operative system (NetOS) [43] combines SDN with NFV in order to obtain a mentality based on software, where any VFs can be added, managed, moved and updated efficiently and in an easy way, no matter what the kind of commodity hardware, the vendor, their place and other variables. It supports OpenFlow protocol and it can interact with the components of NFV architecture. NetOS defines three main components: (i) ‘drivers and devices’ which contains a network abstraction layer which provides an unified southbound API to the physical appliances. (ii) ‘NetOS kernel’ which maintains the whole network state and it has a virtualisation network layer and (iii) ‘user space’ which facilitates the integrations with OSS and controller.

In [44], a reference model that combines SDN with NFV is presented. It proposes a network abstraction model (NAM) that allows for the creation of a single framework. NAM takes into account three requirements: first, it should allow new or existing NFs. Second the framework should be extensible, which requires the accommodation of new SDN/NFV blocks and third NAM should be expressive between the interfaces and the controller. NAM also defines two types of interfaces: configuration and management interfaces. In the same way, Masutani *et al.* [45] present the characteristics that a network node should cover in order to support NFV technology. This project provides the initial prototype of a virtual broadband remote access router (BRAS); implemented with Intel DPDK, OVS and OpenFlow protocol. The use cases cover two kinds of services: Internet connection services and SIP services.

OpenSDNCore [46] is an initiative developed by Fraunhofer FOKUS that creates a prototype implementation to experiment with the capabilities of SDN and NFV technologies. It is aligned with OpenStack and OpenFlow 1.4 in order to support telecom features. Additionally, this prototype integrates with other projects such as OpenEPC and OpenIMSCore.

Furthermore, UNIFY [47] provides an environment to deploy NFs based on the combination of cloud computing and virtualisation. The project intends to design a universal hardware node to deploy these functions in an open environment while decrease the deployment and management costs. UNIFY supports a variety of technologies such as OpenFlow and NFV and is focused on three areas: infrastructure virtualisation, flexible service chaining and finally network service chain invocation. Moreover, T-NOVA project [48] aims to design and implement an NFV/SDN framework to deploy VNFs. This NFs are developed in software and eliminate the need of acquire, install and maintain specialised hardware. The framework will create a marketplace wherein the developers could offer their NFs while allowing customers use these applications or services.

Table 2 shows the initial deployments and tests in order to probe the benefits that telecom industry could obtain from the combination of SDN and NFV.

Current projects try to tackle current needs in telecom environments, especially in fifth generation (5G) networks. An example of which is Mobile and wireless communications Enablers for the Twenty–twenty Information Society (METIS) [49] or Connectivity management for eneRgy Optimised Wireless Dense (CROWD) projects [50]. CROWD networks propose an architecture to enhance the performance on very dense and heterogeneous wireless networks (Dense Nets). CROWD provides dynamic controller placement, dynamic backhaul reconfiguration, energy and MAC optimisation and ensures user quality of experience. For this purpose, this project uses OpenFlow to control and manage the nodes (eNBs and Wifi AP).

METIS [51] aims to break the barriers within 5G mobile and wireless communication system and lay the bases for the standardisation process. It is focused on current and future needs of mobile networks such as low latency, high data rate, low energy consumption, ubiquitous communication, massive machine communication, ultra dense networks and better QoS, among

Table 2 SDN/NFV projects

Project	References	Domain	Description	Features
ConteXtream	[31]	SDN/NFV VN environment	it presents a solution that implements network virtualisation for carrier providers	vSphere environment, multi-hypervisor support, any cloud platform Virtual eXtensible Local Area Network (VxLAN), stateless transport tunneling (STT), Generic Routing Encapsulation (GRE) ODL
EmPOWER	[39]	SDN/NFV testbed	testbed within wireless domain, composed by 30 nodes (Trento). Use cases include resource utilisation and dynamic handover Wifi	FlowVisor, Floodlight Pyretic, OVS, light virtual AP Energino, Click Modular Router
Batallé *et al.*	[40]	routing function virtualisation	it designs a virtual routing function. This function is tested in Mininet simulator	Floodlight, OpenNaaS, Mininet, OVS
Monteleone and Paglierani	[41]	SBC based on SDN/NFV	it proposes a virtualised NF of SBC deployed in Italtel R&D laboratory	carrier grade Linux, operative system of Italtel, VMware vSphere, Hypervisor Cisco Servers
Cannistra *et al.*	[42]	SDN/NFV testbed	three data centres connected in a ring topology (125 km). Use cases include probes to VM migration	OpenFlow, Floodlight, DOVE
NetOS	[43]	SDN/NFV platform	it presents an initial approach in order to provide an SDN/NFV environment (NOS)	OpenFlow, simple network management protocol (SNMP) not specify more tools
Haleplidis *et al.*	[44]	SDN/NFV reference model	it proposes a single framework that combines SDN and NFV	it takes into account: ForCES and Click Modular Router
Masutani *et al.*	[45]	virtual BRAS	it presents an initial design of a virtual BRAS using Intel DPDK	physical or cloud networks, OVS, Linux new application programming interface (NAPI) model. kernel-based virtual machine (KVM) hypervisor
OpenSDN Core	[46]	prototype of SDN/NFV	environment that allows the experimentation with SDN and NFV. Some used cases could include vEPC, vIMS and vRAN	OpenFlow 1.4 OpenStack GTP GRE
Unify	[47]	architecture to NFs	it proposes an architecture to flexible creation and deployment of NFs	OpenFlow, Intel, data plane development kit (DPDK) OpenStack
T-Nova	[48]	NFaaS over virtualised infrastructure	it provides a framework that allows the easy deployment of NFs. It provides an NF marketplace	OpenFlow, OpenStack

others. To cover these needs, METIS might progress on the following areas: radio-links, multi-node/multi-antenna technologies, multi-layer and multi-RAT networks and spectrum usage. METIS also creates new opportunities in five scenarios, for which establish a set of requirements and key performance indicators. These scenarios are:

- 'Amazingly fast' (provide very high data-rates without connectivity delays).
- 'Great service in a crowd' (provide QoE in crowd places such a massive events).
- 'Best experience follows you' (provide services to users on movement such as cars).
- 'Super real-time and reliable connections' (provide reliability and low latency in new applications, such as M2M).
- 'Ubiquitous things communicating' (give services to machines devices).

The advances obtained from METIS could be enhanced with the integration of an SDN/NFV approach. One of the main concerns is the capacity to deploy new services no matter the SP or the location, characteristic that might be covered by NFV. Moreover, SDN has showed the improvement of the network management.

Even though METIS project could be a referential point in 5G technologies, SDN/NFV not only addresses this field but also other areas such as cloud computing and data centres, among others [52]. Indeed, advances in 5G networks have become a prioritised topic in the research agenda, as evidenced the European Commission with the call 'Advanced 5G Network Infrastructure for the Future Internet' [53]. This aims to encourage competitiveness of the network operators within 4G/5G networks by means of novel technologies such as SDN and NFV.

6 Discussion and conclusion

Today, the number of services is growing faster than ever before, and customers want to use these services almost immediately. SDN/NFV could shorten the lifecycle in the development and innovation of new applications. On one hand, the research community aims to accelerate the testing process in order to introduce a new technology. On the other hand, SPs and networking enterprises aim to accelerate the TTM new services and reduce the expenditures, in consequence the final customer could obtain enhanced services.

The lifecycle of network devices includes two important phases: first, the deployment and allocation of resources and second the control and management of the devices. SDN allows a better control of the network, giving programmability capacities and improved network management. SDN also encourages the innovation and facilitates the automation of the network. For its part, NFV increases the network flexibility and reduces the complexity in the deployment of traditional NFs. The new VFs could be provided as a software component.

NFV and SDN are independent and complementary at the same time but the synergy of both technologies could provide an open environment to foster the innovation and decrease the capital and OpEx related to new infrastructure and services, all of this with major control and automation of network resources. For instance, the combination of SDN with NFV enables the dynamic service chaining through its centralised controller. The packet processing can be done as an NF and the controller can programme the flow table in the switch based on subscriber awareness of a given flow. Other application case is related to the deployment of VNs, NFV is responsible of the creation and SDN of the management of them.

There are some challenges to overcome due to the fact that SDN and NFV are involved in a maturing process. The main challenges are related to the migration of VFs between different sites and vendors (mobility and portability), NF management, service continuity and authentication and authorisation methods because of the security problems that introduce new layers. One of the most important challenges is the need to upgrade the current software of the network devices or in the worst case, the change of the entire infrastructure. All stakeholders should create solutions that allow the easy expansion and management of the network devices and applications. Furthermore, it is important to design new reliability and redundancy schemes in order to guarantee the network performance. First, methods to schedule the resources, second the creation of redundancy mechanisms to recover the network status after a failure and third mechanisms for the coexistence with legacy networks. It is important to note that

IET Netw., 2015, Vol. 4, Iss. 5, pp. 255–263

mobile and sensor networks need a better control and manage of data traffic. CROWD and METIS provide some ideas to enhance these kinds of networks; however, there are some problems related to the mobility and resource constraints.

The networking industry is changing towards a software-driven model deployed on standard hardware. This model could align with the current services and products, allowing the customisation of the services and fast scalability, hence accelerating the innovation and reduce the expenditures. This paper presents a chronology of network virtualisation and the description of two key technologies, SDN and NFV. It presents the current projects and applications of SDN in virtualisation field. It also describes the NFV concept and its use cases, as well as the relation between these technologies. This paper is intended to explain the benefits that we could obtain from the combination between SDN and NFV and the challenges embracing this topic.

7 Acknowledgment

This work has been partially supported by the European Commission Horizon 2020 Programme under grant agreement number H2020-ICT-2014-2/671672 - SELFNET (Framework for Self-Organized Network Management in Virtualized and Software Defined Networks). Lorena Isabel Barona López and Ángel Leonardo Valdivieso Caraguay are supported by the Secretaría Nacional de Educación Superior, Ciencia, Tecnología e Innovación (SENESCYT), Quito - Ecuador.

8 References

1 Lara, A., Kolasani, A., Ramamurthy, B.: 'Network innovation using OpenFlow: a survey', *IEEE Commun. Surv. Tutor.*, 2013, **16**, (1), pp. 493–512
2 Valdivieso Caraguay, Á.L., Benito Peral, A., Barona López, L.I., García Villalba, L.J.: 'SDN: evolution and opportunities in the development IoT applications', *Int. J. Distrib. Sens. Netw.*, 2014, **2014**, pp. 1–10
3 Feamster, N., Rexford, J., Zegura, E.: 'The road to SDN: an intellectual history of programmable networks', *ACM SIGCOMM Comput. Commun. Rev.*, 2014, **44**, (2), pp. 87–98
4 McKeown, N., Anderson, T., Balakrishnan, H., *et al.*: 'OpenFlow: enabling innovation in campus networks', *ACM SIGCOMM Comput. Commun. Rev.*, 2008, **38**, pp. 69–74
5 Valdivieso Caraguay, Á.L., Benito Peral, A., Barona López, L.I., García Villalba, L.J.: 'Evolution and challenges of software defined networking'. Workshop on Software Defined Networks for Future Networks and Services, 2013, pp. 47–55
6 ETSI Industry Specification Group (ISG): 'Network function virtualization (NFV) white paper'. SDN and OpenFlow World Congress, Frankfurt-Germany, 2013, pp. 1–16
7 Jain, R., Paul, S.: 'Network virtualization and software defined networking for cloud computing: a survey', *IEEE Commun. Mag.*, 2013, **51**, (11), pp. 24–31
8 Chowdhury, N.M., Boutaba, R.: 'A survey of network virtualization', *Comput. Netw.*, 2010, **54**, (5), pp. 862–876
9 Pfaff, B., Pettit, J., Amidon, K., Casado, M.: 'Extending networking into the virtualization layer'. Proc. of ACM SIGCOMM HotNets, ACM, 2009, pp. 1–6
10 Luo, M.Y., Chen, J., Mambretti, J., *et al.*: 'Network virtualization implementation over global research production networks', *J. Internet Technol.*, 2010, **14**, (7), pp. 1061–1072
11 ETSI Industry Specification Group (ISG): 'Network function virtualization (NFV) architectural framework', October 2013, pp. 1–21
12 ETSI Industry Specification Group (ISG): 'Network function virtualization (NFV) use cases', October 2013, pp. 1–50
13 Consortium, O.S.: 'OpenFlow Switch Specification v.1.3.4', 2014, pp. 1–171
14 'OpenDaylight: Technical Overview'. Available at http://www.opendaylight.org/project/technical-overview, accessed February 2015
15 Voellmy, A., Kim, H., Feamster, N.: 'Procera: a language for high-level reactive network control'. Proc. of the First Workshop on Hot Topics in Software Defined Networks, 2012, pp. 43–48
16 Foster, N., Guha, A., Reitblatt, M., *et al.*: 'Languages for software defined networks', *IEEE Commun. Mag.*, 2013, **51**, (2), pp. 128–134
17 Lee, S.W., Han, S.W., Kim, J.W., *et al.*: 'FIRST: Korean future Internet testbed for media-oriented service overlay network architecture', *J. Internet Technol.*, 2010, **11**, (4), pp. 553–559
18 Kobayashi, M., Seetharaman, S., Parulkar, G., *et al.*: 'Maturing of OpenFlow and software-defined networking through deployments', *Comput. Netw.*, 2014, **61**, pp. 151–175
19 Courtney, M.: 'Service as a standard [software-defined networking concept]', *IET Eng. Technol.*, 2012, **7**, (11), pp. 64–67
20 Sherwood, R., Gibb, G., Yap, K.K., *et al.*: 'Can the production network be the testbed?'. OSDI, 2010, vol. 10, pp. 1–6
21 Salvadori, E., Corin, R.D., Broglio, A., Gerola, M.: 'Generalizing virtual network topologies in OpenFlow-based networks'. Global Telecommunications Conf., 2011, pp. 1–6
22 Sonkoly, B., Gulyás, A., Németh, F., *et al.*: 'OpenFlow virtualization framework with advanced capabilities'. Software Defined Networking (EWSDN), 2012, pp. 18–23
23 Benson, T., Akella, A., Shaikh, A., Sahu, S.: 'CloudNaaS: a cloud networking platform for enterprise applications'. Proc. of the Second ACM Symp. on Cloud Computing, NY, USA, 2011, pp. 1–13
24 Paul, S., Jain, R.: 'OpenADN: mobile apps on global clouds using OpenFlow and software defined networking'. Globecom Workshops, 2012, pp. 719–723
25 Drutskoy, D., Keller, E., Rexford, J.: 'Scalable network virtualization in software-defined networks', *IEEE Internet Comput.*, 2013, **17**, (2), pp. 20–27
26 Matias, J., Mendiola, A., Toledo, N., Tornero, B., Jacob, E.: 'The EHU-OEF: an OpenFlow-based layer-2 experimental facility', *Comput. Netw.*, 2014, **63**, pp. 101–127
27 'VMware® NSX Network Virtualization Design Guide'. Available at https://www.vmware.com/products/nsx/, accessed February 2015
28 'Programmable Flow'. Available at https://www.necam.com/sdn, accessed February 2015
29 Open Networking Foundation (ONF): 'OpenFlow-enable SDN and network function virtualization', February 2014, pp. 1–12
30 'OpenStack: The Open Source Cloud Operating System'. Available at http://www.openstack.org, accessed February 2015
31 'Network Virtualization Report 2014 Edition', SDNCentral. Available at http://www.sdncentral.com, accessed February 2015
32 'Open Platform for NFV Project (OPNFV)'. Available at https://www.opnfv.org/, accessed February 2015
33 Bernardos, C.J., De La Oliva, A., Serrano, P., *et al.*: 'An architecture for software defined wireless networking', *IEEE Wirel. Commun.*, 2014, **21**, (3), pp. 52–61
34 Yap, K.K., Kobayashi, M., Sherwood, R., *et al.*: 'Empowering research in mobile networks', *ACM SIGCOMM Comput. Commun. Rev.*, 2010, **40**, (1), pp. 125–126
35 Haberland, B., Derakhshan, F., Grob-Lipski, H., *et al.*: 'Radio base stations in the cloud', *Bell Labs Tech. J.*, 2013, **18**, (1), pp. 129–152
36 Li, L.E., Mao, Z.M., Rexford, J.: 'CellSDN: software-defined cellular networks'. Technical Report, Princeton University, 2012, pp. 1–6
37 Hawilo, H., Shami, A., Mirahmadi, M., *et al.*: 'NFV: state of the art, challenges, and implementation in next generation mobile networks (vEPC)', *IEEE Netw.*, 2014, **28**, (6), pp. 18–26
38 Courtney, M.: 'Stack them high [software-defined networking]', *IET Eng. Technol.*, 2014, **9**, (11), pp. 80–83
39 Riggio, R., Rasheed, T., Granelli, F.: 'EmPOWER: a testbed for network function virtualization research and experimentation'. Future Networks and Services (SDN4FNS), 2013, pp. 1–5
40 Batalle, J., Ferrer Riera, J., Escalona, E., *et al.*: 'On the implementation of NFV over an OpenFlow infrastructure: routing function virtualization'. Future Networks and Services (SDN4FNS), 2013, pp. 1–6
41 Monteleone, G., Paglierani, P.: 'Session border controller virtualization towards service-defined networks based on NFV and SDN'. Future Networks and Services (SDN4FNS), 2013, pp. 1–7
42 Cannistra, R., Carle, B., Johnson, R., *et al.*: 'Enabling autonomic provisioning in SDN cloud networks with NFV service chaining'. Optical Fiber Communication Conf., 2014, pp. 1–3
43 López, V., González de Dios, O., Fuentes, B., *et al.*: 'Towards a network operating system'. Optical Fiber Communication Conf., 2014, vol. 31, no. 6, pp. 1–3
44 Haleplidis, E., Salim, J.H., Denazis, S., Koufopavlou, O.: 'Towards a network abstraction model for SDN', *J. Netw. Syst. Manage.*, 2014, **23**, (7), pp. 1–19
45 Masutani, H., Nakajima, Y., Kinoshita, T., *et al.*: 'Requirements and design of flexible NFV network infrastructure node leveraging SDN/OpenFlow'. Optical Network Design and Modeling, 2014 Int. Conf. on, 2014, pp. 258–263
46 'OpenSDNCore'. Available at http://www.opensdncore.org/index.html, accessed February 2015
47 'Unify Project'. Available at http://www.fp7-unify.eu/, accessed February 2015
48 'T-Nova Project'. Available at http://www.t-nova.eu/, accessed February 2015
49 'METIS Project'. Available at https://www.metis2020.com, accessed February 2015
50 'CROWD Project'. Available at http://www.ict-crowd.eu, accessed February 2015
51 Osseiran, A., Boccardi, F., Braun, V., *et al.*: 'Scenarios for 5G mobile and wireless communications: the vision of the METIS project', *IEEE Commun. Mag.*, 2014, **52**, (5), pp. 26–35
52 Wang, J., Yan, Y., Dittmann, L.: 'Design of energy efficient optical networks with software enabled integrated control plane', *IET Netw.*, 2015, **14**, (1), pp. 30–36
53 'ICT-14–2014 Call', European Commission'. Available at http://www.ec.europa.eu/research/participants/portal/desktop/en/opportunities/h2020/topics/77-ict-14-2014.html, accessed February 2015