

Ryu SDN Framework— Open-source SDN Platform Software

*Rui Kubo, Tomonori Fujita, Yuji Agawa,
and Hikaru Suzuki*

Abstract

We introduce Ryu SDN Framework, an open source software (OSS) system developed by NTT, and describe our efforts toward promoting its wider adoption. Ryu SDN Framework is a platform that provides tools and libraries for easy use of SDN (software-defined networking). We have taken advantage of the strengths of OSS in developing Ryu, which conforms to the most recent industry-driven OpenFlow specifications. Ryu is used by a variety of experts including SDN application developers, network device developers, and network service maintainers.

Keywords: SDN, OpenFlow, open source software

1. Introduction

Server virtualization and cloud computing technologies have created an unprecedented need for network adaptability, expandability, and flexibility. An approach to solving those problems that has been attracting attention is SDN (software-defined networking), which enables configuration of networks and the programming of network functions by operating software [1].

The network configuration function is divided into a data plane for handling data transfers and a control plane for handling control of the data plane. In the conventional approach, the two planes are tightly linked within individual network devices, so interworking between the control planes of the devices is necessary when constructing a single network with multiple network devices.

In the SDN approach, on the other hand, the data plane and control plane are separated, and their independence is maintained by specifying an application programming interface (API). Using the API to develop programs according to purpose and use enables higher independence of data plane control. A program that uses this kind of API is called an SDN application.

2. Ryu SDN Framework

Ryu SDN Framework (Ryu) is a framework that provides the libraries and tools that are required in order to develop SDN applications (**Fig. 1** and **Fig. 2**). The framework facilitates development by providing the basic functions for controlling the data plane and the functions that are common to SDN applications. The functions that are features of Ryu are described in the following subsections.

2.1 OpenFlow controller function

OpenFlow is a set of API specifications for the control plane and the data plane formulated by the Open Networking Foundation (ONF); it has been attracting broad attention as the most important API specifications for implementing the SDN approach. *OpenFlow switch* is the term used for the data plane that conforms to the specifications, and *OpenFlow controller* similarly refers to the control plane. SDN applications can be implemented by using these functions.

Ryu has the OpenFlow controller function, so SDN applications that use OpenFlow switches can easily be developed using Ryu. Although various versions of the OpenFlow API specifications exist, Ryu is compatible with a greater variety of versions than other OpenFlow controllers (versions 1.0, 1.2, 1.3,

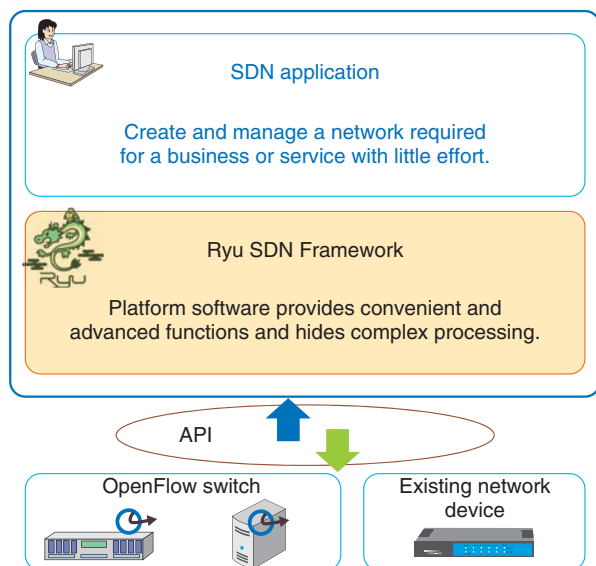


Fig. 1. Overview of Ryu SDN Framework.

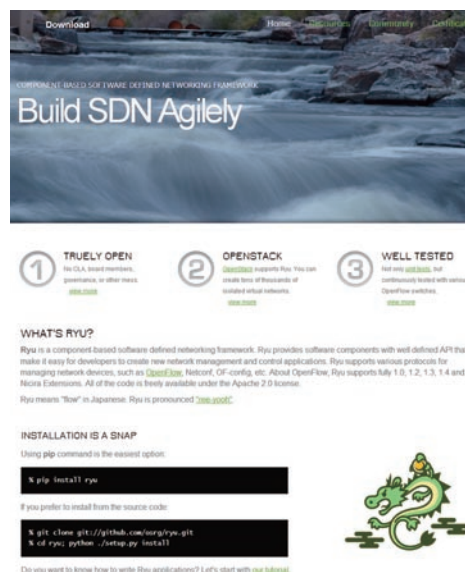
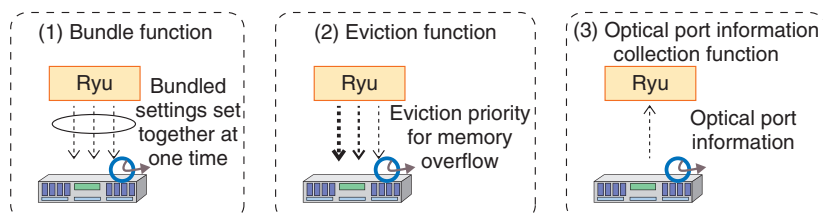
Fig. 2. Ryu public site (<http://osrg.github.io/ryu/>).

Fig. 3. Functions of OpenFlow version 1.4.

and 1.4), so more OpenFlow switches can be used.

The functions of the most recent version, 1.4, which was released in the fall of 2013, are explained as follows and presented in **Fig. 3**.

- (1) Bundle function: This function is for setting multiple commands for an OpenFlow switch at the same time. In the previous versions, it was necessary to confirm the setting of each command individually on the application side. This bundle function facilitates development by making it possible to set multiple commands together as a bundle.
- (2) Eviction function: This function enables processing to be done when memory overflow occurs in OpenFlow switches. The memory capacity for storing the switch settings can vary with the type of machine. In the previous versions, it was necessary to implement the required

processing on the application side for each type of machine in order to prevent memory overflow. This eviction function facilitates development of applications through automatic selection of the eviction command based on the priorities set by the application.

- (3) Optical port information collection function: This enables OpenFlow switches to have 10-Gbit/s or 40-Gbit/s broadband optical ports in the same way as other network devices. With previous versions, it was not possible to obtain the signal wavelength and strength or other information that is specific to optical ports, and it was necessary to use special functions outside the OpenFlow specifications for specific device types to obtain such information. OpenFlow switches that work with this function can obtain the information that is specific to an optical port

in the same way as for an ordinary Ethernet port, reducing the machine-specific implementation required in application development.

2.2 Functions for cooperating with existing network devices

It has been difficult to replace all of the network devices in an existing network with new OpenFlow switches. An OpenFlow switch can transmit various types of data in the same way as the existing network devices, so it is also possible to construct a network in which the devices of the existing network co-exist with OpenFlow switches. For example, when expanding an existing network or dealing with devices that have complex management settings, a practical approach would be to partially and gradually introduce OpenFlow switches by using an SDN application. Ryu provides functions for such introductions of OpenFlow switches in an existing network.

2.2.1 Obtaining management information

Even in existing networks, it is generally common to collect data such as the amount of traffic flow, link status, and other kinds of information from network devices that are compatible with SNMP (Simple Network Management Protocol), NetFlow, sFlow, etc., via network management systems, monitoring systems, or other such means, and to perform total, overall management of the data. The same kinds of information concerning OpenFlow switches can also be collected by the OpenFlow controller function. Ryu provides a function for collecting information from conventional network devices as well as from OpenFlow devices to avoid splitting the management system into two systems when a network configuration includes both OpenFlow switches and conventional network devices. Using Ryu to develop the management system makes it possible to collect information on various network devices other than OpenFlow switches and to perform total, integrated management. When the existing system is taken as the parent system, the functions of the Ryu OpenFlow controller can be called from the existing system so that information collection and total management that includes OpenFlow switches can be performed.

2.2.2 Settings

Some users use provisioning tools for implementing automatic changes in network device ports and bandwidth settings, for example, and also for entering settings. The NETCONF (Network Configuration Protocol) specifications are intended to provide a common procedure for such tasks for conventional network devices.

ONF is also addressing the issue of a common procedure for entering settings, which is not covered by OpenFlow specifications, by establishing the OF-CONFIG (OpenFlow Management and Configuration Protocol) specifications. Ryu accommodates both OF-CONFIG and NETCONF in the switch settings management function, so it is also possible to develop provisioning tools for unified settings changes for OpenFlow switches and existing network devices. The Ryu OF-CONFIG function can also be called from existing provisioning tools.

2.2.3 Exchange of routing data

The exchange of routing information based on the Border Gateway Protocol (BGP) is often used for overall cooperative operation of existing network devices as a single network. For example, data can be efficiently transferred by synchronizing the routing information possessed by network devices in different locations. Because Ryu conforms to the BGP specifications, it is possible, for example, to receive the routing information that is exchanged among existing network devices and send that data to OpenFlow switches. Also, changes made to routing information on the OpenFlow switch side can be sent to existing network devices via Ryu. Thus, when OpenFlow switches coexist with an existing network, routing information can be synchronized by the same mechanism as used previously, and obstacles to the introduction of OpenFlow switches can be reduced.

2.3 Sample applications

Ryu mainly provides tools and libraries that facilitate SDN application development, but sample applications that are easily referenced are also included. The examples include applications that implement firewalls and routers of the kinds that are familiar to most of us as well as applications that implement functions that are often used when constructing networks, such as link aggregation and spanning trees.

3. Efforts to achieve widespread adoption

In addition to developing Ryu, the NTT laboratories are actively encouraging the wider use of this platform. By making Ryu known and available to many people and obtaining feedback on it that can be used to develop an even better platform, we hope to contribute to the overall development of SDN/OpenFlow.

3.1 OSS development

Ryu is open source software (OSS). The source

Ryu Certification

- Overview
- Configuration

Action	OK	ERROR
(Required)	30	26
(Optional)	(9)	(9)
(Optional)	(27)	(26)
set_field	71	99
(Optional)	(71)	(99)
Match	437	265
(Required)	(108)	(0)
(Optional)	(329)	(265)
Total	538	390
(Required)	(311)	(0)
(Optional)	(427)	(390)

Action

	Required	IPv4	IPv6	ARP
OUTPUT	X	OK	OK	OK
PUSH_VLAN	-	OK	OK	OK
PUSH_MPLS	-	OK	ERROR	ERROR
PUSH_PBB	-	ERROR	ERROR	ERROR
PUSH_VLAN (multiple)	-	ERROR	ERROR	ERROR
POP_VLAN	-	OK	OK	OK
COPY_TTL_OUT	-	ERROR	ERROR	
COPY_TTL_IN	-	ERROR	ERROR	
SET_MPLS_TTL	-	OK	OK	OK
DEC_MPLS_TTL	-	OK	OK	OK
PUSH_MPLS (multiple)	-	OK	OK	OK
POP_MPLS	-	OK	OK	OK
PUSH_PBB (multiple)	-	ERROR	ERROR	ERROR
POP_PBB	-	ERROR	ERROR	ERROR

Required ether vlan mpls pbb

Fig. 4. Test results for the OpenFlow switch (<http://osrg.github.io/ryu/certification.html>).

code is published and can be downloaded and used freely under an Apache license [2]. Anyone can incorporate Ryu into their products, commercial or otherwise, and some products that incorporate Ryu are already on the market [3]. The platform was published in 2012, and development has been vigorous ever since, with participation by switch vendors, network providers, systems integrators, academic organizations, and various others.

3.2 Wider use of OpenFlow specifications

The OpenFlow specifications do not necessarily define all functions as essential, and there are some functions whose application depends on the switch. There are also cases in which specific conditions must be satisfied for proper operation, even when conformance to OpenFlow specifications has been achieved. Currently, specific development may be needed for each particular switch because of ambiguities in the OpenFlow specifications themselves.

We are taking advantage of the high degree of Ryu conformity to the OpenFlow controller function specifications to develop a tool for checking the degree of conformance to OpenFlow switch specifications. We are using that tool to test the connection to various OpenFlow switches and are publishing the results on the Ryu website [4] (Fig. 4). Those who are developing SDN applications can understand the

Rybook 1.0 documentation

Next topic
Preface

This Page

Show Source

Quick search

Enter search terms or a module, class or function name

RYU SDN Framework

- Preface
- Switching Hub
 - Switching Hub by OpenFlow
 - Implementation of Switching Hub Using Ryu
 - Execution of Ryu Application
 - Conclusion
- Traffic Monitor
 - Routine Examination of Network
 - Implementation of Traffic Monitor
 - Executing Traffic Monitor
 - Conclusion
- REST Linkage
 - Integrating REST API
 - Implementing a Switching Hub with REST API
 - Implementing SimpleSwitchRest13 Class
 - Implementing SimpleSwitchController Class
 - Executing REST API Added Switching Hub
 - Conclusion
- Link Aggregation
 - Link Aggregation
 - Executing the Ryu Application
 - Implementing the Link Aggregation Function with Ryu
 - Conclusion
- Spanning Tree
 - Spanning tree
 - Executing the Ryu Application
 - Spanning Tree by OpenFlow
 - Using Ryu to Implement Spanning Tree
 - Conclusion
- OpenFlow Protocol
 - Match
 - Instruction
 - Action
- Packet Library
 - Basic Usage
 - Application Examples
- OF-Config Library
 - OF-Config Protocol
 - Library Configuration
 - Usage Example

Fig. 5. E-book reference for Ryu (<http://osrg.github.io/ryu/resources.html#books>).

basic characteristics of various OpenFlow switches by browsing the published results.

This tool is also provided together with Ryu and is being used by many switch developers. Using this tool from the beginning of switch development can raise the level of product conformance with the specifications and aid in product quality management.

3.3 Documentation and publications

Although there was little documentation right after Ryu was published as OSS, more and more documentation is being made available. Currently, an e-book that compiles that documentation is being distributed (Fig. 5). Like the software, the book is also in open development, so interested persons are welcome to access it from any handy terminal.

References

- [1] Y. Nakajima, "Standardization Progress in Software Defined Networking/OpenFlow," NTT Technical Review, Vol. 11, No. 2, 2013. <https://www.ntt-review.jp/archive/ntttechnical.php?contents=ntr201302gls.html>
- [2] Ryu public website, <http://osrg.github.io/ryu/>
- [3] Introduction examples of Ryu, http://osrg.github.io/ryu-book/en/html/introduction_example.html
- [4] Test results for the OpenFlow switch, <http://osrg.github.io/ryu/certification.html>



Rui Kubo

Manager, Smart Navigation Business Division, NTT Resonant Inc.

He received the M.S. in electro-communications from the University of Electro-Communications, Tokyo, in 2003. He joined NTT in 2003. As of July 1, 2014, he moved to NTT Resonant Inc. His research interests include distributed computing and software-defined networking. He is a member of the Japanese Society for Artificial Intelligence.



Tomonori Fujita

Senior Research Engineer, Distributed Computing Technology Project, NTT Software Innovation Center.

He received the B.E. and M.E. from Waseda University, Tokyo, in 1998 and 2000, respectively. He has been with NTT since 2000 and has been engaged in research on operating systems. He is a member of IPSJ (Information Processing Society of Japan) and ACM (Association for Computing Machinery).



Yuji Agawa

Senior Research Engineer, Supervisor, Advanced Information Processing Technology SE Project, NTT Software Innovation Center.

He received the B.S. and M.S. in electrical engineering from Keio University, Tokyo, in 1990 and 1992, respectively. In 1992, he joined the NTT Network Information Systems Laboratories. During 2006–2013, he was engaged in commercial development of the Next Generation Network (NGN). He is a member of the Institute of Electronics, Information and Communication Engineers (IEICE).



Hikaru Suzuki

Vice President, Head of NTT Software Innovation Center.

He received the M.E. in industrial and management system engineering from Waseda University, Tokyo, in 1988. In April 1988, he joined NTT Switching System Laboratory. In 2009, he moved to NTT Communications and in 2012, he moved to NTT Software Innovation Center. He served as vice-chair of the IEICE Technical Committee on Information Networks from May 2008 to May 2010, and as the chair from May 2010 to May 2012. He has contributed to the development of a range of network services such as PHS, 3G-mobile, Free-Phone, and NGN. His research interests include information network services and operations. He is a senior member of IEICE and a member of IEEE (Institute of Electrical and Electronics Engineers).
