sina-pub.ir


# Graph Abstraction Based Virtual Network Management Framework For SDN

Yoonseon Han*, Jonghwan Hyun†, James Won-Ki Hong†
*Division of IT Convergence Engineering, POSTECH
{seon054}@postech.ac.kr
†Department of Computer Science and Engineering, POSTECH
{noraki, jwkhong}@postech.ac.kr

***Abstract*—Currently, Software Defined Networking (SDN) paradigm has attracted a significant interest from industry and academia as a future network architecture. The programmable control plane enables flexible and rapid modifications of network behavior. With OpenFlow, one of the promising SDN protocols, we can provide several advantages to develop advanced network virtualization technologies. In this paper, we propose a graph abstraction based virtual network management framework that covers from tenant requirement specification to virtual network embedding on physical resources. The fundamental idea is to automatically create and manage virtual networks on the top of SDN technologies from high level tenant requirements. We expect that the proposed framework can overcome the limitation of overlay based network virtualization technologies, and leverage the deployment of SDN technologies.**

***Keywords*—*Network Virtualization, Software Defined Network, Graph Abstraction, Virtual Network Embedding***

## I. Introduction

In recent years, Software-Defined Networking (SDN) has received a lot of attention by networking researchers, vendors, standard organizations, data center operators and telcos around the world. SDN has been envisioned to provide various benefits to operators such as programmability, flexibility, elasticity, and reduction of CAPEX/OPEX. SDN is defined as "the physical separation of the network control plane from the forwarding plane, and where a control plane controls several devices" by Open Networking Foundation (ONF) [1]. OpenFlow [2], which is one of the most promising implementations of SDN, enables the network controller to determine the path of network packets through the OpenFlow enabled switches. With OpenFlow, network operators can manage network traffic in a flow level with a centralized global view of the network at the controller. SDN brings various advantages such as programmability, agility, flexibility, reduced CAPEX/OPEX.

Network Virtualization (NV) is a method of providing an illusion of a dedicated network on top of the hardware resources, and it allows the sharing of the same hardware resources among multiple users without any interference among virtual networks. By creating logical Virtual Networks (VNs) that run on top of a physical network, it allows to separate network functionality from physical equipment. NV technology can bring various advantages such as multi-tenancy, flexibility, scalability, agility, and reduced CAPEX/OPEX. Unfortunately, most of commercial NV solutions are focused on data-center networks based on overlay approaches such as VMware NSX [3] and Microsoft Hyper-V [4]. These overlay based NV approaches construct multiple VNs as overlay network using tunnelling method. This approach is simple to deploy, and requires no network configuration changes. However, they suffer from tunneling overhead and requiring special hardware (or software) such as virtual switches or derivers.

SDN and OpenFlow can provide several advantages to develop advanced NV technologies because it provides the abstraction on various switch hardware. With OpenFlow, we can assign a part of physical resources for each VN via segmented flow tables, slices. We call this approches as slice based NV approach. Currently, several solutions based on network slice are available such as FlowVisor [5], OpenVirteX [6], and FlowN [7]. The slice based approaches accompany no tunneling overhead, and provide strong QoS and SLA control. The main shortage is that they require OpenFlow based network infrastructure. Now, these solutions are in incubation or prototyping stage. Moreover, they are suffer from manual VN configuration, lack of VN embedding method, and no automated VN management process.

In this paper, we propose a graph based virtual network management framework based on SDN. The fundamental idea is to automatically manage VNs from high-level tenant requirements. More specifically, the proposed approach will address following problems.

- How to create VNs on top of SDN networks?
- How to specify VN requirements in high level?
- How to compose VN from requirements automatically?
- How to embed the composed VNs optimally?
- How to dynamically re-embed VNs to address requirement changes or network failure?

## II. Our Approach for Network Virtualization

The proposed framework is composed of three main component. *VN composer*, *VN manager*, and *Virtualization layer*. *VN composer* provides a set of user interface to specify tenant requirement in high level, and automatically compose VN that satisfy the requirements. *VN manager* have responsibility to embed and manage the composed VNs into physical resources, and monitoring the requirement violations. Lastly, the main

This work was supported by Institute for Information & communications Technology Promotion(IITP) grant funded by the Korea government(MSIP) (No. B0190-15-2012, Global SDN/NFV OpenSource Software Core Module/Function Development)

978-1-4673-9955-5/16/$31.00 ©2016 IEEE

sina-pub.ir

role of *network virtualization* layer or network hypervisor is to provide address, topology, and policy isolation in terms of network slice. For this purpose, we consider to reuse existing slice based NV tools such as ON.Lab's OpenVirtex. The overall architecture is described in Fig. 1.

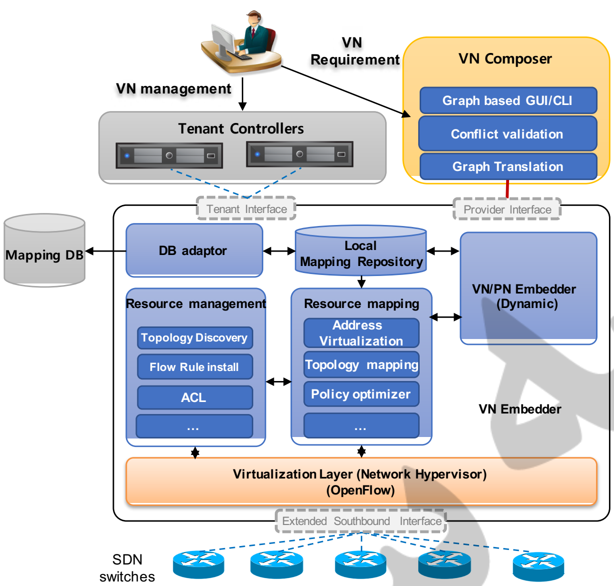


Fig. 1. The overall architecture

## A. *High-level requirement specification*

The starting point of the proposed framework is a high-level virtual network requirement specification. Our framework provides high level network abstraction model such as host, switches, link, and middle-boxes. Using those entities, a tenant can specify their requirements as graph drawing. We believe that the most familiar and intuitive way to express the requirement is drawing. With these interface, tenant can specify own requirements step-by-step rather than designing whole network structure at once that is complex and error-prone task.

## B. *Automated virtual network composition*

The specified requirements are aggregated into a whole VN structure including network address, topology, and policies. To aggregate and translate high level requirements, we plan to use network model based graph translation. Moreover, we expect that a concrete network model can provide capabilities to validate the requirements, and to verify the consistency of the generated VN formally. The translation process is depicted in Fig. 2.

## C. *Optimized virtual network embedding*

One of the important issue for NV is to find out the optical mapping between VNs and physical resources such as network bandwidth, switches' CPU and memory usage, and network traffic patten. Finding a optimal mapping is a NP hard problem, and requires a tremendous computation time. Moreover, we should consider different objectives and characteristics of each tenant. To address these problems, a multi objective heuristic algorithm for VN embedding is necessary.

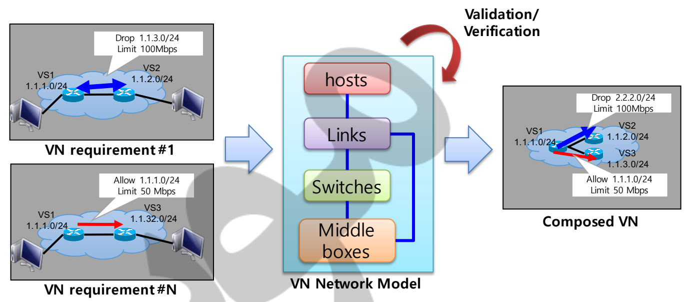


Fig. 2. Automated virtual network composition from requirements

## D. *Dynamic re-embedding and virtual network migration*

The specified requirements of tenants can be changed frequently such an VN scale up/down, and removal. Moreover, we also consider the resilience of VNs from physical network failure. To maintain both VNs and physical networks, we need to a method to dynamically re-embed VNs. To realize this approach, we also need to monitor QoS of VNs and physical resources. When QoS violation and failure is detected, automatically migrate embeded VNs to other places to reduce service downtime as short as possible.

# III. Conclusion and Future Work

In this paper, we proposed a graph based virtual network management framework for SDN. OpenFlow-based network virtualization will be one of the most effective SDN services because it can overcome several limitations of overlay network based network virtualization technologies such as the absence of visibility between virtual networks and physical networks, hard to handle performance degradation and failover issues, and cannot fully manage both virtual and physical networks. By realizing the proposed framework, we expect following contributions, 1) reduced VN design complexity using user friendly VN requirement specification interfaces in high-level, 2) model based consistency and conflict checking of requirements, 3) separation of complex VN embedding and management process from low-level commands and rules, 4) automated VN and physical resource mangement by automating VN embedding and monitoring.

# References

[1] "Open Networking Foundation," https://www.opennetworking.org/, [Online; accessed Oct. 22, 2013].

[2] *OpenFlow Switch Specification Version 1.0.0*, Open Networking Foundation, Std., Dec. 31, 2009.

[3] VMware NSX, "The platform for network virtualization." [Online]. Available: https://www:vmware:com/files/pdf/products/nsx/VMware-NSX-Datasheet:pdf

[4] A. Velte and T. Velte, *Microsoft Virtualization with Hyper-V*, 1st ed. New York, NY, USA: McGraw-Hill, Inc., 2010.

[5] R. Sherwood *et al.*, "Flowvisor: A network virtualization layer," Tech. Rep., 2009.

[6] Al-Shabibi *et al.*, "Openvirtex: Make your virtual sdns programmable," in *Proceedings of the Third Workshop on Hot Topics in Software Defined Networking*, ser. HotSDN '14. New York, NY, USA: ACM, 2014, pp. 25–30.

[7] D. Drutskoy *et al.*, "Scalable network virtualization in software-defined networks," *IEEE Internet Computing*, vol. 17, no. 2, pp. 20–27, 2013.