

# A SFC Network Management System in SDN

Li-Der Chou<sup>(✉)</sup>, Chia-Wei Tseng, Hsin-Yao Chou, and Yao-Tsung Yang

Department of Computer Science and Information Engineering,  
National Central University, Taoyuan, Taiwan  
cld@csie.ncu.edu.tw, cwtseng@g.ncu.edu.tw,  
qooqoonick@networklab.csie.ncu.edu.tw, yao.vct@gmail.com

**Abstract.** With the rapid growth of the Internet, more demand for network resources allocation and management of value-added services are appeared. The new concept of network service architecture, Service Function Chain (SFC), has been proposed. The SFC is a capability that uses Software-defined networking (SDN) technology to create a service chain of connected network services and connect them in a virtual chain. This paper designed a VLAN and OpenFlow based Service Function Chain (VOFSFC) system to improve the flexibility of the network service deployment and increase the utilization of service functions.

**Keywords:** SDN · NFV · SFC · Network management

## 1 Introduction

With the rapid growth of the Internet, the network communication architecture becomes relative complicated and large. Traditional network infrastructure combines complex network functions with routers and switches for support more network services. This make the network becomes more complicated and decrease the performance of network severely. The purpose of SDN is to change the physical, complex network into a virtual, programmable and open network architecture. The SDN architecture decouples the network control plane from the data plane, and control the network by SDN controller. The OpenFlow protocol is a foundational element for building SDN solutions [1, 2].

As the Internet becomes more and more sophisticated, different traffic needs to apply different network service functions. Such as the video streaming service needs to apply the firewall, Video Optimization Server, etc. To satisfy the demands on network design and deployment as well as service application, Internet Engineering Task Force (IETF) is developing a new technology called Service Function Chain [3]. The main purpose of SFC is to lead the traffic flow of users to their required services according to users' demands. The SFC approach promises high flexibility and lower costs for network operator at the same time. The paper proposed a VLAN and OpenFlow based Service Function Chain (VOFSFC) system in the SDN environment. The VOFSFC system can identify different requirements of network traffic, steer/deliver packets to service functions and record information of the traffic by the VLAN tag. The VOFSFC system

provides ability to manage the service function paths, and makes creation, deletion and modification of service function to be easier.

The main idea of this research is adopting the VLAN ID tag to identify different service function paths, performed services chain and services decision-making positions which reduce the load on these service functions in the chain. Our major contributions are summarized as follows:

- We proposed a VOFSFC system to manage service function paths in the SDN environment.
- We implemented a SDN experiment environment for development and validation of SFC deployment and experiments.

The remainder of this paper is organized as follows: in Sect. 2, the background and related works are addressed. Section 3 is to describe the design and implement of the VOFSFC system. Section 4 presents the experimental results. The last section concludes this paper and addresses potential future works.

## 2 Background and Related Works

SDN attracted much attention these years. Many researchers are working on this research area [4–6]. SDN architecture decouples the network control plane from the data plane, so that the network administrator can manage the network by the programmable operators [7, 8]. The OpenFlow protocol is the first SDN language to be productized and implemented [9]. The OpenFlow Controller is the brains of the SDN network that can manage the Flow table on the OpenFlow switch by the OpenFlow protocol. Flow table is composed of many flow-entry, and the flow-entry contains three parts, Rule, Action and Stats. Rule is used to define and identify a flow; Action defines the behavior of packets which was matched by the Flow table; Stats gathers the flow information statistic. OpenFlow controller as the control center can be more intelligent, automated and efficient to manage network. There are many open-source project of OpenFlow controller for network operators and researchers, like Ryu [10], Floodlight [11], OpenDaylight [12] and ONOS [13]. Ryu is an open-sourced Network Operating System (NOS) developed and maintained by NTT (Nippon Telegraph and Telephone) in Japan. Ryu provides software components with well-defined API that make it easy for developers to create new network management and control applications. Therefore this research will use the Ryu to develop the VOSFC system.

SFC is a kind of network technology currently under research and development by IETF which is formulated by the work groups of IETF Service Function Chain Working Group (SFC WG) and Source Packet Routing in Network Working Group (SPRING WG) [14]. SFC is being include in many SDN and network functions virtualization (NFV) use cases and deployments, including data centers and carrier networks [15, 16]. Many research have propose various implementation and application for SFC. Network Service Header (NSH) [17] is an IETF draft pro-posed to address the Service Function Chaining based on the SFC encapsulation to support the SFC architecture as outlined in the RFC7665 [18]. In the framework of NGSON (Next-Generation Service Overlay

Network), Service Overlay Network (SON) is a new network service framework which is formed by the combination of original service-oriented architecture and service delivery platform to realize service functions [19–21]. described a use case of SFC, the authors proposed a security service on-demand system that offers a security service function chain that enables ICT service providers to provision a dynamic and flexible secure service on the SDN network. To achieve the service function deployment, [22] proposed a greedy algorithm for service function placement. It consider the order of service function to decision the place of service function.

This paper addressed a SFC management system to manage service function paths in the SDN environment. In order to realize the SFC deployment in our experiment, the NFV and SDN technologies are both taken into consideration in the proposed VOFSFC system.

### 3 VLAN and OpenFlow Based Service Function Chain System

Figure 1 shows the system architecture of the VOFSFC system. The system is divided into three parts, the first part is OpenFlow Switch (Pica8 Pronto Open Switch) that act as the service function forwarder role in the network. The second part is the VOFSFC

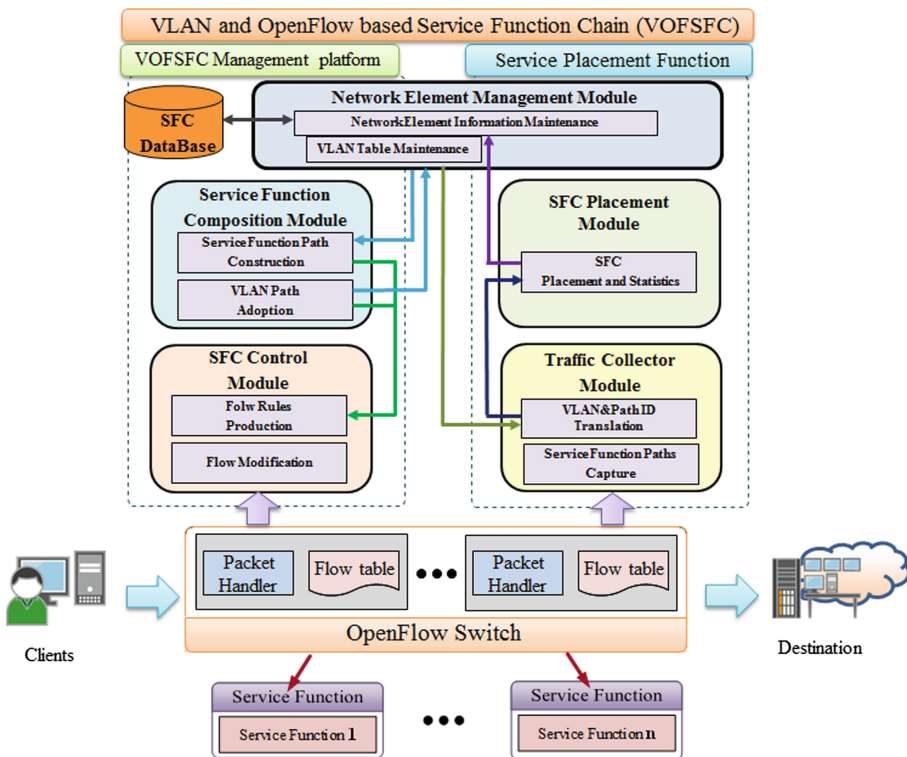


Fig. 1. VOFSFC system architecture

management platform. The third part is the service placement function. There are five different modules in the VOFSFC system.

### 3.1 Network Element Management Module

The Network Element Management Module provides the network element information maintain function and VLAN table maintain function in the VOFSFC system. The network element information maintenance function will regularly access and confirm the *SF\_Infor* and the *Dest\_Server\_Infor* information in the SFC Database. These information will transmit to the service function path construction function and the VLAN&Path ID function to establish the service function paths. The VLAN table maintenance function will communicate with the service function path construction function and the VLAN path adoption function to maintain the VLAN table.

### 3.2 Service Function Composition Module

The Service Function Composition Module provides the service function path construction function and VLAN path adoption function in the VOFSFC system. The service function path construction function will get the service function paths information from the SFC Database and combined with the information obtained by the Network Element Information Maintenance Module to construct the path of the service function to the destination. And request the VLAN Path Adoption function to provide the VLAN\_ID for the service function path. The VLAN path adoption function can receive the *SF\_Path* and *SF\_List* from the service function path construction function. In accordance with the desired SFC path and the demand service functions, calculated the *Path\_ID* for each SFC. The VLAN path adoption function will use this *Path\_ID* to register with the VLAN Table Maintenance function. When the VLAN table maintenance function received queries from the VLAN path adoption function, it will check the VLAN table that the *Path\_ID* is registered in the SFC database or not. If the system have not register before, the system will assign a *VLAN\_ID* to this service function path, and use the *Path\_ID* to register the VLAN table. The VLAN table information will return to the VLAN Path Adoption function and update the SFC Database.

### 3.3 Service Function Chain Control Module

The Service Function Chain Control Module provides flow rules production function and flow modification function in the VOFSFC system. After receiving the information service function path from Service Function Composition Module. The flow rules production function will divide into three cases. One for the packet into the VOFSFC system, add the *VLAN\_ID*, turn to the next service function path; the second from the service function server back to the system packet, through the packet ID on the packet identification, modify the *VLAN\_ID* to send To the next service function; Finally, the service flow will untag the VLAN through the OpenFlow switch and forward to the destination server.

A flow modification function is used to modify the flow rules in the OpenFlow Switch. After receiving the information from the flow rules production function, the flow modification function will add OFPT\_FLOW\_MOD message by OFPFC\_ADD command and remove the old OpenFlow rule through OFPFC\_DELETE. The OpenFlow rule will be configured in the OpenFlow switch. If the Flow table receives the OpenFlow rule from the Flow Modification function, it performs the modification of the Flow table, adds the new flows to the Flow table, and removes the old not applicable flow.

### 3.4 Traffic Collector Module

The Traffic Collector Module provides the VLAN&Path ID Translation function and service function paths capture function in the VOFSFC system. When the packet enters the OpenFlow switch, the Packet Handler on the OpenFlow switch will compare the OpenFlow rule with the VOFSFC system and tag the VLAN\_ID forward to the Service Function Placement Module. When the service function paths capture function receives the packets from OpenFlow switch, it will calculate the *statistic\_VLAN* and *statistic\_Packet* information and send back to the OpenFlow switch. After receiving this packet, the OpenFlow switch will forward it to the next service function based on the VLAN\_ID information. The VLAN&Path ID Translation function will regularly receive the *statistic\_VLAN* and *statistic\_Packet* information from service function paths capture function. Combined with the *SF\_Infor* and the *Dest\_Server\_Infor* provided by the network element information maintenance function, the VLAN&Path ID Translation function can identify the relationship between service function and Path\_ID. Afterwards, the VLAN&Path ID Translation function will generate the *Statistic>Loading* information to the path statistic and prediction function.

### 3.5 SFC Placement Module

The Service Function Placement Module provides the service function placement and statistics functions in the VOFSFC system. After receiving the *Statistic>Loading* information from Traffic Collector Module, the service function placement and statistics functions will count the number of each SFC paths in the system and evaluate the available service functions when new service flows arrive.

In our design, the VOFSFC system provides a Web GUI as user interface. Figure 2 shows the designed VOFSFC system. The graphical user interface allows users to interact with our system where the user can designate the service function path in the experiments or find additional information about SFC status.



Fig. 2. VOFSC system's user interface

## 4 Experiment

The experiment environment is shown in Fig. 3. The OpenFlow Controller is the central manager for this system. The Pica8 OpenFlow switch is act as the service function forwarder in our design. There are two All-in-one Hypervisors worked as the SFC servers to provide service functions. Table 1 summarizes our experimental environment.

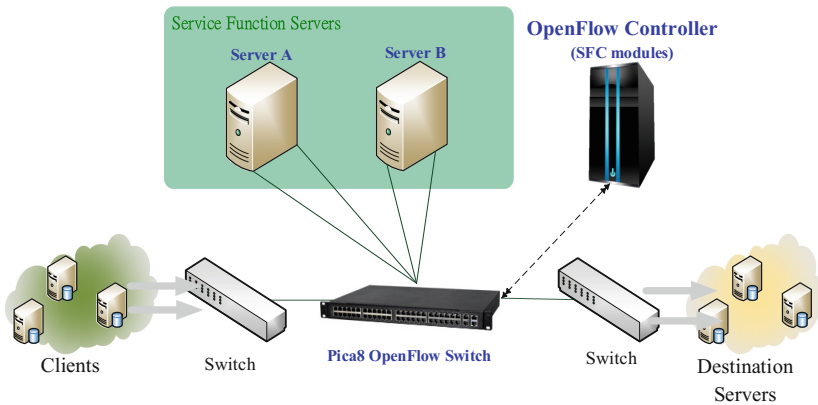


Fig. 3. Experiment environment

When the system is initialized, the OpenFlow controller adds the flow rules of the service function path to the Flow table of the OpenFlow switch. When the client's traffic flow enter the system, the OpenFlow switch will compare the Flow table in the OpenFlow switch and forward to the service function servers based on the flow rules. Then the Flow will start designated SFC routing, since VOFSC system has determined the corresponding SFC routing to the decision, every corresponding service function can

lead flows to next service function after its processing to complete required service chains. After the Flow complete all demand service chains, OpenFlow switch will lead Flow to the destination according its original destination location.

**Table 1.** Hardware and software resource

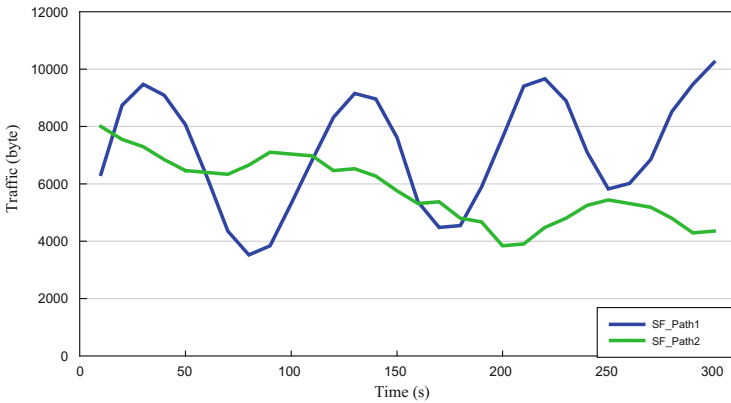
| Hardware/software | SFC server                               | OpenFlow controller (Ryu 3.6)            |
|-------------------|--|--|
| Manufacturer      | ASUSTeK computer INC.                    | Dell INC.                                |
| CPU               | Intel(R) Core(TM) i3-3220 CPU @ 3.30 GHz | Intel(R) Core(TM) i3-4130 CPU @ 3.40 GHz |
| Memory            | DDR-3 1600 4 GB                          | DDR-3 1600 4 GB                          |
| Network interface | TP-LINK TG-3468 Gigabit PCI Express * 2  | TP-LINK TG-3468 Gigabit PCI Express      |
| Operating system  | Ubuntu 12.04.1 LTS                       | Ubuntu 12.04.4 LTS                       |
| OS Kernel         | 3.2.0-29-generic-pae                     | 3.11.0-15-generic                        |

To evaluate the functionality of the proposed VOFSFC system, the experiment designed five different SFC for testing. Table 2 shows the details of different service function path. There are 8 different service functions in this experiment. The existing service functions such as an Service Placement Function (SPF), Monitoring (Mon), Firewall (FW), intrusion detection system (IDS), Web Optimizer (Web), DPI, Video Optimizer (Video) and NAT. In order to achieve the service function placement and traffic statistics, SPF function is necessary for each SFC in our system.

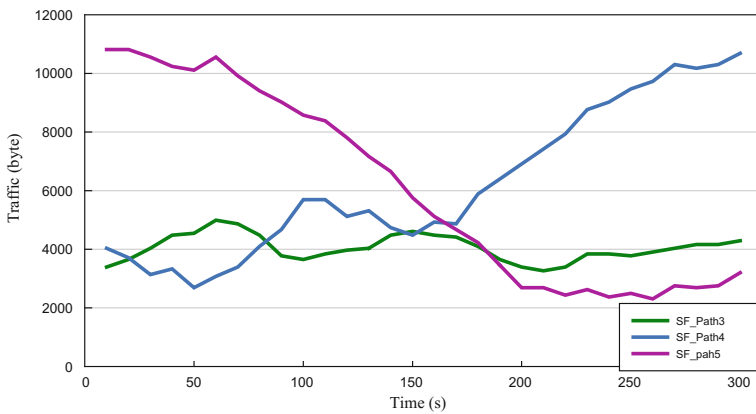
**Table 2.** Service function path table

| SFP          | SFs                           | Path ID |
|--------------|-------------------------------|---------|
| $SF\_Path_1$ | SPF, FW, IDS, NAT             | 23      |
| $SF\_Path_2$ | SPF, Mon, Web, NAT            | 147     |
| $SF\_Path_3$ | SPF, FW, DPI, Video           | 141     |
| $SF\_Path_4$ | SPF, FW, IDS, DPI, Video, NAT | 237     |
| $SF\_Path_5$ | SPF, Mon, FW, Web             | 101     |

In the experiment, the VOFSFC system utilize the service function placement and statistics functions to monitor the status of traffic flow through different SFCs. Figure 4 shows the experiment result between the SF\_Path1 and SF\_Path2. The combination of different service functions is the main factor affects the experiment result. The utilization of the FW and IDS is higher than the Mon and Web service function. Figure 5 shows the experiment result among the SF\_Path3, SF\_Path4 and SF\_Path5. As shown in Fig. 5, the traffic statistics in the SF\_Path4 is higher than SF\_Path3 and SF\_Path5. The result indicates that if a new traffic flow is entered into the system, the SF\_Path3 is the better choice than other paths.



**Fig. 4.** Traffic statistics of  $SF\_Path_1$  and  $SF\_Path_2$



**Fig. 5.** Traffic statistics of  $SF\_Path_3$ ,  $SF\_Path_4$  and  $SF\_Path_5$

## 5 Conclusion and Future Works

This paper proposes a VLAN based Service Function Chain management system in SDN network. The VOFSC system can deploy service functions in the SDN network environment. The system achieves the redirection of network flows and the addition of the needed service functions. The SFC can make a flexible order of the comprehensive services, also practicing SDN/NFV will bring benefits on the operation and deployment of overall service resources.

Our future work will involve more functions and analyses of the VOFSC system. The impact of the VNF movement technology over SDN and study the capacity influenced by different network architecture will be also taken into consideration.



**Acknowledgment.** The work described in this paper was supported in part by Ministry of Science and Technology of the Republic of China. (Project No. 104-2221-E-008-039-MY3 and 105-2221-E-008-071-MY3)

## References

1. Mcdysan D (2013) Software defined networking opportunities for transport. *IEEE Commun Mag* 51(3):28–31
2. OpenFlow. <http://archive.openflow.org/>
3. Service Function Chaining (sfc). <https://datatracker.ietf.org/wg/sfc/documents/>
4. Chou L-D, Tseng C-W, Lai P-H, Hsieh S-Y, Wu M-C (2016) SDN/NFV virtualization testbed with automatic deployment and management functions. In: Proceedings of The second international conference on electronics and software science (ICCESS2016), Takamatsu, Japan, November, pp. 112–121
5. Sieber C, Basta A, Blenk A, Kellerer W (2016) Online resource mapping for SDN network hypervisors using machine learning. In: IEEE NetSoft conference and workshops (NetSoft), Seoul, Korea, pp. 78–82, June 2016
6. Nikbazm R, Dashtbani M, Ahmadi M (2015) Enabling SDN on a special deployment of OpenStack. In: 2015 5th international conference on computer and knowledge engineering (ICCKE), pp. 337–342
7. Casado M, Foster N, Guha A (2014) Abstractions for software-defined networks. *Commun ACM* 57(10):86–95
8. ONF, “SDN architecture,” Open Networking Foundation, Technical report, June 2014. <https://www.opennetworking.org/images/stories/downloads/sdn-resources/technical-reports/TRSD NARCH1.006062014.pdf>
9. McKeown N, Anderson T, Balakrishnan H, Parulkar G, Peterson L, Rexford J, Shenker S, Turner J (2008) OpenFlow. *SIGCOMM Comput Commun Rev* 38(2):69
10. Osrq.github.io, Ryu SDN Framework (2015). <http://osrg.github.io/ryu/>
11. Project Floodlight (2015): Floodlight OpenFlow Controller. <http://www.projectfloodlight.org/floodlight/>
12. Opendaylight.org (2015) OpenDaylight | A Linux Foundation Collaborative Project. <http://www.opendaylight.org/>
13. Onosproject.org (2015) Open Network Operating System. <http://onosproject.org/>
14. Source Packet Routing in Networking (spring). <https://datatracker.ietf.org/wg/spring/documents/>
15. King D, Farrel A, Georgalas N (2015) The role of SDN and NFV for flexible optical networks: current status, challenges and opportunities. In: 2015 17th international conference on transparent optical networks (ICTON), pp 1–6
16. Akhtar N, Matta I, Wang Y (2016) Managing NFV using SDN and control theory. In: NOMS 2016 - 2016 IEEE/IFIP network operations and management symposium, pp 1005–1006
17. Network Service Header. <https://tools.ietf.org/html/draft-ietf-sfc-nsh-11>
18. Service Function Chaining (SFC) Architecture. <https://tools.ietf.org/html/rfc7665>
19. Duan Z, Zhang Z, Hou YT (2003) Service overlay networks: SLAs, QoS, and bandwidth provisioning. *IEEE/ACM Trans Net* 11:870–883 Erl T (2007) SOA: Principles of Service Design, 1st edn., Prentice Hall
20. Pavlovski C (2007) Service delivery platforms in practice. *IEEE Commun Mag* 45(3):114–121

21. Chou L-D, Tseng C-W, Huang Y-K, Chen K-C, Ou T-F, Yen C-K (2016) A security service on-demand architecture in SDN. In: The 7th international conference on information and communication technology convergence (ICTC 2016), Jeju Island, Korea, October
22. Quinn P, Guichard J (2014) Service function chaining: creating a service plane via network service headers. *Computer* 47(11):38–44