

Service Function Chaining on Programmable Data Plane

Gyuyeong Kim and Wonjun Lee
Network and Security Research Lab.
School of Cybersecurity
Korea University, Seoul, Republic of Korea
{gykim08,wlee}@korea.ac.kr

Abstract

Service Function Chaining (SFC) enables advanced in-network packet processing by passing through an ordered set of service functions. However, it is challenging to achieve high performance because chaining functions like service function forwarders generally run on commodity servers, which offer limited throughput and require additional round-trip times. In this paper, we present P4-SFC, a novel SFC architecture that leverages the flexibility of emerging programmable switches for high performance SFC packet processing. P4-SFC achieves low latency and high throughput by migrating the chaining functions to programmable switches. Our benchmark results show that P4-SFC outperforms existing the server-based solution in chain latency.

Keywords: Programmable data plane, Service function chaining, P4

1. Background and Motivation

Service Function Chaining (SFC) [1] has received a lot of attention because it enables network operators to satisfy service requirements through advanced in-network packet processing for modern networks like 5G networks and data center networks. SFC packets are handled by chaining functions, such as the service classifier (SC) and the service function forwarder (SFF). The chaining functions en/decapsulate packets and determine the next service function (SF) in a service chain. Since the chaining functions perform network-intensive jobs, it is important to provide low latency and high throughput for SFC. Unfortunately, the current switch ASICs do not allow users to add a new feature in processing pipelines, and the chaining functions are generally running on commodity servers. This implies the following limitations.

Latency Overhead. Ideally, the packet processing in SFC should be limited to hundreds of nanoseconds, which commodity switches provide. For example, Arista 7050X3 series with the Broadcom Trident-3 ASIC offer the minimum processing delay of 800ns. Unfortunately, in the

server-based SFC, a packet must experience tens to hundreds of microseconds of additional RTTs to be processed in remote SFs and chaining functions. By considering a sub-microsecond delay of the switch, this RTT significantly degrades chain latency, which means the packet processing delay in SFC.

Limited Throughput. A recent approach [2] implements the chaining functions in Open vSwitch (OVS) [4], a vSwitch on the virtualized server where SFs are co-located. Since chaining functions and SFs are in the same node, the chain latency decreases. However, the vSwitch-based approach does not offer high throughput because the vSwitch is still in the server. An optimized server can process only tens of millions of packets per second even with DPDK, a user space library that enables high performance software packet processing [3]. Conversely, modern switching chips like Broadcom Tomahawk can process billions of packets per second.

Bare-Metal Incompatibility. One pitfall of the vSwitch-based SFC is that it cannot support bare-metal (BM) SFs. The approach implicitly assumes that all SFs are virtualized. Although recent SFs have been virtualized, there still exist many legacy SFs running on dedicated BM hardware. If the network uses the vSwitch-based SFC, applicable SFs are limited. Otherwise, the BM SFs must be replaced to the virtualized ones at additional cost.

2. P4-SFC Design

Traditional switching chips provide only fixed functions. A new feature like SFC cannot be implemented without burdensome capital and engineering costs. In recent years, we witness a rapid growth of programmable switching chips. Unlike the fixed function ASICs, programmable switch ASICs like Barefoot Tofino and Cavium Xpliant allow users to customize the packet processing pipeline with high-level programming language like P4. We can manipulate packet headers and perform customized operations by designing Match-Action (M-A) tables.

Motivated by the flexibility of programmable switches, we design P4-SFC, a high performance SFC design leveraging programmable switches. P4-SFC migrates the chaining functionality to emerging programmable switches. Since the switch can act as a SC

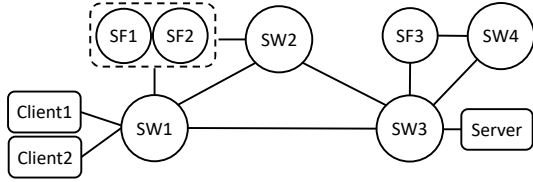


Fig. 1 Example for P4-SFC.

and SFF simultaneously, SFs and switches are the basic building blocks for the network. This brings the following benefits.

- We inherit the high throughput of switch hardware. Billions of SFC packets can be processed per second with a sub-microsecond per-packet processing delay.
- Since the chaining functions run on the switch, the packet does not experience RTTs to be processed in the chaining function server. This greatly reduces the chain latency.
- Unlike the vSwitch-based SFC, the P4-SFC does not place chaining functions and SFs together. Therefore, BM SFs are also compatible.

Table 1 shows the differences between SFC designs.

Table 1: Achieved requirements by SFC designs

Requirements	Server	vSwitch [2]	P4-SFC
Low latency	X	○	○
High throughput	X	X	○
BM compatibility	○	X	○

Fig. 1 illustrates an example of P4-SFC. We can see that no server exists for chaining functions. Instead, all of switches can perform the chaining functionality. The dashed line surrounding SF1 and SF2 represents a virtualized server. SF3 runs on a dedicated BM server and is blind to SFC. In the server/vSwitch-based SFCs, a service proxy server must exist between SF3 and SW/S4 to process packets at SF3. In P4-SFC, virtualized SFs like SF1/SF2 and BM SFs like SF3 are not distinguished. It is enough to interconnect the switch and the SF. This suggests that P4-SFC is compatible with BM SFs.

3. Performance Evaluation

Implementation. We have implemented P4-SFC using P4 with BMv2. We employ Network Service Header (NSH) in RFC8300 for the SFC header. To forward SFC packets, we implement an tunneling header including the destination ID and next hop ID fields. The first one indicates the next SF to visit and the other one denotes the next physical node to visit.

M-A tables for P4-SFC are implemented in the ingress pipeline. We have implemented three tables. SFC Classifier reads the DSCP field in the IP header, and assigns Service Path Index (SPI) to the packet. In addition, the table encapsulates the packet with NSH and the tunneling header. SFF-SFC Next table reads

SPI and updates the fields in the tunneling header. SFF-SFC Egress table reads the fields in the tunneling header and forwards the packet to the output port.

Benchmark Results. To evaluate the performance of P4-SFC, we generate a baseline network topology using Mininet where two switches, classifiers, SFFs, and SFs between a client and a server. With P4-SFC, this topology can be simplified as a topology where only two switches and SFs exist. In this benchmark, the client sends a 1.5KB ping packet to the server 100 times. We measure the RTTs of the packets (i.e. chain latency) and obtain statistics.

Table 2: Chain latency statistics

	Baseline	P4-SFC
Maximum	17.27ms	17.06ms
Minimum	6.64ms	2.88ms
Average	8.45ms	3.88ms

Table 2 shows the benchmark results. For the maximum latency, we find that P4-SFC outperforms the baseline slightly. For the minimum and average latencies, P4-SFC achieves the significantly better performance than the baseline. For example, P4-SFC is better than the baseline by 2.18x in the average latency. This suggests that SFC on programmable dataplane can improve the chain latency by reducing the number of hops that SFC packets should visit.

4. Conclusion

In this paper, we proposed P4-SFC, a SFC architecture that provides high performance by leveraging programmable switches. We have implemented a P4-SFC prototype using P4 language and demonstrated that P4-SFC delivers low latency through targeted benchmarks. For the future work, we will consider high availability mechanisms for P4-SFC.

Acknowledgements

This research was partly supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIT) (No. 2017-0-00195, Development of Core Technologies for Programmable Switch in Multi-Service Networks), and Next-Generation Information Computing Development Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science and ICT (No. 2017M3C4A7083676). Wonjun Lee is the corresponding author.

References

- [1] J. M. Halpern and C. Pignataro, "Service Function Chaining (SFC) Architecture." RFC 7665, Oct. 2015.
- [2] "Service function chaining in openstack." <https://docs.openstack.org/networking-sfc/latest/>, 2018.
- [3] R. G., "Open vswitch with dpdk overview." <https://software.intel.com/en-us/articles/open-vswitch-with-dpdk-overview>, 2016.