國立清華大學
NATIONAL TSING HUA UNIVERSITY

碩士論文

Design and Implementation of SDN-based QoS Management System

基於軟體定義網路之服務品質管理系統設計與研製

所別	資訊工程研究所
學號	104062564
姓名	李紀萱 (Chi-Hsuan Li)
指導教授	黃能富博士 (Dr. Nen-Fu Huang)

中華民國一〇六年七月

Abstract

Keywords -

中文摘要

Acknowledgement

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In recent times, researchers have shown an increasing interest in evolving network infrastructures. Software-defined networking (SDN) and network functions virtualization (NFV) are key roles for this evolution. SDN [1–4] has been widely studied for almost a decade since the first OpenFlow [5, 6] article had be presented in 2008. The main concept of SDN is separating data plane and control plane to enable smart control on switch and give a brand-new viewpoint on network research, and makes innovation on industries.

As SDN was developed, NFV [7–9] has been introduced by telco operators at the same time. The network services offered by operators previously performed by specific hardware appliances and it is difficult to decrease the OPEX and CAPEX on service deployment and management. In this context, NFV is proposed to innovate in the service delivery arena. The concept of NFV is to reduce the coupling between network functions (NFs) and hardware devices. Virtual Customer Premise Equipment (vCPE) [10], in particular virtual residential gateway (vRGW) [11], is one of the network services which benefited from NFV [12].

In the progress of vCPE development, the SDN is not involved at first. Most of previous researches focused on other technology to virtualize and deploy the CPE node [13–18]. Cloud4NFV [19, 20], proposed by Portugal Telecom, started to use SDN technology on designing virtual CPE management and organization (MANO) platform for telco cloud. Italy Telecom also proposed NetFATE [21], which is a network function deploy-to-edge model in which the NFs are designed by SDN and perform by SDN switch. Inspired from these two frameworks, our laboratory, High Speed Network Labtory (HSNL), also proposed a vCPE framework and a few network functions, attempting to replace hardware-based CPE [22, 23].

However, these SDN-involved vCPE research most focused on how SDN benefits the design of NFV MANO [24, 25] platform or traffic steering between CPE nodes, not the CPE network function itself. When the NFV is deployed at network edge and performed by SDN switch, there will be restriction on the OpenFlow Table [26]. In this paper, we proposed a multiple OpenFlow table mechanism to implement network functions and explain how to use it to resolve the table restriction. We also evaluate the new VNF implemented by the proposed mechanism, and compare with the single-table mechanism and hardware CPE. This new VNF can also be deployed to the network edge by the previous HSNL vCPE framework.

This paper is structured as follows. Chapter 2 briefly introduces SDN technology, NFV architecture, the OpenFlow protocol, related studied of vCPE framework and the previous HSNL vCPE framwork. In Chapter 3, we will review the NF design from the concept of SDN-enabled [27] architecture, and then move on describing our proposed multiple flow table management mechanism, which achieved vRGW functions. Chapter 4 turns to analyze the performance of vCPE network function what we proposed and compare to single table NF and traditional network devices, followed by Conclusion and proposed future works in the last chapter.

Chapter 2

Related Work

Chapter 3

System Implementation

3.1 Overview of Network Functions

Our network functions (Fig. 3.1) are designed with SDN-enabled NFV architecture concept [27], using the synergies between computer infrastructures (NFVs) and network infrastructures (NFVIs) [28,29]. An NFV is a VNF controller, mainly used for addressing stateful processing and NFVI is an SDN switch used for stateless processing.

3.1.1 Stateful Processing Component

This component is used to control the workflow, maintain the state associated with the VNF, and provide an interface for service providers or customers to configure and update the behavior of the stateless datapath processing component. We used an SDN controller to implement the VNF controller; and notably, we use southbound APIs of the SDN controller framework to manage the interface between the stateful and stateless components with the OpenFlow protocol, which was originally designed for this purpose.

3.1.2 Stateless Processing Component

Stateless processing component is implemented by SDN datapath resources and is optimized for data plane traffic processing. Because an SDN switch can be decoupled

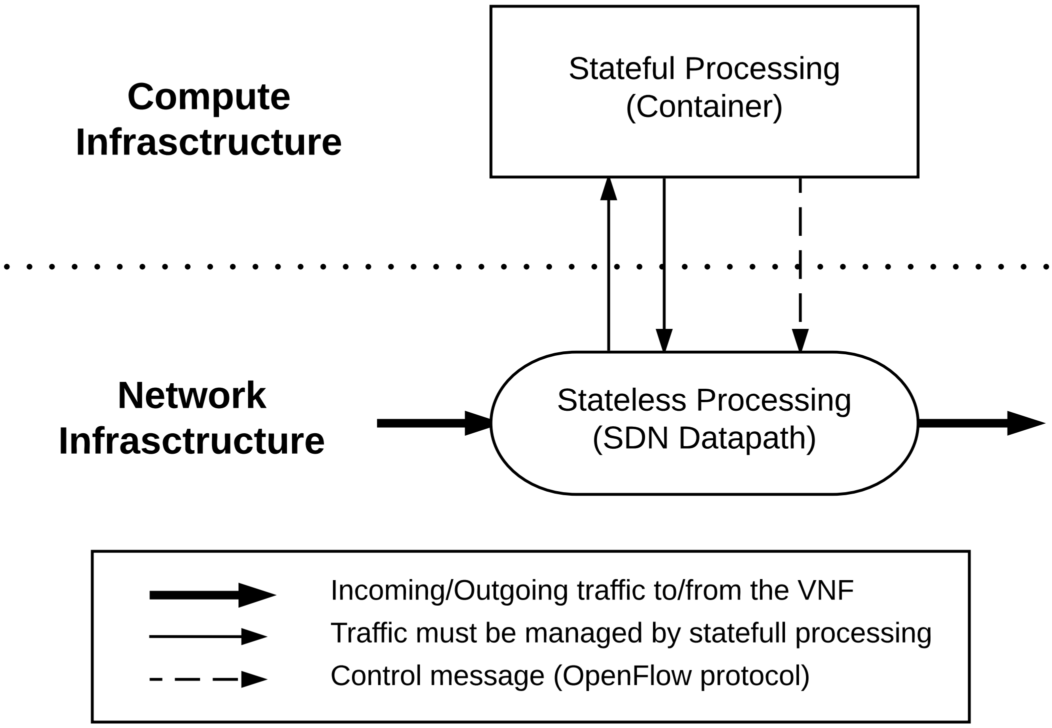

Figure 3.1: Overview of network functions.

with control plane and data plane, the switch can accept the control messages from the stateful processing component.

By using the advantages of this architecture, we can assign stateless or light-weight state work to the SDN switch (e.g., packet filtering and packet counting) to reduce the load on the computing resources. If we want to update our service, we are required to update only the stateful component, because the stateless component merely follows the commands from the stateful component.

3.2 Multiple Flow Table Strategy

In section 3.1, we introduce the vCPE service design architecture. The network functions are managed through cooperation between the SDN controller on the cloud

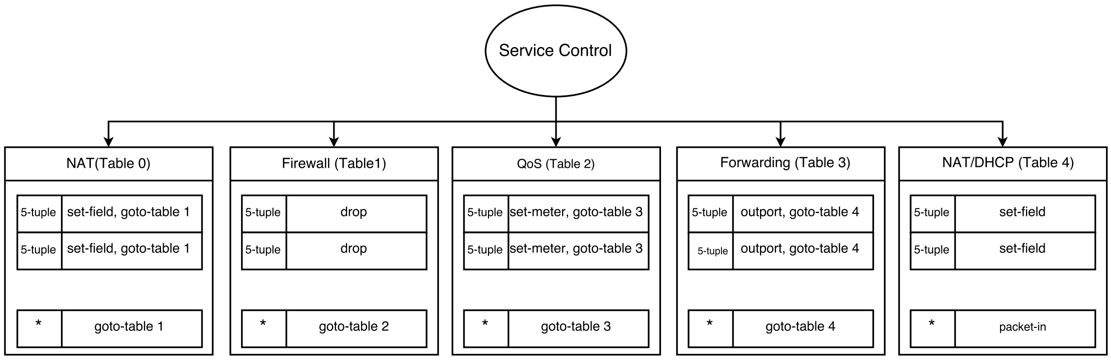

Figure 3.2: Flow table order of vCPE service.

and SDN switch at the local network gateway. The controller transforms the network functions into a series of OpenFlow rule requests and sends it to the SDN switch. Following the orders from the controller, the SDN switch inserts the rules into its flow tables, examines the incoming packets against the flow entry match fields, and executes the actions in matching rules. The flow table [30] defines all matching and corresponding processing, thus playing an important role in the executive network function.

We found that a single flow table restricts the implementation of our network functions. In [26], two conditions under which a single flow table is too restrictive were reported. The first is a condition where a single packet must perform independent actions based on matching with different fields. The second is a condition where the packet requires two-stage processing. To satisfy both conditions, we implemented the network functions by using a multiple flow table strategy.

Before we discuss about the multiple flow table strategy, we introduce the pipeline of OpenFlow flow table first [6]. The processing of each packet always starts at the first flow table. When being processed by a flow table, the packet is matched the flow entries of the flow table and adds corresponding action to the instruction set. The packet can executes the instrcution set immediately, or execute after finishing the journey in switch. A flow entry can direct a packet to next table by go-to action. In our multiple flow table management model, we set the “go to next table” action as the table-miss action. Therefore, the packet is processed table-by-table in a certain sequence.

In a multiple flow table strategy, it is most important to determine which flow table the rules should be inserted into. We used the network function as a demarcation, that is, SDN applications responsible for specific network functions inserted rules into one specific flow table to enable us to focus on the design of the network function itself. However, the order of the flow table and the sequence of the network functions become crucial. This can be addressed by considering the type of match and action in the rules generated by the network function.

The network functions of vCPE services are the firewall service, NAT, DHCP, forwarding, and QoS. The order of each function was determined as shown in Fig. 3.2 (note that the flow tables are counted from zero). Each packet In the following sections, we introduce the method of implementing these network functions, the type of rules to be inserted into the SDN switch, and the effect of these rules on deciding the order of the flow tables.

3.3 Service Control

Service control is used to enable or disable services. To enable a service, the table-miss rule should be modified. A packet-in rule is always placed in the flow table of the last active service as a table miss in case there is no corresponding rule. To enable the service chain, the rules of each service except the last service contain an additional action, “go to next flow table”, which enables the packets to continue to pass through all active services.

To disable a service, we must not only modify the table-miss rule but also add an enforce rule. Each enforce rule has maximum priority with the action, “go to next flow table”. It indicates that packets still pass through the disabled service’s table, but ignore other rules and proceed to the next flow table.

3.4 Network functions

3.4.1 Firewall

The firewall service can dynamically block traffic and prevent the packets from causing a packet-in event. On the dashboard, we can specify the blocking policies. There are three kinds of policies:

1. block any traffic from a source IP or destination IP address;
2. block traffic based on known layer 4 protocols, such as SSH and HTTP;
3. block traffic to customize layer 4 ports of a host.

For different policies, the controller applies corresponding rules to the SDN switch. After the policies are set, the blocking rules are immediately installed. Subsequently, any traffic that satisfies the blocking criteria is dropped. Normal traffic is unaffected.

As shown in Table 3.1, all the actions of flow entries are dropped. The first rule illustrates that SSH connection with the source IP address 192.168.2.1 is blocked. The second rule indicates that the flow entry blocks the Telnet protocol.

In our multiple flow table model, the firewall service is located in flow table 1 because once packets are detected by the blocking rules, they do not need to be applied to any other services. The packets that satisfy the blocking rules are immediately dropped, and their journey in the flow table ends. The other unblocked packets pass all blocking rules

Table 3.1: Firewall rules in Flow Entry

IP proto	IP src	IP dst	L4 sport	L4 dport	action
TCP	192.168.2.1	*	*	22	drop
TCP	*	*	*	23	drop

Symbol * represents wildcard (matches any value).

and finally satisfy the table-miss rule, which allows the packets to proceed to the next flow table. The action of the firewall is different from those of other services, because in other services, irrespective of the actions taken with the packets, the packets must proceed to the next flow table.

Chapter 4

Evaluation

Chapter 5

Conclusion and Future Work

There has been a significant increase in using SDN technology to develop the vCPE. However, the studies are most focused on vCPE orchestration, deployment and management, not the vCPE functions. The purpose of this paper was therefore to discuss about the development of CPE function with the SDN technology. The proposed a multiple flow table management mechanism to implement the vRGW functions and unrestricted the single flow table limitation. The results of the experiment show that the new VNF provides better performance compete over single table SDN application and even the physical CPE.

Further research is needed, however, the multiple flow table mechanism need more flow rules in the SDN switch. The mechanism use more space in flow tables to gain more functionality. Also, the order of network functions in flow table must be fixed and reduce the flexity. As a future work, we plan to study further on the optimization of multiple flow table mechanism. We are still at an early stage of this approach and the full potential is yet to be revealed.

Bibliography

[1] N. McKeown, “Software-defined networking,” *INFOCOM keynote talk*, vol. 17, no. 2, pp. 30–32, 2009.

[2] O. N. Fundation, “Software-defined networking: The new norm for networks,” *ONF White Paper*, vol. 2, pp. 2–6, 2012.

[3] N. Feamster, J. Rexford, and E. Zegura, “The road to SDN,” *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 2, pp. 87–98, Apr. 2014.

[4] D. Kreutz, F. M. V. Ramos, P. E. Verissimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, “Software-defined networking: A comprehensive survey,” *Proceedings of the IEEE*, vol. 103, no. 1, pp. 14–76, Jan. 2015.

[5] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, “OpenFlow,” *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 2, p. 69, Mar. 2008.

[6] B. Pfaff, B. Lantz, B. Heller *et al.*, “Openflow switch specification, version 1.3. 0,” *Open Networking Foundation*, 2012.

[7] M. Chiosi, D. Clarke, P. Willis, A. Reid, J. Feger, M. Bugenhagen, W. Khan, M. Fargano, C. Cui, H. Deng *et al.*, “Network functions virtualisation: An

introduction, benefits, enablers, challenges and call for action," in *SDN and OpenFlow World Congress*, 2012, pp. 22–24.

[8] NFV ISG, "GS NFV-SWA 001 V1.1.1 - Network Functions Virtualisation (NFV); Virtual Network Functions Architecture," ETSI, Tech. Rep., Dec. 2014.

[9] R. Mijumbi, J. Serrat, J.-L. Gorricho, N. Bouten, F. D. Turck, and R. Boutaba, "Network function virtualization: State-of-the-art and research challenges," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 236–262, 2016.

[10] P. Minoves, O. Frendved, B. Peng, A. Mackarel, and D. Wilson, "Virtual CPE: Enhancing CPEś deployment and operations through virtualization," in *4th IEEE International Conference on Cloud Computing Technology and Science Proceedings*. IEEE, Dec. 2012.

[11] Z. Bronstein and E. Shraga, "NFV virtualisation of the home environment," in *2014 IEEE 11th Consumer Communications and Networking Conference (CCNC)*. IEEE, Jan. 2014.

[12] NFV ISG, "GS NFV 001 V1.1.1 - Network functions virtualisation (NFV); use cases," ETSI, Tech. Rep., Oct. 2013.

[13] M. Ibanez, N. M. Madrid, and R. Seepold, "Virtualization of residential gateways," in *2007 Fifth Workshop on Intelligent Solutions in Embedded Systems*. IEEE, Jun. 2007.

[14] M. Ibanez, N. M. í nez Madrid, and R. Seepold, "Security management with virtual gateway platforms," in *2009 Third International Conference on Emerging Security Information, Systems and Technologies*. IEEE, 2009.

[15] B. Zamaere, L. Da, and E. Kullberg, “On the design and implementation of a virtualized residential gateway,” in *2012 2nd International Conference on Consumer Electronics, Communications and Networks (CECNet)*. IEEE, Apr. 2012.

[16] N. Herbaut, D. Negru, G. Xilouris, and Y. Chen, “Migrating to a NFV-based home gateway: Introducing a surrogate vNF approach,” in *2015 6th International Conference on the Network of the Future (NOF)*. IEEE, Sep. 2015.

[17] F. Sanchez and D. Brazewell, “Tethered linux CPE for IP service delivery,” in *Proceedings of the 2015 1st IEEE Conference on Network Softwarization (NetSoft)*. IEEE, Apr. 2015.

[18] R. Bonafiglia, S. Miano, S. Nuccio, F. Risso, and A. Sapio, “Enabling NFV services on resource-constrained CPEs,” in *2016 5th IEEE International Conference on Cloud Networking (Cloudnet)*. IEEE, Oct. 2016.

[19] J. Soares, M. Dias, J. Carapinha, B. Parreira, and S. Sargento, “Cloud4nfv: A platform for virtual network functions,” in *2014 IEEE 3rd International Conference on Cloud Networking (CloudNet)*. IEEE, Oct. 2014.

[20] J. Soares, C. Goncalves, B. Parreira, P. Tavares, J. Carapinha, J. P. Barraca, R. L. Aguiar, and S. Sargento, “Toward a telco cloud environment for service functions,” *IEEE Communications Magazine*, vol. 53, no. 2, pp. 98–106, Feb. 2015.

[21] A. Lombardo, A. Manzalini, G. Schembra, G. Faraci, C. Rametta, and V. Riccobene, “An open framework to enable NetFATE (network functions at the edge),” in *Proceedings of the 2015 1st IEEE Conference on Network Softwarization (NetSoft)*. IEEE, Apr. 2015.

[22] C.-W. Lin, “A Novel Virtual CPE Architecture and Service for Enterprises with SDN Network Technologies,” Master’s thesis, National Tsing Hua University, No.101, Sec. 2, Guangfu Rd., East Dist., Hsinchu City 300, Taiwan, 2016.

[23] N.-F. Huang, C.-W. Lin, S.-J. Wu, C.-H. Li, and I.-J. Liao, “A novel virtual cpe architecture and service for enterprises with sdn network technologies,” in *PROCEEDINGS OF THE 9TH IEEE INTERNATIONAL CONFERENCE ON UBI-MEDIA COMPUTING” UMEDIA-2016”*, 2016, pp. 104–109.

[24] NFV ISG, “GS NFV-MAN 001 V1.1.1 - Network Functions Virtualisation (NFV); Management and Orchestration,” ETSI, Tech. Rep., Dec. 2014.

[25] ——, “GS NFV-EVE 005 V1.1.1 - Network Functions Virtualisation (NFV); Ecosystem; Report on SDN Usage in NFV Architectural Framework,” ETSI, Tech. Rep., Dec. 2015.

[26] Open Networking Fundation, “The benefits of multiple flow tables and ttps,” ONF, Tech. Rep., 2015.

[27] J. Matias, J. Garay, N. Toledo, J. Unzilla, and E. Jacob, “Toward an SDN-enabled NFV architecture,” *IEEE Communications Magazine*, vol. 53, no. 4, pp. 187–193, Apr. 2015.

[28] NFV ISG, “GS NFV 003 V1.2.1 - Network Functions Virtualisation (NFV); Terminology for Main Concepts in NFV,” ETSI, Tech. Rep., Dec. 2014.

[29] ——, “GS NFV-INF 004 V1.1.1 - Network Functions Virtualisation (NFV); Infrastructure; Hypervisor Domain,” ETSI, Tech. Rep., Jan. 2015.

[30] M. Kuźniar, P. Perešíni, and D. Kostić, "What you need to know about sdn flow tables," in *Passive and Active Measurement*. Springer Science + Business Media, 2015, pp. 347–359.