Automated Overlay Virtual Networking Manager for OpenFlow-Based International SDN-Cloud Testbed

Junsik Shin and JongWon Kim(✉)

School of Information and Communications,
Gwangju Institute of Science and Technology (GIST), Gwangju, Korea
{jsshin,jongwon}@nm.gist.ac.kr

Abstract. With the development of virtualization and cloud technology, cloud data centers have been emerging for ICT infrastructure market. In order to link resource pools (i.e., servers, storages, and switches) over distributed data centers, tunneling-based overlay networking can maintain L2 connectivity with reasonable scalability. However, in properly managing all the switches and tunnels, the complexity of underlying network topology causes uncontrollable difficulties. Thus, in this paper, we discuss how to automatically manage overlay virtual networking by employing an OpenFlow-based OvN-Manager (Overlay vNetworking Manager) tool. The verification results for implemented OvN-Manager tool are also presented over OF@TEIN international SDN-Cloud testbed.

Keywords: OpenFlow-enabled software-defined networking · Future internet testbed · DevOps-based automation · Tunneling-based inter-connection · Overlay virtual networking

1 Introduction

The hyper-scale cloud data centers from Amazon, Google, and Microsoft are already established across the world and have been providing public cloud services [1]. These cloud data centers are equipped with a large number of servers, storages, and switches, distributed over multiple data centers. Due to their scales (>10,000 servers per center) and underlying network technology [2, 3], they are exposed to complexity problems in networking aspects. In linking resource pools among cloud data centers, it is important to maintain connectivity with reasonable scalability. However, the complexity of underlying network topology can cause uncontrollable difficulties in properly inter-connecting resource pools together.

Meanwhile, the emerging SDN (software-defined networking) paradigm is increasingly embracing OpenFlow and Open vSwitch (OVS) as a de-facto protocol interface and virtual switch, respectively. Aligned with this trend, by leveraging hyper-convergent SmartX Boxes and OpenFlow-enabled OVS, GIST and OF@TEIN collaborators have been building and operating an international OF@TEIN SDN-Cloud testbed [4]. Diverse operational experience is built to address the real-world challenges including the complex overlay tunneling management.

Y. Zhang et al. (Eds.): CloudComp 2015, LNICST 167, pp. 3–11, 2016.
DOI: 10.1007/978-3-319-38904-2_1

Thus, in this paper, we discuss how to improve overlay virtual networking management by utilizing an automated OpenFlow-based OvN-Manager (Overlay vNetworking Manager). The proposed OvN-Manager is implemented and verified over OF@TEIN international SDN-Cloud testbed. Then, in Sect. 2, we introduce the OF@TEIN international SDN-Cloud testbed. In Sect. 3, we detail the design and implementation of automated OvN-Manager. We also provide the functionality verification results for the proposed OvN-Manager tool. Finally, we conclude this paper in Sect. 4.

2 OF@TEIN: International Distributed SDN-Cloud Testbed

2.1 SDN-Cloud Playground Interconnecting Multiple OpenFlow Islands

The OF@TEIN project (from July 2012) aims to provide OpenFlow-enabled SDN testbed (i.e., playground) for international researchers [4–6]. As shown in Fig. 1, an initial 6-site testbed has grown to more than 8 sites with hyper-convergent SmartX Boxes. Note that SmartX Box refers to a hyper-convergent resource box, which integrates compute, storage, and networking resources in a single box.

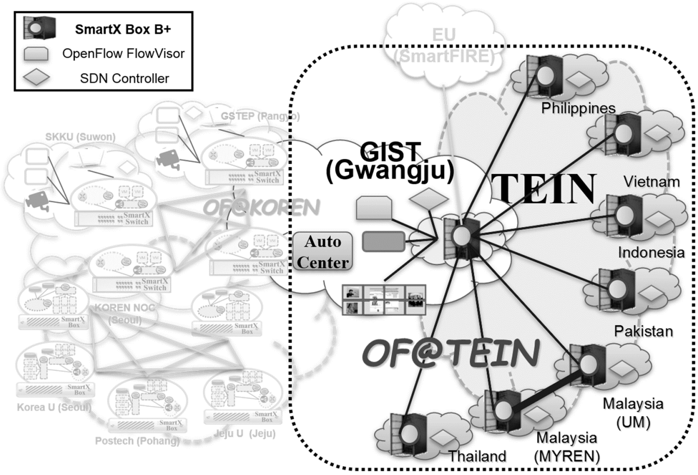

Fig. 1. OF@TEIN testbed infrastructure (June 2015).

To link SmartX Boxes distributed in multiple international sites, virtual overlay networking with NVGRE(Network Virtualization using Generic Routing Encapsulation)/VXLAN(Virtual eXtensible Local Area Network) tunnels establishes OpenFlow-based inter-connection over underlying TEIN (Trans Eurasian Information Network) network. Figure 2 illustrates the operation concept behind OF@TEIN testbed setup and configuration. Each site has three types of nodes. First, nodes under VMs at the end of each site contain hypervisors to provide VMs (virtual machines). OpenFlow Switch nodes are managed by user controllers that are policed by FlowVisor [7]. Also, Capsulator nodes are in charge of NVGRE/VXLAN tunnelling. Note that, in early days of

OF@TEIN, some switches/capsulators are supported by hardware switches and FPGA-accelerated servers, but all hardware switches are now replaced by OVS virtual switches.

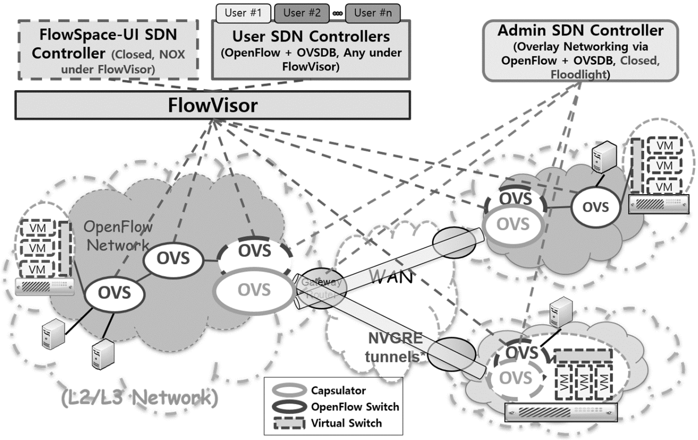

Fig. 2. OF@TEIN sites and operation setup (concept version).

By replacing networking devices from hardware to software, as depicted in Fig. 3, all SmartX Boxes have the same virtual networking topology as long as there are no site-specific modifications. Xen-hypervisor VMs connect to "xenbr0" bridge (for Internet connectivity) and "xenbr1" bridge (for VM to VM connectivity). For data traffic between VMs, bridges "br1", "br2", and "brcap" are placed and subsequently connected beyond "xenbr1". Furthermore, "br1" and "br2" are connected to user's choice of controllers, e.g., ODL (Open Daylight) SDN Controller [8]. With its own SDN controllers and open-shared bridges (e.g., "br1" and "br2"), users can do their own SDN experiments over their overlay networks. However, as shown as a line from Admin's ODL SDN Controller to "brcap" bridge in Fig. 3, "brcap" connects only to Admin's ODL SDN Controller, which manages all the NVGRE/VXLAN tunnels.

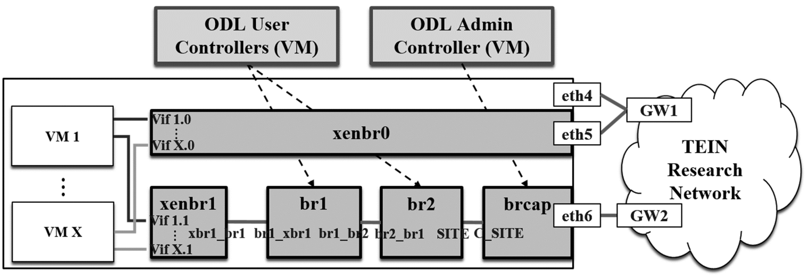

Fig. 3. Common bridge configuration of SmartX Box (Type B+).

2.2 Automated Inter-Connection Provisioning Tools

To provision overlay virtual networking for OF@TEIN playground, we should take care of combined configurations for bridges, tunnels, and flows. It is apparent that managing them is very difficult without an automated tool. Thus, to assist the configuration of tunnel-based inter-connections, in 2014, we deployed a script-based automated inter-connection tool [9]. This inter-connection tool is focused on linking multi-point OpenFlow islands by utilizing OVSDB (Open vSwitch Data Base) configuration protocol [10] to automate the configuration procedures for all OVS-based bridges and tunnels. Note that, in Fig. 2, bridges and tunnels are matched with OpenFlow Switch nodes and Capsulator nodes.

However, although the existing tool in [9] can help the automated provisioning, it is limited in terms of software flexibility since it narrowly focuses on how to automate with scripting. First, with the existing tool, it is hard to dynamically manipulate diverse networking configurations. It simply combines together pre-arranged automated site-specific scripts for all SmartX sites. Only those who know the details of all configurations, commands, and APIs are possible to modify the automated scripts. Next, the existing tool lacks fine-grained configuration capability and aims only to automate the provisioning procedures. Specialized configurations in building bridges, making flows, and creating tunnels are not supported. Thus, the operators should manually configure fine-grained parameters, whenever needed. Finally, there is no automated tool to find and recover inter-connection problems, which are occurred in the OF@TEIN playground. The operators can notice inter-connection problems based on user's claims during their experiments. Even though the existence of inter-connection problems is noticed, the operators have to spend a lot of time to figure out the root cause. Besides, even after the operators find out the root cause, the existing tool could not cover the recovery configurations. Thus, finding and recovering inter-connection problems are sometimes slower than the brute-force automated provisioning procedures that flush all existing configuration and reconfigure them. These inflexibilities of existing tool verify the observations such as “hard to manipulate the tool”, “lack of fine-grained configuration capability”, and “no support for finding and recovering inter-connection problems”, which puts additional burden to the operators of OF@TEIN playground.

3 Overlay VNetworking Manager

3.1 Design

To overcome the limitations of existing tool discussed earlier, we have refactored it for flexible inter-connection management under the name of OvN-Manager. The OvN-Manager tool is designed to satisfy the following requirements.

- All configuration procedures are automated from creating the virtual switches (i.e., bridges) to downloading flow rules.
- Anybody can easily design his/her own overlay virtual networking and change its configuration.

- The inter-connection status and resource are repeatedly monitored at each site and the spotted problem is reported for automatic recovery attempts. Also, the operators are notified about the spotted problems and their recovery progress.
- Easy-to-use configuration is supported for bridges, ports, tunnels, and flows at all sites in a fine-grained manner. Also, according to inter-connections among sites, flows can be dynamically created and modified.

The important design concept of OvN-Manager is networking template for flexible provisioning and management. To realize flexible configuration for the operators, the OvN-Manager tool configures each site according to the specified networking template, which contains the customized details. The OvN-Manager tool has only one common code to automate different provisioning cases based on template files. Figure 4 represents an example networking template that is being used for managing the SmartX Box in Pakistan site. The networking template contains detailed configurations for bridges, DPID (data path identifier), ports, and so on. With this networking template, the SmartX Box in Pakistan site can be automatically configured for overlay virtual networking as represented in Fig. 3. Furthermore, we manage an additional tunneling list file for automated tunnel management. All tunnels required for overlay virtual networking are listed in this file.


```
<BRIDGE>
        <NAME>xenbr0</NAME>
        <PORT>eth5</PORT>
</BRIDGE>

<BRIDGE>
        <NAME>xenbr1</NAME>
        <PATCH_PORT>
                <FROM>xbr1_br1</FROM>
                <TO>br1_xbr1</TO>
        </PATCH_PORT>
</BRIDGE>

<BRIDGE>
        <NAME>br1</NAME>
        <DPID>00:00:11:11:11:11:11:07</DPID>
        <PORT>eth0</PORT>
        <PORT>eth1</PORT>
        <PORT>eth2</PORT>
        <PORT>eth3</PORT>
        <PATCH_PORT>
                <FROM>br1_xbr1</FROM>
                <TO>xbr1_br1</TO>
        </PATCH_PORT>
        <PATCH_PORT>
                <FROM>br1_br2</FROM>
                <TO>br2_br1</TO>
        </PATCH_PORT>
</BRIDGE>
<BRIDGE>
        <NAME>br2</NAME>
        <DPID>00:00:11:11:11:11:11:08</DPID>
        <PATCH_PORT>
                <FROM>br2_br1</FROM>
                <TO>br1_br2</TO>
        </PATCH_PORT>
        <PATCH_PORT>
                <FROM>GJ</FROM>
                <TO>C_GJ</TO>
        </PATCH_PORT>
        <PATCH_PORT>
                <FROM>MYREN</FROM>
                <TO>C_MYREN</TO>
        </PATCH_PORT>
</BRIDGE>

<BRIDGE>
        <NAME>brcap</NAME>
        <DPID>00:11:11:11:11:11:11:06</DPID>
        <PATCH_PORT>
                <FROM>C_GJ</FROM>
                <TO>GJ</TO>
        </PATCH_PORT>
        <PATCH_PORT>
                <FROM>C_MYREN</FROM>
                <TO>MYREN</TO>
        </PATCH_PORT>
</BRIDGE>
```

Fig. 4. OvN-Manager networking template for Pakistan site.

3.2 Implementation and Working Procedure

The OvN-Manager tool, implemented in a bash shell script, is working at a centralized management box in GIST. From this management box, the OvN-Manager tool provisions, monitors, and recovers the overlay virtual networking of all remote SmartX Boxes in OF@TEIN playground. The implemented OvN-Manager tool works by following the procedure depicted in Fig. 5. This tool operates in 3 different main phases: Bridge, Tunnel, and Flow Management Phases. Figure 5 represents Bridge-Tunnel-Flow Management Phases with detailed working procedure. After completing all 3 phases, the checked errors are saved into a report file. This report file is automatically sent to the operators by e-mail. Also, the OvN-Manager tool recycles again after 30 s.

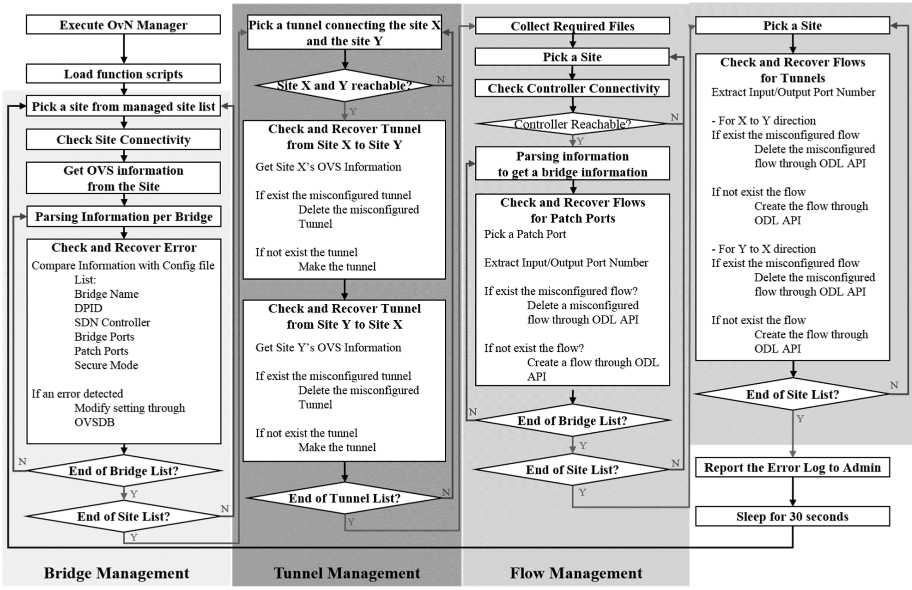

Fig. 5. OvN-Manager: Flow diagram for working procedure.

First, in Bridge Management Phase, the manager collects current OVS configurations from all sites through the OVSDB protocol. All bridges in each site are automatically checked by comparing against listed bridges in the networking template one by one. If any listed bridge is missing, it is automatically created or recovered.

Next, Tunnel Management Phase takes care of managing all tunnels listed in the tunneling list file. In this phase, unlike Bridge Management Phase, each tunnel should be checked at both tunneling endpoints to ensure bi-directional connectivity. The manager first checks the connectivity (e.g., ping) and OVS configuration in one direction. After that, it also checks in the reverse direction. The spotted problems are fixed by configuring the tunnel patch ports through the OVSDB protocol. Note that, since all bridges and tunnels are instantiated with OVS virtual switches and all sites are internationally distributed, the OvN-Manager tool mainly utilizes the OVSDB protocol to configure overlay virtual networking for remote sites.

Lastly, in Flow Management Phase, all flows are dynamically enabled according to the OVS configuration and the OpenFlow flowtable at each site. To manage the flows effectively, we divide Flow Managements Phase into two distinct steps. One step manages flows between internal bridges and another step covers only tunneling-related flows. Initially, the manager checks the connectivity between bridges at each site and the ODL SDN controller, because all flows are managed by the SDN controller. Next, all flows between internal bridges are checked by comparing with the OVS configuration and flow tables originated from the SDN controller. Then, the missing or misconfigured flows are automatically configured through ODL APIs. After finishing the internal flow management for all sites, tunneling-related flow management begins. Again, flows should be checked in both directions by following similar checking and recovering mechanism for internal flow management. Also, all flows are automatically arranged according to the existing rules on port names. For example, if all packets coming from “xbr1_br1” should be forwarded to “br1_xbr1”, the flow from “xbr1_br1” to “br1_xbr1” is defined and configured. Note that, in this regard, the applicability of OvN-Manager tool is restricted to OF@TEIN playground yet.

3.3 Evaluation: Features and Limitations

The OvN-Manager tool provides several features for the OF@TEIN operators, coming from its functionalities, to effectively manage testbed infrastructure. The operators can provision overlay virtual networking. The template-based configuration alleviates the burden of operators in modifying networking configurations. The problem recovery and reporting can also improve the management efficiency by reducing trouble-shooting time. Also, due to template-based automation support, the OvN-Manager tool can be generalized. By decomposing into Bridge-Tunnel-Flow Phases, the OvN-Manager tool can leverage specific functionalities of each isolated phase. It can be easily generalized by plugging with other tools. Thus, Bridge, Tunnel and Flow Phases give improved scalability by increasing the number of manageable nodes at a time. The time-consuming procedure of the OvN-Manager tool is collecting information through OVSDB, SSH, and ODL APIs. In each phase, all boxes provide the same type of information to the manager. Thus the manager can simultaneously collect information from distributed SmartX Boxes. Moreover, we can easily add additional functionality due to phase-based procedures.

On the other hand, the OvN-Manager tool is limited, mostly from the implementation. The OvN-Manager tool can address tunnel-based connectivity for overlay virtual networking. However it is suitable only for overlay virtual networking in the OF@TEIN testbed. Similarly, the operator may have to spend time to apply the OvN-Manager tool on one’s own environment.

3.4 OvN-Manager: Functionality Verification

To verify the monitoring and recovering problems, we apply a simple experiment to the SmartX Box at Pakistan site. We detach the SDN controller from all bridges. Then, according to the Pakistan networking template, we define additional bridge “testbr” and a patch port connecting “TEST_A” to “TEST_B”. Also, another experiment removes a tunnel connecting between SmartX Boxes at Malaysia site (MY2) and Pakistan site (PK1).

Finally, the reports from OvN-Manager are represented as shown in Fig. 6. The OvN-Manager tool can detect and distinguish problem types in a fine-grained level (i.e., Bridge/Port/Tunnel/Flow/Controller problems, repectively), and send a report to the operators. If the default recovery fails, the OvN-Manager tool includes recovery failure notification. Additionally, the OvN-Manager tool repeats its operation every 30 s, which is sufficiently freqent for dynamic operator intervention.

Fig. 6. Problem Reports from OvN-Manager.

4 Conclusion

In this paper, we introduce OvN-Manager, an automated tool for overlay networking management, which can check and recover problems for multi-point international OF@TEIN sites. In future, we plan to improve its recovery features by adding VMs and package recovery and integrate the OvN-Manager tool with GUI-based efficient visibility [11].

Acknowledgments. This work makes use of results produced by the SmartFIRE project, which is supported by the International Research & Development Program of the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT and Future Planning (MSIP, Korea) (Grant number: K2013078191) and the Seventh Framework Programme (FP7) funded by the European Commission (Grant number: 611165).

References

1. Right Scale: Right Scale 2014: State of the cloud report. Survey Report (2014)
2. Koomey, J.G.: Growth in data center electricity use 2005 to 2010. Analytical Press, New York (2011)
3. Microsoft: Microsoft had more than 1 million servers in 2013. http://news.microsoft.com/2013/07/08/steve-ballmer-worldwide-partner-conference-2013-keynote/

4. Kim, J., et. al.: OF@TEIN: An OpenFlow-enabled SDN testbed over international SmartX Rack sites. In: Proceedings of the APAN–Networking Research Workshop, Aug 2013
5. Risdianto, A.C., Na, T., Kim, J.: Running lifecycle experiments over SDN-enabled OF@TEIN testbed. In: Proceedings of the IEEE International Conference on Communications and Electronics, July-Aug. 2014
6. Risdianto, A.C., Kim, J.: Prototyping media distribution experiments over OF@TEIN SDN-enabled testbed. In: Proceedings of the APAN Network Research Workshop, Nantou, Taiwan, Aug. 2014
7. Sherwood, R., et al.: Flowvisor: A network virtualization layer. Technical Report Openflow-tr-2009-1, Stanford University, July 2009
8. Medved, J., Varga, R., Tkacik, A., Gray, K.: OpenDaylight: Towards a model-driven SDN controller architecture. In: Proceedings of the IEEE International Symposium on A World of Wireless, Mobile and Multimedia Networks, June 2014
9. Na, T., Kim, J.: Inter-connection automation for OF@TEIN multi-point international OpenFlow islands. In: Proceedings of the International Conference on Future Internet Technologies (CFI 2014), Tokyo, Japan, June 2014
10. Pfaff, B., Davie, B.: The open vSwitch database management protocol. In: IETF RFC 7047, Dec. 2013
11. Risdianto, A.C., Kim, J.: Flow-centric visibility tools for OF@TEIN OpenFlow-based SDN testbed. In: Proceedings of the International Conference on Future Internet Technologies (CFI 2015), Seoul, Korea, June 2015