Open Source Software behind NTT Network Services

Hideki Shina, Masahito Muroi, Hiroaki Kobayashi, Hirokazu Takahashi, Tomoya Hibi, Takeshi Kinoshita, Jun-ya Kato, Tomonori Fujita, and Wataru Ishida

Abstract

Open source software (OSS) has been gaining importance in many industries, including network services. Members of the NTT laboratories have been involved in numerous activities in OSS communities both as users and as active contributors by doing coding and by providing requirements for telecom operators. These efforts are expected to give us a significant advantage over our competitors. This article introduces the involvement of the NTT laboratories in OSS community activities related to OpenStack Blazar, Lagopus, and GoBGP.

Keywords: Blazar, Lagopus, GoBGP

1. Transformation for the development of network services

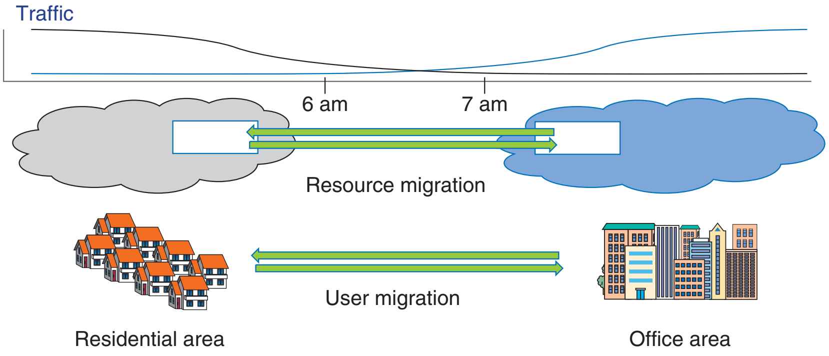
The use of open source technologies has been increasing in various fields, including the telecommunications field, which traditionally relied on the use of well-integrated proprietary software and hardware from specific vendors. Open source software (OSS) has been adapted in network infrastructures just as it was adapted in information technology infrastructures. Current trends in network infrastructures are shifting to utilizing white box (non-branded) systems that are composed of standardized hardware, OSS, and sometimes self-developed technologies. In line with these trends, NTT has been very active in OSS communities, which should contribute to improving the speed and flexibility of development of our network infrastructures.

The NTT laboratories develop various open source products and contribute to many OSS communities, including OpenStack—a major OSS for cloud infrastructure, Open Networking Foundation (ONF)—an organization for standardizing software-defined networking (SDN), and Open Platform for Network Functions Virtualization (OPNFV)—an organization that facilitates the development and evolution of network functions virtualization (NFV) components across various open source ecosystems. These efforts help us to tailor OSS for our services. The NTT laboratories' activities in OSS communities are introduced in the following section.

2. Blazar

Blazar [1] is a service for reserving OpenStack resources such as Compute. It can be used for efficiently managing ownership of limited resources and guaranteeing resource availability for highly prioritized services.

The NTT Software Innovation Center (NTT SIC) and NTT Network Service Systems Laboratories (NTT NS Labs) are jointly promoting the Blazar project. NTT SIC has a lot of experience in developing OSS such as OpenStack. NTT NS Labs, on the other hand, has developed many network services in production. The synergies gained by the collaboration of these two organizations have been accelerating the development of Blazar.



Reserve resources beforehand to guarantee
sufficient system capacity for the increasing traffic.

Fig. 1. Blazar use case: Resource migration.

2.1 NTT activities

The Blazar project was initiated in 2013. The project was named Climate at that time and renamed Blazar later. The first version of Blazar was released with OpenStack Icehouse in 2014. Around that time, the OPNFV Promise project [2] was organized for defining the requirements of resource reservation in NFV and accelerating implementation of related open source projects. The NTT laboratories and NTT DOCOMO have been participating in it. Members of the Promise project had been looking for an open source project that could potentially satisfy Promise requirements for years to come. They eventually approached the Blazar project members, and we began contributing to it. We have since made numerous contributions, which has led to our leading the Blazar project as a Project Team Lead and as core reviewers in the latest release cycle.

2.2 Recent Blazar releases

In March 2017, a new version of Blazar was released for the first time in three years with OpenStack Ocata. This version of Blazar allows operators to reserve computing resources in a host unit. New features such as instance-based reservation and a dashboard interface have since been developed and included in the latest version of Blazar, which was released with OpenStack Pike in August 2017.

We must pay close attention to maintaining consistency between Blazar and other core components such as Nova. In May 2017, we had discussions with Nova project members at the OpenStack Summit Boston, and we agreed that Nova would focus on the current use of resources, while Blazar would focus on the future use of resources. Such decisions are very important for keeping OpenStack simple and well organized.

2.3 Use cases in NFV

The OPNFV Promise project has decided to use Blazar. The following use cases are assumed based on actual operations of telecom operators:

(1) Resource migration: Resources can be reserved based on the prediction of daily traffic. For example, traffic increases in office areas in the daytime and in residential areas at night. An operator can use Blazar to reserve a sufficient amount of resources in office areas in the daytime and in residential areas at night (**Fig. 1**).

(2) Maintenance work: Resources can be reserved that will be used during planned maintenance work such as software updates.

(3) Guaranteed provisioning: Services that need a large amount of resources can fail because of a lack of resources while starting up. Sufficient resources can be reserved beforehand to prevent such failures.

2.4 Future roadmap

Blazar was approved as an official OpenStack project in September 2017. This will accelerate its development. To satisfy the needs of users—especially research institutes and telecom operators—we have been developing new features and plan to release a new version with OpenStack Queens in February 2018. In this release, we are focusing on improving reliability by adding new features such as monitoring

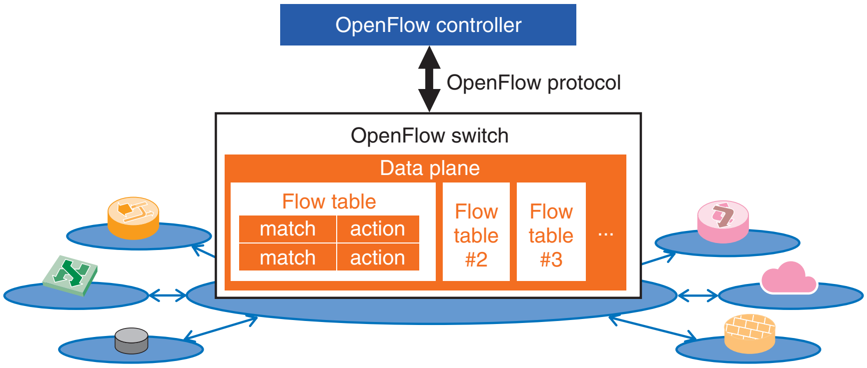


Fig. 2. OpenFlow switch.

of reserved resources and availability zone support [3]. We also plan to support new resources such as network and storage resources in the future.

3. Lagopus

Lagopus [4] is a software switch that enables fast packet forwarding on a general-purpose server while having the highest compatibility with OpenFlow Switch Specification 1.3. Lagopus and other OpenFlow switches operate in accordance with instructions coming from OpenFlow controllers, which specify match conditions and actions (i.e., what actions to take against which types of packets) (**Fig. 2**). Lagopus supports Ethernet, IPv4 (Internet protocol version 4), MPLS (multiprotocol label switching), PBB (provider backbone bridging), and other types of protocols and is also equipped with quality of service functionalities.

The development of Lagopus was initiated by NTT in 2013 with the aim of expanding the applicability of SDN and NFV, both of which have since become widely used technologies in use cases ranging from datacenters to wide area networks, and in 2014 it became open source. Lagopus employed Data Plane Development Kit (DPDK), an input/output library for fast packet handling that became available as OSS just before the development started. The open source Lagopus performed over-10-Gbit/s packet forwarding and processing with 1 million flow table entries, an achievement that was thought to be difficult at that time and thus attracted lots of attention. It now provides 40-Gbit/s packet forwarding and supports fast interconnections with virtual machines as well as a wide variety of tunneling protocols.

3.1 Development of Lagopus

Open-sourcing Lagopus made it possible to achieve cooperation from various organizations and individuals in and outside NTT. It has been used in research and development aimed at commercial uses as well as in field tests exploring novel networks that make the most of OpenFlow capabilities. In general, one type of network equipment can be connected with many other types in a wide variety of uses, which makes it very difficult to carry out interoperability testing exhaustively. By conducting a number of field tests in cooperation with developers, however, we have continuously proved Lagopus's interoperability with other equipment in many practical use cases.

For example, Interop Tokyo, an annual exhibition on information and communication technologies, was a good opportunity to feature Lagopus for that purpose. In the three consecutive years from 2015, it was used in ShowNet, a dedicated network showcasing cutting-edge technologies while providing Internet access to both exhibitors and participants. In each year, Lagopus successfully proved interoperability in different use cases and contributed to the operation of novel networks.

In 2015, Lagopus served as one of the switches that constituted SDN-based Internet exchange (SDN-IX) in ShowNet. We collaborated with members of a Japan-Europe joint research project called NECOMA (Nippon-European Cyberdefense-Oriented Multilayer threat Analysis) and achieved not only layer-2 switching but also interconnection of virtual local area networks and segregation of malicious traffic (**Fig. 3**), winning a special award for ShowNet in the category of Software Defined Infrastructure.

In 2016, we operated Lagopus so as to provide interconnections of virtual networking functions.

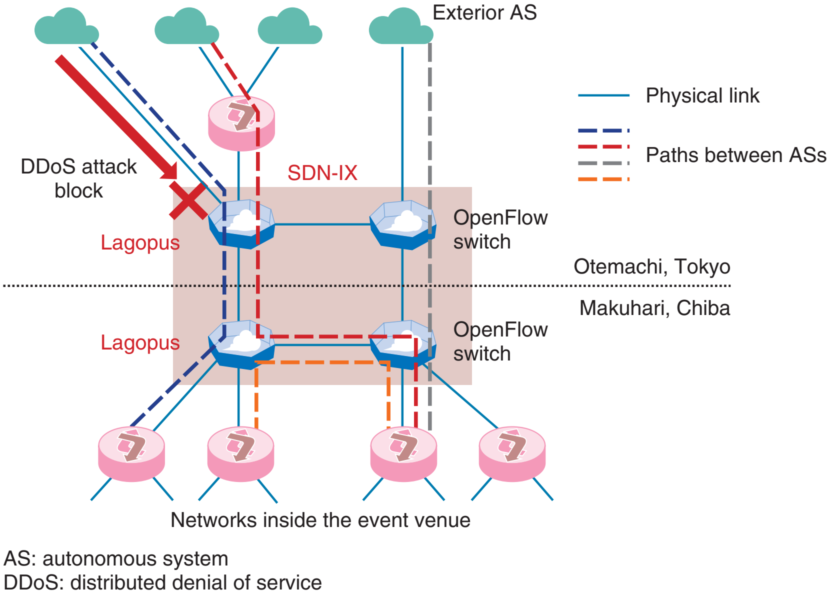


Fig. 3. SDN-IX.

Lagopus used virtual network interface cards (NICs) rather than physical NICs to achieve flexible and fast traffic control. This functionality, with DPDK employed, had been developed for Lagopus and was later contributed to the DPDK community.

In 2017, Lagopus demonstrated tunneling functionality and its interoperability. OpenFlow 1.3 does not support tunneling, which makes it difficult to achieve flexible flow control in an overlay-type network. The tunneling functionality that had been added to Lagopus as an enhancement of OpenFlow 1.3 successfully proved to expand the applicability of the specification.

All of these efforts have helped to enhance Lagopus's functionalities and improve its interoperability and stability.

3.2 Evolving Lagopus

We are now working to equip Lagopus with routing functionalities. Although we have confirmed through the experiments and discussions with outside users that flexible flow control supported by OpenFlow was highly useful, we have also come to conclude that interconnection with existing routers is necessary for Lagopus to be utilized in a commercial network. Against this background, we began developing routing functionalities with the help of the Lagopus community. The initial version was released as OSS in August 2017. The functionalities currently available for Lagopus Router are limited to some simple ones. However, we are making progress in enhancing them for the purpose of achieving a router that enables flexible traffic control using both existing routing protocols and OpenFlow-like match-action instructions.

4. GoBGP

Border Gateway Protocol (BGP) is one type of protocol for performing routing exchange. It has become widely used for routing exchange on the Internet between organizations such as Internet service providers and corporate enterprises known as autonomous systems. BGP has been applied in a variety of use cases, so it is not limited to the Internet. For example, some operators use BGP to exchange label information on a carrier backbone to provide IP virtual private network (VPN) connections, or they apply BGP to network control in datacenters to achieve a distributed arrangement of computing resources such as servers while maintaining scalability. These use cases demonstrate the expansion of BGP application fields.

GoBGP [5] is a BGP implementation written in the Go language, a programming language developed by Google. Its language specifications include standard functions for describing parallel processing, and an extensive library of functions has been prepared. The

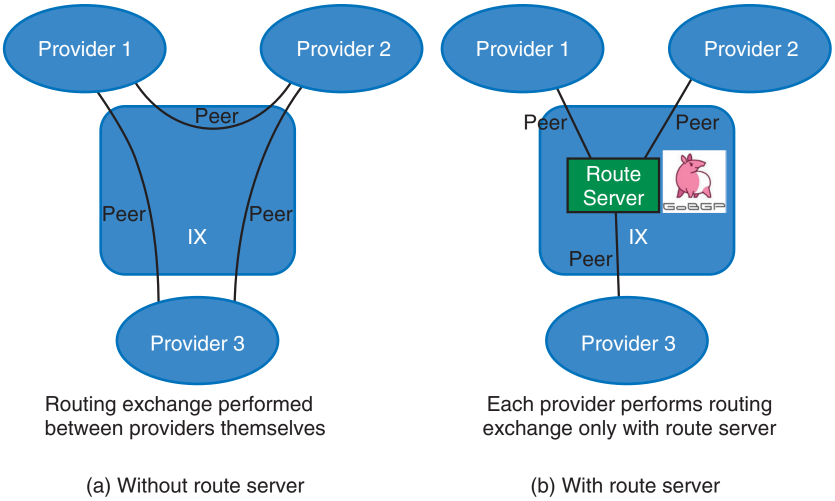


Fig. 4. Role of route server in Internet exchange (IX).

Go language is especially popular in the system programming field. Additionally, because today's server hardware is mounted with multicore central processing units, GoBGP has been designed for high performance by leveraging the features of the Go language and abundant hardware resources.

The main functions currently supported by GoBGP are listed below:

- Provides both application programming interface (API) (gRPC) and command line interface as control interfaces
- Supports a vendor-independent configuration model (OpenConfig)
- Monitoring (MRT: Multi-Threaded Routing Toolkit; BMP: BGP Monitoring Protocol)
- Route reflector (includes Add-Path function)
- Route server
- Policy control
- Resource public key infrastructure (RPKI)
- Flowspec
- VPN functions (Ethernet VPN, layer-3 VPN, virtual routing and forwarding)
- Graceful restart
- Operating system (OS) data plane control (zebra* linking)

In short, GoBGP already supports a variety of functions, but given that BGP use cases are much broader, we are working to add more useful functions for many users while promoting the participation of outside developers by developing GoBGP as OSS.

4.1 GoBGP activities

(1) API for smooth linking with external systems

GoBGP has been commercially deployed as a route server function in the RouteFEED service of the Japan Network Access Point (JPNAP) provided by INTERNET MULTIFEED CO. RouteFEED is a service that mediates routing exchange between various types of providers connected to JPNAP. Without RouteFEED, such providers have to perform routing exchange between themselves, but in the new system, a provider only has to establish a BGP connection (peer) with the route server to perform routing exchange with another provider (**Fig. 4**). The route server must set policy information for each peer, so scalability with respect to the number of peers is required. We have applied high-performance GoBGP to meet this requirement. Additionally, a GoBGP API function for linking with an external operation system is being used to automate [6] complicated setting operations performed manually in the past.

(2) Integration for white box switch control

A white box switch is hardware equipped with general-purpose forwarding-type application-specific integrated circuits (ASICs). The network OS (software) for controlling this hardware can be selected according to the application. An open network OS is often based on Linux. A BGP router can be configured by combining GoBGP with a white box switch

* Zebra: A function included in routing implementations such as Quagga and FRR (Free Range Router). It includes a function for reflecting received routing information in the routing table of the OS kernel in charge of forwarding.

mounted with a Linux-based network OS.

To transfer packets, forwarding equipment must write forwarding rules in a routing table called a forwarding information base (FIB). In GoBGP, there is an FIB operation mechanism linked to the zebra function that enables routes received by BGP to be reflected in the FIB as forwarding rules. For network OS control that targets forwarding ASICs, setting rules in the FIB of that hardware enables the configuration of a BGP router capable of hardware forwarding.

(3) Application to container-oriented virtual network control

Calico [7] is a powerful OSS package for configuring a virtual network required for distributed execution of multiple containers. It supports major container orchestrators such as Kubernetes, Docker, and Mesos and can control connections between various types of systems. BGP has been used for IP connection control between containers, and while BIRD has been traditionally used as a BGP function, GoBGP, which excels in scalability, can also be applied to large-scale use cases.

4.2 GoBGP future outlook

Use cases applying BGP will continue to expand, and the emergence of new protocol extensions can be expected. We aim to apply GoBGP to a variety of use cases through the participation of outside developers in an open-development environment. We also aim to promote the growth of the business market using OSS and to achieve technical advances and invigorate business toward creation of a software-based network infrastructure.

References

[1] Blazar, https://docs.openstack.org/blazar/latest/

[2] Promise, https://wiki.opnfv.org/display/promise/Promise

[3] Blueprints for Blazar, https://blueprints.launchpad.net/blazar

[4] Lagopus, http://www.lagopus.org/

[5] GoBGP, https://osrg.github.io/gobgp

[6] Press release issue by NTT on September 30, 2016.
http://www.ntt.co.jp/news2016/1609e/160930b.html

[7] Calico, https://www.projectcalico.org/

Trademark notes

Hideki Shina

Senior Research Engineer, Supervisor, NTT Network Service Systems Laboratories.

He received a B.E. and M.E. in information and computer engineering from Kanazawa Institute of Technology in 1994 and 1996. He joined NTT Network Service Systems Laboratories in 1996 and studied a distributed and high-availability platform for switching systems. He is a member of the Institute of Electronics, Information and Communication Engineers (IEICE) of Japan.

Masahito Muroi

Researcher, NTT Software Innovation Center.

He received an M.E. in computer science from the University of Electro-Communications, Tokyo, in 2011. He joined NTT the same year. He is now working as a cloud architect for NTT's public/private cloud. He is also a Project Team Lead in the OpenStack Blazar project and one of the core reviewers in the OpenStack Congress project. He has been working on the development of NTT's cloud with OpenStack since the Diablo release in 2011.

Hiroaki Kobayashi

Researcher, NTT Network Service Systems Laboratories.

He received a B.E. and M.E. in computer and information sciences from Tokyo University of Agriculture and Technology in 2011 and 2013. He joined NTT Network Service Systems Laboratories in 2013. He specializes in the area of distributed computing, telecom networks, and NFV. He is an active contributor to the OpenStack and OPNFV communities. He is a core developer of the OpenStack Blazar project and a contributor to the OPNFV Promise project. He is a member of IEICE.

Hirokazu Takahashi

Senior Research Engineer, NTT Network Innovation Laboratories.

He received a B.E. and M.E. in electrical engineering from Nagaoka University of Technology, Niigata, in 2000 and 2002. He is currently researching software architectures for network systems and high-performance packet processing techniques.

Tomoya Hibi

Researcher, NTT Network Innovation Laboratories.

He received a B.E. and M.E. in computer science and engineering from Toyohashi University of Technology, Aichi, in 2010 and 2012. His current research is focused on software networking and packet processing techniques.

Takeshi Kinoshita

Research Engineer, NTT Network Innovation Laboratories.

He joined NTT Optical Network Systems Laboratories in 1996, where he studied optical fiber communication systems in access networks and their management. He then moved to the research and development center of NTT WEST, where he was responsible for developing and introducing commercial services, including broadband internet access, IPv4/v6 VPN, and wide area Ethernet. He also worked on reducing the energy consumption of various components in datacenters such as servers, network equipment, and air conditioning systems. His current research interests include SDN, NFV, and other network softwarization related technologies. He is also involved in standardization efforts for various standards organizations.

Jun-ya Kato

Senior Research Engineer, NTT Software Innovation Center.

He joined NTT Information Sharing Platform Laboratories in 2002, where he studied next-generation Internet protocol IPv6. He joined the architecture design project for IPv6 Internet service provider (ISP) support on NGN (Next Generation Network) in 2008. He also technically supported NTT operating companies participating in World IPv6 Day and World IPv6 Launch, which were global IPv6 deployment events in 2011 and 2012, respectively. He then moved to the Technology Development Department, NTT Communications, where he developed network security services such as DDoS (distributed denial of service) protection in the ISP backbone. His current research is focused on SDN and distributed computing platforms.

Tomonori Fujita

Senior Research Engineer, NTT Software Innovation Center.

He received a B.E. and M.E. from Waseda University, Tokyo, in 1998 and 2000. He joined NTT in 2000 and has been researching operating systems. He is a member of the Information Processing Society of Japan and the Association for Computing Machinery.

Wataru Ishida

Researcher, NTT Software Innovation Center.

He received a B.S. in information science in 2011 and an M.E. in engineering from the University of Tokyo in 2013. He joined NTT Software Innovation Center in 2013 and has been working on the development of SDN software, in particular, Ryu SDN Framework and GoBGP.