A low workload operation method for SDN switch replacement that prevents wiring mistakes

Che Huang1a), Tomoya Hatano1, Takashi Yamada1, Tatsuya Shimada1, and Tomoaki Yoshida1

1NTT Access Service Systems Laboratories, NTT Corporation, 1–1 Hikari-no-oka, Yokosukashi, Kanagawa, 239–0847, Japan
a) sei.kou.mv@hco.ntt.co.jp

Abstract: In order to dramatically reduce the operation work needed for switch replacement, we propose a new operation method that allows connections among arbitrary port pairs. We build and test a system that implements our proposal. The system detects wiring mistakes, and supports replacement with different or same switch models. A wiring experiment shows our proposal can enhance wiring efficiency.

Keywords: Switch replacement, Automation, Configuration, Wiring

Classification: Network management/operation

References

[1] https://www.statista.com/statistics/821488/japan-broadbandsubscription-number-by-type/
[2] https://www.statista.com/statistics/1021346/ftth-customers-ofchina-telecom/
[3] https://www.ieee802.org/21/doctree/2006_Meeting_Docs/2006-11_meeting_docs/802.3ah-2004.pdf.
[4] Katiyar, R., Pawar, P., Gupta, A., and Kataoka, K., “Auto-configuration of SDN switches in SDN/non-SDN hybrid network” In: Proceedings of the Asian Internet Engineering Conference. 48-53(2015).
[5] Gebauer, O., Ohms, J., Wermser, D., and Wähling, S. O., “Mechanisms for the Automated Setup of Software-Defined Networks” ITG-Fachbericht-Mobilkommunikation–Technologien und Anwendungen 263(2016).
[6] Nippon Telegraph and Telephone Corporation, Ryu Topology Viewer. https://github.com/osrg/ryu/blob/master/doc/source/gui.rst.
[7] McKeown, N., Anderson, T., Balakrishnan, H., Parulkar, G., Peterson, L., Rexford, J., … and Turner, J. “OpenFlow: enabling innovation in campus networks” ACM SIGCOMM Computer Communication Review 38.2, 69-74 (2008).
[8] J. Kani, M. Yoshino, K. Asaka, H. Ujikawa, T. Yamada, K. Nishimoto, K. Suzuki, and A. Otaka, “Flexible access system architecture(FASA) to support diverse requirements and agile service creation,” J. Lightwave Technol. 36, 1510–1515 (2018).
1 Introduction

Many network devices such as network switches are used in data centers and telecommunications carriers. Due to finite device lifetimes, EOL, network migration, etc., a lot of network device replacement work is required. Currently, the number of FTTH subscribers in Japan and China exceeds 30 million and 100 million, respectively [1, 2]. According to the current GE-PON [3] specifications and Eq. (2), the access network of Japan uses at least 39062 optical line terminals (OLT), and 1220 (39062/32) network switches (32 port) that aggregate OLTs. Assuming such devices must be replaced once every 10 years, approximately 4028 network devices must be replaced annually, which incurs a lot of operation work.

Maximum number of subscribers per OLT =

\[
12 \times \text{(PON-PKG number per OLT)} \times 2 \times \text{(number of ports per PON-PKG)} \times 32 \quad \text{(Maximum number of subscribers per port)}
\]  

(1)

Number of OLTs = 30million / Maximum number of subscribers per OLT  

(2)

As shown in Fig. 1 (a), replacing a network switch (Initial tasks such as OS and IP of new switch are assumed to have been completed) is generally performed as follows.

1. Prior-design work by the operator
   - Step 1.1. Operator designs the number of wires between the new switch (D) and the adjacent switches (A, C). Example: Two between Switch A and D.
   - Step 1.2. Operator creates a database of connection port numbers between switch D and switch A, C. Example: connect Switch A port 1 with Switch D port 2.
   - Step 1.3. Operator creates a database for configuration (routing) for switch D and switch A, C based on step 1.2. Example: Switch D, if src Mac=11:11, then forward to port 2.

2. Wiring and setting work by operator
   - Step 2.1. Operator performs wiring between the switches based on step 1.2.
   - Step 2.2. Operator configures the switches based on step 1.3.

All these steps require a lot of operation work. To raise operation efficiency, many automatic technologies have been proposed [4, 5]. However, these technologies address only the initial network switch setting tasks, no technologies have been proposed to assist overall network switch replacement.
In order to dramatically raise operation efficiency, we propose a novel switch replacement operation method that allows connections among arbitrary port pairs. Applying this proposal, operator can connect switch D and switch A, C using any available physical ports in step 2.1 without wiring mistakes. Moreover, as shown in Fig. 1 (b), we introduce a system that can automatically complete steps 1.2, 1.3, and 2.2 by simple command inputs. The proposed system is explained in Chapter 2. Experiments and results are described in Chapter 3.

2 Proposed System

To ensure versatility, our system provides the function of wiring mistake detection, and supports replacement between different switch models (different communication speed and number of wires) as well as the same model. In order to automate steps 1.2, 1.3 and 2.2 in Fig. 1 (b), the proposed system needs to automatically generate configuration that reflect the physical topology.

Our solution includes the processing procedure and functions shown in Fig. 2 (a). The proposed procedure follows Fig. 1 (a) for ease of explanation.

- First, the proposed system accepts the replacement command (ID of old and new switch) from operator.
- Topology Detection then automatically acquires all feasible network paths by using the Link Layer Discovery Protocol (LLDP). This makes it possible to detect wiring mistakes in step 1.1.
- Next, Switch Information Acquisition automatically acquires and analyzes the configuration of switches A, B and C. The configuration for bidirectional communication flow via switch A, B and C can be extracted by using an identifier such as source MAC address.
- Next, Switch Configuration Generation automatically acquires the network topology and physical communication speed between switch D and switch A, C. The new network paths are automatically designed, and the extracted communication flows are allocated to the new network paths in order. Based on the allocation, the configuration of switch D is generated while the configuration of switches A and C is regenerated.
Finally, Switch Configuration Setting automatically sets switches D, A and C by using generated configuration.

As shown in Fig. 2 (b), we implemented the Auto Configuration Interface on the OSS called Ryu Topology Viewer [6]. The system uses OpenFlow [7] to realize Software-defined networking (SDN). Recently, virtualization of the access network using SDN is underway as well as the core network [8]. By using SDN, we can centrally control multi-vendor devices with the same control software without having to create device-specific implementations. In addition, we can update functions simply by modifying the control software. As a result, it becomes possible to realize and modify automation functions at low cost.

3 Experimental evaluations
We evaluated the proposed system using replacement experiments between the same and different switch models. The conventional method was used as the benchmark in the wiring experiments.

3.1 Same switch model replacement
This experiment evaluated the system shown in Fig. 3 (a) consisting of 3 physical servers (Ubuntu 16.04.5 LTS) and 4 Pica8 OpenFlow switches (P-3290). The configuration server hosted the proposed system to manage the 4 OpenFlow switches (A, B, C and D) by connecting the physical NIC of the server and the management ports of the Openflow switches with cables and L2 switch. Four virtual machines were prepared for each of the two physical servers (Host A and Host B). Each virtual machine has its own physical port on the physical server. In this experimental environment, the physical servers and switches had the physical port speed of 1Gbps.

Four configurations were tested as shown in Fig. 3 (b). The experimental method is explained for configuration No.2 of Fig. 3 (b).

- Connect switches A and B (same for D and C) with 2 cables.
- Use 2 virtual machines in Host A (a1, a2) and Host B (b1, b2) respectively.
- Configure routing for switches A, B and C so that the virtual machines of Host A and B could communicate(Ping) with each other.
- Connect any two ports of switches A and D (same for D and C) with 2 cables.

In the above state, input the command to initiate replacement of switch B with D. After the replacement is completed, the configuration of switches A, B, and C and communication status between the virtual machines are checked to see if replacement was completed correctly.

The experiment confirmed that the replacement was successful for all configurations tested. We used iperf to observe the throughput between the virtual machines during replacement. As shown in Fig. 3 (c), throughput decreased to 660 Mbps only at the 7th second after the command was entered, and then returned to the normal level.

3.2 Different switch model replacement
This experiment evaluated the system shown in Fig. 3 (d). The difference from the
setup shown in Fig. 3 (a) is use of a Dell OpenFlow Switch (S4148F-ON) as switch E. All ports of switches A and C connected to switch E were 10 Gbps ports. We tested the 5 configurations shown in Fig. 3 (e). The proposed system was evaluated in the same way as in 3.1, and the effectiveness of the system was again confirmed.

![Experimental environment for same switch model replacement](image1)

![Experimental environment for different switch mode replacement](image2)

![Experimental methods for same switch model replacement](image3)

![Experimental methods for different switch mode replacement](image4)

![Result of throughput measurement by iperf](image5)

![Comparison of wiring time between the conventional method and the proposed method](image6)

### 3.3 Wiring experiment

We prepared two Pica8 OpenFlow switches (P-3290) and 10 LAN cables on a desk for the wiring experiment. Eight participants were asked to individually perform the wiring using the conventional method and the proposed method.

- **Conventional method:** We prepared a piece of paper with wiring instructions. The complete instructions for connecting the two switches with the 10 cables consisted of 10 lines of text. Instruction example: connect port 1 of Switch A with port 2 of Switch B. The participants completed the wiring according to the instructions. However, participants did not check if the wiring was done correctly.

- **Proposed method:** The participants connected any port pairs of the two
switches using the 10 cables without fixed instructions.

We measured the time taken by each participant to connect all port pairs for the two methods and used it as the completion time of wiring. A comparison of the completion time between the two methods is shown in Fig. 3 (f). It shows that the proposed method reduced the time to complete wiring by 44% on average. In addition, we also checked for the wiring mistakes after the participants completed the wiring. We found that 2 of the 8 participants using the conventional method made wiring mistakes; the proposed method caused no wiring mistakes.

4 Conclusion
In this paper, we proposed a switch replacement operation method that allows connections among arbitrary port pairs. Our proposed method can dramatically reduce the operation work of switch replacement without wiring mistakes. We implemented a system that provides the function of wiring mistake detection, and supports replacement with different or same switch models. The effectiveness of the proposed system was verified by experiments. The proposed operation method was shown to reduce the wiring time. In future research, we will consider replacement of different numbers of switches for large-scale network migration.