Network Operation System for the IT/Network Integrated Control Plane

SHIMONISHI Hideyuki

Abstract

The OpenFlow technology that can control the switch function from an external control server is beginning to attract attention. We offer a method that builds a virtual infrastructure by creating a virtualized control plane on this control server. This paper describes the architecture of the overall virtual infrastructure and the network operating system as the control server integration platform.

Keywords

OpenFlow, infrastructure virtualization, network virtualization
new-generation network, FutureInternet

1. Introduction

Since Internet appeared on the scene and was released for commercial use, it has been used for various applications. Internet is used not only for Web services as computer-to-computer communication that was initially assumed but also for voice communication and video distribution. Various technologies have been added to the Internet technology. As a result, the layer structure that was simple has become more complex. Costs for building, maintaining, and managing the system have increased and it has become to take time to add new features and services to the system [1)]. Even though a new feature is developed, a long times and costs are required to add the feature to the commercial Internet system and start a new service.

To reconsider the Internet infrastructure for which innovations taking time, discussions on FutureInternet such as those in NSF and FIND in the United States, in FP7 in Europe, and those on the new-generation network in Japan have become very active in recent years [1-3)]. As the environment for large-scale demonstration of the new technologies consequently created, large-scale test beds such GENI [4)] are starting to be built and operated.

Under such circumstances, we are examining the architecture of a new Internet infrastructure [5, 6)]. Our goal is not to achieve a "one fits all" type infrastructure like the IP technology. Rather, it is to allow for easy building of a new virtual infrastructure through programmability and virtualization technology to realize a versatile Internet infrastructure. We expect that the versatile infrastructure can keep on evolving changing its form from time to time without relying on particular technologies over years.

The architecture that we offer will basically have programmability for the network feature on the control server using the OpenFlow technology [7)], but not on the switch. Further, the control server will control OpenFlow as well as various infrastructure resources such as wireless and virtual machine (VM) in an integrated manner to allow for building the virtual infrastructure on the control server. This paper offers the "network operating system" as the software platform on the control server to build and operate the virtual infrastructure easily. The network operating system offered will not only play the role of the OpenFlow controller, typically NOX [8)] but also have a feature that provides the control module group with physical resources converted into virtual resources as services and a feature that isolates and manages the control module group per virtual infrastructure as its characteristics so as to build the virtual infrastructure and, as the integrated control plane, control OpenFlow as well as resources including the virtual machine.

This paper describes the summary of relevant research trends, the architecture of the virtual infrastructure offered, and the network operating system to achieve it below. Finally, a trial system of the network operating system and application implementation on the system will be briefly explained.

2. Relevant Trends

To examine the architecture of a new network infrastructure, various proposals have been given in the relevant research field [1-9]. This paper does not discuss each research but raises the following four points as the common requirements [9].

(1) Programmability

To build a new infrastructure with the users' original features, it is necessary that the features of the overall infrastructure including the network can be designed like software. OpenFlow is a technology allowing for realizing switch programmability on the control server and provides for high programmability on the control server.

(2) Virtualization

To ensure programmability in the real infrastructure, it is important that multiple users share the physical infrastructure and the virtualized ones are operated separately by the users.

(3) Openness

Programmability and virtualization must be open so that anyone can use them easily. In the field of computers, operating systems have evolved so that hardware can be easily used and innovations are enabled freely. In the same way, a similar scheme is important for network equipment.

(4) Modularization

It is important that various network equipment is modularized so that users can freely customize the network infrastructure. It is critical for making innovations more active that new features can be easily realized (i.e. mashup) by freely combining the modules developed by various users together.

3. Architecture of the Virtual Infrastructure Offered

This section explains the architecture offered based on the requirements described in section 2.

(1) Integrated Control Plane

Fig. 1 illustrates the concept of the integrated control plane.

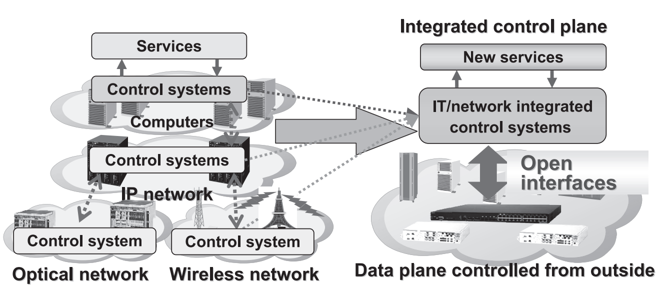

Fig. 1 Integrated control plane.

Typically, a network system is divided into the data plane that actually processes data and the control plane that controls such processing (the management plane controlling it also exists). For each system, as shown on the left in Fig. 1, respective systems such as optical transport and wireless network are independent of each other and the respective control planes are basically loosely coupled. Therefore, unified programmability and virtualization cannot be provided for the overall infrastructure containing various systems as elements. This makes building the virtual infrastructure difficult.

Then, we offer an architecture that totally integrates the control planes as shown on the right in Fig. 1 and integrates and controls various types of physical infrastructures from this integrated control plane. As discussed in section 1, programmability is provided on the control plane and the building environment for the unified virtual infrastructure is provided on the control plane.

(2) Infrastructure Virtualization

Fig. 2 illustrates infrastructure virtualization using the integrated control plane. To realize multiple virtual infrastructures on a single physical infrastructure, a virtual resource group (e.g. virtual switch and virtual machine) sliced out of the physical infrastructure is provided for each virtual infrastructure. By combining each virtual resource group with the original control plane, a virtual infrastructure optimized for each application can be built. For example, if a virtual infrastructure for a sensor network is necessary, the necessary virtual resources such as the virtual machine and virtual switch are sliced out of the physical infrastructure. By combining the control programs for sensor network route control, virtual machine control, etc. into the integrated control plane, a virtual infrastructure is formed.

4. Network Operating System

The software platform of the integrated control plane for building the virtual infrastructure as described above is the network operating system. As illustrated in **Fig. 3**, the network operating system is the middleware on the control servers that controls the physical resources including the OpenFlow switch. It provides for open APIs for controlling physical resources such as control modules for route control, network measurement, etc. that have been virtualized. Not only to play the role of the OpenFlow controller but also to build the virtual infrastructure, and then to control the virtual machine and

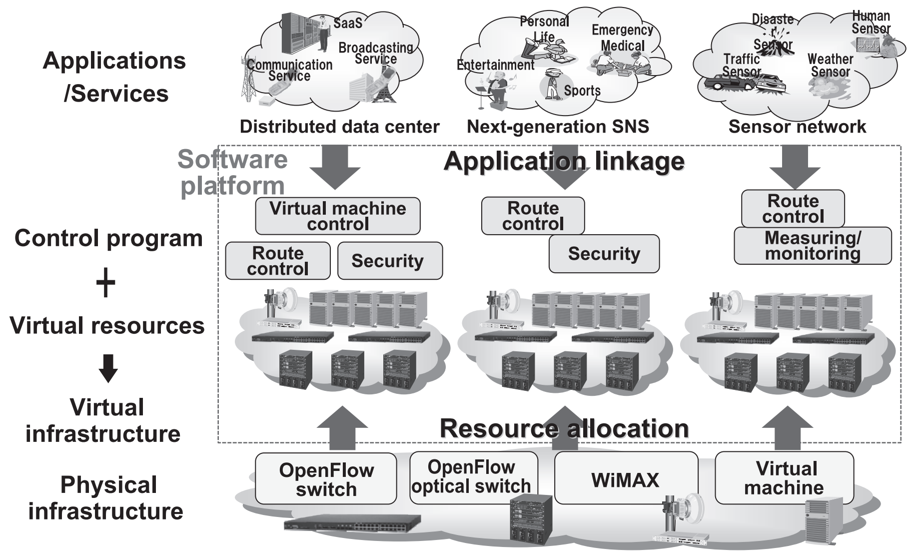

Fig. 2 Building a virtual infrastructure.

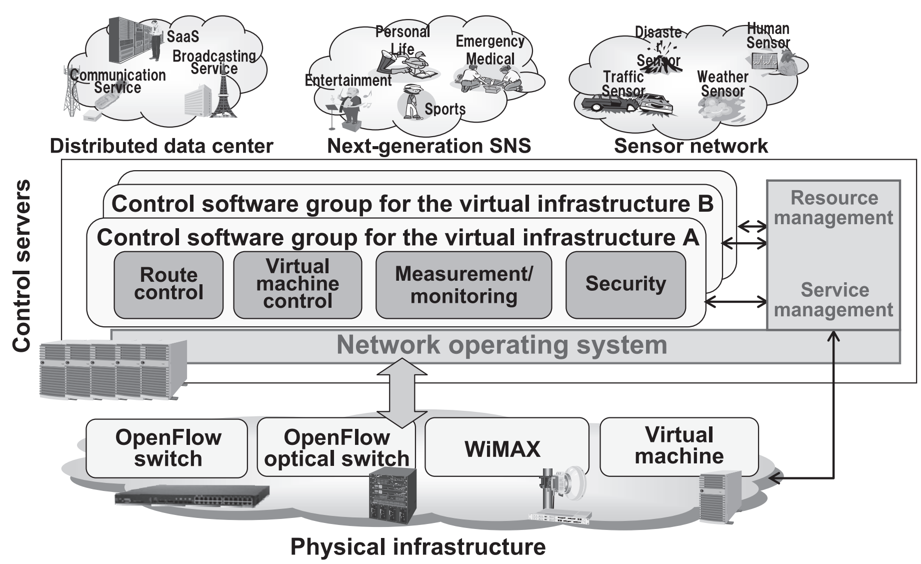

Fig. 3 Network operating system.

other resources in addition to OpenFlow, the network operating system has a control structure that links control modules and services hierarchically as illustrated below. With this structure, a virtual infrastructure can be easily built by combining multiple modules together.

(1) Services

Services are defined to give manipulation to resources including virtualized physical resources such as the virtual

switch and virtual machine as well as logical resources such as the network topology and flow path that are created by control modules.
For example, service "OpenFlow switch" gives the virtual switch as the virtual resource and the API for setting a flow for the switch. Service "Flow path" gives the logical resource as an aggregate of path for the flow and the API for manipulation such as path creation and deletion.
On the network operating system, a feature is provided that gives unique name spaces for these services. Therefore, on the network operating system being offered, services are defined to have the following:

1) Service name
2) Data structure associated with the virtual resource/logical resource
3) API for manipulating the resource

The network operating system being offered, as illustrated in **Fig. 4** , has a control structure, which hierarchically combines a structure, in which the control module performs processing using the virtual resource/physical resource as a service, with another structure, in which the control module creates a new service as a result of processing.

(2) Control Modules

Control modules are the entities on which network control algorithms for topology discovery, path creation, etc. are implemented. These include those which control the network as well as server resources for server load balancing, virtual machine migration, etc. As shown on the bottom in Fig. 4, control modules that are directly connected to physical resources such as the switch and provide them as

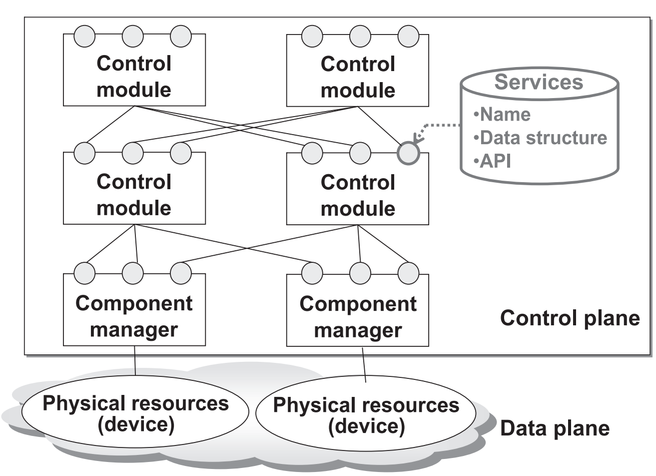

Fig. 4 Basic structure of the control model.

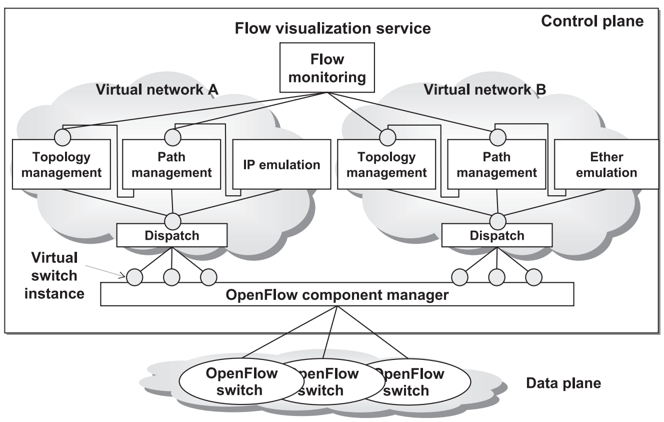

Fig. 5 Building a virtual infrastructure.

virtual resources are referred to as component managers in particular. For example, the OpenFlow component manager is connected to multiple OpenFlow switches via TCP/SSL and plays the role of providing other control modules with a virtual resource such as bus creation, so-called "virtual switch instance."

(3) Building a Virtual Infrastructure

With the control structure as described above, the network operating system being offered builds virtual infrastructures. For example, as illustrated in **Fig. 5** , a virtual infrastructure is built by stacking control modules from the service group, that is, the virtual switch instances provided by the OpenFlow component manager as the base point. Further, by dividing the service described above into several services and configuring multiple control module groups, multiple virtual infrastructures are built.

5. Prototype and Evaluation

To demonstrate the architecture being offered, we are producing the network operating system and various control module groups on the operating system experimentally and evaluating them. This section explains its outline.

The prototype system, as illustrated in **Fig. 6** , consists of the control server on which the network operating system is implemented and the OpenFlow switch group and server group controlled by them. On the network operating system, a group of control modules such as Topology discovery (topology management), Path manager (route setting/management), Path resolver (route calculation), IP emulation (network emulation

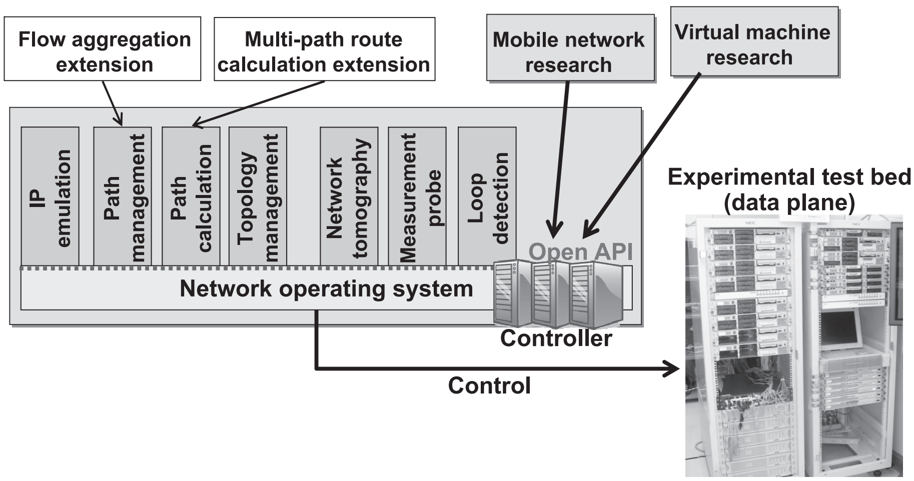

Fig. 6 Test bed configuration.

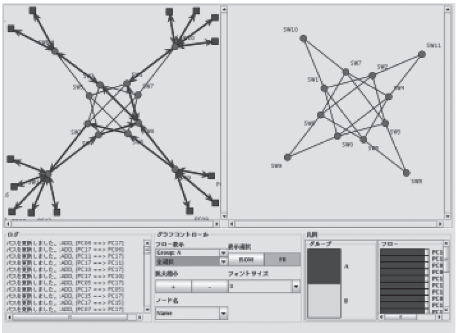

Fig. 7 Visualization of the virtual network.

via Ether/IP) is implemented to experiment the following network features:

(1) Visualization of the Virtual Network

As illustrated in **Fig. 7**, the topology of the virtual network and the flow on it are dynamically visualized. In this example, two virtual networks are built on a single physical network. Load balancing is applied to each TCP/IP flow on the tertiary hypercube network.

(2) Flow-based Mesh Network

To realize a large-scale mesh network to which load balancing is applied on a flow basis, a large-scale multi-path route calculation algorithm is offered and implemented as an extension of the Path manager module [11]. Additionally, the flow aggregation system is offered to reduce the switch route table and implemented as an extension of the Path manager module [10].

(3) Automation of Building the Control Network

An OpenFlow-native building system is offered for the control network between the control server and OpenFlow switch [12].

(4) Network Tomography

An active probe linked with arbitrary test flow path setting through OpenFlow and a quality degraded part estimation system linked with Topology discovery/Path manager are offered [13].

(5) Flow Loop Detection

Loop detection based on passive measurement per flow and a loop removal system per flow is offered [14].

6. Conclusion

This paper described openness, virtualization, modularization, and programmability as the requirements for realizing a versatile Internet infrastructure and offered an architecture based on them. Further, a network operating system for the integrated control plane as the technical core was proposed. The network operating system offered in this paper is a software platform for the integrated control plane that controls overall virtual infrastructure including the OpenFlow switch. For demonstration, we produced the network operating system and various control module groups experimentally and evaluated them. We reported the outline in this paper.

References

1) "AKARI,"
http://akari-project.nict.go.jp/
2) NSF NeTS FIND Initiative,
http://www.nets-find.net/
3) Seventh Framework Programme (FP7),
http://cordis.europa.eu/fp7/dc/index.cfm
4) "GENI,"
http://www.geni.net/
5) Hideyuki Shimonishi, Shuji Ishii, "A network OS for integrated control plane and its application to OpenFlow controller," Technical report of IEICE, NS2009-162, 2010 (JAPANESE).
6) Hideyuki Shimonishi and Shuji Ishi, "Virtualized network infrastructure using OpenFlow," in proc. of BcN workshop, invited talk #2, 2010.
7) N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "Openflow: enabling innovation in campus networks," SIGCOMM Comput. Commun. Rev., vol.38, no. 2. 2008.
8) NOX | An OpenFlow Controller,
http://noxrepo.org/wp/
9) Toshiyuki Kanoh, "JISEDAI Internet JITSUGEN NI MUKETA GIJUTSU DOUKOU TO NEC NO TORIKUMI (Technology trend toward the realization of the next-generation Internet and NEC's approach)," Dempa Shinbun, 7th Jan, 2010 (JAPANESE).
10) Yasunobu Chiba, Hideyuki Shimonishi, Yusuke Shinohara, "A Proposal of Flow Entry Reduction Scheme for Flow-based Networks and Its Implementation on OpenFlow-based Network," Technical report of IEICE,, NS2009-163, 2010.
11) Yuusuke Shinohara, Yasunobu Chiba, Hideyuki Shimonishi, "An Efficient Multipath Routing Algorithm for Datacenter Networks and its Implementation on OpenFlow-based Network," Technical report of IEICE, NS2009-164, 2010.
12) Toshio Koide, Hideyuki Shimonishi, "A study on the automatic construction mechanism of control network in OpenFlow-based network," Technical report of IEICE, NS2009-165, 2010.
13) Youichi Hatano, Yasuhiro Yamasaki, Hideyuki Shimonishi, "An Efficient Measurement Method Flow Placement for QoS Degradation Locating on OpenFlow-based Network," Technical report of IEICE, NS2009-166, 2010.
14) Nobuhiko Itoh, Yasuhiro Yamasaki, Hideyuki Shimonishi, "An Efficient Calculation of Passive Type Loop Detection Method and its Implementation on OpenFlow-based Network," Technical report of IEICE, NS2009-167, 2010.

Author's Profile

SHIMONISHI Hideyuki
Principle Researcher
System Platforms Research Laboratories