

# Hardware/Software Co-design Technology for Network Virtualization

*Takahiro Hatano, Koji Yamazaki, and Akihiko Miyazaki*

## Abstract

Network virtualization technology is a topic of great interest because of its ability to emulate network features in software. In this article, we present a technique for adding programmable hardware acceleration to network virtualization equipment, which is then used in coordination with software to preserve the software's flexibility while raising the limits on its performance.

*Keywords: network virtualization, convergence of software and hardware, FPGA (field-programmable gate array)*

## 1. Introduction

As the performance of commodity servers has improved, we have set out to rapidly provide services with reduced operating and capital expenditures by employing network virtualization technology. Software-defined networking (SDN) is representative of this; it separates the features that control a communication device from the device's data transfer functionality, enabling network configurations and settings to be centralized in and automatically controlled by software. To apply this SDN technology to carriers' wide area networks, NTT has taken the lead in the accelerated research, development, and open-source release of an SDN application to the world. Our application has two components: the Ryu SDN Framework [1] and Lagopus [2]. The Ryu SDN Framework is an SDN controller that provides convenient tools and libraries for easily setting up an SDN. Lagopus is a high-performance SDN software switch with capabilities and features that have broad applications from datacenters to wide area networks.

## 2. Development of hardware accelerator for network virtualization

Network features that have traditionally been pro-

vided by dedicated hardware can now be implemented in software by network virtualization technologies on commodity servers. As the use of network virtualization becomes more common, we can expect commodity servers to be equipped with many network features that will increase the load on the servers' central processing units (CPUs), which will be responsible for handling network-related processing. If we want to meet the demands for even more advanced network services and systems, we will need technology that can alleviate this additional CPU load in order to increase the power/performance ratio of processing on commodity servers and thus provide a stable environment for running more software.

The NTT Device Innovation Center has been involved in researching and developing hardware/software co-design technologies to help hardware and software work together. We developed a hardware accelerator (HWA) [3, 4] that implements the concepts shown in **Fig. 1**, improving performance and reliability while preserving the flexibility of the highest-performance SDN software switch in the world, Lagopus.

## 3. HWA properties

The system configuration of an SDN software

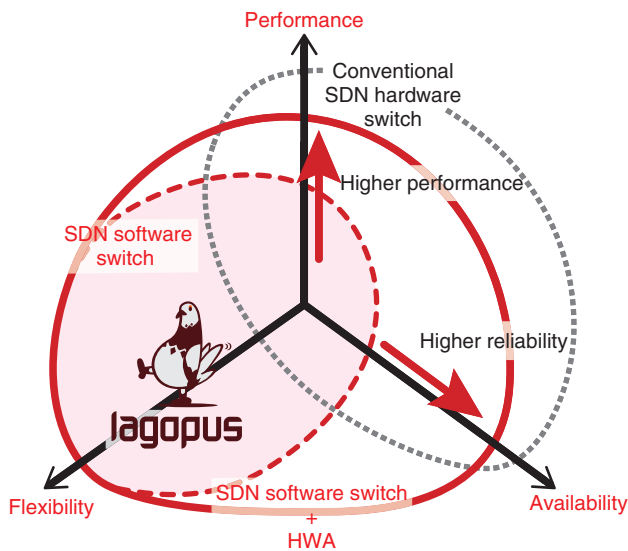


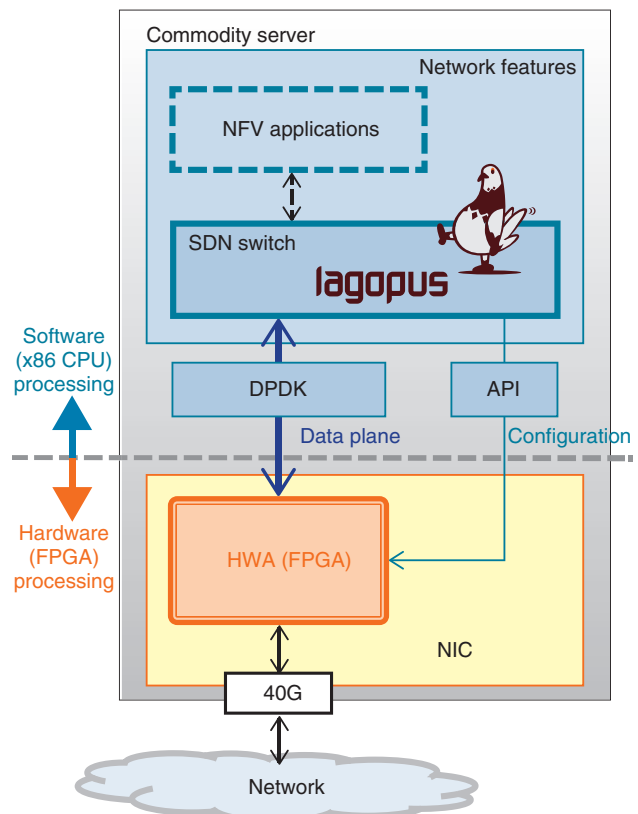
Fig. 1. Conceptual diagram of hardware accelerator.

switch with the HWA we developed is shown in **Fig. 2**. The system is characterized by a reconfigurable HWA—specifically, a field-programmable gate array (FPGA)—placed on a commodity server’s network interface card (NIC). Hardware ordinarily takes much longer to develop than software. However, by using an FPGA, which can be overwritten with features implemented in a program, we can not only speed up the development process, but we can also update network features and quickly make changes to fix problems that arise later.

The SDN software switch system in **Fig. 2** forwards network packets through the FPGA on the NIC to the software that will process them. In addition to the NIC’s ordinary packet processing, the FPGA has implemented preprocessing features to assist with the first part of the heavy processing in Lagopus’s packet forwarding component (i.e., the data plane). The HWA’s preprocessing cannot directly communicate with—and is independent of—Lagopus’s packet processing. This allows Lagopus to retain its flexibility while reducing the software load and further improving the performance of the SDN software switch. Specifically, we have developed the following three features.

#### (1) High-speed flow dispatcher

High-speed packet processing on a software switch requires evenly distributing the work among the processing units (cores) in a multi-core processor (CPU), which can process multiple instructions in parallel.



API: application programming interface  
DPDK: Intel® Data Plane Development Kit  
NFV: network functions virtualization

Fig. 2. System configuration for SDN software switch.

There usually is no mechanism for distributing the packet load in advance, so packets must be sent to a designated CPU core to be divvied up in software. This could cause packets to pile up on a single CPU core, creating a bottleneck, and in the worst-case scenario leading to packet loss. The high-speed flow dispatcher feature assigns packets to FPGA data structures (queues) based on the packet flow data received by the hardware, as shown in **Fig. 3**. Direct memory access (DMA) is then used to transfer packets to the main memory regions used by the CPU cores. In this way, the total processing load is distributed and smoothed out across all CPU cores without wasting software cycles.

#### (2) Flow director

In conjunction with the high-speed flow dispatcher feature, we need a way for software to freely specify which flows to assign along with their corresponding DMA queues. The flow director feature allows Lagopus to accomplish this by communicating with the

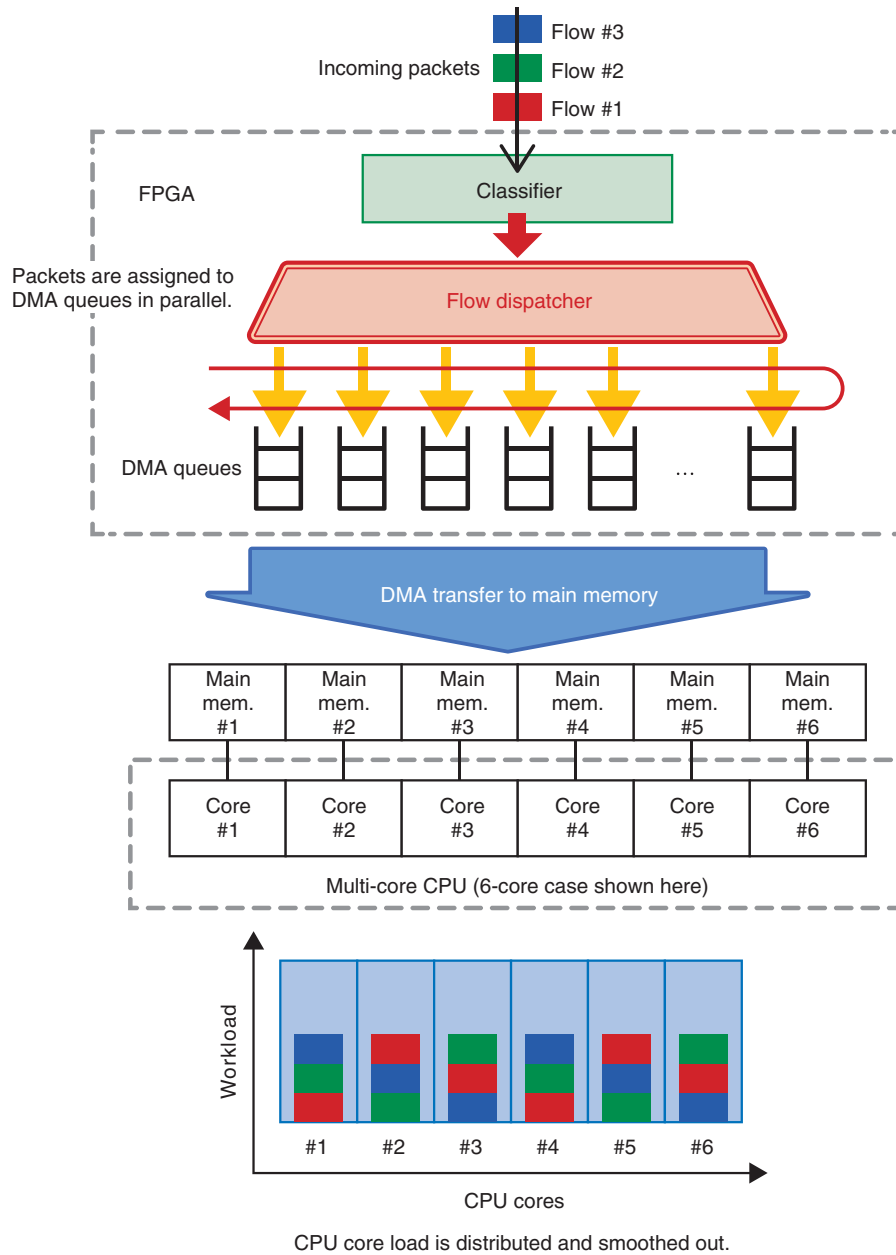


Fig. 3. High-speed flow dispatcher.

hardware through an application programming interface (API). The flow director feature can be used to improve performance. Lagopus can figure out how flows should be assigned to CPU cores, so flows do not need to be identified twice, and the corresponding processing can be skipped (**Fig. 4**). This gives Lagopus enough spare time to process short packets more efficiently.

### (3) Packet mirroring

We will need to make it easier to track down the cause of network problems in the future because we expect them to become more complex with advances in network virtualization. To that end, we have implemented packet mirroring in our HWA, enabling virtual networks to be monitored at all times without placing any burden on the software. This feature copies incoming packets within the FPGA and forwards

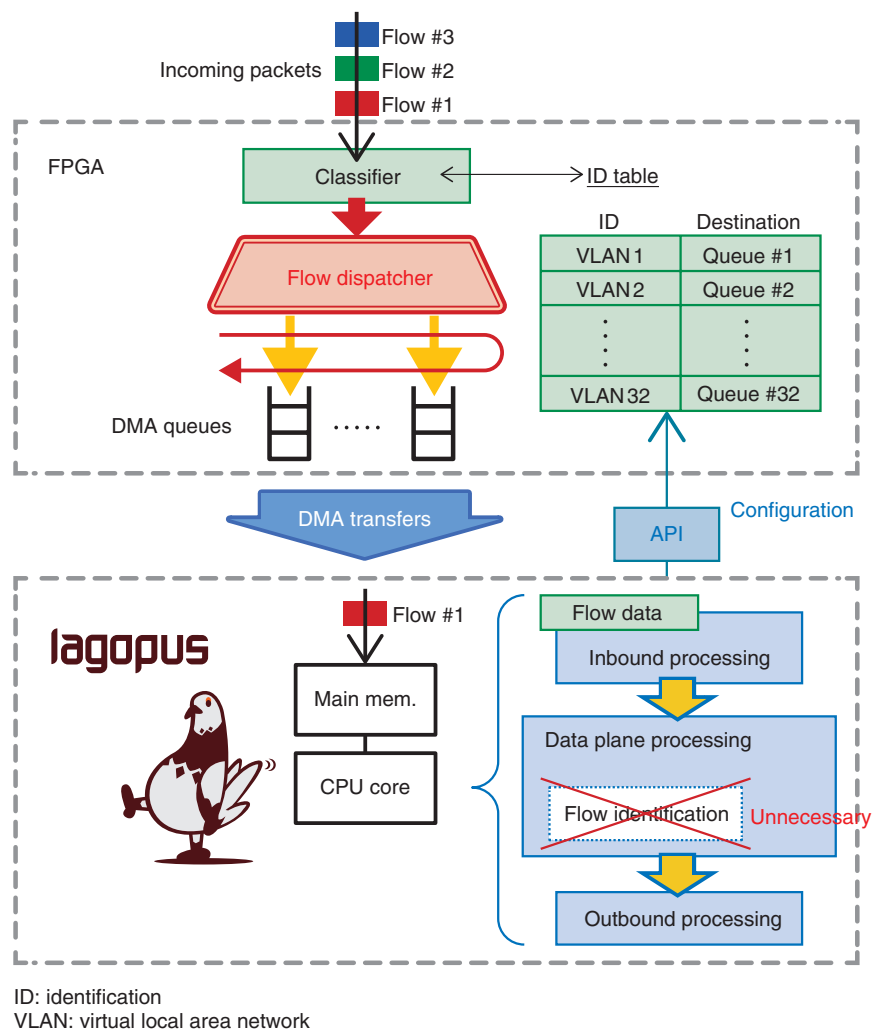


Fig. 4. Flow director.

them unmodified to an external device for analysis (Fig. 5). Problems can then be analyzed, and signs of possible problems can be detected without affecting normal packet processing.

#### 4. Trial and evaluation of systems with hardware acceleration

In collaboration with Xilinx Inc., we tested a prototype of our SDN software switch system equipped with an HWA that had all of the aforementioned features. We then checked the performance gains achieved through our hardware/software co-design. Our tests used the latest FPGA devices and tools (i.e., SDNet\*) capable of implementing cooperative hardware/software behavior. The results of evaluating the

packet-forwarding performance are plotted in Fig. 6(a), and the measured power consumption is in Fig. 6(b). The data show that short packets were forwarded more quickly when hardware acceleration was enabled. Furthermore, when using eight CPU cores (the maximum), our system was able to process packets longer than 384 bytes at a wire rate of 40 Gbit/s. We observed that the HWA only used 19 W—less than 5% of the 425 W used by an entire commodity server—while processing packets at 40 Gbit/s, confirming that we can achieve this wire rate with only a modest increase in power consumption. By using Lagopus to switch the HWA's packet copying feature

\* SDNet (Software Defined Specification Environment for Networking): Xilinx's design tool for packet processing.

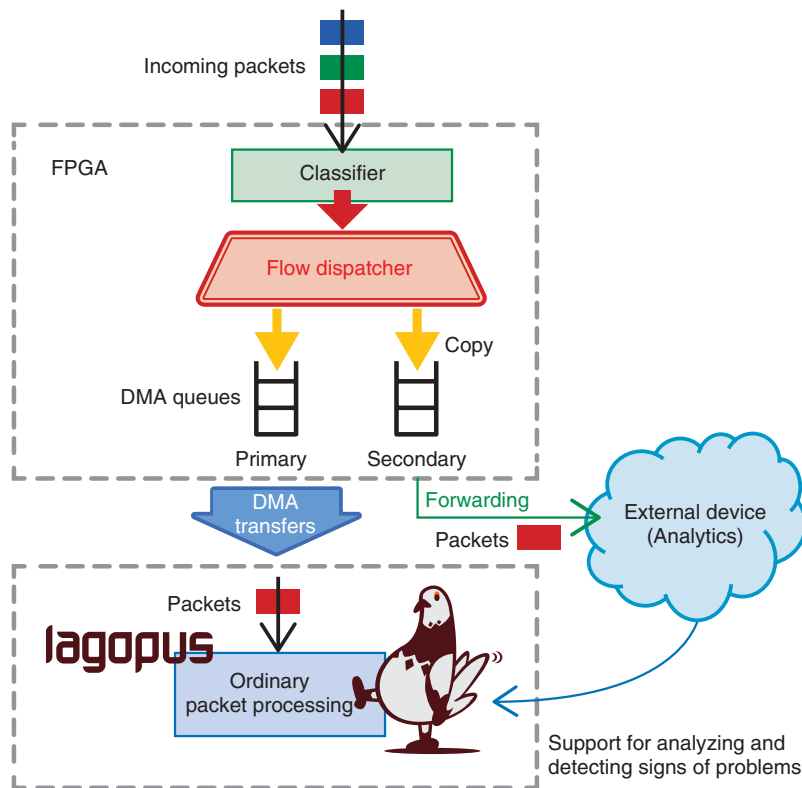
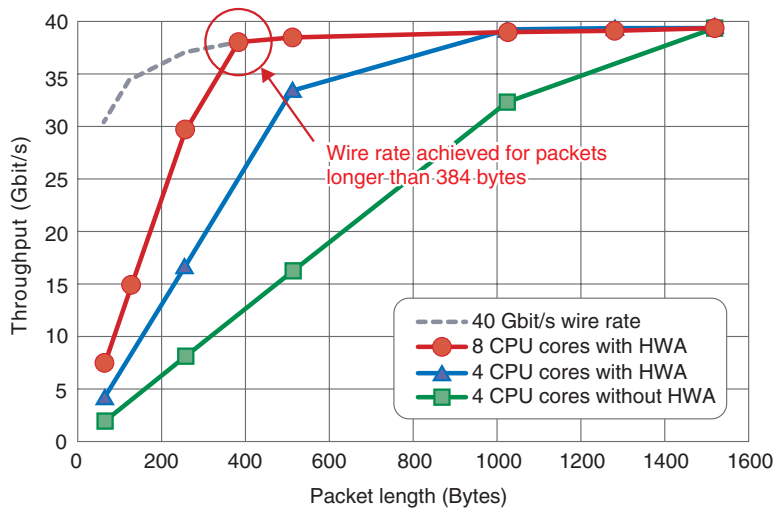
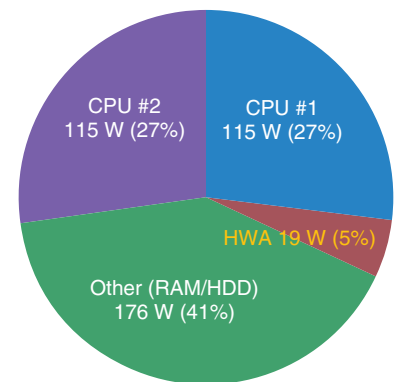


Fig. 5. Packet mirroring.



(a) Packet forwarding performance



(b) Power consumption at wire rate of 40 Gbit/s

Fig. 6. Results of prototype performance tests.

on and off via its API, we also confirmed that the feature did not affect primary packet processing in our prototype.

## 5. Future plans

In the future, we will work on improving efficiency even further by tackling the problem of accepting and processing packets shorter than 384 bytes at the full wire rate of 40 Gbit/s. To further improve reliability, we will experiment with the use of packet mirroring to implement network monitoring and analysis features and thus help improve the operational aspects of network virtualization. We also plan to actually implement network functions virtualization (NFV) applications with all of these features and develop a system for trial use.

In this way, we will continue to improve the performance and reliability of our network virtualization technology, establish hardware/software co-design as a fundamental technique for network virtualization, and commercialize our technology for practical use on carrier networks.

## References

- [1] Website of Ryu, <http://osrg.github.io/ryu/>
- [2] Website of Lagopus switch, <https://lagopus.github.io/>
- [3] Y. Nakajima, H. Takahashi, K. Ishiguro, and K. Yamazaki, "It's Kind of Fun to Do the Impossible with DPDK," DPDK Summit 2015, San Francisco, USA, 2015.
- [4] K. Yamazaki, Y. Nakajima, T. Hatano, and A. Miyazaki, "Lagopus FPGA - A Reprogrammable Data Plane for High-performance Software SDN Switches," Proc. of HOT CHIPS 27, pp. 10–19, Cupertino, USA, 2015.



**Takahiro Hatano**

Senior Research Engineer, Metro-Access Network Device Project, NTT Device Innovation Center.

He received a B.E. and M.E. in electronics, information and communication engineering from Waseda University, Tokyo, in 1990 and 1992. He joined NTT LSI Laboratories in 1992. In 2010, he moved to the Visual Communication Division of NTT Bizlink, where he developed and managed the video distribution services for live viewing on the NGN (Next Generation Network). He has worked on developing hardware/software co-design technology for future networks. He is a member of the Institute of Electronics, Information and Communication Engineers (IEICE).



**Akihiko Miyazaki**

Senior Research Engineer, Supervisor, Metro-Access Network Device Project, NTT Device Innovation Center.

He received a B.S. and M.S. in pure and applied sciences from the University of Tokyo in 1991 and 1993. In 1993, he joined NTT LSI Laboratories, where he initially engaged in research and development of layout CAD algorithms and systems. In 2000, he moved to NTT Communications and worked on developing application-platform services for enterprise customers. In 2004, he moved to NTT Microsystem Integration Laboratories and developed MAC (media access control) LSIs for access networks. He is a member of IEICE and the Physical Society of Japan.



**Koji Yamazaki**

Senior Research Engineer, Metro-Access Network Device Project, NTT Device Innovation Center.

He received a B.S. in environmental information science from Keio University in 2000. While in school, he founded a venture company with his colleagues and after operating that, he received a Master of Arts and Sciences (information studies) in interdisciplinary information studies from the University of Tokyo in 2004. Since joining NTT, he has been researching and developing large-scale integrated circuits (LSIs) and related computer aided design (CAD) systems. In 2009, he was engaged in developing a large-scale distributed data processing platform for cloud computing. His current interests include SDN/NFV and integration of high performance routers and switches.