2016 IEEE 24th International Conference on Network Protocols (ICNP)
Workshop on COntrol, Operation and appLication in SDN protocols (CoolSDN 2016)

Role-based Campus Network Slicing

Chien-Hsin Chen[1], Chien Chen[2], Ssu-Hsuan Lu[3], and Chien-Chao Tseng[4]
Department of Computer Science, National Chiao Tung University, Hsinchu, Taiwan
{chienhsin.cs02g[1], shlu[3]}@nctu.edu.tw, {chienchen[2], cctseng[4]}@cs.nctu.edu.tw

***Abstract*—The Internet has evolved greatly in this decade. For economic benefits, the management of Internet networks needs to adapt to mass changes without huge modifications in hardware and software. Especially in campus networks, there are a variety of users and devices with different Quality of Service (QoS) requirements and management policies. Software-defined networking (SDN) decouples the data and control planes. The separated control plan resides on a centralized controller to make networks easier to manage. In this paper, a role-based SDN campus network slicing is proposed by utilizing an authentication controller and the virtualization technology of FlowVisor to divide the campus network into several virtual networks according to the types of users. However, the mapping between the devices' MAC addresses and the users' roles results in a heavy loading problem on FlowVisor. Therefore, a VLAN-based slicing is introduced in this framework to offload the workload of FlowVisor for classifying packets into corresponding slices. Packets are appended with VLAN tags for recognizing types of users to lower the overhead of FlowVisor. Analyses of experimental results show that compared with MAC-based slicing, VLAN-based slicing can reduce the flow setup latency by 14% to 60% depending on the number of devices.**

***Index Terms*—software defined networking, network virtualization, FlowVisor, QoS, network management.**

I. Introduction

Today's campus networks are facing major challenges such as mobile clients, huge video bandwidth demand, and the growing number of connected devices and applications. Therefore, a campus network management system needs to support diverse sets of users, devices, applications, and ways of connecting efficiently. A typical campus network serves many different kinds of users, including faculty, students, guests, medical facilities, laboratories, libraries, police departments, restaurants, bookstores, etc. These individual tenants may need bandwidth or delay guarantees. Some tenants are required to have different access privileges such as in Table I. Thus the campus network needs to isolate traffic among multiple tenants and operate logical networks over a single physical network. At the same time, it requires an integrated network management to provide different access control and QoS support for different groups of users.

TABLE I
Different access privileges

Tenants	Faculty	Student	Guest
Policy	Accessible everywhere No time limit	Limited access No time limit	Limited access Limited time

The goal of this paper is to achieve traffic isolation by slicing a campus network into several logical networks according to the users' roles. Furthermore, with network virtualization we can apply different policy control to the different groups of users. In networks today, isolation is usually achieved through Virtual Local Area Networks (VLAN). VLAN provides a way to separate different classes of packets in the campus network. However, current VLAN virtualization technologies have some limitations and problems [17]. The enormous and complex manual configurations as mentioned in [2], [8], [17], [19] for using VLANs is against our goal to make management of campus network easier for administrators. Nowadays, VLAN can be dynamically assigned to users according to the users' credentials through IEEE 802.1X [7]. Operators use VLAN as a group identifier to apply different policies to different groups. Although dynamic VLAN assignment is powerful, it requires a series of configurations, which are error-prone and get more complex as the size of the network grows. In order to provide the flexibility of virtual network creation for different groups of users, the Software-Defined Network (SDN) [10], [18] becomes an emerging network architecture to have the capabilities to easily virtualize a network. SDN allows network administrators to manage network services through high-level functionalities. SDN decouples control plane from the forwarding plane to a separate device which is called the *controller*. The control plane has the ability to program the forwarding plane through the OpenFlow protocol [11], [12]. The OpenFlow, which is defined and promoted by the Open Networking Foundation (ONF) [3], [6], is a standard communication interface between the control plane and the forwarding plane in SDN. It provides an open protocol to program the forwarding plane. By using SDN, the network administrators have the benefits of faster deployment and easier configurations of the networks.

Some works have been done on network virtualization by using OpenFlow [11], [12], [14], [15], [16]. One of them is FlowVisor [16], which is a special controller aiming to deal with network virtualization based on SDN architecture. FlowVisor can slice a physical network into several virtual networks which are called *slices*. The slice is composed of all packets matching the specific headers, and each slice has its own controller which simplifies the management of how to send the packets. The information about the slices' controllers and the matching rules is stored in the flowspace table of FlowVisor. FlowVisor maintains the control isolation and traffic isolation between slices. It delegates the control of each slice to a different controller, so traffic in multiple slices can run simultaneously without interfering with each

978-1-5090-3281-5/16/$31.00 ©2016 IEEE

other. FlowVisor virtualizes the network control by rewriting messages from controllers and forwarding messages from switches to the corresponding controllers.

In the campus network, there are several different kinds of roles for users, such as faculty, students, and guests as shown in Fig. 1. In this paper, a role-based campus network slicing which combines FlowVisor and an authentication controller is introduced. The slicing technology of FlowVisor is used to provide isolation between slices [12]. The campus network provides wireless access so that different roles of users may send traffic through any wireless Access Point, thus slices cannot be decided just by slicing the topology. Therefore, FlowVisor needs to do MAC-based slicing by recording the mapping between the MAC addresses of devices and the types of users. Because the users may carry different devices at different times, this mapping should be dynamical. Therefore, an authentication controller is applied to our framework to help FlowVisor classify packets into different slices. The authentication controller authenticates users and classifies their devices into different slices.

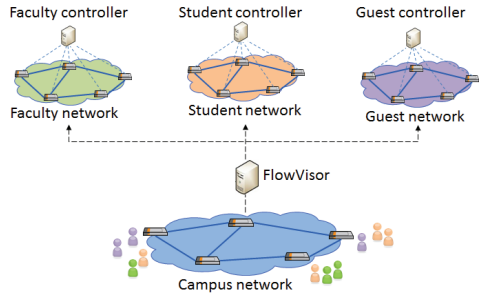


Fig. 1. Campus Network

However, with a growing number of devices in the campus network, the size of the mapping table in FlowVisor will increase enormously. Because the lookup time increases linearly with the size of the mapping table, the latency of processing a new flow will increase rapidly as the number of active devices grows. Therefore, the FlowVisor loading increases along with the number of users' devices.

To solve this heavy loading problem in FlowVisor, a VLAN-based slicing is developed by appending VLAN tags to packets to represent the roles of users. When a user is authenticated by the authentication controller, the authentication controller configures the switches to append the VLAN tags to packets which are sent from the user. Then, FlowVisor can send messages related to the packets to controllers according to the VLAN tags. In this way, the number of mappings which are used for classification in FlowVisor is equal to the kinds of roles, which is much smaller than the number of devices.

The main objective of our proposed framework is to differentiate the network behaviors of different types of users. The proposed system can provide different kinds of QoS by assigning different types of queues in OpenFlow switches for different slices. By using these queues, the traffic of different slices can be treated differently by the forwarding plane. For example, the faculty can have the best QoS by dedicating to the queues with the specific bandwidth guarantee.

Some experiments were performed to demonstrate the proposed framework. The experimental results revealed that VLAN-based slicing can let FlowVisor easily classify packets and reduce latency. Latency can be reduced by 14% to 60%.

The remainder of this paper is organized as follows. Section 2 presents some related work. Section 3 describes the proposed slicing technology, and Section 4 shows some experimental results. Finally, conclusions and future work are presented in Section 5.

II. Related Work

FlowVisor does not control the forwarding plane but hands the control over to the corresponding controller. In this way, it is easy to add or delete a slice because only the controller that is responsible for the slice needs to be turned on or turned off. The controller controls all the packets in the slice's flowspace. A flowspace entry describes the mapping between a specific header and the corresponding slice that owns the flowspace. When FlowVisor receives a Packet-In message, it looks up the flowspace table and finds the flowspace entry that matches the packet's header, and sends that message to the slice's controller.

Procera [9] is a SDN network management platform deployed on the Georgia Tech campus network. Procera separates the authenticated traffic and unauthenticated traffic into two different VLAN domains. But the policies that operators can create are restricted to the policy language. In addition, the complexity of the virtualization is not hidden to operators.

FlowN [4] is an overlay approach to realize virtualization. Although it has more functionality than FlowVisor, it has worse performance than FlowVisor when the size of virtual networks is less than 100. Moreover, the logic of virtual networks can only be designed under the FlowN framework. OpenVirtex [1] is another network virtualization framework developed from the design of FlowVisor. It allows tenants to specify their own IP addresses and topologies. However, it eliminates the ability to create a virtual network from packets matching a specific header.

III. Role-Based Network Slicing

In the campus network, there are many users with different roles. They also have different access rights and QoS requirements, which results in the complexity of network management. This section introduces the proposed role-based network slicing which assigns different network slices to different groups of users. An SDN campus network is deployed by combining multiple controllers and FlowVisor to achieve an efficient network slicing.

A. Our Framework

This study proposes a framework to logically slice a SDN campus network into several virtual network slices based on the roles of users, such as faculty network, student network, and guest network. Slices can be managed independently by

their own controllers. As shown in Fig. 2, our framework slices the SDN campus network through FlowVisor. FlowVisor creates slices of network resources and delegates control of these slices to different controllers. FlowVisor is located between the campus network and controllers. In this way, the complexity of virtualization is hidden from the controllers. Only the network resources and devices that belong to the same slice are visible to the controller. In a typical campus network, a user carries multiple devices to connect to the campus network through different connectivity options and physical ports. In order to classify different devices to different types of users dynamically, an authentication controller is introduced in our framework (in Fig. 2). An authentication controller is applied to authenticate users and classify their devices into corresponding slices. When a user's devices join the campus network, the authentication controller authenticates the user and decides which slice the devices belong to. Then it adds the user's devices to the flowspace of the associated slice. In addition, user controllers are responsible for managing their slices. An authentication controller and several user controllers form a *controller set*.

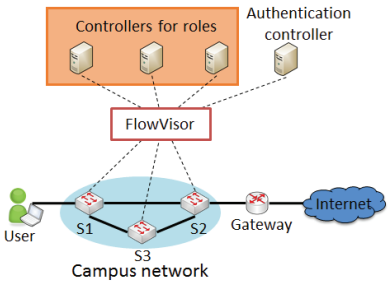


Fig. 2. Framework with an Authentication Controller

B. MAC-based Slicing

FlowVisor defines a slice as a set of flows, which is called the flowspace. The flowspace can be defined by a collection of packet headers. Therefore, one simple way to implement role-based campus network slicing is mapping a group of devices' MAC addresses to form a slice. For each device, the authentication controller will insert a MAC and slice ID mapping into the flowspace table in FlowVisor. Therefore, to establish a new SDN flow, the first Packet-In message after the device has been authenticated should try to match the device's MAC address in the flowspace table of FlowVisor to decide which user controller the device should be sent to. However, with the growing number of devices in the campus network, the size of the flowspace table will increase enormously. Because the lookup time of the flowspace table increases linearly with its size, the latency of processing a new flow will increase rapidly as the number of active devices grows.

C. VLAN-based Slicing

To reduce lookup time on FlowVisor, the size of the flowspace table on FlowVisor needs to be reduced. A VLAN tag which is defined in IEEE 802.1Q can be used as an identifier to represent a type of user. Once the packet is appended with the VLAN tag, FlowVisor can simply classify among these identifiers. Because the authentication controller maintains the mapping between MAC addresses of devices and types of users, the authentication controller installs rules into OpenFlow switches to add the VLAN tags associated to packets. Instead of inserting flowspace entries in the flowspace table of FlowVisor like MAC-based slicing, the authentication controller installs the authentication rules into switches during the authentication process.

In this way, the flowspace of each slice is simplified to just one entry. The number of flowspace entries can be reduced to the number of roles of users. For example, there are three kinds of roles in our campus network, faculty, student, and guest, so only three entries are added to FlowVisor.

FlowVisor maintains the settings of the slices. If FlowVisor receives packets that do not have VLAN tags, it sends the packets to the authentication controller. Otherwise, FlowVisor sends the packets to the controller according to their VLAN tags. The authentication controller installs rules to switches to add the VLAN tags to the packets when the packets enter the campus network and remove the VLAN tags when the packets leave the campus network to their destinations.

Figure 3 shows an example of rules that the authentication controller installs on switches. When a user is authenticated, the authentication controller installs rules on two switches. One is the switch (S1) that the user is connected to. The first rule on S1 is to add corresponding VLAN tags to the packets sent from the user and to force the packets to lookup the table again, and the second rule makes sure that packets with corresponding VLAN tags will be sent to the corresponding controller if they do not match any other flow entries. The other one is the switch that connects to the gateway (S2). The reason for installing rules on S2 is to provide access from Internet to users. Therefore, the first rule on this switch is to add VLAN tags to all packets that come from the Internet and target the user. Similarly, packets with VLAN tags will be resubmitted in order to send to the corresponding controller when they do not match any other flow entries.

S1 Flow Table

Src. MAC	Dest. MAC	VLAN ID	Action	Priority
User MAC	*	None	Push VLAN tag, Resubmit	Highest
*	*	VLAN tag	Send to Controller	Lowest

S2 Flow Table

Src. MAC	Dest. MAC	VLAN ID	Action	Priority
Gateway MAC	User MAC	None	Push VLAN tag, Resubmit	Highest
*	*	VLAN tag	Send to Controller	Lowest

Fig. 3. Flow Table for Authentication Rules

Because hosts cannot recognize packets with VLAN tags, FlowVisor needs to strip VLAN tags when packets go to external ports. Therefore, the routine where FlowVisor handles FlowMod messages received from the control plane is modified. Two cases have been considered: First, if the action of the FlowMod message is unicast, FlowVisor checks whether

the output port is an external port by using network topology which is maintained inside FlowVisor. If the output port is an external port, a new action is added to the FlowMod message for stripping the VLAN tag. Otherwise, no additional action is added to the rule. Second, if the action of the FlowMod message is broadcast and is applied for ARP request packets, FlowVisor removes the original action and creates multiple actions to send to all physical ports except the ingress port. Moreover, an action is added for stripping the VLAN tags and the order of the actions is specified as follows. The actions that output to the internal ports are executed first, and the VLAN tags stripping action is performed next. The actions that output to the external ports are executed last.

Figure 4 shows the Message Sequence Diagram of the proposed VLAN-based slicing. When a user tries to access the SDN campus network, the packet does not have the VLAN tag. Therefore, FlowVisor sends the Packet-In message to the authentication controller. After the authentication controller authenticates this user, the authentication controller installs rules on the switches. Then, the packets are appended with the VLAN tags, and FlowVisor can recognize the user of the packets by looking up the flowspace table using the VLAN tag and sends it to the user's controller. The user controller then directs the packet to its destination by sending the FlowMod message. FlowVisor will rewrite this FlowMod message by uniting the rule's match with the associated flowspace. In this case, the flowspace is the flows that match a specific VLAN tag, so a match to the VLAN ID field will be added by FlowVisor.

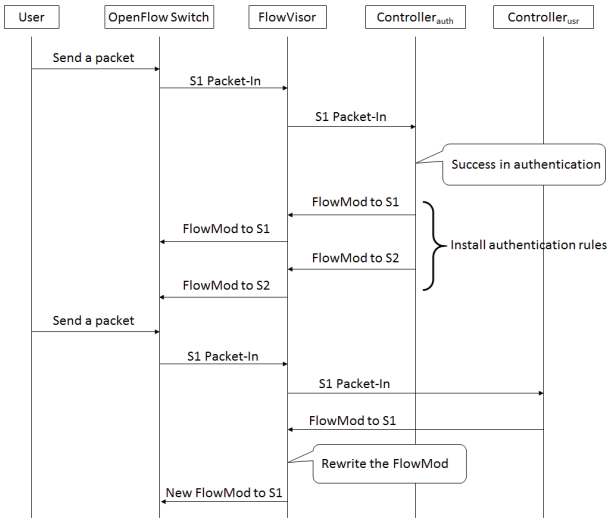


Fig. 4. Message Sequence Diagram

D. Intra-slice Communication

In this subsection, an example of intra-slice communication is provided. As shown in Fig. 5, the process of how two faculty users (F1 and F2) communicate with each other is demonstrated. Assume that F1 and F2 are authenticated. Therefore, the authentication rules for F1 and F2 (rules A, B, C, and D) have been installed on S1 and S2 respectively. The process of how F1 initiates a connection to F2 is described as follows:

1. The first packet sent from F1 matches rule (A) and is tagged with the VLAN tag of faculty. Then, the packet is resubmitted to match the flow table again.
2. The packet matches rule (B) in the second look-up and is encapsulated into the Packet-In message. Afterward, the message is sent to the control plane. It matches the flowspace of the faculty slice in FlowVisor and is sent to the faculty controller.
3. The faculty controller sends FlowMod messages to install rules (E) and (F) on the switches on the path from F1 to F2. FlowVisor modifies the messages by: (i) adding a match to the *VLAN ID* field in rules (E) and (F), and (ii) stripping the VLAN tag in (F) when the packet is sent to the external port.
4. The packet is forwarded to F2 after the rules are installed.

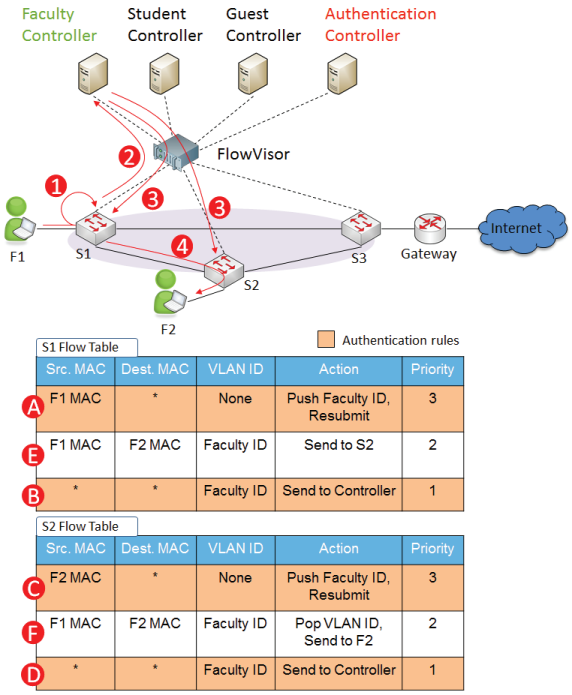


Fig. 5. Intra-slice Communication

E. Internet Access

Figure 6 shows an example to demonstrate how a faculty user (F1) accesses the Internet. Assume that F1 is authenticated, so the authentication rules (rules A, B, C, and D) have been installed. The process is described as follows:

1.-2. These two steps are the same as the first and second steps of intra-slice communication.
3. The faculty controller sends messages to install rules (E) and (F) on the switches on the path from F1 to the gateway. FlowVisor modifies the messages by: (i) adding

a match to the *VLAN ID* field in (E) and (F), and (ii) stripping the VLAN tag in (F) when the packet is sent to the gateway.

4. The packet is forwarded to the gateway after the rules are installed.

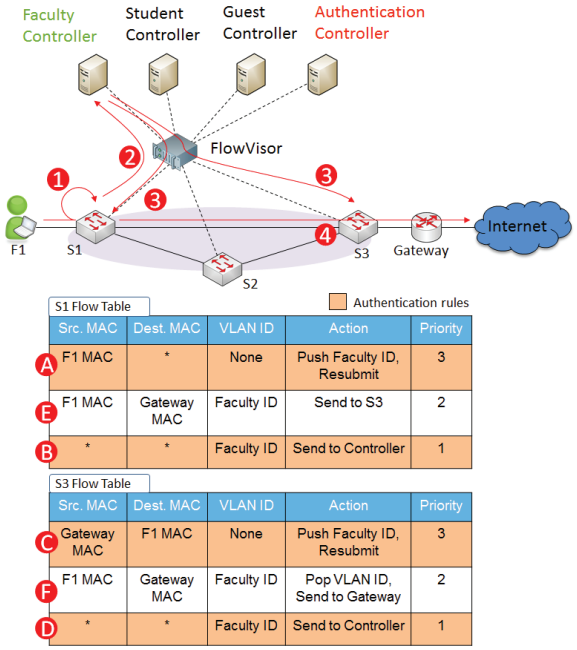


Fig. 6. Internet Access

IV. SIMULATION RESULTS

In this section, some experiments are performed to evaluate the performance of the proposed slicing technology which uses VLAN tags for identifying users. The latency of processing a new flow was compared between the proposed VLAN-based slicing and MAC-based slicing to show the effect of VLAN-based slicing.

A. Simulation Environment

Mininet [13] was used to generate network topologies and hosts for the experiments. FlowVisor, Mininet, and two instances of controllers were run on the same virtual machine which was installed on a computer with one i5 CPU 2.6 GHz processor and 8G memory. The virtual machine ran Ubuntu 12.04 with 4 GB memory and 3 processors.

Floodlight [5] was used as the controller. Floodlight ran a forwarding module which installs rules for the unicast flow to direct the packets from the source to the destination.

B. Latency

The experiment was run under a use case, which has two slices– student and faculty. The size of the flowspace table of the MAC-based slicing is double the number of hosts. However, the number of flowspace entries was 2 for the VLAN-based slicing, which is equal to the number of slices. Authentication rules have been installed for hosts.

Hosts were programmed to generate flow requests by sending ICMP requests to the gateway simultaneously. The ICMP requests were turned into Packet-In messages and were sent to FlowVisor. FlowVisor then looked up the flowspace table to decide where the messages should be forwarded. Once the corresponding controllers installed the forwarding rules, hosts could receive the ICMP replies. The latency for flow setup is defined as the time that a host took to receive the first ICMP reply. Each experiment was run 30 times and the average time of latencies was calculated.

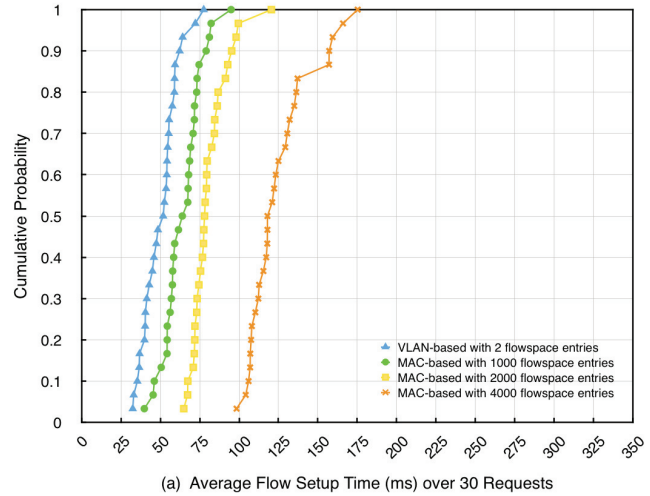
Figures 7(a), 7(b), and 7(c) show the latency time of flow setup latencies over 30, 60, and 90 requests respectively. In Fig. 7, the x-axis represents the average flow setup time over different number of requests, and the y-axis represents the cumulative probability of average flow setup latencies. The latencies of VLAN-based slicing are compared to those of MAC-based slicing with a various number of flowspace entries. The four lines represent VLAN-based slicing with only 2 flowspace entries, and MAC-based slicing with 1000, 2000, and 4000 flowspace entries.

As seen in Fig. 7, with the same number of requests, the latencies of setting up a flow for VLAN-based slicing are lower than those for MAC-based slicing. The average flow setup latency grows with the size of the flowspace table in the MAC-based slicing, because FlowVisor uses linear search for the flowspace table. Therefore, when the number of hosts is greater, the latency also will be increased.

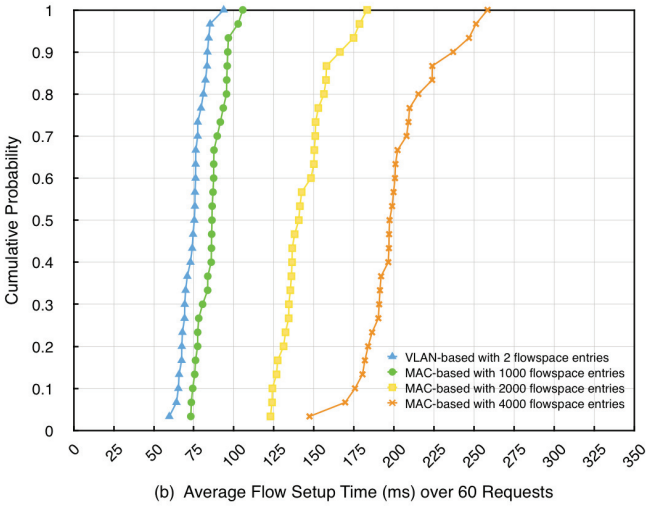
Furthermore, the requests are queued in FlowVisor, so the late arriving flow setup request should wait in the queue until the previous ones have been processed. The penalty of the increasing number of requests also grows as the size of the flowspace table enlarged. For example, considering the two lines of MAC-based slicing with 1000 and 2000 flowspace entries, the gaps between them become bigger as the number of requests grows. Compared with MAC-based slicing, VLAN-based slicing can reduce the flow setup latencies by 14% to 60% on average.

V. CONCLUSIONS

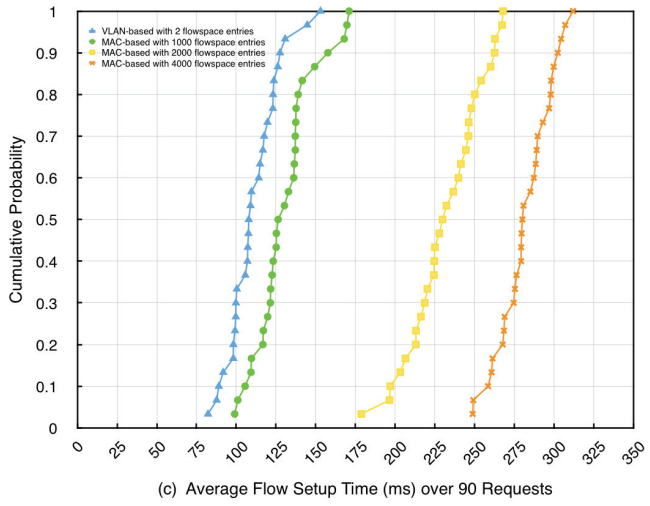
In this paper, a role-based campus network slicing is developed by utilizing FlowVisor. Moreover, an authentication controller is applied to authenticate users and maintain the mapping between devices and types of users. The VLAN tags are appended to packets, and FlowVisor can recognize roles of users according to the VLAN tags and send packets to corresponding controllers. With VLAN-based slicing, the lookup time of FlowVisor does not increase linearly along with the number of devices as it is with MAC-based slicing. The experimental results illustrate that the latency of processing a new flow for VLAN-based slicing can be reduced by 14% to 60% compared to that of MAC-based slicing depending on the number of devices used in a campus network. In the future, it is hoped that the proposed system can be deployed in real campus networks to efficiently manage them.



(a) Average Flow Setup Time (ms) over 30 Requests



(b) Average Flow Setup Time (ms) over 60 Requests



(c) Average Flow Setup Time (ms) over 90 Requests

Fig. 7. CDF of Average Flow Setup Latency

ACKNOWLEDGMENT

This research is supported in part by the Ministry of Science and Technology of Taiwan under Grant: MOST 104-2622-8-009-001, and is also supported by D-Link.

REFERENCES

[1] Ali Al-Shabibi, Marc De Leenheer, Matteo Gerola, Ayaka Koshibe, Guru Parulkar, Elio Salvadori, and Bill Snow. Openvirtex: make your virtual sdns programmable. In *Proceedings of the third workshop on Hot topics in software defined networking*, pages 25–30. ACM, 2014.

[2] Martin Casado, Michael J Freedman, Justin Pettit, Jianying Luo, Nick McKeown, and Scott Shenker. Ethane: Taking control of the enterprise. In *ACM SIGCOMM Computer Communication Review*, volume 37, pages 1–12. ACM, 2007.

[3] Neda Cvijetic, Akihiro Tanaka, Philip Ji, Karthik Sethuraman, Shinsuke Murakami, and Ting Wang. Sdn and openflow for dynamic flex-grid optical access and aggregation networks. *Lightwave Technology, Journal of*, 32(4):864–870, 2014.

[4] Dmitry Drutskoy, Eric Keller, and Jennifer Rexford. Scalable network virtualization in software-defined networks. *Internet Computing, IEEE*, 17(2):20–27, 2013.

[5] Floodlight. http://www.projectfloodlight.org/projects/.

[6] Open Networking Fundation. Software-defined networking: The new norm for networks. *ONF White Paper*, 2012.

[7] IEEE 802.1 Working Group et al. IEEE 802.1 xport based network access control, 2001.

[8] Stephen Gutz, Alec Story, Cole Schlesinger, and Nate Foster. Splendid isolation: A slice abstraction for software-defined networks. In *Proceedings of the first workshop on Hot topics in software defined networks*, pages 79–84. ACM, 2012.

[9] Hyojoon Kim and Nick Feamster. Improving network management with software defined networking. *Communications Magazine, IEEE*, 51(2):114–119, 2013.

[10] Keith Kirkpatrick. Software-defined networking. *Communications of the ACM*, 56(9):16–19, 2013.

[11] Nick McKeown, Tom Anderson, Hari Balakrishnan, Guru Parulkar, Larry Peterson, Jennifer Rexford, Scott Shenker, and Jonathan Turner. Openflow: enabling innovation in campus networks. *ACM SIGCOMM Computer Communication Review*, 38(2):69–74, 2008.

[12] Seokhong Min, Seungju Kim, Jaeyong Lee, Byungchul Kim, Wontaek Hong, and Jonguk Kong. Implementation of an openflow network virtualization for multi-controller environment. In *Advanced Communication Technology (ICACT), 2012 14th International Conference on*, pages 589–592. IEEE, 2012.

[13] Mininet. http://mininet.org/.

[14] Qinglei Qi, Wendong Wang, Xiangyang Gong, and Xirong Que. A sdn-based network virtualization architecture with autonomie management. In *Globecom Workshops (GC Wkshps), 2014*, pages 178–182. IEEE, 2014.

[15] Elio Salvadori, R Doriguzzi Corin, Attilio Broglio, and Matteo Gerola. Generalizing virtual network topologies in openflow-based networks. In *Global Telecommunications Conference (GLOBECOM 2011), 2011 IEEE*, pages 1–6. IEEE, 2011.

[16] Rob Sherwood, Glen Gibb, Kok-Kiong Yap, Guido Appenzeller, Martin Casado, Nick McKeown, and Guru Parulkar. Flowvisor: A network virtualization layer. *OpenFlow Switch Consortium, Tech. Rep*, 2009.

[17] Yasuhiro Yamasaki, Yoshinori Miyamoto, Junichi Yamato, Hideaki Goto, and Hideaki Sone. Flexible access management system for campus vlan based on openflow. In *Applications and the Internet (SAINT), 2011 IEEE/IPSJ 11th International Symposium on*, pages 347–351. IEEE, 2011.

[18] Soheil Hassas Yeganeh, Amin Tootoonchian, and Yashar Ganjali. On scalability of software-defined networking. *Communications Magazine, IEEE*, 51(2):136–141, 2013.

[19] Minlan Yu, Jennifer Rexford, Xin Sun, Sanjay Rao, and Nick Feamster. A survey of virtual lan usage in campus networks. *Communications Magazine, IEEE*, 49(7):98–103, 2011.