Software-Defined Networking (SDN) : A Reference Architecture and Open APIs

Myung-Ki Shin, Ki-Hyuk Nam, Hyoung-Jun Kim

ETRI, 161 Kajong-Dong, Yusong-Gu, Daejeon, 305-350, Korea
{mkshin,nam,khj}@etri.re.kr

Abstract – Recently, Software-defined networking (SDN) is emerging and intensively discussed as one of the most promising technologies to make networks programmable and virtualizable, so that SDN could simplify network operations, reduce cost and accelerate service delivery. Even if openness which gives network customers choice to build best-of-breed their networks is one of main properties of SDN, open architecture, interfaces, APIs, etc. are not fully investigated yet. In this paper, we discuss SDN reference architecture and APIs for open networking environment.

***Keywords* – Software-defined networking (SDN), Southbound, Northbound, East-west bound interfaces, Network virtualization, Programmbility**

I. INTRODUCTION

Up to now, there have been no big changes on traditional networking technologies since 1970. That is, intelligence resided largely at the end point of network, the computers, while network equipments such as routers and switches were relatively dumb post offices of various size, mainly confined to reading address and transferring packets of data to adjacent system. In addition, network control and data planes were developed to be tightly coupled with a closed, proprietary way. In this regard, network complexity and management and operation cost more increase in networks, whenever new services, technologies or hardwares were newly adopted and deployed in networks.

Recently, software-defined networking (SDN) technologies are emerging and intensively discussed including mobile, data centers, enterprise networks, etc. as one of the most promising technologies to introduce and realize network virtualization. SDN is a new networking technology which allows centralized, programmable control planes and data plan abstraction, where control and data planes are separated, so that network operators and/or service providers can control and manage directly their own virtualized resources and networks without recognizing detailed hardware technologies. To achieve this, with SDN, control and data planes are separated which allows control to be directly programmable and manageable in a centralized manner and data plane to be simplified and abstracted rather than specialized hardware.

The reasoning that SDN is necessary for network operators/ service providers are summarized as follows :

- The increase of network intelligence and resource sharing in carrier networks : a lot of the information is stored and processed on computers out on the carrier networks (e.g., data center, clouds, etc.). Therefore network operators/service providers need to use SDN technologies to more easily and efficiently control and manage the intelligence and resources on their networks.
- The need for network programmability : network operators/service providers need more intelligent control systems to directly orchestrate the behavior of thousands of routers and switches.
- Emerging service-aware networking in networks : As customers want to establish their specific purpose services (e.g., on-demand "express lanes" with guaranteed QoS for voice and data traffic, that is, time-sensitive), network operators/service providers should provide a certain method for interaction between the services and network infrastructure and then the services should be securely isolated from other existing customers' traffic.
- High complexity of operations and management in netwotks: As network complexity more increases in networks, network operators want to reduce management and operation complexity with software-defined networking rather than configured networks

II. SDN REFERENCE ARCHITECTURE AND OPEN APIS

Nowdays, SDN technologies are very popular and there exists many mechanisms to realize it. In this paper, we tried to gather a common concept and understating on SDN technologies comparing with traditional networking technologies [1][2]. Simply, current networking technology is characterized with (1) control and data planes tightly coupled, with close, proprietary ways, (2) distributed control systems of network equipments, and (3) use of a single physical network infrastructure. By comparing with them, software-defined networking is a new networking approach and its key attributes include: (1) separation of control and data planes, (2) centralized, programmable of network equiments, and (3) Support of multiple, isolated virtual networks [3]. Figure 1 illustrates a reference architecture and relevant APIs for SDN.

A. Programmable Control Plane

With SDN, control and data planes are separated, and control is directly programmable in a centralized manner. It is required that SDN has three open interfaces, including southbound, northbound, and east-west bound interfaces for control planes. Each interface should be designed with the following requirements.

- Southbound interface : an interfaces between programmable control plane (e.g., controller) and data plane. Objectives of southbound interfaces are discussed as follows
  - Programmability and quick re-configurability : Southbound interfaces should support flexibility in the control plane and make it possible to easily adopt new control schemes on networks. The southbound interface

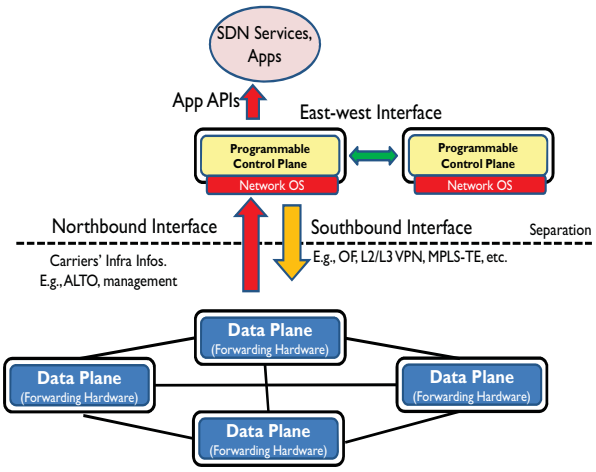

Figure 1 SDN reference architecture and APIs

is used for easy and rapid creation of virtual networks and dynamic reconfiguration of them.

- Sharing of resources : Southbound interfaces should allow the abstraction of physical resources' characteristics so that controllers and/or applications can access the capabilities of resources by using this interface.
- Traffic isolation : Southbound interfaces should provide the secure isolations among multiple, virtual networks such as performance and security aspects.
- Network abstraction : Southbound interfaces should abstract the information of physical network resources and support the simplified or the higher level interfaces for resource control, in order to disengage the virtual networks from the complexity characteristic of the physical network resources.

○ Northbound interface : an interface between applications/services and network infrastructure (or controllers). Objectives of northbound interfaces are discussed as follows :

- Routing-related requirements : Southbound interfaces should provide the following well-defined routing–related information from network infrastructure or controllers for services/applications (e.g., topology discovery, traffic engineering, delay, jitter, QoS, etc.)
- Management-related requirements : Southbound interfaces should provide the following well-defined management–related information from network infrastructure or controllers for services/applications (e.g., resource, energy use, monitoring, maintenance, accounting, etc.)
- Policy-related requirements : Southbound interfaces should provide the following well-defined policy–related information from network infrastructure or controllers for services/applications (e.g., access control, security, etc.)

○ East-west interface : an interface between controllers. Objectives of east-westbound interfaces are discussed as follows.

- Intra-domain, inter-domain, scalability, interoperability : deployability, etc.

B. SDN Languages and Formal Specification

It is required that SDN has formal specification method and languages for application/service development.

With SDN operators and application/service providers can introduce a new capability easily by writing a simple software program. To design and implement networks that conform to the design goals of software defined network topology, the structure and behavior of the networks need to be formally specified to prevent from misinterpreting of the intended meanings and to avoid inconsistency in the networks. In SDN, incomplete or malicious programmable entities could cause break-down of underlying networks shared by heterogeneous devices and stake-holders. Formally verifiable networking in software defined networking can reduce any inconsistency or misunderstanding of the meaning of components and mechanisms because formal specification removes ambiguity in the informal specifications. Furthermore, formal specification can be applied to verification methods such as theorem proving, process algebraic analysis, model checking, and static analysis.

C. Data Plane Abstraction

It is required that SDN has well-defined, common data plane models, rather than specific hardware, which includes :

○ Packet forwarding abstraction models (Ethernet, IPv4, IPv6, etc.)

○ Circuit switching abstraction models (Optical, MPLS, etc.)

○ Wireless integration, characterization of wireless interfaces, flows, handover support

○ Evolved packet core, LTE support

III. Conclusions

This paper discusses a reference architecture and open APIs for SDN. Further works will be discussed with analysis of SDN use cases and detailed requirements on each standard interface. In particular northbound interface will be intensively investigated for ecosystem of SDN components/modules (i.e., controller, data forwarding hardware, etc.).

Acknowledgements

This research was supported by the ICT Standardization program of KCC (Korea Communications Commission) (2012-PK-10-18).

References

[1] OpenFlow Switch Specification version 1.3, Open Networking Foundation, http://www.opennetworking.org/, 2012.

[2] Open Networking Foundation, "OpenFlow/Software-Defined Networking (SDN)," https://www.opennetworking.org/.

[3] M-K. Shin, ITU-T Draft Recommendation Y.FNsdn, "Framework of Software-Defined Networking (SDN) for Future Networks," (work-in-progress), 2012.

[4] M-K. Shin et al., IETF I-D, "Formal Specification of Software-Defined Networking (SDN)," draft-shin-formal-specification-01.txt, (work-in-progress), 2012.