Special Issue

Wouter Tavernier*, Bram Naudts, Didier Colle, Mario Pickavet, and Sofie Verbrugge

Can open-source projects (re-)shape the SDN/NFV-driven telecommunication market?

DOI 10.1515/itit-2015-0026
Received June 30, 2015; revised July 17, 2015; accepted July 20, 2015

Abstract: Telecom network operators face rapidly changing business needs. Due to their dependence on long product cycles they lack the ability to quickly respond to changing user demands. To spur innovation and stay competitive, network operators are investigating technological solutions with a proven track record in other application domains such as open source software projects. Open source software enables parties to learn, use, or contribute to technology from which they were previously excluded. OSS has reshaped many application areas including the landscape of operating systems and consumer software. The paradigm shift in telecommunication systems towards Software-Defined Networking introduces possibilities to benefit from open source projects. Implementing the control part of networks in software enables speedier adaption and innovation, and less dependencies on legacy protocols or algorithms hard-coded in the control part of network devices. The recently proposed concept of Network Function Virtualization pushes the softwarization of telecommunication functionalities even further down to the data plane. Within the NFV paradigm, functionality which was previously reserved for dedicated hardware implementations can now be implemented in software and deployed on generic Commercial Off-The Shelf (COTS) hardware. This paper provides an overview of existing open source initiatives for SDN/NFV-based network architectures, involving infrastructure to orchestration-related functionality. It situates them in a business process context and identifies the pros and cons for the market in general, as well as for individual actors.

Keywords: Software-Defined Networking, Network Function Virtualization, open-source software.

***Corresponding author: Wouter Tavernier,** Ghent University – iMinds, Department of Information Technology, Gent, Belgium, e-mail: wouter.tavernier@intec.ugent.be
Bram Naudts, Didier Colle, Mario Pickavet, Sofie Verbrugge: Ghent University – iMinds, Department of Information Technology, Gent, Belgium

ACM CCS: Networks → Network services → Programmable networks

1 Introduction

The increased softwarization and programmability have changed the telecom landscape significantly. Whereas communication networks previously were under the rigid control of vendor-specific solutions or tightly standardized protocols, Software-Defined Networking (SDN) enabled network operators to configure the control of their networks through their-own custom software. Network Function Virtualization (NFV) pushes this even further by making it possible to code data plane behavior in software, enabling it to run on general purpose server hardware rather than on expensive vendor-controlled hardware platforms.

This openness and control provides unseen perspective in avoiding vendor-based monopolies or vendor lock-in for network providers and operators. However, if not steered carefully, a new threat could be that the software-driven telecom landscape might again fall into the hands of a limited number of players with strong software skills and/or departments. A similar situation occurred in the domain of operating system software for pcs, which was dominated for years by a few big players such as Microsoft or Apple. The introduction of open source (OS) software (OSS) projects, however has made the OSS market more democratic and innovative, enabling billions of end-users to use, extend and research, for example Linux-based OSes for professional, research or educative purposes. Through the increasing importance of software, similar conditions and opportunities might arise for OSS in telecom networks.

The goal of this paper is to clarify the role of existing and potential open source projects within this context of SDN and NFV-driven communication networks. We will investigate the following questions. Which projects have the potential to impact the industry, who are the prominent players, what are their motivations, what are the opportunities for novel parties or projects? In order to tackle these in a structured way, Section 2 introduces the architectural

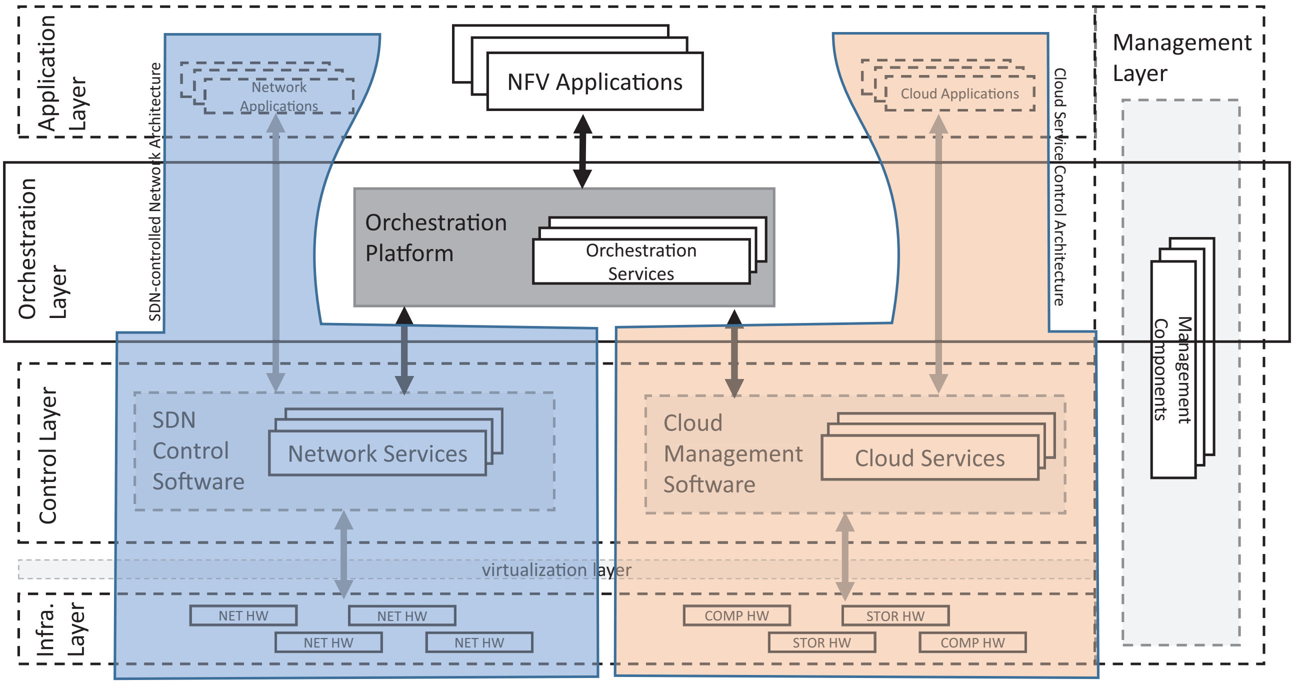

Figure 1: Network and Cloud control architectures.

context of telecom and cloud networks and introduces important SDN- and NFV-related concepts. In Section 3 we investigate the role of OSS projects by providing an overview of existing projects, identifying their main contributions and dependencies, as well as the resulting gaps within the SDN/NFV open-source software landscape. In addition, an analysis is made of the impact of the range of license policies these projects might have and the role that standards development organizations can play. At last, the paper concludes with some lessons learned and potential future topics in Section 4.

2 SDN/NFV network architecture

This section shortly sketches the main functional components and layers in the SDN control architecture of a modern telecom network supporting NFV in relationship to the most important standardization bodies, in order to provide context for the SDN/NFV open source projects which will be analyzed later. Particular attention will be given to the concept of Service Function Chaining, as it plays a crucial role in linking services to the virtualization of network functions.

2.1 Overarching architecture

Modern network architectures are structured into multiple functional layers of smaller components. This modular approach reduces complexity, enhances component reusability, and enables multiple migration paths towards future architectures. Recent softwarization and virtualization tendencies have only further accumulated the decomposition of functional components and layers within architectures. By decoupling the forwarding- from control functionality, SDN transformed previously monolithic switches/routers into two multiple independent components. Server- and network virtualization mechanisms on their turn introduced additional functional splits which isolate data plane functionality of its underlying hardware platform.

NFV brings together two areas: i) the (software-driven[1]) control of communication networks, and ii) the control of cloud (service) platforms. Both control architecture are depicted in Figure 1. The first (marked in blue) is in charge of controlling network of switching and routing equipment, the second (in orange) is in charge of creating and exposing cloud networks, i. e. a network of re-usable

1 In the context of this paper we focus on SDN-controlled networks, although traditional distributed routing protocols could also be considered as the control layer of communication networks.

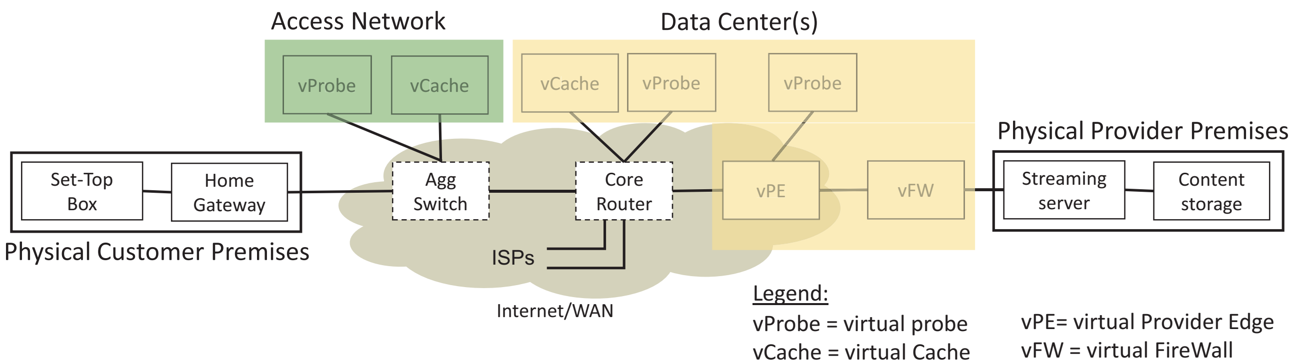

Figure 2: Service Function Chain for multimedia delivery.

compute and storage servers for the purpose of, e. g., building web services. The control architecture of both domains follows a roughly similar 3-layered approach, as depicted in Figure 1. At the lowest layer, infrastructure resources form the physical foundation on top of which services are provided. Communication networks rely on network hardware such as switches and routers, Cloud infrastructures rely on (interconnected) compute and storage hardware (servers). A second layer, the Control Layer interconnects the components of the infrastructure layer via their north-bound interface (e. g., OpenFlow for network control) in order to provide control-level services such as topology management or datastore services. Virtualization technology introduces a sublayer in between the infrastructure layer and the control layer, either at the device level, enabling one device to be segmented in multiple logical devices (e. g., in the case of server virtualization using Xen or KVM virtualization technology, see Figure 4, or at the level of multiple devices by horizontally segmenting or slicing an entire collection of devices (e. g., in the case of SDN-networks using FlowVisor [35]). At the highest layer, components of the application layer build further on control layer services to program client applications. A traffic engineering application might be defined on top of the SDN-control layer, while a Hadoop cluster might be an application on top of the cloud platform. Orthogonal to the horizontal layers, management functionality might be required to configure any of the components at the infrastructure, control or application layer for example to ensure policies or security-related options.

2.2 Service Function Chains

Telecom operator services such as VPN-, telephone- or content services are usually composed of a combination of packet-processing L2-L7 network functions (NF) or service functions, e. g., firewalling, intrusion prevention or server load balancing. The merit of NFV is to virtualize these NFs enabling to deploy them on multi-purpose hardware (servers). Service Function Chaining emerged as a way to describe the traffic steering in between the necessary NFs building the service. Traffic steering could refer to a simple sequence of involved functions, or chain, but also might include a complex mesh-like interaction between these components, requiring a forwarding overlay between the network functions (leading to the term Service Graph, or VNF Forwarding Graph).

To make the concept more concrete, we may consider a Service Function Chain (SFC) for monitored multimedia delivery as depicted in Figure 2. The setup enables a CDN to provide high-quality video streaming to the customer from various locations. Monitoring probes (monitoring NFs) are deployed to monitor performance. The Content Storage NF provides a video catalogue to the customer, and the Streaming Server NF is in charge of indexing the storage, encoding the video in the necessary formats and attaching Digital Rights Management (DRM) protection to the content. Dynamic content is mostly served from the content-originating servers, while static content might be cached at the edge. The vPE (virtual Provider Edge) function handles multimedia requests of customers and thus acts as a CDN request router. In order to minimize delay, the CDN caches are distributed in the ISP networks close to the edge (in the access network). An HTTP Client or dedicated Streaming Client on the Set-Top Box will enable to consume video stream at the end-user premises.

SFCs pose challenges to the communication network interconnecting NFs as well as to the cloud infrastructure on which individual NFs are deployed. In the considered example, caches should only be deployed on server resources which are close to the customer (e. g., in the access network) or caches which are reachable with minimal delay and maximal bandwidth across geographically spread datacenters. As a result, to deploy telecom services as Service Function Chains, a concerted control of network and server resources is required. To adequately provision SFCs, orchestration functionality is needed, having a complete view on available network as well as on compute and storage resources. Such orchestration components are able to make an informed decision on which infrastructure should be used for which NF or link within the SFC. The provisioning process itself can then be further delegated to the already existing network and cloud control platforms. As illustrated in Figure 1, the resulting Orchestration Layer is placed on top of the control layer, and thus is responsible for: i) translating SFC requirements into resource requirements in terms of network and server resources, ii) mapping required resources to infrastructure, and iii) maintenance of the service life-cycle for instantiation, re-provisioning and tear-down of services and associated VNFs.

2.3 ETSI ISG NFV MANO architecture

The virtualization of data plane functionality and associated network and control infrastructures are studied within the Industry Study Group Network Function Virtualisation (ISG NFV) organized by ETSI [9]. The ETSI group has proposed a fine-grained NFV MAnagement and Orchestration (MANO) architecture for tackling the control of SFCs, as illustrated in Figure 3. In this architecture, orchestration functionality is incorporated in the Network Function Virtualization Orchestrator (NFVO) which interfaces to a set of databases: i) the Network Service catalogue, ii) the VNF catalogue, iii) NFV instances database, and iv) the NFVI resources database. The first two store templates and provisioning scripts of Network Services and Virtual Network Functions, while the latter two store information about already deployed VNFs and associated infrastructure resources (network, compute and storage hardware). The NFVO is responsible for all life-cycle management incl. instantiation, scale-in/out and termination of Virtual Network Function Forwarding Graphs (referring the generalized concept of SFCs). The NFVO interfaces with Virtualized Infrastructure Manager and the VNF Manager. The first refers to the combination of network and server control systems, while the second refers to the management of individual VNFs, e. g. with respect to starting or stopping the instance. Deployed VNFs might have Element Managers which are responsible for FCAPS-related management of instantiated VNFs.

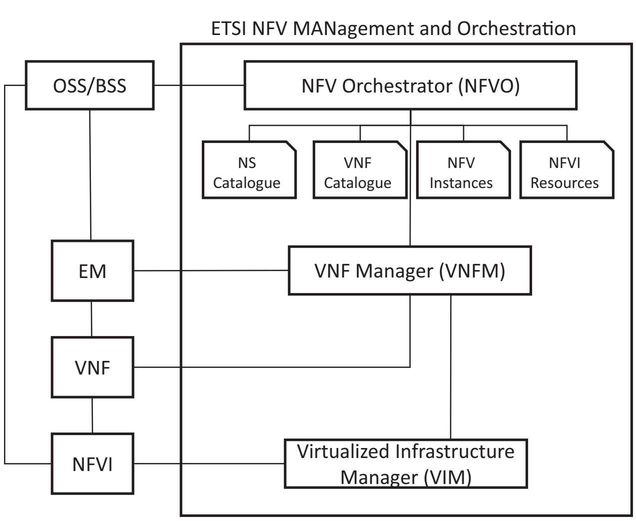

Figure 3: ETSI NFV MANO architecture.

Technical architectures are specifications that require a translation into real life business processes such as those for service provisioning and -management. To maximize the efficiency of these processes, the employees of the network operations centers use software that is able to automate part of their workflows. The next section describes the shift from the conventional situation in which that software is closely tied to the hardware towards a modular system in which both can evolve independently due to the advances in virtualization technology as described above.

3 The role of open source software

For decades, network operators have ordered network equipment from vendors which have delivered specialized solutions. The separation of responsibilities allows operators to focus on service provisioning and management while the development of network solutions and at least part of the legal liability for equipment failures remains with the vendors. This separation has been proven successful and both network operators and network equipment vendors have structured their organizations accordingly. E. g., vendors have developed hardware-, protocol- and software engineering skills for the development of specialized networking equipment, while network operators train their staff in the configuration of the network devices in order to provision and manage services. Over the years, the vendor market has been consolidated with as result that a few large players dominate the market. It has, however, also created a dependence on equipment vendors as the network operators rely on them to provide innovative network solutions that fit their needs.

The move towards softwarization and programmable networks that is proposed by SDN- and NFV advocates has as goal to reshape that relationship. The previously described high-level architecture makes, by providing standardized interfaces, an independent evolution of each part possible and allows for developers to start creating their own solutions. Network equipment vendors, however, typically wish to sell the code they develop. As such they sharply restrict access to the source code of their software products to firm employees and contractors. As a consequence only insiders have the information to modify and improve that proprietary code further. In sharp contrast, the source code of OSS is accessible to all and for free[2]. As such, anyone with the proper programming skills and motivations can use and modify any OSS written by anyone.

Multiple actors are exploring potential positions with respect to OSS in order to benefit from the promises of SDN and NFV. Network operators might reduce costs because of their reduced dependency on vendor equipment and associated software solutions (license costs), as they get new opportunities with increased flexibility in controlling and managing their networks and services. Server- and chip manufacturers potentially face new markets by the possibility to sell servers or chips enabling to run functionality (VNFs) which was previously reserved for dedicated vendor-branded/flavored hardware. Software companies may develop new network control solutions for network operators or software infrastructure supporting efficient VNF development or deployment. Academia might investigate new control frameworks as well as mechanisms in order to optimize performance of softwarized network functions. Last but not least, network equipment vendors will try to protect their existing market-share by proposing themselves SDN/NFV-based solutions to their customers (potentially by re-branding existing solutions).

2 OSS or free software are two terms for the same thing. The term open implies that the source code is visible or available but the terminology should be interpreted wider as next to mere inspection access it also conveys to recipients the perpetual right to fork covered code and use it without additional fees [20]. Free does not refer to the monetary aspect, it refers to the license under which the software is distributed which guarantees a certain, specific set of freedoms as conceived by Richard Stallman in his definition [11]. It refers to the freedoms granted to its receiver, which are: (1) freedom to run the program in any place, for any purpose and forever, (2) freedom to study how it works and to adapt it, (3) freedom to redistribute copies and (4) freedom to improve the program and to release improvements to the public.

3.1 Overview of open source software projects

In order to investigate the impact of OSS on the SDN/NFV-driven telecom market, we examine how well the components of the sketched architecture are covered by existing OSS projects and how it relates to various actors. The table in Figure 4 gives an overview of some existing OSS-projects with respect to the architectural role, their license type, main drivers of the projects, as well as their activity in terms of number of GIT repository commits during the last year[3], and the number of unique developers (which is the sorting criterion of the table, as it gives an indication of the actual degree of support).

NFV is mainly driven by the potential of running network functions on low-cost multi-purpose server hardware. OSS such as DPDK and netmap focusing on acceleration of the packet processing on commodity hardware may be interpreted as incentives from Intel to stimulate the market in developing efficient VNFs, increasing the likelihood of selling x86 hardware. In order to optimally deploy and isolate given VNFs on a single server, Virtual Machine (VM) technologies such as Xen and KVM or software container-based technologies such as Docker[4] are heavily promoted by Linux OS software companies, putting Linux in pole position for NFV deployments. On the side of network virtualization, OpenVirtex has emerged as the de facto OSS standard for virtualizing SDNs, succeeding FlowVisor [35]. This enables network infrastructure providers to provide their services to multiple customers.

SDN and Cloud control software has been a very active area last years. SDN control solutions initially emerged from startups closely related to academia, such as NOX, POX, Floodlight, as well as programming frameworks enabling higher-level abstractions such as the Frenetic-related of languages (e. g., Frenetic, Pyretic, Merlin). However in a later phase, multiple industrially-supported OSS projects were started such as Ryu, ONOS, and in particular OpenDaylight, focusing on the unique requirements of network operators (i. e. five-nines availability, highly tested, scalability, reliability and complex networking). OpenDaylight originated from the Linux Foundation and is pushed by a very large consortium of telecom- and hardware vendors, operators and software companies, each of

3 reflecting the number of source code changes

4 Very recently the Open Container Project (opencontainers.org) has been proposed as the future standard containers by major cloud providers and software industry

Name	License	Scope	Description	Driver(s)	# recent commits	# authors	reference
Application Layer							
Click Modular Router	MIT/BSD-like	VNF	Development framework for software routers and packet-processing logic on Linux.	UCLA,MIT	127	99	[3]
linux firewalling	GPLv2	VNF	Linux kernel components for packet filtering/mangling + network addr/port translation.	netfilter			[17]
opentosca	Apachev2	service	Model tool and service orchestration tool for cloud-focused TOSCA services	Uni. Stuttgart	72	6	[28]
Open vSwitch	Apachev2	VNF	Production quality, multilayer, software OF switch	VMware (Nicira)			[30]
Orchestration Layer (NFVO)							
juju	GPLv3	cloud	Service orchestration and management tool supporting scaling	Canonical (Ubuntu)	515	69	[13]
openMANO	Apachev2	NFV	Reference implementation of reference architecture for ETSI NFV MANO	Telefonica	100	6	[25]
Cloudify	Apachev2	cloud	TOSCA-based cloud orchestration software platform.	Cloudify, GigaSpaces		58	[4]
OPNFV	Apachev2	NFV	Carrier-grade, integrated, platform to accelerate the introduction of new NFV products and services.	Top industry in NFV	252	9	[31]
opencontrail	Apachev2	NFV	NFV orchestrator and SDN controller.	Juniper, NTT, AT&T, Mirantis	3820	126	[22]
opencloud	Apachev2	cloud+NFV	Research platform for IaaS, cloud and nfv service orchestration	ON.Lab, PlanetLab, Internet2	1042	75	[23]
Control Layer (VIM)							
frenetic+pyretic	GPLv4	network	SDN programming language family	Uni. Princeton, Uni. Cornell			
OpenStack	Apachev2	compute	Cloud computing software/IaaS platform	Rackspace, Red Hat, SUSE, HP, Rackspace, AT&T, IBM, Intel	27575	915	[27]
CloudStack	Apachev2	compute	Cloud computing software for creating, managing, and deploying infrastructure cloud services	Apache Software Foundation	3995	347	[1]
NOX/POX	Apachev2	network	SDN control platform written in C/C++ and Python and corresponding API's.	VMware (Nicira)		25	[19]
OpenDaylight	EPLv1	network	SDN control platform with many plugins supporting cloud platforms.	Brocade, Cisco, Citrix, Dell, Ericsson, HP, Intel, Red Hat	2939	158	[24]
Ryu	Apachev2	network	SDN control platform written in Python	NTT	431	79	[34]
Floodlight	Apachev2	network	SDN control platform written in Java		282	69	[32]
ONOS	Apachev2	network	Carrier-grade, distributed SDN control platform.	ON.lab, AT&T, NTT, SK Telecom, Ciena, Cisco, Ericsson, Fujitsu, Huawei, Intel, NEC	3731	51	[21]
Management Layer							
opennms	AGPL	network	enterprise grade network monitoring and network management platform	OpenNMS, OGP	3666	105	[26]
puppet	Apachev2	cloud	Configuration management utility for distributed deployment of software.	Puppet labs			[33]
chef	Apachev2	cloud	Configuration management utility for distributed deployment of software.	Chef			[2]
Virtualization layer (NFVI)							
FlowVisor	None	network	Network hypervisor creating rich "slices" of network resources and delegates control of each slice to a different controller	ON.Lab		7	[10]
OpenVirtex	Apachev2	network	Network hypervisor enabling full virtualization of SDN networks	ON.Lab			[29]
KVM	GPLv2 o.a.	compute	virtualization infrastructure for the Linux kernel that turns it into a hypervisor	Red Hat			[16]
Xen	GPLv2	compute	hypervisor using a microkernel design, providing services that allow multiple computer operating systems to execute on the samecomputer hardware concurrently	Citrix, Linux Foundation	910	23	[36]
Docker	Apachev2	compute	project that automates the deployment of applications inside software containers, by providing an additional layer of abstraction and automation of operating-system-level virtualization on Linux	Docker, Canonical, Fedora, OpenStack, Red Hat	7305	1041	[7]
Infrastructure layer							
netmap	BSDv3	network	Framework for efficient line-rate raw packet I/O from user space in Linux and FreeBSD.	Uni. di Pisa			[18]
dpdk	BSDv3	network	Set of data plane Linux kernel libraries and network interface controller drivers for fast packet processing on x86 hardware.	Intel, 6WIND	1509	128	[8]

Figure 4: Selection of open source software projects in the SDN/NFV space.

them trying to defend its position within the roaring market, resulting into a massive code-base with plenty of sub-layers, projects and plug-ins. In the data center market, OSS projects such as OpenStack or CloudStack for the control of processing, storage and networking resources are adopted today and heavily supported by OS software- and server hardware industry. In fact, OpenStack has become the preferred Virtual Infrastructure Manager (VIM) in the commercial SDN/NFV solutions of major telecom vendors such as Cisco, Ericsson or Alcatel-Lucent.

OSS at orchestration layers in the SDN/NFV architecture is a very recent phenomenon, getting impulses from the telco side as well as from the cloud industry. The ETSI ISG NFV MANO work plays an important role in this space and has set up a Proof of Concept-zone[5] promoting multi-vendor or open ecosystems integrating components from different players. The openMANO project led by Telefonica is one of the smaller OSS initiatives providing real support for NFV orchestration. The OPNFV initiative builds further on existing industry-driven projects such ODL and OS, and targets to become de facto standard open source NFV platform, initially focusing on the NFVI and VIM components of the ETSI MANO architecture. In a similarly way, Juniper has opened and extended its Contrail SDN-control platform to OpenContrail, increasingly supporting orchestration of NFV-focused scenario's. Juju and Cloudify are traditional cloud service orchestration platforms which are extending their scope towards NFV orchestration. OpenCloud emerges from on.lab as research-driven alternative for cloud and NFV orchestration.

NFV orchestration is closely related to the specification of Service Function Chains and VNFs, as well as the process for their (re-)deployment and configuration. Currently there is no software standard of how VNFs should be packaged, deployed and configured. However, a range of potential building blocks are currently available in the OSS domain. Pure packet-processing functionality is available either in the Linux kernel netfilter modules, as software switching functionality such as Open vSwitch, or as a set of configurable packet-processing components in the Click Modular Router. ClickOS [14] is a research OSS initiative driven by NEC to support Click-based VNFs on top of light-weight Xen virtualization. Configuration and state-handling is covered by the Click handler system. Another research-driven OSS is OpenNF [12] project, providing a control API for Floodlight controllers supporting state migration between NF instances. The ESCAPE emulator [6] resulting from the EU UNIFY project [5] provides an alternative orchestration framework supporting Click-based VNFs in a Mininet environment. On the level of VNFs, when focusing for example on the SFC of Figure 2, OSS implementations can be found for many higher layer services such as video streaming, caching and probing (cfr. executables of MistServer, Varnish Cache and Stream Surfer).

In order to have a fully automated and carrier-grade orchestration framework, industry-wide OSS standards for VNF characterization, packaging, and management will be needed, in addition to VNF repository/catalogue[6]. With respect to VNF configuration, OSS packages such as OpenNMS, Puppet and Chef might play important roles. The first is an OSS/BSS framework supporting carrier-grade network monitoring and management of a wide range of equipment, while the latter two tools are highly used and supported for supporting the automated software roll-out and configuration on distributed systems. With respect to the specification of SFCs, there is currently no OSS standard to do this. One option would be to build further on, e. g., the OpenTOSCA implementation, and extending the OASIS TOSCA for supporting SFCs. TOSCA is a language for describing the topology of cloud based web services, their components, relationships, and processes that manage them.

3.2 Open source licenses

Each of these projects are distributed under a specific open source license. Through the license, the author gives permission for the receiver of the program to exercise these freedoms, adding also any restrictions that the author may wish to apply. Although there are many open source licenses, the important ones can be divided in two large categories and within each category only a few licenses are in widespread use. The first category comprises licenses that do not impose special conditions on the second redistribution these are referred to as permissive licenses. Among these licenses, the Berkeley Software Distribution (BSD) license is the best known. The only obligation it imposes is to credit the authors. Other popular licenses in this category are the MIT license, the Apache License Version 2 and the Eclipse Public License Version 1.0. All of these licenses include a limitation of guarantee which is really a disclaimer in order to avoid legal claims. The second category, strong licenses, impose conditions in the event of wanting to redistribute the software, aimed at ensuring compliance with the license,s conditions following the first redistribution. The General Public License of the GNU project (GNU GPL) is the most popular and well-known in the world of OSS. It allows modifications to be made without any restrictions to the source code, however it is only possible to redistribute code licensed under the GPL integrated with other code if it has a compatible license. Another popular strong licenses are the Affero GPL which considers the case of programs offering services via computer networks.

5 http://www.etsi.org/technologies-clusters/technologies/nfv/nfv-poc

6 The NF market place is one of the expected outcomes of the EU T-NOVA project [37].

3.3 Diffusion and adoption of open source

The main consequence of offering software under an open source license is that it is not possible to make much money from its redistribution. Curiously, several of these open source projects draw members from across the industry including network equipment vendors. The OpenStack community with over 500 companies supporting the OpenStack foundation is such an example. From the vendor's side, the choice for collaborating in an OSS project together with its direct competitors can be understood as a strategy to reduce market fragmentation and to distribute development costs in emerging markets with uncertain business potential. Based on stable releases of the projects, the open source licenses allow vendors to develop applications that interface with the software as well as to offer technical support or to extend the guarantees. From the network operator's side, the availability of OSS means that there are no longer black boxes that need to be fitted together. Parts of programs can be integrated, without reverse engineering, to obtain an integrated product without constraints to a single vendor. Also, having the source code relieves the network provider from dependence on the long product cycles of vendors. The provider can modify the program for own use as often as needed and use this freedom to spur innovation and meet rapidly changing user needs.

For network operators to be successful with the adoption of OSS, a high level of organizational change will be required. Most network operators do not train staff in the skills needed to extend open source project with own custom solutions that meet the operator's requirements. When software is developed internally and that software is running on commodity hardware, the organization will need to find ways to mitigate the risk of both hardware- and software failures. To do so, close communication and a common understanding of the problem between the development team and the network architects, engineers and operators should be developed as well as related project management skills.

If network operators move towards SDN and NFV, it will disrupt the business model of equipment vendors. As specialized, integrated network solutions will be largely interchanged for open source implementations of NFs running on top of commodity hardware the traditional business model will be challenged. At the same time, network vendors will face a rising need for software development and related services such as application development. To prepare for this change, several vendors are among the prominent investors in open source projects such as OpenStack, OpenDaylight and OPNFV.

3.4 Standardization development organizations

Standards development organizations (SDOs) such as the Internet Engineering Task Force (IETF) and the European Telecommunications Standards Institute (ETSI) play a critical role in the telecommunications market. Both SDN and NFV have triggered significant interest in these standardization organizations.

The standardization process assures quality standards development but this process typically takes many years, preventing operators who are willing to venture into new domains from doing so in a fast pace. This is in contrast with the dynamics of open source software projects which can create a de-facto market-based consensus which is realized in a product (an OSS project) even before a standardization organizations has agreed on a standard. The question therefore arises how a standardization organization can be relevant for an open source software project.

The answer is probably very close to the initial goal of the standards development organizations: to facilitate communication and ensure good governance. As such, a first priority should be to ensure that OSS projects have a voice in the standards development organizations to establish a feedback loop between both communities. Once established, the provided governance structures by the standards development organizations can be a way to avoid (unintentional) overlap and ensure compatibility between OSS projects. SDOs can provide support for an effective development process for both new contributions as well as maintenance, updates and releases and non-technical challenges such as business, management, strategic and legal processes.

One particularly relevant example is the establishment of the Open Networking Foundation (ONF) which has been active in the promotion of SDN via the development of open standards. As the OpenFlow protocol evolved very quickly (and not always in a backward-compatible way), the ONF focused on market development for the protocol, architecture and OpenFlow controller (the ONF did not develop a reference implementation of their own). These activities are atypical for a traditional SDO but can be relevant as an example to show how the gap between both worlds can be bridged.

4 Conclusion

The paradigm shift in telecom networks towards SDN and NFV introduces possibilities for OSS projects. Network operators may decrease their dependency on network equipment vendor solutions, by replacing them through free software implementations running on low-cost multi-purpose hardware. These hardware vendors may benefit by selling more hardware for purposes which were previously reserved for dedicated vendor hardware solutions. Software industry faces a new customer base for control and orchestration-related software solutions, as well as platforms increasing the performance of NFs on COTS hardware. Vendors are looking for new ways in maintaining their customer base through strong participation in large OSS projects such as OpenDaylight, OpenStack and OPNFV. Although many OSS components are available supporting the SDN/NFV model, OSS standards are still missing with respect to SFC and VNF specification, configuration and deployment. In order to be successful within the new SDN/NFV trend with OSS, operators, as well as vendors will need to adapt their organizational structure, increasing their software skills and focus on active OSS contribution. Standards development organizations on the other hand will need to adopt their processes to bridge the gap between both worlds by focusing on optimization of communication and active development of governance structures for OSS projects.

References

1. Apache CloudStack, http://cloudstack.apache.org.
2. Chef, https://www.chef.io.
3. Click Modular Router, http://read.cs.ucla.edu/click.
4. Cloudify, http://getcloudify.org/.
5. Császár, A., et al. Unifying cloud and carrier network: Eu fp7 project unify. IEEE/ACM UCC 2013.
6. Csoma, A., et al. ESCAPE: Extensible service chain prototyping environment using mininet, click, netconf and pox, ACM SIGCOMM 2014.
7. Docker, https://www.docker.com.
8. DPDK, http://www.dpdk.org.
9. ETSI ISG NFV homepage. http://www.etsi.org/technologies-clusters/technologies/nfv.
10. FlowVisor, https://github.com/opennetworkinglab/flowvisor.
11. Free Software Foundation, Free Software Definition.
12. Gember-Jacobson, A., et al. OpenNF: Enabling innovation in network function control, ACM SIGCOMM 2014.
13. Juju Charms, https://jujucharms.com.
14. Martins, J., et al. ClickOS and the art of network function virtualization, 11th USENIX NSDI 2014.
15. Frenetic, http://frenetic-lang.org.
16. KVM, http://www.linux-kvm.org.
17. Netfilter.org project, http://netfilter.org.
18. Netmap, http://info.iet.unipi.it/~luigi/netmap/.
19. NOX, http://www.noxrepo.org.
20. Open Source Initiative, http://www.opensource.org.
21. ONOS, http://onosproject.org.
22. OpenContrail, http://www.opencontrail.org.
23. OpenCloud, http://www.opencloud.us.
24. OpenDaylight, http://www.opendaylight.org.
25. OpenMANO, https://github.com/nfvlabs/openmano.
26. OpenNMS, opennms.org.
27. OpenStack https://www.openstack.org.
28. OpenTOSCA – Open Source TOSCA Ecosystem, http://opentosca.org.
29. OpenVirtex, http://ovx.onlab.us.
30. OpenvSwitch, http://openvswitch.org.
31. OPNFV, https://www.opnfv.org.
32. Project Floodlight, http://www.projectfloodlight.org.
33. Puppet labs, https://puppetlabs.com.
34. Ryu, http://osrg.github.io/ryu.
35. Sherwood, R., et al. Flowvisor: A network virtualization layer, OpenFlow Switch Consortium, Tech. Rep, 2009.
36. Xen Project, http://xenproject.org.
37. Xilouris, G., et al. T-NOVA: a marketplace for virtualized network functions, Networks and Communications (EuCNC), 2014 European Conference on. IEEE, 2014.

Bionotes

Dr. Wouter Tavernier
Ghent University – iMinds, Department of Information Technology, Gaston Crommenlaan 8/201, B-9050 Gent, Belgium
wouter.tavernier@intec.ugent.be

Wouter Tavernier received a M. Sc. in Computer Science in 2002 and a Phd. degree in Computer Science Engineering in 2012, both from Ghent University (Belgium). In addition he received a B. Sc. degree in Philosophy in 2003 from K.U.Leuven (Belgium). After a two-year period as business analyst at Accenture, he joined the Internet-Based Communication Networks and services group of Ghent University in 2006 to research future internet-topics including resiliency of L2 and L3 networks, geometric routing, machine learning for routing, and Software-Defined Networks and Network Function Virtualization. Last years his focus was on the design of algorithms for service orchestration and decomposition in SDN/NFV contexts. Most of this work is performed in the context of European projects such as TIGER, ECODE, SPARC, EULER, UNIFY and SONATA.

Mr. Bram Naudts
Ghent University – iMinds, Department of Information Technology, Gaston Crommenlaan 8/201, B-9050 Gent, Belgium
bram.naudts@intec.ugent.be

Bram Naudts received an MSc degree in applied economics: business engineering specialized in operations management from Ghent University (Ghent, Belgium) in 2011. He joined the same university in 2011, where he joined the IBCN group of the INTEC department at the Faculty of Engineering and Architecture. His current research focuses on techno-economic evaluation of communication network architectures and services.

Prof. Dr. Ir. Didier Colle
Ghent University – iMinds, Department of Information Technology, Gaston Crommenlaan 8/201, B-9050 Gent, Belgium
didier.colle@intec.ugent.be

Didier Colle is full professor at Ghent University since 2014. He was associated professor since 2011 at the same university and received a PhD degree in 2002 and a M. Sc. degree in electrotechnical engineering in 1997 from the same university. He is group leader in the iMinds Internet Technologies department. He is co-responsible for the research cluster on network modelling, design and evaluation (NetMoDeL) inside the Internet Based Communication Networks and Services (IBCN) research group. This research cluster deals with fixed internet architectures and optical networks, green-ict, design of network algorithms and techno-economic studies. His research is mainly conducted inside international (mainly European), national and bilateral research projects together with the industry. This research has been published in more than 300 international journal and conference articles and has resulted in more than 15 PhD degrees.

Prof. Dr. Ir. Mario Pickavet
Ghent University – iMinds, Department of Information Technology, Gaston Crommenlaan 8/201, B-9050 Gent, Belgium
mario.pickavet@intec.ugent.be

Mario Pickavet received an M.Sc. and Ph.D. degree in electrical engineering, specialized in telecommunications, from Ghent University in 1996 and 1999, respectively. Since 2000, he is professor at Ghent University where he is teaching courses on discrete mathematics, broadband networks and network modeling. He is leading the research cluster on Network Design, Modeling and Evaluation, together with Prof. Didier Colle. In this context, he is involved in a large number of European and national research projects, as well as in the Technical Programme Committee of a dozen of international conferences. He has published about 100 international journal articles (IEEE JSAC, IEEE Comm. Mag., Journal of Lightwave Technology, Proc. of the IEEE, Photonic Network Communication, ...) and over 300 publications in international conference proceedings. He is co-author of the book 'Network Recovery: Protection and Restoration of Optical, SONET-SDH, IP, and MPLS'.

Prof. Dr. Sofie Verbrugge
Ghent University – iMinds, Department of Information Technology, Gaston Crommenlaan 8/201, B-9050 Gent, Belgium
sofie.verbrugge@intec.ugent.be

Sofie Verbrugge received an MSc degree in computer science engineering from Ghent University (Ghent, Belgium) in 2001. She obtained the PhD degree from the same university in 2007 for her thesis entitled "Strategic planning of optical telecommunication networks in a dynamic and uncertain environment". Since 2008, she has been working as a researcher affiliated to iMinds, a research institute to stimulate ICT innovation in Flanders, where she is a coordinator for the techno-economic research within the Internet Based Communication Networks and Services group (IBCN). Since October 2014 she is appointed as an associate professor in the field of techno-economics at Ghent University. Her main research interests include infrastructure as well as operational cost modeling, telecom service and network deployment planning, advanced evaluation techniques including real options and game theory. She has been involved previously in several European as well as national research projects in these domains, including the COST-action Econ@tel on telecommunication economics. She led the work package on "Business modeling" with the European FP7 project ICT-OASE and is currently working within the FP7 project ICT-Flamingo.