2019 IEEE 12th International Conference on Cloud Computing (CLOUD)

FAVE: Bandwidth-aware Failover in Virtualized SDN for Clouds

Heesang Jin†, Gyeongsik Yang†, Bong-yeol Yu, Chuck Yoo
Department of Computer Science and Engineering
Korea University
Seoul, Republic of Korea
{*hsjin, ksyang, byyu, chuckyoo*}*@os.korea.ac.kr*

***Abstract*—Network virtualization based on SDN has gained attention in cloud networking. However, existing studies have not provided any failover technique in the event of physical link failure. We propose FAVE, which provides seamless failover and bandwidth-aware protection. FAVE carefully allocates backup routes to handle both failure and interference between tenants. Evaluation shows that FAVE is effective. To our knowledge, FAVE is the first attempt to address failover in virtualized SDN environments.**

***Keywords*-Software-defined networking; Network virtualization; Failover; Traffic engineering;**

I. Introduction

Network virtualization (NV) becomes necessary in datacenters to provide network connections between users' virtual resources, such as virtual machines or containers [1]. Among the various schemes of NV, the use of software-defined networking (SDN) has opened a new direction for NV. SDN is a promising technology that separates the packet forwarding and control functions (e.g., routing) of a switch and centralizes the control function into a controller. The controller enables central management and abstracts various physical network switches. SDN-based NV (SDN-NV) has been proposed [2]; it creates multiple virtual networks for tenants with SDN. Various datacenters have adopted SDN-NV, as it allows tenants to directly set the forwarding routes between virtual machines [3].

Several SDN-NV studies have introduced extensions of NV services, such as address [4] and topology virtualization [2]. Yet, there are very few studies that consider the reliability of SDN-NV. In particular, because cloud networking fabric frequently suffers from link failure [5], handling link failure is the requirement of cloud network services [6]. However, to the best of our knowledge, existing SDN-NV studies have not addressed the failover to recover the link failure.

†The first two authors contributed equally to this paper.

*This work was supported by Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korean government (MSIT) (No. 2015-0-00288, Research of Network Virtualization Platform and Service for SDN 2.0 Realization, and No. 2015-0-00280, (SW Starlab) Next generation cloud infra-software toward the guarantee of performance and security SLA).

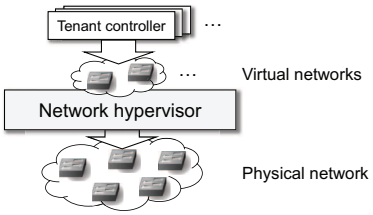

Figure 1: SDN-based network virtualization

In traditional networking, to cope with link failure, failover techniques have been proposed. These techniques install a backup route to forward packets in case of failure. However, in SDN-NV, such failover techniques can cause throughput degradation. Through an experiment, we observe that when the existing failover technique is implemented in SDN-NV, the throughput of tenants can significantly degrade (discussed in detail in Section II-C).

To solve the aforementioned problem, this paper presents FAVE, which provides seamless failover. FAVE is seamless in that it hides all the physical link failure events from tenants and achieves failover so that the tenants do not experience performance degradation even in cases of link failure. In addition, FAVE provides bandwidth-aware protection, which means that the backup route for the failover is calculated with the bandwidth requirements from tenants. This scheme is designed to avoid interference between the backup route and the forwarding routes from tenants.

The remainder of this paper is organized as follows. Section II presents the background and motivation of this work. Section III presents the design of FAVE. The evaluation results are provided in Section IV. Finally, the conclusions and future work of this study are discussed in Section V.

II. Background and Motivation

A. Background: SDN-based Network Virtualization

SDN-NV comprises two components as shown in Fig. 1: a network hypervisor (NH) and tenant controllers. The NH is the main component of SDN-NV and creates virtual switches comprising virtual networks for tenants [2]. Tenant controllers control the created virtual switches, and ONOS and OpenDayLight are well-known examples. The tenant controller calculates a route, which is the end-to-end packet

2159-6190/19/$31.00 ©2019 IEEE
DOI 10.1109/CLOUD.2019.00092

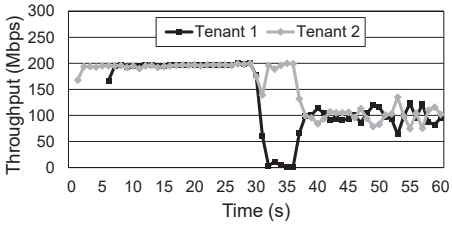

Figure 2: Throughput of each tenant when the protection technique is applied

forwarding path between hosts within a virtual network. The calculated route also determines the flow rules. These flow rules are then sent to the NH.

An advantage of SDN-NV is the address isolation between tenants [4], which is achieved by translating the virtual IP and MAC addresses into the corresponding physical IP and MAC addresses. This enables tenants to select arbitrary addresses for their virtual networks.

B. Background: Failover

Restoration and protection are two well-known network failover techniques [7], both of which assume that the network has redundant forwarding routes between hosts to address link failure. The restoration technique calculates and installs the forwarding routes between hosts after a failure occurs. In contrast, protection installs the backup route using a routing algorithm such as shortest path algorithm before a failure occurs. Although the failover techniques are classical research topics in the networking field, existing NHs do not support the aforementioned techniques despite the necessity for highly reliable networking service.

In this paper, FAVE adopts the protection technique because protection is known to guarantee bounded failover time for reliable user service [7].

C. Motivation: Need for Bandwidth-awareness

We conduct experiments to demonstrate the need for bandwidth-awareness for FAVE. We set up two tenants in a 4-ary fat-tree topology, each with a linear virtual network topology (five switches) connecting two virtual machines. Every physical link is permitted to transmit at a maximum throughput of 200 Mbps. The workload is that two tenants send UDP traffic of 200 Mbps for 60 s. Then, we make one physical link fail at 30 s so that the route of tenant 1 (edge layer in fat-tree topology) becomes unavailable. Figure 2 shows the throughput of two tenants. Tenant 1 suffers a network breakdown for 5 s (30-35 s).

We find that this is because the backup route of tenant 1 overlaps with the route of tenant 2, and the root cause is that the routing algorithm does not consider whether the route is occupied by another tenant. From 40 s, after the switches handle two tenants, tenant 1 can use the same route as tenant 2. However, both tenants undergo 50% throughput degradation. To overcome this problem, FAVE introduces bandwidth-awareness to the protection technique, which is explained in Section III.

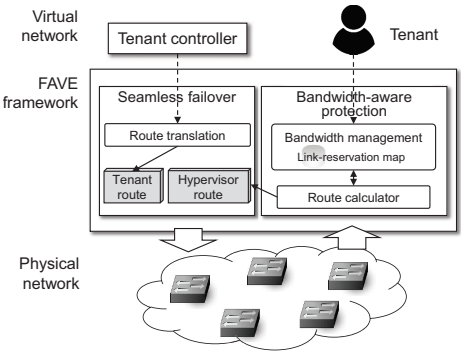

Figure 3: Components of FAVE

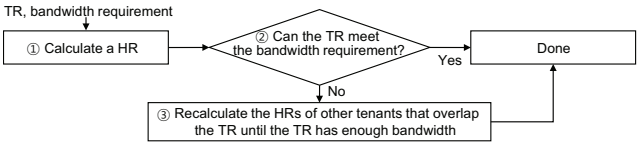

Figure 4: Flowchart of route calculator

III. Design

Figure 3 shows the two main components of FAVE: seamless failover and bandwidth-aware protection. Seamless failover enables physical link failover in the NH and hides the failure event from tenants, and bandwidth-aware protection calculates additional routes for failover by considering the bandwidth requirements of tenants.

A. Seamless Failover

Seamless failover first receives a route calculated from a tenant controller. As this route is calculated within the virtual network, the address or switch should be translated to be compatible with the physical network. Route translation module (Fig. 3) performs this operation. We refer to the tenant-created route as "tenant route" (TR). In addition, FAVE creates an additional backup route, called "hypervisor route" (HR), which is calculated from bandwidth-aware protection component (discussed in Section III-B). When the HR and TR are ready, they are installed in physical switches. As a result, if the TR fails, the HR is used as a backup route for the TR, and the tenant remains unaware of the failure event.

B. Bandwidth-aware Protection

The purpose of bandwidth-aware protection component is to calculate HRs considering TRs and bandwidth requirements from tenants. Note that FAVE employs existing SDN controllers such as ONOS, and they do not accept bandwidth requirements. Therefore, FAVE adds bandwidth management module (Fig. 3) that receives the bandwidth requirements from the tenants and stores them. In addition, to calculate the HRs while considering the bandwidth of physical links,

bandwidth management module maintains link-reservation map that stores the currently available bandwidth of each link.

Route calculator module (in Fig. 3) performs two major tasks per TR as Fig. 4: 1) generating a HR and 2) checking whether the TR can forward packets according to its bandwidth requirement. First, based on the bandwidth requirement, this module calculates a HR using the shortest path algorithm (①). The algorithm excludes two kinds of links in the calculation: 1) links that are already contained in the TR because the calculated HR should not overlap with the TR for failover and 2) links whose available bandwidth is insufficient to meet the bandwidth requirement. In this manner, the calculated HR can function as the backup route of the TR, and it can meet the bandwidth requirement.

Route calculator also checks whether the TR can forward packets while meeting the bandwidth requirement (②). The TR forwards the packets before link failure, so the TR should be able to satisfy the bandwidth requirement. However, this cannot be achieved when the existing HRs of other tenants overlap with the TR because the HRs interfere with the TR in the case of link failure. To avoid the interference, route calculator considers existing HRs to determine whether the TR can meet the bandwidth requirement. If not, this module recalculates the HRs to other paths until the TR meets the bandwidth requirement (③).

IV. Evaluation

Here, we present the evaluation results in terms of seamless failover and bandwidth-awareness. FAVE is implemented on OpenVirteX [2] and evaluated with the same topology as described in Section II-C. We deploy ONOS as a tenant controller. All results are measured three times, and the average values are presented in this paper.

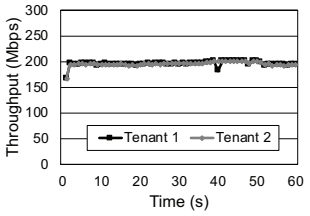

Figure 5: Throughput measured on the end hosts during the experiment

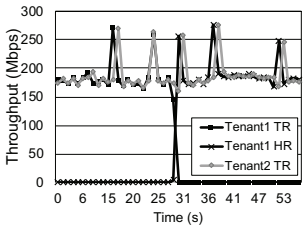

Figure 6: Per-route throughput

A. Seamless Failover

Figure 5 shows the throughput of each tenant during the 60 s experiment. Although we break the link for tenant 1 at the 30 s, Fig. 5 shows that tenant 1 does not suffer any network breakdown. Moreover, both tenants do not suffer any performance degradation despite physical network failure. This result shows that FAVE successfully performs seamless failover that hides physical failures, and it also provides reliable virtual network services.

B. Bandwidth-awareness

Figure 6 shows the per-route throughput (TR and HR per tenant) measured at the core switches in the fat-tree topology. It shows that when the link of tenant 1 fails at 30 s, the HR of tenant 1 forwards packets instead of its TR. Thus, the throughput remains nearly the same even in the failure. In addition, because FAVE allocates the HR of tenant 1 onto another route to avoid performance interference with the TR of tenant 2, tenant 2 does not experience performance degradation, either. The comparison of these results with Fig. 2 shows that FAVE can handle link failure and avoid interference between tenants. The throughput of tenant 1 and 2 remains unaffected, even in the event of a failure.

V. Conclusion

In this paper, we propose FAVE, seamless failover technique that considers the bandwidth requirements of tenants. FAVE stabilizes the throughput of tenants when link failures occur. FAVE is implemented on an open-source NH, and its core functionalities hide physical failures from tenants to avoid throughput interference between tenants.

In the future, we plan to integrate FAVE with traffic load balancing techniques in the virtualized SDN environment.

References

[1] T. Koponen, K. Amidon, P. Balland, M. Casado, A. Chanda, B. Fulton, I. Ganichev, J. Gross, P. Ingram, E. Jackson *et al.*, "Network virtualization in multi-tenant datacenters," in *11th USENIX Symposium on Networked Systems Design and Implementation (NSDI 14)*, 2014, pp. 203–216.

[2] A. Al-Shabibi, M. De Leenheer, M. Gerola, A. Koshibe, G. Parulkar, E. Salvadori, and B. Snow, "OpenVirteX: Make your virtual SDNs programmable," in *Proceedings of the third workshop on Hot topics in software defined networking*. ACM, 2014, pp. 25–30.

[3] G. Yang, B. Y. Yu, W. Jeong, and C. Yoo, "FlowVirt: Flow rule virtualization for dynamic scalability of programmable network virtualization," in *11th IEEE International Conference on Cloud Computing, CLOUD 2018*. IEEE Computer Society, 2018, pp. 350–358.

[4] B.-y. Yu, G. Yang, K. Lee, and C. Yoo, "Aggflow: Scalable and efficient network address virtualization on software defined networking," in *Proceedings of the 2016 ACM Workshop on Cloud-Assisted Networking*. ACM, 2016, pp. 1–6.

[5] A. Roy, D. Bansal, D. Brumley, H. K. Chandrappa, P. Sharma, R. Tewari, B. Arzani, and A. C. Snoeren, "Cloud datacenter SDN monitoring: Experiences and challenges," in *Proceedings of the Internet Measurement Conference 2018*. ACM, 2018, pp. 464–470.

[6] R. Buyya, S. N. Srirama, G. Casale, R. Calheiros, Y. Simmhan, B. Varghese, E. Gelenbe, B. Javadi, L. M. Vaquero, M. A. Netto *et al.*, "A manifesto for future generation cloud computing: research directions for the next decade," *ACM computing surveys (CSUR)*, vol. 51, no. 5, p. 105, 2018.

[7] P. C. da Rocha Fonseca and E. S. Mota, "A survey on fault management in software-defined networks," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2284–2321, 2017.