An Efficient Traffic-aware Scaling Mechanism for Service Function Chains in NFV

Venkatarami Reddy Ch, Bheemarjuna Reddy Tamma and Antony Franklin A
Department of Computer Science and Engineering, Indian Institute of Technology Hyderabad, India
Email: [cs17resch01007, tbr, antony.franklin]@iith.ac.in

***Abstract*—Using the Service Function Chaining (SFC), traffic is steered to a specific set of Service Functions (SFs) based on its type and policy. To handle the massive increase of user traffic by using current deployment models is costly and time consuming. By leveraging advanced technologies such as Network Functions Virtualization (NFV) and Software Defined Networking (SDN), SFs can be deployed and operated more quickly with effective resource utilization which is essential for 5G networks. In this paper, we propose an efficient scaling mechanism for SFC in a holistic way by considering how each SF is involved in various SFCs and ratio of traffic flow between different SFs.**

I. Introduction

Service Function Chaining (SFC) is an ordered set of Service Functions (SFs) which guides subsequent traffic through them to provide different end-to-end services. In traditional networks, each SF is built on middleboxes which are deployed and maintained manually based on requirements at fixed locations. Due to the rapid explosion of mobile and IoT devices, the amount of traffic getting generated from users is increasing significantly. In order to handle such massive increase in the user traffic, network operator installs more middleboxes. However, this approach is not efficient with respect to installation time and cost. With emergence of technologies such as Network Functions Virtualization (NFV) and Software Defined Networking (SDN), it is possible to dynamically create virtualized instances of SFs (*i.e.,* Virtual Network Functions (VNFs)). These VNFs in turn, interconnect to create SFCs which efficiently utilize the resources, there by lowering the OPEX (OPerational EXpenditures) and CAPEX (CAPital EXpenditures).

The main components of SFC architecture defined by the IETF SFC WG [1] and the ONF organization [2] are classifier, service function forwarder, SF and SFC proxy. The SFs in a SFC may be overloaded and behave unexpectedly due to increase in the user traffic, thereby resulting in a performance deterioration to the service continuity. Hence, it is required to monitor the health of these SFs in frequent interval and do scaling to maximize service availability to end users. Existing works in literature, perform scaling at the SF level (*i.e.,* scaling the SF which demands more resources) in isolation based on a specific metric (*i.e.,* CPU load or network traffic) is greater than or lower than predefined threshold [3]. This may result in the adverse effect on the SFCs passing through that particular SF.

If a SF shared by multiple SFCs is overloaded or down, then subsequent SFs are also affected. Without considering this if we do the scaling, immediate scaling of subsequent SFs in less time interval may cause service interruption because SDN controller has to propagate new rules to different components in the short intervals of time. Hence, considering the participation count of a SF (*i.e.,* how many SFCs need this SF) in order to mitigate the overloaded scenario is of supreme importance.

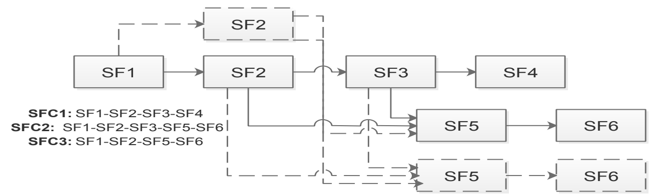

Fig. 1: Scaling of SFs in a holistic way of multiple SFC scenario.

II. Proposed Work

To address the aforementioned problem, in this work, we propose scaling mechanism in a holistic way by considering how each SF is involved in various SFCs. While taking the decision for scale out or scale in, the updated orchestrator will also consider the current threshold values of other subsequent SFs and their dependencies into account and do scaling the required SFs so that services can be provided without any interruption. This can be achieved by setting different threshold rules to different NFs and considering the fraction of traffic flows between SFs.

For example, as shown in Fig. 1, it consists of three different SFCs where each SFC comprises a particular set of SFs connected in order. Each square represents a SF connected to another SF. SF2 is shared by three different SFCs. When SF2 is overloaded, orchestrator creates a new instance of SF2 and also two other instances SF5 and SF6 based on fraction of traffic flows between SFs and current threshold values of it as shown in dotted squares in the same figure.

III. Way Forward

Currently, we are creating an experimental platform for creating NFV enabled network using the FOKUS Open Baton toolkit on top of OpenStack cloud infrastructure. OpenDaylight controller is being used for SDN.

References

[1] E. Halpern, J. and E. C. Pignataro, “Service Function Chaining (SFC) Architecture,” RFC 7665, DOI 10.17487/RFC7665, October 2015.
[2] “ONF White Paper. ONF TS-027: L4-L7 Service Function Chaining Solution Architecture,” June 2015.
[3] A. M. Medhat *et al.*, “Orchestrating Scalable Service Function Chains in a NFV Environment,” in *IEEE conference on NetSoft*, July 2017.