Attachment A: Branching-Aware Service Function Placement and Routing in Network Function Virtualization

Issues with Traditional Middle-boxes: In traditional enterprise networks, network functions, namely middle-boxes, are purpose-built in proprietary appliances. Network traffic flows are steered through middle-boxes of respective functions in a predetermined order to enhance network security (e.g., Firewall, Intrusion Detection System (IDS)), and/or network performance (e.g., Load Balancers that balance the traffic load). In the example shown in Fig.1, the traffic flow from working-from-home employees to the company server needs to go through three middle-boxes: an NAT (Network Address Translator), a Firewall, and an IDS for enhanced security. The number of middle-boxes in a typical enterprise network are considerable and can be in the same order of magnitude as rudimentary network components (i.e., switches and routers). This type of purpose-build middle-boxes leads to huge challenges to the network management. Given the wide deployment of middle-boxes, the malfunctions of middle-boxes have became a significant source of network failures in both the dimensions of scale and frequency. A bug in a load balancer update at *Google* caused the service degradation for up to 40% of *Gmail* customers in 2012 [1]. According to a recent field study, middle-boxes contribute to up to 43% of high-severity incidents in data centers [2]. These issues are further aggravated by the fact that middle-boxes are error-prone and hard to repair as it is required to possess ad-hoc expertise for the configuration/repairment of proprietary middle-boxes.

Network Function Virtualization: Network Function Virtualization (NFV) is considered to be the foremost solution to address above issues [3]. The basic idea of NFV is to create a software instance to implement the functionality of each middle-box, which can be considered as a virtualized appliance. Deploying a middle-box (say a Firewall) in NFV is transformed to running a virtual function (i.e., a Firewall software) on any commodity hardware (that replaces purpose-built proprietary appliances). The advantage of NFV may be conveyed in an analogy: assume that an iPhone's functionalities are all virtualized as a software instance on (cost-efficient) commodity hardware, then iPhone software failures can be repaired by simply rebooting the software instance; and iPhone hardware failures can be addressed either by fixing the commodity hardware (without the need for ad-hoc knowledge) or running an instance on another hardware.

Service Function Placement and Routing in NFV: Consequently, in NFV, network traffic flows need to pass through a group of virtual software instances (instead of middle-boxes) in a given order. In the literature, the set of virtual instances are referred to as a *service function chain* [3], and are represented with a *Forward Graph*. For instance, the service function chain that corresponds to Fig.1 can be captured by a forward graph in Fig. 2, where I, O corresponding to the entry (i.e., home) and exit point (i.e., company server) of the chain, and three functions a, b, c represent the functionalities of NAT, Firewall and IDS in Fig.1, respectively. To instantiate the service function chain, it has to be mapped to the physical networks that consist of connected commodity hardware. An example of the physical network is shown in Fig. 3, where there are six commodity servers (each node has have 100 units of computing resources except Node B) that are connected by eight links (each link has 10 Gb/s of bandwidth except Link A-B). Given a service function chain, it is critical to decide where to place the required virtual functions, i.e, the *placement* problem, and how to steer the traffic to pass through the placed functions, i.e. the *routing* problem. An example of placement and routing decision is shown in Fig. 4: functions f_1 to f_3 are created in Nodes A, C, and E, respectively (assuming that the entry and exit points are fixed at Nodes A and F, respectively); and the traffic flow are steered through Nodes A, C, E, F in sequence as indicated by the routing path in the dash line in Fig. 4. Note that link A-B was avoided in the routing decision as which has no sufficient bandwidth. In our prior work, we have studied the primitive version of the placement and routing problem in NFV.

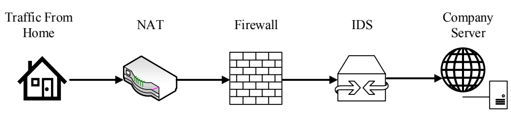

Fig. 1. Example of Chained Services through Middleboxes

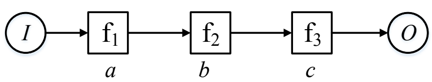

Fig. 2. An Example of Forward Graph

Branch-Awareness in NFV: In the literature, it is generally assumed that the network flow continues to *one* and *exact one* next function (e.g., function f_2 in Fig. 2) after being processed by one network function (i.e., function

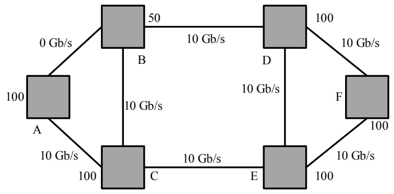

Fig. 3. An Example of Physical Network

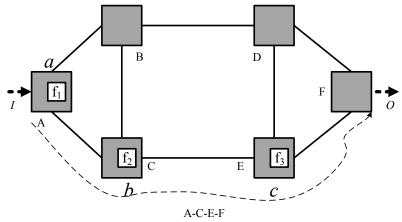

Fig. 4. An Example of Function Placement and Routing

f_1). This assumption naturally leads to a straightforward linear chain of virtual functions as in Fig. 2. Our recent investigation, however, has identified a considerable number of use cases with presence of traffic branching, which render above assumption no-longer valid. These use cases can be classified into three major categories. First, certain applications distinguish and ”label” traffic flows, and flows with different labels are treated in different manners. For instance, a DPI (Deep Packet Inspector) in Fig. 5 classifies the traffic flow into video and non-video traffic where only the former is further forwarded through a video optimizer (VO) function. Second, with the presence of load balancer, network flow can fork into multiple branches (e.g., Web traffic and Non-Web traffic) with reduced traffic rate per branch as shown in Fig. 6. Third, recent study in [4] reveals the possibility of parallel operating network functions to reduce the overall network delay. As a result, the network flow can travel through multiple parallel branches instead of a chain of functions. For instance, a parallelism of functions in Fig. 2 results into three branches in Fig. 7. Given above new uses cases, to ensure that NFV is still a viable solution for future enterprise network, there is a need to investigate the problem of Branching-Aware Service Function Forward Graph Placement and Routing in network function virtualization, which is the major goal of the proposed project.

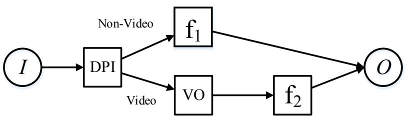

Fig. 5. Traffic Classification

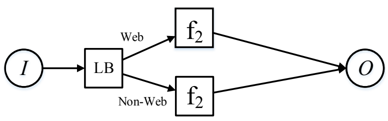

Fig. 6. Load Balancing

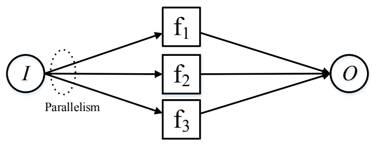

Fig. 7. Parallelism

Plan of Study: Note that above examples appear to be simple only for the ease of reading. In reality, the optimal placement and routing decision can be very challenging. The difficulties of the virtual function placement and routing lies on two facts: first, to achieve the optimal decision, the placement and routing have to be jointly considered. Simply addressing the placement problem and the routing problem in sequence only leads to sub-optimal solution; second, the added branch-awareness indicates that the forward graph is a complex mesh topology (in contrast to a linear chain topology in the literature), which leads to significant complexity to the placement and routing decision. In fact, the studied problem belongs to NP-Complete problems, which are considered to be the hardest type of problems in Computer Science. Given its NP-Completeness, in this project, we plan to develop an *optimal* solution based on the Integer Linear Programming (ILP) transformation that jointly addresses the placement and routing problem. The ILP model can then be resolved with a state-of-the-art ILP solver software - *IBM CPLEX* to find an optimal solution. We will also seek time-efficient near-optimal solutions based on approximation algorithms, which have a guaranteed near-optimal performance. Overall, we expect the solutions found in this project will address the branch-awareness issues that are raised in recent use cases to make NFV a viable solution in the long run.

References

[1] “Why gmail went down: Google misconfigure load balancing servers(updated).” [Online]. Available: http://arstechnica.com/

[2] R. Potharaju and N. Jain, “Demystifying the dark side of the middle: A field study of middlebox failures in datacenters,” in *Proceedings of the ACM IMC*, 2013, pp. 9–22.

[3] NFV, “http://www.etsi.org/technologies-clusters/technologies/nfv,” 2019.

[4] C. Sun, J. Bi, Z. Zheng, and H. Yu, “Nfp: Enabling network function parallelism in nfv,” in *Proceedings of ACM Specical Interest Group on Data Communication Conference*, 2017, pp. 43–56.