Management Requirements for ClickOS-based Network Function Virtualization

Lucas Bondan, Carlos Raniery Paula dos Santos, Lisandro Zambenedetti Granville
Institute of Informatics – Federal University of Rio Grande do Sul
Av. Bento Gonçalves, 9500 – Porto Alegre, Brazil
Email: {lbondan, crpsantos, granville}@inf.ufrgs.br

***Abstract*—Network Functions Virtualization (NFV) is a new approach to design, deploy, and manage network functions. In a recent past, such functions used to be implemented at hardware. This approach, besides effective, presents many disadvantages such as increased operational costs, difficulties to scale up or down the network, and deploy new functions. The rise of virtualization technologies, on the other side, provides new ways to rethink about network functions. Instead of specialized and expensive hardware, multiple network functions can share the same commodity hardware, thus contributing to a better utilization of resources. Besides its advantages, NFV is still on its early stages of employment. Important aspects are not yet being investigated by the research community. For example, to this date, the management requirements of NFV remain unclear. Therefore, the present paper addresses this subject, it presents a realistic network function request, which is used to identify management requirements in the context of a specific NFV enabler platform called ClickOS.**

***Index Terms*—Network Functions Virtualization, Network Management, Function Requests**

I. Introduction

Network Function Virtualization (NFV) [1] is a novel network paradigm that separates data plane software from the underlying hardware. Different than Software Defined Networking (SDN) [2], which deals with control plane through more mature technologies like OpenFlow [3], NFV is still in its infancy, struggling to establish itself as viable way to reduce costs of network deployment and maintenance, *i.e.,* CAPEX and OPEX [4].

Industry, academia, and standardization bodies have shown increased interest in NFV. That can be observed, for example, in important consortiums formed by network vendors, the proliferation of papers and conferences about NFV, and on the NFV-related work under development in the European Telecommunications Standards Institute (ETSI) and on the attempt to create a NFV research group (RG) in the Internet Research Task Force (IRTF).

As in any new networking technology, network management aspects are crucial for the success of NFV. However, despite the increased interest, network management has been neglected in current NFV efforts. Because we believe that network management cannot be an afterthought, in this paper we deal with the issue of identifying NFV management requirements in the context of a specific NFV enabler platform, *i.e.,* ClickOS [5].

Because NFV is still in its infancy, as mentioned before, there is no widely deployed NFV platform. In fact, several platforms seem to be under development, but most of them is unavailable or not based on open source software. ClickOS is a fortunate exception in this landscape. Based on open source software, ClickOS is frequently considered an NFV enabler. Although far from being an NFV materialization as mature and concrete as OpenFlow is for SDN, ClickOS allows us to identify the challenges that network administrators wanting to operate a network with NFV will face.

NFV allows several different networking scenarios, including those with complex relationships between network operators and service providers. Although we recognize that complex scenarios would emerge to support relevant business models, we concentrate our investigation in a simpler scenario where the network administrator is interested in operating its network using NFV by deploying virtualized functions hosted inside the managed network itself. Our methodology to identify management requirements is based on using ClickOS over real virtualized servers and observing the challenges that the operator faces to take advantage of NFV's advertised benefits.

Our investigation starts with a network setup request, which describes the network functions and their relationships in a test network scenario. Our network setup request is introduced in Section II. In Section III, we present our ClickOS environment, describing in details the network infrastructure and the set of operations on top of ClickOS used in our investigation to materialize the network setup request. The list of management requirements is then extracted from the activities the network operator carried out over ClickOS. In Section IV, we present and discuss such a list of management requirements. Finally, we present our conclusions and discuss opportunities of future work in Section V.

II. Network Setup Request

NFV uses virtualization technologies to deploy network functions (NFs) (*e.g.,* firewalls, IDSes, load balancers). In comparison to traditional hardware-based networks, NFV has the advantage of decreasing operational costs, since multiple NFs can share the same commodity hardware. Another benefit of NFV is that it provides a more dynamic environment where NFs can be quickly scaled up or down to address changing demands.

In NFV, virtualized network functions (VNFs) act as building blocks that are connected and orchestrated from a Management System. Through this system, a network operator is able to manage the functions' life-cycle (*e.g.,* instantiation, scaling, termination), as well as to define the chain of VNFs that creates more sophisticated network functionalities. This chain of functions is defined through VNF Forwarding Graphs (VNF-FG) [6].

In this work we use ClickOS as the platform for NFV provisioning. Although other alternatives (*e.g.,* using containers or hypervisors) are available to achieve the same goal, ClickOS is the most prominent of them. ClickOS is a minimalist operating system based on the Click Modular Router [7], focused on supporting typical network requirements such as high throughput, low delay, and isolation. In its current version, ClickOS supports a significant variety of network functions, including traffic shaping, network monitoring, and DDoS prevention.

A. Perimeter Network Design

To identify management requirements in the context of ClickOS-based NFV, we consider the scenario where a network operator needs to deploy a perimeter network – also know as Demilitarized Zone (DMZ) – to provide a protected environment for the organizations' services. A DMZ is used to offer some of an organization's services (*e.g.,* Web server, e-mail server, VoIP server) to an untrusted external network such as the Internet. Hosts placed the DMZ have only limited connectivity to services running inside the internal network. This approach allows the internal network to be protected in the case of an intruder compromising any DMZ's host.

In order to deploy a DMZ, the network operator needs to use multiple and disparate NFs. The most basic NF in such an environment is firewalling, used to filter incoming and outgoing traffic based on specific rules. In simple DMZs, a single firewall is used to protect the internal network from the public one. In this paper, we consider the case where a more secure setup is requested, composed of two firewalls as presented in Fig. 1.

Four NFs are highlighted in the gray boxes of the network setup request of Fig. 1: firewall, load balancer, IDS sensor, and NAT. From top to bottom, the first firewall filters traffic to allow only HTTP/S communications. Any packet that does not match this filtering rule is discarded. All traffic allowed by the firewall is then captured by an IDS sensor, which is used to monitor network traffic looking for malicious activities. Also inside the DMZ, a load balancer distributes user requests among Web servers, according to a probability distribution function, thus ensuring that no Web server becomes overloaded.

A second firewall separates the DMZ from the internal network. This firewall is a second barrier in case of an intrusion in the DMZ, thus preventing important services inside the organization (*i.e.,* LDAP, Application Server, and Database) of becoming compromised. Since Web servers at the DMZ and Application Server (responsible for the business logic) at the internal network communicate using multiple protocols (*e.g.,* SOAP, JMS, RMI), the second firewall must allows traffic of these protocols. A good practice of network security is to place a second IDS sensor inside the internal network to detect attacks coming from insiders (*e.g.,* hosts within the internal network). In this way, the sensor is also less prone to attacks directed to the IDS itself. Finally, a traditional NAT is placed inside the internal network for translating public IP addresses to the corresponding local ones.

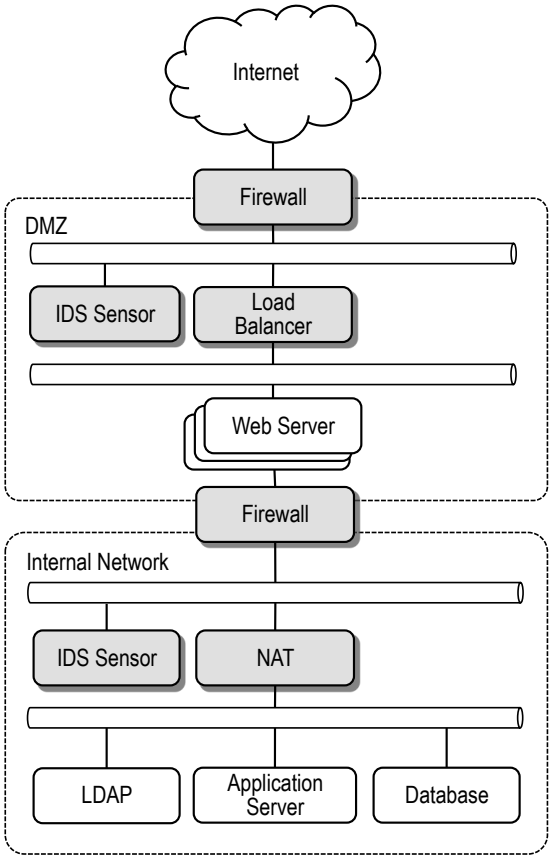

Fig. 1. DMZ Topology Request

In the next section, we describe in details all the steps performed to deploy the network setup request presented in this section, highlighting the main characteristics related to the NFV concepts applied in our experimentation.

III. Network Infrastructure Deployment

In order to deploy the network request presented in the previous section, we employed two VNF servers: one responsible for hosting the DMZ functions, and a second VNF server responsible for hosting the internal network functions. This approach enabled us to isolate both networks, thus supporting an increased security level for the organization's services. This setup is presented in Fig. 2.

Each VNF server consists of a XenServer hypervisor running on top of a Linux-based OS. A generic ClickOS image is compiled for each hosting OS, and is initialized according to a predefined configuration template. In this work, we used

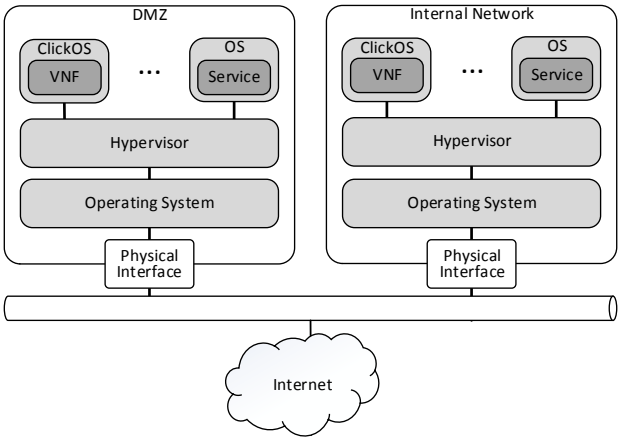

Fig. 2. Topology Request Deployment with ClickOS

a template consisting of one CPU, 12MB of memory, and up to 2 virtual network interfaces depending on the function. For example, firewall functions require two interfaces: one for incoming traffic, and the other for outgoing.

In ClickOS, network functions are defined using the network elements provided by the Click Modular Router. These definitions are represented in description files, which are interpreted and executed by ClickOS. Network operators manages function's lifecycle through the Cosmos tool, which implements the communication interface between the user domain and the ClickOS domain.

Once the VNFs are defined and ready for execution, the network operators needs to specify how to interconnect them (*i.e.,* define the VNF-FG). The connection between two or more NFs can be either physical or virtual (*e.g.,* through bridges). Virtual interfaces (*i.e.,* between the VNF server and ClickOS VMs), can be created using XenServer. The VNF-FG used in this work is presented in Fig. 3, with dark lines representing real connections, and dashed lines representing logical connections.

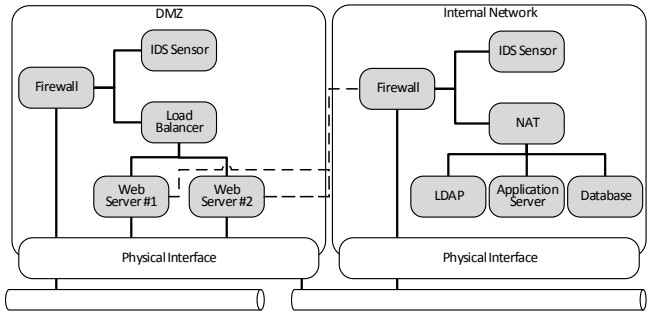

Fig. 3. VNF-FG of the Resulting Topology Deployment

In the evaluated scenario, the first VNF is a stateless firewall located in the DMZ. This firewall is used to filter incoming traffic, allowing only HTTP/S packets. We use a network bridge to connect the physical network interface of the VNF Server #1 to the virtual interface of the ClickOS VM hosting the firewall function. A second network bridge is used to connect the load balancer to the virtualized Web servers. In this work, we deployed two Web servers on the DMZ to process user requests. These servers are connected to the physical interface using a third bridge. We decided to use virtualized Web servers to facilitate the deployment of the network setup request, otherwise it would be necessary to employ additional physical interfaces and network devices. This decision, however, doesn't have any impact on the management requirements presented in the next section.

Inside the internal network, a second firewall is used to allow network traffic related to specific protocols. This firewall is connected with the VNF Server #2 physical interface, and performs the first packet processing in the internal network. A network bridge is used to connect the firewall's output with the virtualized IDS sensor and NAT. Finally, the output interface of the NAT function is also connected using a network bridge with three virtualized services inside the internal network: a LDAP, an Application Server, and a Database.

All the connections (physical or virtual) are part of VNF-FG of the service provided by the network setup scenario. When the service be required, it will be provided using the VNF-FG corresponding to this service, *i.e.,* , the request will be processed following all the network elements that compose the respecting VNF-FG. Thus, the creation of the VNF-FG is a key factor in the deployment of a NFV scenario, allowing a set of NFs be composed to provide a whole service.

In next section, we will discuss all the difficulties faced with the deployment of the proposed network setup on the presented NFV infrastructure. The main objective is to derive the management requirements based on these difficulties, composing a first summary about NFV management requirements.

IV. Management Requirements

The deployment of a DMZ using ClickOS revealed significant difficulties in the adoption of current NFV technologies. Based on such difficulties we identified a list of important management requirements for properly maintaining NFV-based networks. These requirements are discussed in the following. Our objective with this list is to provide starting point for network administrators interested in using NFV.

A. VNF Server Configuration

The current landscape of NFV lacks a common platform for NFs embedding. Network administrators are obligated to use multiple solutions in order to configure and maintain a VNF server. In the case of ClickOS, for example, the VNF server should support specific libraries to build the virtual images as well as an instance of the Citrix XenServer hypervisor to host those images [5]. These solutions, however, are not created with integration in mind, thus leading to an additional effort for the network administrator.

B. VMs Instantiation

While there is a good support for virtualizing traditional OSs (*e.g.,* Linux, Windows, and OSX) using Xen technology, its not the case for ClickOS. Changes on its compilation

process to improve network performance prevent existing tool (*e.g.,* XenCenter and XenManager) to be used. The lack of standardized communication methods between these tools and ClickOS images, led the ClickOS maintainers to develop a specific tool, called Cosmos, to provide user interaction. In this context, VMs instantiation should be supported by appropriate management solutions, able to provide feasible and effective methods for the network operators to configure and instantiate ClickOS VMs.

C. Infrastructure Deployment and VNF Location

The provision of NFV-based networks requires an infrastructure of VNF servers properly configured. VNFs connections (*i.e.,* network bridges) are usually manually defined by the network administrator using the XenServer CLI tool. This tool, besides effective, doesn't provide a complete view of the network for the network administrators. Connections between VNF servers are individually defined, using command line instructions for that. Systems focused on the the network administrator needs, like graphical network representation, can be helpful in the infrastructure deployment process, and consequently on its management.

D. Network Functions Design and Deployment

NFs are defined in ClickOS using a configuration language based on the Click Modular Router elements. Each element refers to a basic network operation, such as IP filtering, traffic shaping, and address translation. The way such elements are connected represents the processing flow applied to incoming/outgoing network packets. Although NFs can be manually defined, the usage of a high-level and visual design tool, such as Clicky [8], improves the process of creating new functions. Besides making it easier for network administators to design NFs, Clicky doesn't support VNF deployment. It forces administrators to design a NF in one tool and manually apply them using the Cosmos interface.

E. VNF Monitoring

Network monitoring is one of the main tasks performed by network administrators. In the NFV context, ClickOS VMs can be monitored using the Xen console, which displays information regarding to network traffic and Click operation. Although this approach may be enough for small networks, as the number of VNFs increases, new methods becomes necessary. Moreover, considering the scenario where a NFV orchestrator is responsible for migrating VNFs, it becomes easy for network administrators to lose control of VNFs location [9].

F. VNF Reconfiguration

In order to reach higher performance levels, network administrators are constantly reconfiguring network functions. Such reconfigurations may be performed in ClickOS VM instances by using the Cosmos tool. However, Cosmos only provides access to individual VMs, while some scenarios might benefit from a batch reconfiguration of the network. This approach can, for example, handle dynamic provisioning changes and improve the overall management task by saving time from the network administrators.

Once summarized and discussed the list of management requirements derived from our experimentation, in the next section we present some conclusions about this work. Moreover, we present our perspectives for future work based on the results obtained with the experiments presented in this paper.

V. Conclusion and Future Work

In this paper we presented an initial effort to identify NFV management requirements. Our investigation is based on the deployment of a NFV request in a ClickOS-based infrastructure. Besides using a specific technology (*i.e.,* ClickOS), we believe that the identified requirements are still valid for other NFV platforms and more sophisticated network scenarios. As future research, we will design an integrated management system based on the identified requirements. The objective of this system is to promote the adoption of NFV by network operators. For example, we plan to provide the level of functionalities supported by XenCenter, assist the design of VNF-FG, and the development of new network functions through visual interfaces.

Acknowledgements

The authors thanks to João Martins and Filipe Manco from NEC Europe for the support with creation and configuration of ClickOS images.

References

[1] White Paper, "Network Functions Virtualisation (NFV)," *ETSI NFV ISG White Paper*, no. 1, pp. 1–16, 2012.

[2] N. Feamster, J. Rexford, and E. Zegura, "The Road to SDN: An Intellectual History of Programmable Networks," *ACM Sigcomm Computer Communication*, vol. 44, no. 2, pp. 87–98, 2014.

[3] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "OpenFlow: Enabling Innovation in Campus Networks," *SIGCOMM Computer Communications Review*, vol. 38, no. 2, pp. 69–74, Mar. 2008.

[4] Y. Jarraya, T. Madi, and M. Debbabi, "A Survey and a Layered Taxonomy of Software-Defined Networking," *IEEE Communications Surveys Tutorials*, vol. PP, no. 99, pp. 1–1, 2014.

[5] J. Martins, M. Ahmed, C. Raiciu, V. Olteanu, and M. Honda, "ClickOS and the Art of Network Function Virtualization," *USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, 2014.

[6] White Paper, "Network Functions Virtualisation (NFV) - Use Cases," *ETSI NFV ISG*, vol. 1, pp. 1–50, 2013.

[7] E. Kohler, R. Morris, B. Chen, J. Jannotti, and M. F. Kaashoek, "The Click Modular Router," *ACM Trans. Comput. Syst.*, vol. 18, no. 3, pp. 263–297, Aug. 2000.

[8] Clicky. (http://www.read.cs.ucla.edu/click/clicky) Accessed on August, 3.

[9] S. Wright, D. Clarke, P. Runcy, K. Siva, and M. Mularczyk, "Virtual Function State Migration and Interoperability," *NFV ISG Proof of Concept Proposal*, pp. 1–5, 2013.