

## 오픈소스 플랫폼을 통한 SDN 구축

송병후\*, 김상영\*, 송준석<sup>0</sup>, 김경태\*, 윤희용\*\*

<sup>0</sup>성균관대학교 정보통신대학 전자전기컴퓨터공학과

\*\*성균관대학교 소프트웨어대학 소프트웨어학과

e-mail: {by911129, impsoft, alskpo}@skku.edu<sup>0</sup>, kyungtaekim76@gmail.com\*, youn7147@skku.edu\*\*

## SDN deployment Via an Open Source Platform

Byung-Hoo Song\*, Sang-Young Kim\*, Jun-Seok Song<sup>0</sup>, Kyung-Tae Kim\*, Hee-Yong Youn\*\*

<sup>0</sup>Dept. of Electrical and Computer Engineering, Sungkyunkwan University

\*\*Dept. of Software, Sungkyunkwan University

### ● 요약 ●

본 논문은 SDN을 가상머신을 통하여 구현하는 방식에 대해 서술한다. SDN은 최근 들어 네트워크 분야의 패러다임으로 부각되었으며 여러 분야에 적용되고 있다. OpenDayLight는 SDN을 구축하기 위한 오픈소스 플랫폼으로 SDN과 NFV를 모두 제공하는 점에 있다. 본 논문에서는 OpenDayLight를 통하여 컨트롤러를 구축하고 Mininet를 통하여 스위치 구축을 통해 SDN 환경을 구축하는 방법에 대하여 서술한다.

**키워드:** SDN, OpenDayLight, Mininet

## I. Introduction

Software Defined Networking(SDN)은 H/W에서 하던 일은 S/W에서 처리하는 네트워크를 제어하는 패러다임으로 큰 주목을 받고 있다.[1] SDN은 중앙의 컨트롤러와 스위치 이렇게 크게 2가지의 구조로 구성되며 컨트롤러를 통하여 전체 네트워크를 제어하고 스위치를 통하여 데이터를 수집한다.[2] 최근 SDN의 연구로는 전체 단일 컨트롤러의 구조에서 컨트롤러가 고장 날 경우 전체 네트워크가 다운되는 문제점 및 동기화에 관한 연구와 데이터의 처리를 위한 처리속도 및 확장성에 관한 연구가 진행되고 있다. [3, 4] 이러한 SDN을 구축하기 위한 플랫폼으로는 컨트롤러부분에는 OpenDayLight, FloodLight, NOX등 여러 플랫폼이 존재하며 스위치를 구축하기 위한 플랫폼으로는 Mininet등이 있다. [5] Mininet은 개인 pc에서도 손쉽게 가상 네트워크 환경을 구성하여 성능평가를 테스트할 수 있는 오픈소스 플랫폼이다.

본 논문에서는 SDN의 환경을 구축하기 위해서 컨트롤러의 구축은 OpenDayLight를 통하여 구축하고 Mininet을 통하여 스위치를 구축하는 방법론에 대하여 서술한다.

## II. Preliminaries

### 1. Related works

#### 1.1 SDN Topology

SDN 네트워크의 Topology 구성은 다양하며 Switch의 호스트의 접속 수 및 컨트롤러 및 스위치의 개수 등 다양한 변인 요소가 존재한다. 먼저 single형은 N개의 host가 하나의 switch로 연결되는 형태이고 Linear은 N개의 스위치에 M개의 host가 연결되는 Topology 형태이다. 다른 형식의 Topology 형태는 Tree, Tours 등 다양하며 single형과 linear의 토폴로지 형태는 Figure 1과 같다.

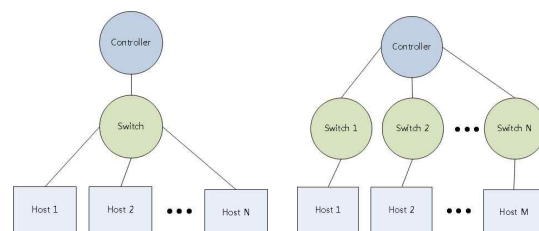


Fig. 1. SDN Topology

### III. The Proposed Scheme

최근 네트워크의 패러다임인 SDN 환경을 구축하기 위해 먼저 컨트롤러를 위한 OpenDayLight, 스위치 구축을 위한 Minet을 설치한다. OpenDayLight를 통하여 Minet을 시작하고 컨트롤러에 연결한다. Minet을 컨트롤러에 연결한 후에 확인을 하면 Figure 2와 같은 Topology를 확인할 수 있다.

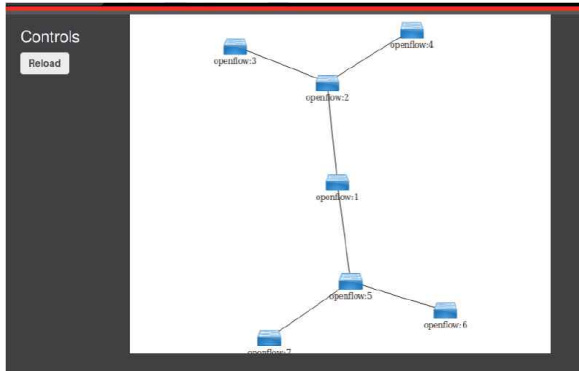


Fig. 2. 오픈소스 플랫폼을 통해 구축한 Topology

Figure 2에는 스위치만 존재하며 호스트는 확인되지 않는다. 호스트는 호스트가 데이터를 보낼 경우에 LLDP 방식에 의해서 결정된다. Host가 데이터를 보낼 경우 Figure 3와 같은 SDN 환경이 구축되었음을 알 수 있다.

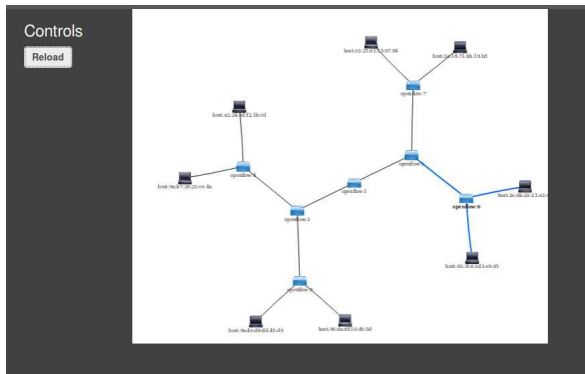


Fig. 3. SDN 환경 구축

### IV. Conclusions

본 논문에서는 일반적으로 많이 사용하는 오픈소스 플랫폼인 OpenDayLight, Minet을 통한 SDN 구축방안에 대하여 서술했다. 향후 연구로는 SDN을 통한 IoT 환경 구축 및 SDN의 단일 컨트롤러의 문제점을 해결하기 위한 동기화에 관한 연구를 진행하고자 한다.

### Acknowledgment

본 연구는 Institute for Information & communications Technology Promotion(IITP) grant funded by the Korea government(MSIP) (No.B0717-16-0070), Science and Technology (2016R1A6A3A11931385), the second Brain Korea 21 PLUS의 일환으로 수행되었음.

### References

- [1] Haleplidis, Evangelos, et al. Software-defined networking (sdn): Layers and architecture terminology. No. RFC 7426. 2015.
- [2] Kreutz, Diego, et al. "Software-defined networking: A comprehensive survey." Proceedings of the IEEE 103.1 (2015): 14-76.
- [3] Yan, Qiao, et al. "Software-defined networking (SDN) and distributed denial of service (DDoS) attacks in cloud computing environments: A survey, some research issues, and challenges." IEEE Communications Surveys & Tutorials 18.1 (2016): 602-622.
- [4] Huang, Tao, et al. "A Survey on Large-scale Software Defined Networking (SDN) Testbeds: Approaches and Challenges." IEEE Communications Surveys & Tutorials (2016).
- [5] Brooks, Michael, and Baijian Yang. "A Man-in-the-Middle attack against OpenDayLight SDN controller." Proceedings of the 4th Annual ACM Conference on Research in Information Technology. ACM, 2015.