Design and Develop an OpenFlow Testbed within Virtualized Architecture

Pang-Wei Tsai, Yu-Ting Lai, Pei-Wen Cheng,
Chu-Sing Yang
Institute of Computer and Communication Engineering,
Department of Electrical Engineering,
National Cheng Kung University
Tainan, Taiwan
{pwtsai, lynette, pwncheng, csyang}@ee.ncku.edu.tw

Mon-Yen Luo
Department of Computer Science
& Information Engineering,
National Kaohsiung University of Applied Sciences
Kaohsiung, Taiwan
myluo@kuas.edu.tw

***Abstract*—Virtualization has brought the revolution of utilizing computing elements, and also incurred the trends of cloud computing with distributed data centers built for enterprises. Recently, the network virtualization has become an issue for decreasing the cost and adding the flexibility of networking inside either the service platform or the infrastructure. Hence, research and education demands for testing new ideas about Software Defined Network in testbed have become more and more popular. This paper proposes a time- and resource-shared network testbed which integrates with Emulab system, Xen hypervisor and OpenFlow. This OpenFlow Testbed aims to provide an isolated, clean, quick-built and independent environment for each user to emulate OpenFlow networks. The implementation also has been verified with real use cases which are tutorials of courses in college. Users can build OpenFlow environment upon the shared and virtual infrastructure for network research and education purposes. Results also revealed that not only this network emulation testbed meets most requirements that real machine environment could provide, but also acquires benefits from virtualization.**

Keywords—Emulab; Network Virtualization; OpenFlow; Software Defined Network; Testbed; Xen

I. INTRODUCTION

Virtualization technology utilizes the hardware resources of computing infrastructure, and then brought out the rise of the information industry. Recently, the Software Defined Network (SDN) is often discussed as a developing direction toward to the next generation of Internet. With the possibility of self-designed control policy, technical developers can easily do modification on networks. For information and cloud computing industry, migrating to SDN would reduce the cost of networking and also stably deliver the services. Even though software defined network has become a popular trend, it is still on the way to be standardized by Open Network Foundation [1]. Many providers are hard to claim their SDN-related developments are ready for production use, except Google[2]. Academic institutes and enterprises are aspiring in developing both control policy and data processing logics currently. For this purpose, building a SDN testbed is a necessary step for concrete feasibility in study and research.

The design aspiration in this paper is to deploy an OpenFlow testbed within virtualized architecture. The designed OpenFlow testbed offers a platform for SDN emulated environment. Besides, it can provide not only virtualized node but also topology-configurable network and easily re-configurable features. With the benefits of quarantined, closed and controllable environment of virtualized architecture, end-users can build OpenFlow environment with time- and space-shared network emulator based on Emulab system. With integrated virtual infrastructure and control framework, end-users can easily apply their requirements in virtualized environment.

II. RELATED WORK

A. Network Testbed and its Building Concepts

A testbed provides an environment for building emulated or duplicated system to process scenarios. It allows user to acquire actual (instead of simulated) performance measures. For instance, Emulab[3] proposed an environment with servers and switching routers to place nodes atop one another. Moreover, several projects in GENI[4], such as OFNLR[5] and ProtoGENI[6], deployed national networking testbed with integrated OpenFlow Switch. RISE[7], a wide-area hybrid OpenFlow network testbed, is to support a variety of OpenFlow network experiments based on JGN-X[8] in Japan. These testbeds aim to build large-scale and geographically distributed physical OpenFlow experiments for research.

However, the complexity of resource requesting is a time-consuming process. For short-term and small-scale verification in networking research, a testbed with fast-deployment and quick-replication environment is more important. A testbed with the high speed in creating, deleting and replicating the experimental environment is more convenient for students and researchers to conduct their experiments. Based on these benefits mentioned above, this paper proposes an OpenFlow network emulation TestBeD (OFTBD) with a simple and full-

This research is financially supported by the National Science Council of Taiwan under grants No. 102-2219-E-006-001 and the Headquarter of University Advancement at National Cheng Kung University.

virtualized environment. OFTBD provides exclusive virtual machines, OpenFlow switches and POX controller for each experiment. This topological and OpenFlow-enabled testbed aims to provide researchers and educators an initial verification of new network protocol or testing of network control policy.

B. *Related Techniques: Xen, OpenFlow and POX*

1) Xen: Xen is a hypervisor providing services through a virtualized infrastructure. It manages virtual resources with high availability over whole system. The first version of Xen is developed by University of Cambridge Computer Laboratory under the GNU[9] General Public License. The implementation in this paper uses Xen hypervisor to allocate virtual machines as nodes due to its efficiency and flexibility of controllability over the hardware resource.

2) OpenFlow: OpenFlow is proposed by Stanford University in 2009, is an open standard to deploy innovative protocols in production networks. This protocol aims at creating virtual environments for innovations in parallel with the production network.

3) POX Controller: Networking Operating System[10] is initially developed side-by-side with OpenFlow researches. As being the first OpenFlow controller and being released as open sourced project, NOX has been used and deployed in various researches. POX[11] acceded to NOX with its code architecture refined to be light and to offer a platform for rapid developing a prototype of network control software in Python language.

III. System Design

The implemented testbed in this paper includes a virtualized testing environment combined with Emulab system, Xen hypervisor, software-based OpenFlow switches and POX controller. It is developed with its experimental platform based on Emulab system and that enabled users to easily design network topology through the front-end web application. The backend control framework of testbed would then allocate resources including virtual machine nodes, individual OpenFlow switches, POX controller to the experiment according to user's request and configurations.

As shown in Fig.1, the system architecture inherits the front-end web applications and the back-end database schema. The process of creating an experiment with OpenFlow switches on the testbed is shown as Fig.2. User uses NetBuild[12] to design network topology by using icons such as nodes and network elements and to generate Network Simulator file (NS file[13]), which specifies the properties of all the nodes, networks as well as the connections between nodes. Then, the NS file would be modified and parsed into an acceptable format for Xen hypervisor to build the requested experiment by several implemented scripts. Here, a new option that extra nodes are created as OpenFlow switches and POX controller instead of deploying original LANs among connected nodes is provided to carry out the SDN experiments. Therefore, the system will automatically revise the NS file, adding requisition for emulated OpenFlow switches and POX controller after user confirms and submits the NS file. An implemented API translator in backend server is able to create and configure virtual machines by using Xen API[14]. These translators are a series of scripts combined with Emulab framework to translate and parse the request from user. In addition, Xen hosts also support different OS images as templates used for nodes with different purposes. User owns multiple choices of nodes in different types with pre-installed operating system to build required environments. After the experiment is built on Xen hosts, the system would initialize configuration of each node based on its operating system, especially those nodes within emulated OpenFlow network. In the end, user account on frontend system will be copied to each node, and also user access permission will be extended to enhanced privilege.

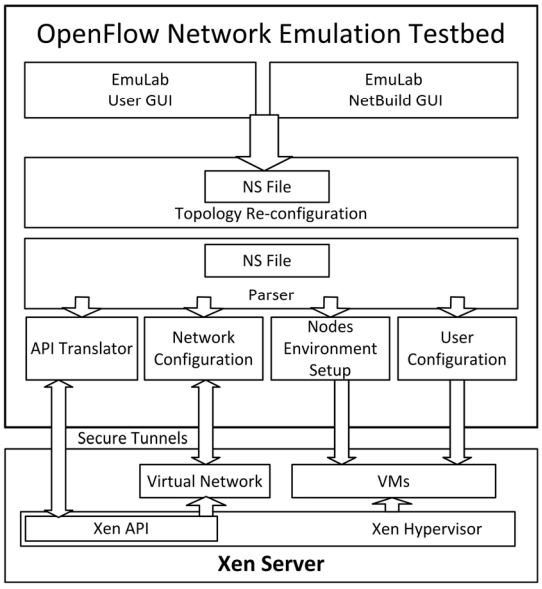

Fig.1. OFTBD System architecture

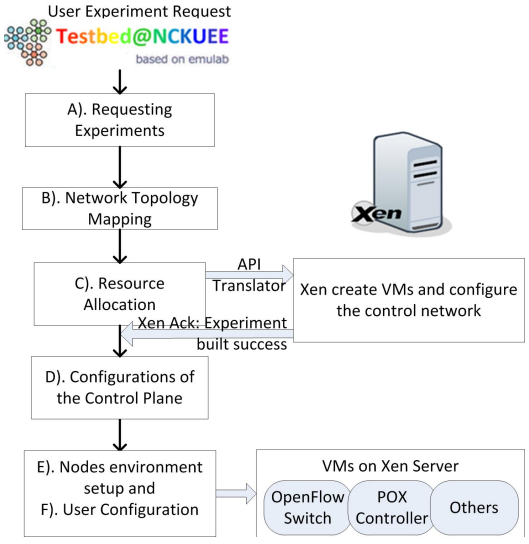

Fig. 2. Process of building experiment by user

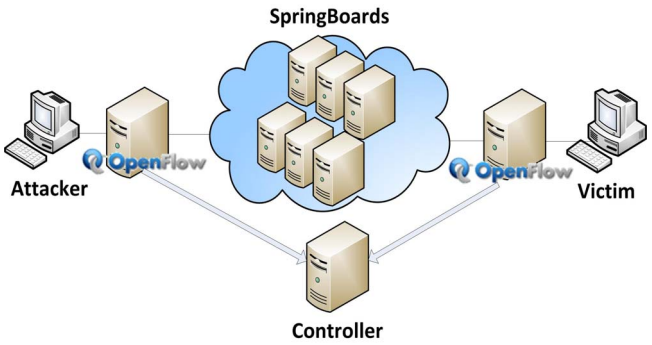

Fig. 3. Experiment of DDos Simulation

TABLE 1. SYN FLOOD MEASURED DATA RATE OF VITIM NODE

	CPU Loading (%)	Measured Data Rate (KBytes/sec)
Exp3.1	34.1	157
Exp3.2	34.4	195

IV. VERIFICATION AND RESULTS

The implemented testbed is evaluated with three different real network applications. These scenarios proved that it can reach desired requirements with the advantages of quick-built and easy-replication.

A. Experiment Scenarios

In this paper, authors use three different network security related scenarios for verification: firewall-related experiment, Malware detection experiment and Distributed Denial of Service (DDoS) experiment.

1) Firewall-related Experiment: There are a web server and a ftp server. The web server is free to access but ftp server is restricted for most nodes in experiment.

2) Malware Detection Experiment: Several nodes are implanted with the backdoor program "Rootshell". Operator can use NetCAT[15] and other tools to remotely control therse nodes through Rootshell interface.

3) DDoS-related Experiment: The experiment is shown in Fig.3. Attacker installed with DDoS attack tool TFN2K[16] remotely triggers the springbords to send SYN packets. The cpu utilization and network traffic of the victim are measured. To compare the performance of software-based OpenFlow switch with software-based router, 2 experiments—Exp 3.1 and 3.2 have been conducted. Exp 3.2 which has the same topology as Exp 3.1 but replaces OpenFlow switch with software-based router installed on virtual machine.

B. Evaluation and Discussion.

For firewall-related experiment and Malware detection and defense experiment, OFTBD is proofed that it can function as well as a network and network security testbed. For the DDoS-related experiment, Table 1 shows the average measured CPU loading and receiving data rate during 60 minutes of SYN flood attacking period. According to experiment result that both of the topology with software-based OpenFlow switch and with software-based router can reach desired requirement, and measured data also shows that the CPU usage surges up and inbound traffic rises rapidly.

V. CONCLUSIONS

This paper proposes an OpenFlow Network Emulation Testbed which integrated with virtual architecture and software defined network concepts. With this quarantined and controllable platform, researchers and students can build OpenFlow environment with resource-shared hardware based on the Emulab system for network research and education. OFTBD provides each experiment with exclusive software-based OpenFlow switches and POX controller. Furthermore, an implemented overlay which includes Emulab framework and Xen plays a significant role on distribution and re-distribution of virtual resources. Users send request for creating experiment and the OFTBD framework will setup virtual machines and virtual network through constructing a group of nodes with pre-installed OpenFlow network automatically. The result of experiment showed that emulated environment is quite capable and substitute for research and education.

ACKNOWLEDGEMENT

Authors would like to have many thanks to researchers of National Center for High-Performance Computing for their great help on this research.

REFERENCES

[1] Open Network Foundation, ONF. Available: https://www.opennetworking.org

[2] U. Hoelzle. (2012). OpenFlow @ Google. Available: http://opennetsummit.org/talks/ONS2012/hoelzle-tue-openflow.pdf

[3] M. Hibler, R. Ricci, L. Stoller, J. Duerig, S. Guruprasad, T. Stack, et al., "Large-scale virtualization in the Emulab network testbed," presented at the USENIX 2008 Annual Technical Conference on Annual Technical Conference, Boston, Massachusetts, 2008.

[4] C. Elliott, "GENI - global environment for network innovations," 33rd IEEE Conference on Local Computer Networks (LCN 2008), 2008.

[5] GENI OpenFlow Backbone Deployment at National LambdaRail. Available: http://groups.geni.net/geni/wiki/OFNLR

[6] ProtoGENI. Available: http://groups.geni.net/geni/wiki/ProtoGENI

[7] Y. Kanaumi, S. Saito, E. Kawai, S. Ishii, K. Kobayashi, and S. Shimojo, "RISE: A Wide-Area Hybrid OpenFlow Network Testbed," Ieice Transactions on Communications, vol. E96b, pp. 108-118, Jan 2013.

[8] National Institute of Information and Communications Technology, "JGN-X". Available: http://www.jgn.nict.go.jp/english/index.html

[9] GNU Operating System. Available: http://www.gnu.org

[10] N. Gude, T. Koponen, J. Pettit, B. Pfaff, M. Casado, N. McKeown, et al., "NOX: towards an operating system for networks," SIGCOMM Comput. Commun. Rev., vol. 38, pp. 105-110, 2008.

[11] POX controller. Available: http://www.noxrepo.org/pox/about-pox/

[12] Emulab NetBuild Documentation. Available: https://wiki.emulab.net/wiki/netbuilddoc

[13] Network Simulator version 2, NS2. Available: http://www.isi.edu/nsnam/ns/

[14] XenAPI Documentation. Available: http://www.xen.org/files/XenCloud/ocamldoc/apidoc.html

[15] Netcat, http://netcat.sourceforge.net

[16] Tribe Flood Network, http://www.cert.org/incident_notes/IN-99-04.html