Click-on-OSv: A Platform for Running Click-based Middleboxes

Leonardo da Cruz Marcuzzo[‡], Vinicius F. Garcia[‡], Vitor Cunha[†], Daniel Corujo[†], Joao P. Barraca[†], Rui L. Aguiar[†], Alberto E. Schaeffer-Filho[*], Lisandro Z. Granville[*], Carlos R. P. dos Santos[‡]
[*]Institute of Informatics – Federal University of Rio Grande do Sul – Porto Alegre, Brazil
[†] Department of Electronics, Telecommunications and Informatics – University of Aveiro – Aveiro, Portugal
[‡]Department of Applied Computing – Federal University of Santa Maria – Santa Maria, Brazil

***Abstract*—In this paper, we present a modern platform for running Virtualized Network Functions based on the Click Modular Router. The proposed platform leverages of current technologies – such as DPDK, OSv, and REST Web Services – to fulfill the ETSI requirements for Network Functions Virtualization. The obtained results show the feasibility of the platform to consolidate disparate network functions, while demonstrating flexibility from a management point-of-view.**

I. Introduction

The increasing dependency on communication infrastructures by modern companies, industries, and the government has led to challenging issues for both network researchers and practitioners. Although sophisticated solutions have been proposed in recent years, limitations of the current infrastructures are a major obstacle to their wide adoption. For example, in the specific case of the Internet ossification, the use of middleboxes hinders the employment of new protocols in the network.

In this context, Network Functions Virtualization (NFV) [1] is an interesting approach to tackling the problem of ossification by using virtualization techniques. In NFV, network functions commonly performed by middleboxes are virtualized and deployed on commodity hardware. This enables several advantages for both developers and ISPs, for instance, the reduction of capital and operational expenditure (CAPEX and OPEX), and the possibility of sharing physical resources between different service providers.

In order to promote the adoption of NFV, the European Telecommunications Standards Institute (ETSI) has defined a set of requirements (*e.g.*, portability, high performance, and optimized resources usage) that must be supported by platforms used for running the virtualized functions. However, NFV is still in its infancy and lacks a solid solution that fulfills these requirements. Thus, this paper presents a modern platform that leverages of state-of-the-art technologies for executing Virtualized Networking Functions (VNFs).

The paper is organized as follows. Section 2 presents the internal architecture of the proposed platform, while in Section 3 we present both an evaluation scenario and obtained results. Finally, we conclude this paper in Section 4.

II. Platform Architecture and Test Scenario

Based on the analysis of ETSI's NFV specifications [4] we identified three main components that must exist in platforms

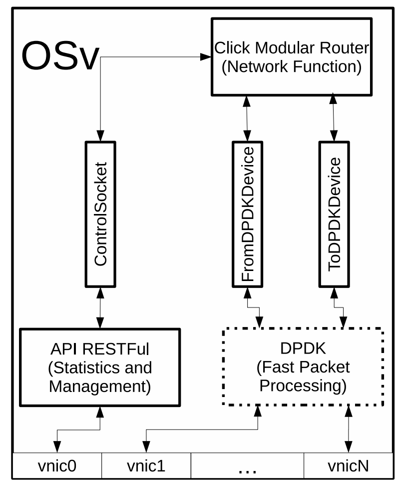

Fig. 1. Click-on-OSv architecture

created to run VNFs: the operating system itself, a packet processing engine, and a packet accelerator. Based on these categories, disparate solutions were evaluated in order to select the most promising ones.

The proposed platform consists of a virtual machine image, created based on the minimalist operating system OSv [3]. The network functions are implemented using the complete set of elements available in the Click Modular Router [2]. Intel's DPDK was used to provide a faster input/output interface, thus improving packet processing performance and throughput. Figure 1 shows the internal architecture of the developed platform.

For the operating system, an unikernel OS – *i.e.*, OSv – was chosen because such systems are specifically designed to run a single application and are more suited for virtualized environments than traditional operating systems. OSv supports two modes of operation: Paravirtualization (PV) and Hardware assisted Virtualization (HVM). Further, OSv is lightweight and presents an integrated management interface. On the proposed platform, its main responsibilities also include handling all the management actions, as well as providing a basic environment for the execution of both Click-based functions and DPDK.

We also used a packet accelerator – *i.e.,* Intel DPDK – to achieve a better performance (especially when Virtio is used)

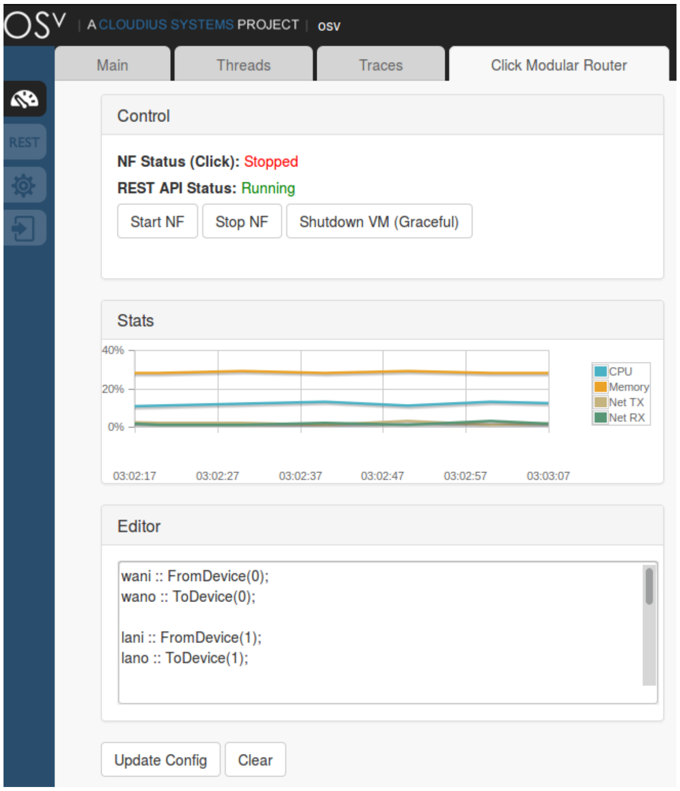

Fig. 2. Platform Web Interface

in comparison with the default input/output elements used by the user-level version of Click. Finally, DPDK also demonstrated to be an adequate solution to enable the communication of Click Modular Router with the OSv kernel network stack.

The Click Modular Router is responsible for performing the packet processing defined in a configuration file (representing the virtual function). We choose Click for the execution of the virtual functions because it has an extensive list of elements, performs well when used with DPDK and has been extensively validated by the academia in the past few years.

Finally, as shown in Figure 2 we added new functionalities to the native OSv Web interface. These modifications enable the network operator to: (i) gather statistics of the running functions; (ii) modifying the VNF's implementation; (iii) controlling both the VMs and NF's lifecycle. The same data and operations are also available through a REST interface, thus facilitating the integration of the proposed platform to current management and orchestrator systems (*e.g., OpenStack*).

To evaluate the platform, a simple scenario was created based on the RFC 2544. It consists of two virtual machines (*i.e.*, a transmitter and a receiver) connected to the platform through network bridges. The network function running on the platform was a packet forwarding, so that the maximum throughput could be achieved. The tests were performed on an Intel Xeon E5620@2.4GHz server, running Debian 8 and the Xen hypervisor. A dedicated physical CPU core and 192Mb of memory were used for each VM.

We also evaluated another platform, called ClickOS [5], on

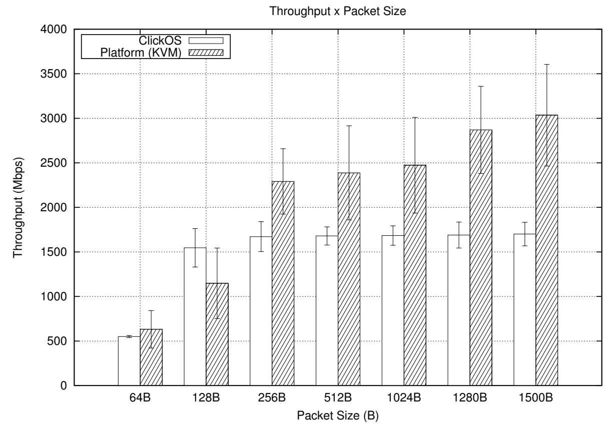

Fig. 3. Throughput

the same scenario. ClickOS is one of the most preeminent platforms available up to this date. The obtained results, presented in Figure 3, demonstrate the higher performance of our proposed platform when compared to ClickOS. We plan to use the same scenario for our demonstration, albeit running more complex network functions, such as a firewall and a proxy.

III. Conclusion

We presented in this paper a new minimalist platform for running Click-based network functions. The platform was created by the integration of existing solutions, each one chosen to meet specific requirements of the NFV specification. We were able to successfully run network functions with a reasonable performance. We also developed new functionalities not available on other platforms, such as a REST and Web interfaces for managing VNFs' lifecycle and monitoring resources' usage. As future work, we aim to improve the platform's performance and include new features, such as VNF live migration.

IV. Acknowledgment

This work has been partially supported by the project "Um Ecossistema para Funes Virtualizadas de Rede" (# 0409/2016), from CAPES Brazil, program FCT 039/2014.

References

[1] M. Chiosi, "Network functions virtualisation: An introduction, benefits, enablers, challenges and call for action," *ETSI White Paper*, 2012.

[2] R. Morris, E. Kohler, J. Jannotti, and M. F. Kaashoek, "The click modular router," *ACM SIGOPS Operating Systems Review*, vol. 33, no. 5, pp. 217–231, 1999.

[3] A. Kivity, D. Laor, G. Costa, P. Enberg, D. Marti, and V. Zolotarov, "Osvoptimizing the operating system for virtual machines," in *In Proc. USENIX Annual Technical Conference (ATC) (Philadelphia, PA, June 2014), USENIX Association*, pp. 61–72.

[4] ETSI, "Network functions virtualisation requirements," ETSI ISG, Tech. Rep. GS NFV 004 v1.1.1, October 2013.

[5] J. Martins, M. Ahmed, C. Raiciu, V. Olteanu, M. Honda, R. Bifulco, and F. Huici, "Clickos and the art of network function virtualization," in *11th USENIX Symposium on Networked Systems Design and Implementation (NSDI 14)*. Seattle, WA: USENIX Association, 2014, pp. 459–473. [Online]. Available: https://www.usenix.org/conference/nsdi14/technical-sessions/presentation/martins