Challenges and Solutions for a Flexible High-Performant SDN Hypervisor

Nemanja Đerić, Andreas Blenk, Arsany Basta, Wolfgang Kellerer
Chair of Communication Networks
Department of Electrical and Computer Engineering
Technical University of Munich, Germany
Email: {nemanja.deric,andreas.blenk,arsany.basta,wolfgang.kellerer}@tum.de

***Abstract*—The virtualization of SDN networks boosts flexibility in communication networks further by combining the programmability of SDN with the flexible sharing of the infrastructure due to network virtualization. The main element to realize this combination is the SDN network hypervisor interacting on the control traffic between tenant SDN controllers and the shared physical infrastructure. We outline the current challenges of designing an SDN hypervisor, and we present updates on HyperFLEX, a reliable and flexible SDN hypervisor, which targets at solving the existing challenges. Furthermore, we provide a short outlook into future research directions, such as providing switch diversity abstraction and precise estimation of control plane resources that are needed to realize a flexible and efficient virtualization of SDN networks.**

I. Introduction

Hypervisors were originally developed for the virtualization of computers, with the goal of running one or more virtual machines completely isolated on one host [1]. This concept of sharing computers is seen as the main driver for the success of today's cloud business: virtualization enabled an easy way of sharing unused computer resources. Similarly, Network Virtualization (NV) in combination with Software-Defined Networking (SDN) is seen as one key enabler to tackle the still inherent problems of today's communication infrastructure: inefficient network resource utilization, application and service-unaware protocol design, etc.

Whereas resource sharing as provided by NV improves resource utilization, SDN provides an easy and standardized way (e.g., by using OpenFlow (OF) [4]) to tweak the use of networking resources towards the demands of network and application services. To realize this promising combination, SDN network hypervisors have been introduced in literature [2], [3], [5]; however, research on SDN network hypervisors still faces many challenges. While most virtualization solutions are designed for rather static network scenarios, flexible virtualization mechanisms working in a dynamic manner are missing in literature. Beside, existing SDN network hypervisor architectures are not well-equipped to handle unpredictable network operation due to switch diversity or unpredictable behavior of softwareized network components. In the following, we will discuss more in detail the existing challenges of a flexible SDN network hypervisor design.

This work has been performed in part in the framework of the CELTIC EUREKA project SENDATE-PLANETS (Project ID C2015/3-1) and is partly funded by the German BMBF (Project ID 16KIS0473), and in part in the framework of the EU project FlexNets funded by the European Research Council under the European Unions Horizon 2020 research and innovation program (grant agreement No 647158 - FlexNets).

Flexible and Dynamic Network Abstraction. Network abstraction is one key feature of network virtualization; it abstracts network characteristics, hence, potentially simplifies the management and operation of the network. Examples on network abstraction are abstracted network topologies like the big switch abstraction. Here, the tenant sees the network only as one big node as all intermediate links are abstracted. For instance, in case of operating a firewall, the tenant does not have to take care of complex routing tasks and can focus on deploying rules on network traffic that enters the network. To reduce the workload of tenants, network operators should provide their tenants with a simplified version of the physical SDN network. While the network abstraction has already been deployed for static use cases, the dynamic adaptation of network abstractions is rather unexplored. Beside, mechanisms that efficiently resolve problems such as switch diversity are not yet integrated into SDN network hypervisors.

Flexible and Dynamic Isolation. To guarantee tenants a predictable network operation of their virtual networks, SDN hypervisors need to provide efficient isolation mechanisms. Whereas data plane isolation mechanisms have already received a lot of attention, efficient control plane isolation concepts are still in an early phase. In contrast to traditional NV, tenants can also interfere on the shared control plane of virtualized SDN networks. As the control plane performance impacts the data plane performance in SDN networks, an uncontrolled control plane operation can lead to cross-effects between vSDNs: e.g., one vSDN might over-utilize the hypervisor resources, hence, directly impacting all other tenants. New and flexible control plane isolation mechanisms are needed that are particularly designed for dynamic scenarios.

II. State of the art & HyperFLEX Concept

FlowVisor (FV) [3] is the first developed SDN hypervisor, and it introduced the idea of the *flowspace*: the flowspace is a subset of the all available header fields which are used in OF protocol. The virtualization of topologies is realized by assigning each vSDN a non-overlapping part of the flowspace. FV supports isolation in the data and control plane, however, arbitrary virtual topologies and efficient mechanisms for dynamic operations are missing.

OpenVirtex [5] supports arbitrary vSDN topologies. However, [5] does not provide any isolation concepts neither for the data plane nor the control plane. Accordingly, an SDN hypervisor that provides arbitrary topologies with guaranteed performance is not yet well explored in literature.

HyperFLEX: Towards a Flexible SDN Hypervisor. In the current state of the art, efficient and dynamic isolation of the control plane resources is typically overlooked, especially resources of the hypervisor. Furthermore, all of the SDN hypervisor architectures are either realized as hardware extensions of SDN switches or as software instances on servers. These shortcomings were the main drivers for the creation of the *HyperFLEX* framework [6]. In the following, we present key features of *HyperFLEX*:

Hypervisor Function Decomposition: Instead of having a centralized hypervisor entity, the hypervisor can be decomposed into required virtualization functions. They can be distributed and instantiated according to the tenant requirements. *Function Flexibility:* Decomposed hypervisor functions can be flexibly realized as either software instances or hardware extensions. *Control Plane Isolation:* Hypervisor resource isolation - e.g. CPU of a hypervisor can be isolated in either software by dropping OF messages or in hardware by policing the control plane throughput of tenants. *Adaptation Support:* The *HyperFLEX* framework has been further extended in [7] where a migration protocol is used to handover control plane connections between hypervisor instances - supporting adaptation at run-time. *Performance Guarantees: HyperFLEX* monitors hypervisor resources (e.g. CPU) as well as the control plane performance of tenants (e.g. latency and loss of control messages). Based on the available resources, *HyperFLEX* performs admission control during run-time: virtual networks are only embedded if there are enough available resources [8].

III. Features Under Development

Resolving Switch Diversity. Although SDN promises a unified switch abstraction, the performance of its realization still depends on the switch hardware. Accordingly, SDN controllers might need to differentiate among switches from different vendors as they clearly exhibit a huge difference in performance [9], [10]. Furthermore, in order to reduce the production cost, some of the vendors even allow divergence from OF specifications [9]. SDN network hypervisors can be used to abstract and address switch diversity. In order to resolve these issues, performing automated benchmarks and modeling of switches is a necessity. Based on the gathered information, the hypervisor abstraction function could then resolve the switch diversity. There could be many benefits of addressing switch diversity, e.g. supporting different OF versions, ensuring that a switch is not over-utilized, compensation of violation of OF specification, etc.

Towards Control Plane Guarantees. Furthermore, SDN hypervisors are usually realized as software instances on servers with a limited amount of resources. It was shown that with the increase of control plane traffic the resource utilization is increased as well as the processing delay [12]. In order to prevent the exhaustion of hypervisor resources, *admission control* is needed to estimate how many virtual networks could be embedded and which virtual networks can be embedded. However, one of the most commonly overlooked challenges is the estimation of resources that an SDN hypervisor needs when facing complex network abstraction and control plane isolation tasks. In [11] authors used machine learning to estimate the hypervisor CPU utilization based on the number of OF messages per second; however, their conducted study was rather simple. The authors studied only setups with simple vSDN network topologies and without any control plane isolation mechanisms; furthermore, they only investigated centralized network hypervisors. Our goal is to design an estimation system for SDN hypervisors facing complex SDN network scenarios. Such scenarios should include dynamic virtual network requests, varieties of abstract network topologies, the need for precise control plane performance guarantees, physical network failures etc. We would like to integrate mechanisms that reliable estimate the needed hypervisor resources and integrate these mechanisms into *HyperFLEX*, as the current admission control relies on offline benchmarks of simple networking scenarios.

IV. Open Source

As of June 2017, HypeFLEX is available as open source on https://github.com/tum-lkn/HyperFLEX/.

References

[1] A. Kivity, Y. Kamay, D. Laor, U. Lublin, and A. Liguori, "kvm: The linux virtual machine monitor," in *Proc. Linux Symp.*, 2007, vol. 1, pp. 225–230.

[2] A. Blenk, *et al.*, "Survey on network virtualization hypervisors for software defined networking." in *IEEE Comm. Sur. & Tut. 18.1* (2016): 655–685.

[3] R. Sherwood, *et al.*, "Flowvisor: A network virtualization layer." in *OpenFlow Switch Consortium, Tech. Rep 1* (2009): 132.

[4] N. McKeown, *et al.*, "OpenFlow: enabling innovation in campus networks." in *ACM SIGCOMM Comput. Commun. Rev. 38.2* pp. 69–74, 2018.

[5] A. Al-Shabibi, *et al.*, "OpenVirteX: Make your virtual SDNs programmable." in *Proc. HotSDN,* ACM, 2014.

[6] A. Blenk, A. Basta, and W. Kellerer. "HyperFlex: An SDN virtualization architecture with flexible hypervisor function allocation." in *Proc. IFIP/IEEE Int. Symp. IM Netw.,* 2015, pp. 397-405.

[7] A. Basta, *et al.*, "Towards a dynamic SDN virtualization layer: Control path migration protocol." in *Proc. Int. Workshop Manage. SDN NFV Syst. CNSM*, Barcelona, Spain, 2015, pp. 1-6.

[8] A. Basta, *et al.*, "HyperFlex: Demonstrating control-plane isolation for virtual software-defined networks." in *Proc. IFIP/IEEE Int. Symp. IM Netw.,* 2015, pp. 1163-1164.

[9] K. Maciej, P. Peresini, and D. Kostic. "What you need to know about SDN control and data planes." in No. EPFL-REPORT-199497. 2014.

[10] C. Rotsos, *et al.*, "OFLOPS: An Open Framework for OpenFlow Switch Evaluation." in PAM. Vol. 7192. 2012.

[11] C. Sieber, *et al.*, "Online resource mapping for SDN network hypervisors using machine learning." in *Proc. NetSoft Conference and Workshops.* IEEE, 2016.

[12] Basta, Arsany, et al. "Logically Isolated, Actually Unpredictable? Measuring Hypervisor Performance in Multi-Tenant SDNs." arXiv preprint arXiv:1704.08958 (2017).