NTT DATA's Efforts for OpenFlow/SDN

Hiroshi Nagasono

Abstract

This article discusses the recent technological trend of OpenFlow/SDN and NTT DATA's efforts related to it for future business. Software Defined Networking (SDN) is currently in the spotlight for its capability to control networks flexibly without being bound by conventional network technologies. NTT DATA has been researching and developing OpenFlow, which is technology for achieving SDN, and has developed its own OpenFlow controller.

1. Introduction

TCP/IP (transmission control protocol, Internet protocol) and Ethernet are currently the current main computer network technologies. To exchange signals, these technologies use various transmission protocols standardized by international organizations such as the Internet Engineering Task Force (IETF) and the Institute of Electrical and Electronic Engineers (IEEE), and such protocols are based on the OSI (Open Systems Interconnection) Reference Model^{*1} that describes the communication layers and their functions. Network devices produced by different manufacturers can communicate with each other by complying with these standards.

Most device manufacturers develop network device software along with the hardware to enable the hardware to perform operations appropriate for each communication layer. The device manufacturers also implement their own networking functions in addition to the standard protocols to add value to their products and thus make them more competitive. However, this makes it difficult to replace devices that have exclusive functions with ones from other manufacturers, so network owners may need to keep on using the same manufacturer's devices. This situation is called *vendor lock-in*. Another issue with the current network is the difficulty of functional enhancement. The requirements for information systems are becoming more complicated and advanced, and you may want to upgrade your network to respond to such requirements. However, the functional limitation of network devices sometimes inhibits network upgrading because the devices do not allow network configurations unsupported by the device manufacturers. For network device users who wish to configure the network as they wish, the only option is to wait for the device manufacturers to support their desired functions.

One solution that can address these problems with conventional networks is Software Defined Networking (SDN) based on OpenFlow [1].

2. OpenFlow/SDN

2.1 Definition of SDN

Let us clearly define SDN before proceeding to discuss OpenFlow. As the name implies, SDN is a system architecture in which networking is defined by software. Although the name SDN has become widely known only recently, the software defining architecture itself has been in use for some time. A conventional network device comprises hardware and software. However, users could not previously define a network by themselves because the device manufacturers did not publish the application programming interface (API) for controlling the hardware.

OpenFlow was introduced in response to such situations. It is one of the technologies capable of

^{*1} OSI Reference Model: A model with a set of layers that define the different communication functions to be used on the networks and devices connected to the networks. Suggested by the International Organization for Standardization.



Fig. 1. Basic behavior of OpenFlow.

achieving SDN. It is not a method of networking that provides specific functions, such as L2 (Layer 2) switching or IP routing. In fact, OpenFlow is simply an interface to be installed on a network device. OpenFlow specifications are defined by the Open Networking Foundation (ONF) [2], which promotes the spread and standardization of SDN. The specifications cover the components and the basic functions of a switch as well as the OpenFlow protocol for managing an OpenFlow switch from a remote controller. OpenFlow opens up access to the API controlling the hardware—even though the API is not disclosed by the device manufacturers—and enables users to define networking by themselves.

2.2 Control of OpenFlow switches

Although the OpenFlow switch specifications are standardized, the controller's behavior is not. The communication unit handled by OpenFlow is called a *flow*. The behavior of a conventional network device is as follows (here an L2 switch is used as an example): the L2 switch identifies the transmitting destination's physical port using a destination media access control (MAC) address^{*2} as an identifier and submits frames from the physical port (In the case of an L3 switch, the destination IP address is used as an identifier.). Like conventional network devices, OpenFlow

uses the same header information including a destination IP address and an originating MAC address as components; however, the difference is that Open-Flow can handle the combination of such components as an identifier. For example, in OpenFlow it is possible to configure an identifier as a flow submitted from a specific originating MAC address to a specific destination IP address, or as TCP port 443 communication received from a specific physical port. When a flow is identified, the OpenFlow switch performs control operations such as packet transmission and packet correction on the flow. The information for controlling this flow is called the *flow entry*, and the information is stored in a *flow table* within the Open-Flow switch. Controllers usually create a flow entry in response to a request from an OpenFlow switch and update the flow table in the switch (Fig. 1).

A series of such control operations enables an OpenFlow switch to emulate the functions of a conventional network device such as an L2 switch and router. Furthermore, it also makes possible network configurations that used to be unachievable by conventional network devices such as L2 segments

^{*2} MAC address: A physical address assigned to network device hardware such as a local area network (LAN) card to identify each node on the networks.

configured using only MAC addresses without using a virtual local area network (VLAN).

2.3 Implementation of OpenFlow/SDN

Practical usage of OpenFlow is now available to provide flexible network control. There are two configurations for implementing OpenFlow: hop-by-hop and overlay.

2.3.1 Hop-by-hop configuration

In the hop-by-hop configuration, the controller centrally manages the end-to-end communication by setting all the network devices via the OpenFlow switch. This configuration makes best use of the following OpenFlow features.

(1) Flexible routing

Flexible routing appropriate for the communication characteristics is possible such as routing a high-priority communication to a physical path that has sufficient bandwidth while routing lower priority ones to other paths.

(2) Traffic monitoring

Since an OpenFlow switch is capable of managing the flow status by using a flow table, it is possible to recalculate routes when traffic congestion is anticipated.

2.3.2 Overlay configuration

In the overlay configuration, servers and network devices at the edge of datacenters are managed by OpenFlow, and communication to the edge is managed by the conventional networks. The overlay configuration also uses tunneling for communication between the originating and destination edges.

(1) Existing facilities can be used.

Not every network device on the communication path needs to be configured by the OpenFlow switch. OpenFlow can be used partially within the conventional network. If Open vSwitch, which is an Open-Flow-compliant software switch, is embedded in a virtual server, such servers can function as edge devices; this allows network management by Open-Flow.

(2) The amount of routing information can be limited.

Network controllers store edge information and logical network routing information, but not physical routing information between edges. This reduces the number of items of routing information managed by the controller.

3. Virtual network controller

NTT DATA is in a position to provide information

system solutions as a global information technology (IT) innovator. We have recently received requests from our corporate customers for network configurations that cannot be achieved with conventional network devices. We intend to utilize OpenFlow/SDN to achieve the optimum network solution that can accommodate such requests. To achieve this, we have developed our own virtual network controller and started providing it. It has the following features: (1) Easy to customize

The controller is composed of two types of software: NetworkOS (NOS) and NOS-Application (NOS-AP) (**Fig. 2**).

NOS-AP is software for defining networking; it can perform routing based on its own algorithm and provide logical configuration management. NOS-AP can also be coordinated with higher systems that handle other network management tasks such as OpenStack. On the other hand, NOS provides functions commonly required in controllers based on the OpenFlow specifications, such as functions for generating Open-Flow protocols and handling events. Along with the software, we provide NOS-API for efficiently controlling NOS from NOS-AP. NOS-API helps reduce the controller development costs and shorten the development period.

(2) Enables a small start

As mentioned earlier, we offer the hop-by-hop and overlay configurations for implementing OpenFlow. One barrier to the upgrading of network technologies is that existing network devices with limited capabilities are still usable, so delaying upgrading is seen as a viable option. The overlay configuration of OpenFlow/SDN can solve such problems by enabling network technology to be upgraded while leaving existing network devices untouched. These network devices can be changed into OpenFlow-compliant devices one by one as required, so it is possible to configure the network with advanced technologies in stages (**Fig. 3**).

(3) High reliability

Because the OpenFlow/SDN architecture allows central control of networks, the controller can become a single point of failure. Large, complicated networks such as the public cloud or the networks of service providers require a mechanism for flexibly upgrading performance and functions. A virtual network controller secures network availability through the highavailability cluster configuration. Furthermore, if controller functions are deployed in a distributed manner, the controllers can also offer scalability to the network.



Fig. 2. Software configuration of the virtual network controller.



Fig. 3. Stepwise implementation of OpenFlow/SDN.

4. Concluding remarks

OpenFlow/SDN used to be applied mainly to datacenter networks as technology for virtualizing networks in order to catch up with more advanced server technology. However, there are a number of other areas where OpenFlow/SDN technology can be applied to solve problems that cannot be solved using conventional network technology. NTT DATA will enhance the functions and capabilities of its virtual network controller to solve such problems.

References

- OpenFlow: Realization of the New Network which Reverses an Old Concept (in Japanese). http://thinkit.co.jp/book/2012/02/01/3150
- [2] Open Networking Foundation. https://www.opennetworking.org/



Hiroshi Nagasono

Senior Expert, Systems Engineering Business Unit, System Platform Sector, NTT DATA Corporation.

He received the M.E. degree in mathematics from Nagoya University, Aichi, in 1998. He joined NTT in 1998. From 2002 to 2007, he was in the R&D Center of NTT EAST. He developed network service systems, such as IP-telephony. Since moving to NTT DATA in 2007, he has developed some network solutions, such as the CTI system, IP-FAX, and OpenFlow/SDN controllers.