

## PAPER

# Networking Experiment of Domain-Specific Networking Platform Based on Optically Interconnected Reconfigurable Communication Processors

Masaki MURAKAMI<sup>†a)</sup>, *Student Member*, Takashi KURIMOTO<sup>‡</sup>, *Member*, Satoru OKAMOTO<sup>‡</sup>, Naoaki YAMANAKA<sup>‡</sup>, *Fellows*, and Takayuki MURANAKA<sup>††</sup>, *Nonmember*

**SUMMARY** A domain-specific networking platform based on optically interconnected reconfigurable communication processors is proposed. Some application examples of the reconfigurable communication processor and networking experiment results are presented.

**key words:** domain specific, reconfigurable processor, in-network processing, in-network computing

## 1. Introduction

Domain-specific computing [1] is one approach to greatly improving computing server performance by specially designing computing server architecture for a particular application domain. To provide a domain-specific computing platform, reconfigurable interconnection among central processing units (CPUs), graphics processing units (GPUs), tensor processing units (TPUs), field-programmable gate arrays (FPGAs), and memories is most important to specify an architecture for the target application. Computing resources such as CPUs, GPUs, TPUs, and FPGAs and memories can be made into resource pools, users can build their special purpose computing architecture using resource pools with configuring the interconnection.

This concept can be applied as an analogy to the networking. Domain-specific networking can be defined as providing a specially designed network slice for a particular service network from networking resource pools. The networking resource pools are composed with application-specific integrated circuits (ASICs) for high-performance packet routing/switching, FPGAs for reconfigurable special purpose high-performance packet processing, network processing units (NPU) for high-performance functional packet processing, CPUs for flexible and complex functional packet processing, and electrical/optical switching fabrics for interconnecting among packet processing resources. All resources are programmable, reconfigurable, and can process data packets. We have proposed a recon-

figurable communication processor (RCP) as an integrated ASICs/FPGAs boards and NPUs/CPUs boards using interconnecting electric packet switching fabrics [2]–[4]. RCPs can be connected via optical networks and can provide networking resource pools to users. Users can design their special purpose network slices from the networking resource pools.

The RCP concept can be applicable not only the networking function but also the in-network processing function and in-network computing function. Our conventional works related to RCP [2]–[4] has proposed RCP architecture and resource management, but has not examined the feasibility of domain-specific networking platform. In this paper, we present the concept of RCP as a network node of domain-specific networking platform and application examples of RCPs for providing a smart and connected community (S&CC) [5]. Results of the networking experiment are also presented. Results of the networking experiment are also presented.

The rest of this paper is organized as follows. In 2 design concept of the reconfigurable processor is described. Prototype implementation of RCP is presented in 3. Experimental results are shown in 4. The conclusion is drawn in 5.

## 2. Reconfigurable Communication Processor Concept

### 2.1 Related Concepts (1) Photonic Network Processor

In order to cope with traffic growth and service diversity, a photonic network processor (PNP) has been proposed to construct a reconfigurable optical transport node [6]. Figure 1(a) shows a conceptual view of PNP. In PNP, digital signal processors (DSPs), ASICs, FPGAs, NPs, CPUs, and packet switches (SWs) are connected with a reconfigurable optical interconnect system. These devices will be implemented on one board in the future. Therefore, multiple PNPs can be connected with an optical interconnect system and construct a node system such as an optical cross-connect (OXC) and a reconfigurable optical add/drop multiplexer (ROADM). The important thing is that multiple PNPs provide DSP pools, NP/CPU pools, and SW pools at the board level, at the node level, and at the network level over the reconfigurable optical network (Fig. 1(b)).

Manuscript received August 10, 2022.

Manuscript revised November 16, 2022.

Manuscript publicized February 15, 2023.

<sup>†</sup>The authors are with Keio University, Yokohama-shi, 223-8522 Japan.

<sup>††</sup>The author is with Alaxala Networks, Kawasaki-shi, 212-00581 Japan.

a) E-mail: masa809m@keio.jp

DOI: 10.1587/transcom.2022EBP3131

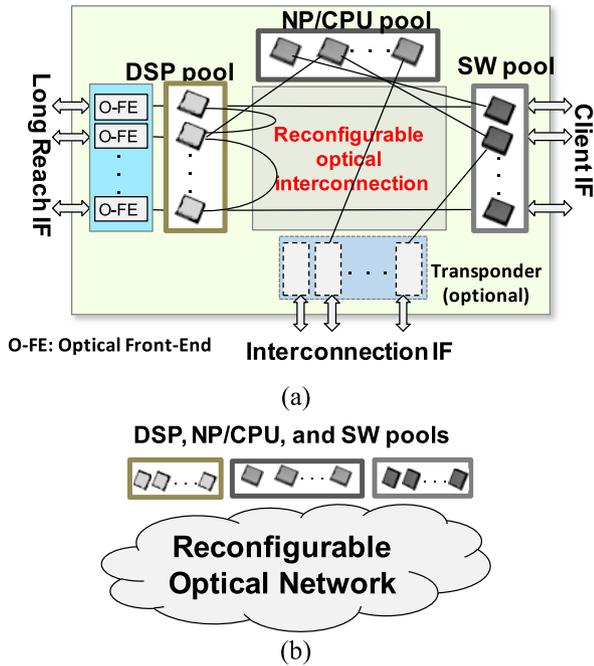


Fig. 1 The photonic network processor concept. (a) A board level PNP. (b) Resource pool over the reconfigurable optical network.

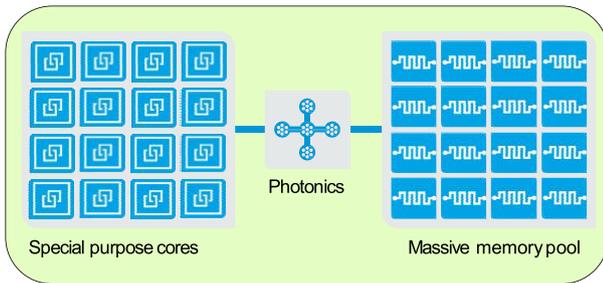


Fig. 2 The Machine architecture [9].

2.2 Related Concepts (2) Disaggregated Computing

In computing area, the concept of disaggregated computing [7], [8] has the intent to break the boundaries of current compute, memory, network and storage components built in a hard, unique and tightly connected unit. It brings benefits of high-performance computing, fine-grained technology upgrade cycles, fine-grained resource allocation, and access to a larger amount of memory [7].

One of the architectures for realizing disaggregated computing is “The Machine” proposed by Hewlett Packard. Figure 2 shows an architecture of The Machine [9]. The Machine consists of special purpose processor cores, a massive memory pool, and photonics. Cores and memory pools are interconnected by photonics. The special purpose processor cores will be constructed with CPUs, GPUs, TPUs, FPGAs, and so on. Therefore, The Machine can be applicable as a domain-specific computing platform.

The photonics in The Machine is not only limited in the

board and system level but extended to the wide area optical network. The transmission delay over the wide area network, i.e., latency, will provide a fatal impact on high performance computing. However, latency tolerant computing technologies [10], [11] will cover the networking latency.

2.3 Reconfigurable Communication Processor

A reconfigurable communication processor (RCP) is designed as a first step toward realizing the PNP concept. Disaggregated computing concept combines computing resources such as special purpose processor cores and a memory pool with photonics. On the other hand, an RCP concept that combines various transport processing functions based on ASICs, FPGAs, NPU, and CPU with switching devices. The design target of RCP is shown below.

- Support 400 Gbps class interface cards.
- Support a kind of IP router node, a kind of Ethernet switch node, and a kind of multi-protocol label switching (MPLS) node by reconfiguring the programmable data-plane.
- Support variety of network functions such as wire-rate flow counter, deep packet inspection, network slicing, tunneling (data packet encapsulation/decapsulation), etc. by reconfiguring the programmable data-plane path.
- Support in-network processing function by applying CPUs to not only the data-plane processing resource but also the computing resource.

These design targets require both high-speed processing and flexibility to the data-plane. The RCP has been designed with co-design of ASICs, FPGAs, NPs, and CPUs as in the right place.

We have developed two types of reconfigurable modules. They are a reconfigurable processing module (RPM) based on ASICs, FPGAs, and NPs, and a reconfigurable service module (RSM) based on CPUs and NPs. As shown in Fig. 3, a Tbps class switching module interconnects RPMs and RSMs. RCP consists of RPMs, RSMs, and switching module. RPM corresponding to over 100 Gbps multiple communication protocols. RSM providing various functions. Multiple service slices can be provided by combining RPMs and RSMs.

Figure 4 shows a conceptual diagram of RPM. FPGA-based flexible distributors are the key for providing scalable processing capacity. ASIC-based packet forwarding engines and flexible search engines provide data-plane packet routing/switching functions. NPU provides high-performance network functions. Figure 5 shows a concept diagram of RSM. NPU/CPU have GB class memories and storage for providing in-network processing and in-network computing. Especially, CPU with storage has capability of running a computer operating system (OS) e.g., Linux. Therefore, virtual machines (VMs) and containers can run in RSM. This means that RCP can be work as a network functions virtualization (NFV) platform.

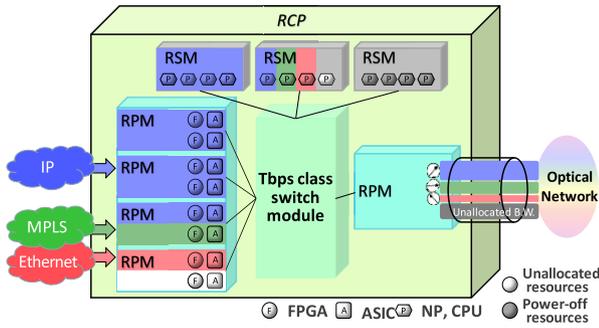


Fig. 3 The Reconfigurable Communication Processor architecture.

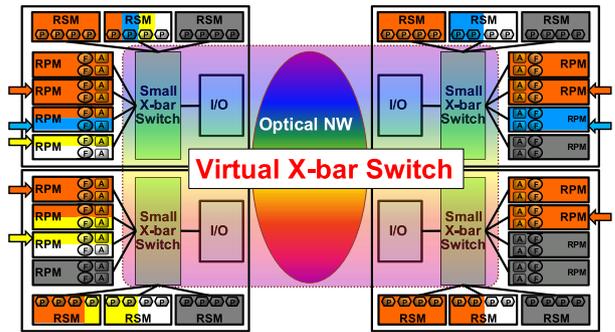


Fig. 6 The virtual cross-bar switch over the optical network.

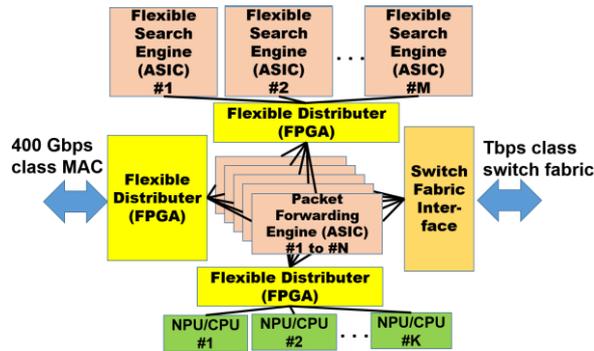


Fig. 4 The conceptual diagram of the RPM board.

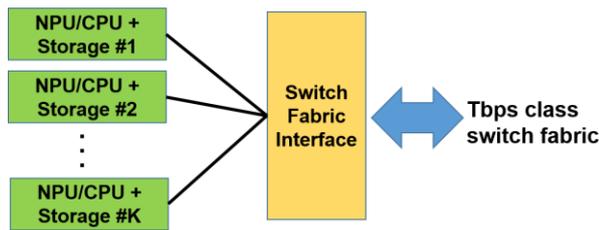


Fig. 5 The conceptual diagram of the RSM board.

The Tbps class switch module which is a crossbar (X-bar) type packet switch may restrict scalability of RCP. To break the restriction, a virtual X-bar switch over the optical network is applied. As shown in Fig. 3, RCP has interface to the optical network. Therefore, small X-bar switch can connect to other X-bar switches and make an equivalent large X-bar switch, i.e., the virtual X-bar switch. Figure 6 shows an example of the virtual X-bar. An RPM pool and a RSM pool are constructed over the optical network. RPMs and RSMs are selected from the pool and configured to the specific service. RCPs with optical networks are act as a domain-specific networking platform, an NFV platform, in-network processing platform, and in-network computing platform.

### 3. Prototype Implementation of RCP

#### 3.1 200 Gbps/400 Gbps Prototype RPM

As a first step, a 200 Gbps class RPM prototype is imple-

Table 1 FPGA specs of the flexible distributor.

#	FPGA spec	
	Item	Description
1	Vendor	Xilinx
2	Series	Virtex UltraScale+
3	Process	16 nm
4	Flip-Flops	2,364 K
5	Look up tables	1,182K
6	Total Block RAMs	75.9 Mb
7	Ultra RAMs	270 Mb
8	DSP slices	6,840

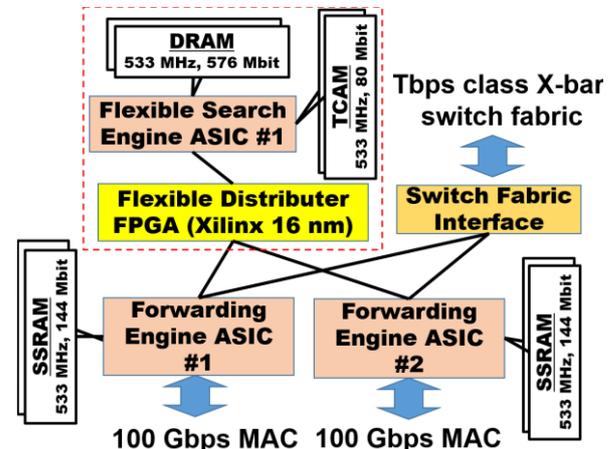


Fig. 7 The main diagram of the prototype RPM.

mented. Alaxala’s AX8600R system [12] is used as a base system. It has multiple 100 Gbps packet forwarding engine ASICs and a 3.2 Tbps X-bar packet switch fabric. Therefore, a flexible distributor FPGA and a flexible search engine ASIC are added to the base system as a daughter board. The applied FPGA spec is shown in Table 1.

Main RPM system diagram is shown in Fig. 7. The main functional modules are forwarding engines, flexible distributors, and flexible search engine. Forwarding engines forward packets incoming from the 100 Gbps MAC accord-

ing to the results of flexible search engines. Flexible search engines search the packet header to find the packet’s forwarding destination and the contents of the packet header update. Then, flexible search engines give instructions to forwarding engines according to the search results. Flexible distributors are located between forwarding engines and flexible search engine. Flexible distributors instruct flexible search engines about search keys and search methods. By using a reconfigurable FPGA, it is possible to provide the instructions for various protocols. Figure 8 shows a photograph of the developed 200 Gbps flexible distributor daughter board. The purpose of this prototype is to examine the 200 Gbps capacity of the flexible distributor FPGA, therefore two 100 Gbps MACs are directly connected to the packet forwarding engine ASICs.

This 200 Gbps class RMP prototype is extended to 400 Gbps by applying four forwarding engine ASICs.

### 3.2 CPU-Based RSM

Prototype RSMs for AXR8600R are developed in both NPU-based and CPU-based. The CPU-based RSM is attractive to develop in-network processing and computing application. Figure 9 shows a developed CPU-based RSM. 8 core Intel Xeon processor, 16 GB RAM, and 200 GB storage are implemented on the board. One 10 Gbps Ethernet (10GE) interface is implemented for connecting to the internal X-bar switch fabric and two GbE interfaces are implemented for connecting to outer servers or controllers. Linux (Ubuntu 14.04LTS and 16.04LTS) OS is successfully worked in the CPU-based RSM.

As an alternative to the Intel CPU-based RSM, MIPS (32 core, clock 1.2 GHz, and 32 GB RAM)-based RSM is also developed. DPI and IPsec hardware accelerators are also implemented in RSM. This RSM can be applicable to several stateful service function processing such as carrier

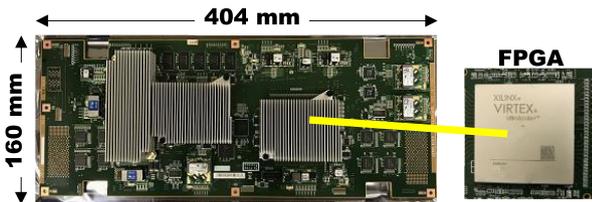


Fig. 8 Photograph of the developed 200 Gbps flexible distributor daughter board.

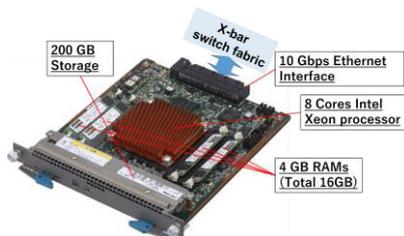


Fig. 9 Photograph of the developed Intel CPU-based RSM prototype.

grade network address translation (CGNAT), secure tunnels, and session border control (SBC).

### 3.3 Linux PC-Based RPM/RSM Emulators

Networking experiment requires many RPMs and RSMs. Therefore, a Linux server-based RPM and RSM emulators are developed. Packet processing functions of RPM are implemented as an emulator program in VM. Linux VMs are also worked as RSM. VMs in the server and Ethernet switch are worked as emulated RCPs. 10GE interfaces are used in the emulator.

## 4. Experimental Demonstrations

In this section, demonstration of 400 Gbps class reconfigurable data-plane processing with the 400 Gbps prototype RPM is described. Moreover, three application demonstrations and one networking demonstration are presented.

### 4.1 Demonstration of 400 Gbps Class Reconfigurable Data-Plane Processing

400 Gbps traffic rate was measured by connecting a 400 Gbps measuring instrument and an RCP which implemented a 400 Gbps RPM prototype with a 400 Gbps link. Figure 10 shows results of measuring maximum traffic rate and CFP8 (centum gigabit form-factor pluggable 8) optical waveform. This result indicates our prototype achieved 400 Gbps transmission with good waveform.

### 4.2 Application Demonstrations

The S&CC network proposal [5] uses optical interconnection to link objects, humans, and applications to a network that offers sophisticated processing functions. S&CC requires that application and processing functions be provided with sophisticated resource integration because each application has different requirements in terms of processing power, data amount, and response time. In other words,

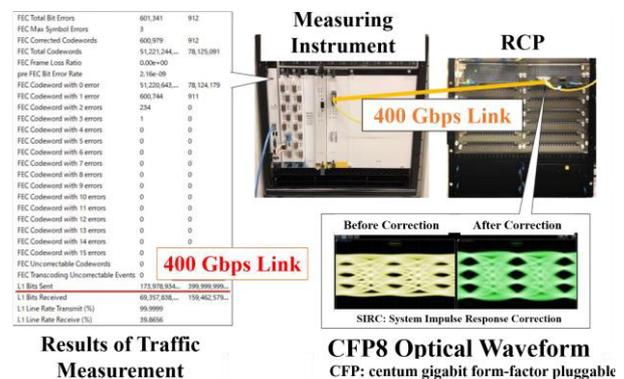


Fig. 10 Results of measuring traffic rate and monitoring optical waveform of a 400 Gbps RPM prototype.

the domain-specific networking platform is required to realize the S&CC network. A possible solution is to combine resources through optical interconnection dynamically. Therefore, the RCP concept can fit as one of the solution candidates. To meet S&CC requirements, we examine three types of applications. The first is wire-rate processing with service reconfiguration. The second is in-network processing/computing. The last is the resource pool feature.

4.2.1 Service Reconfiguration by RPMs

Figure 11 shows an experimental setup. Two RCPs were in NICT Tokyo, Japan and SC18 [13] NICT booth Dallas, USA, respectively. 100GE circuit is set between two locations by JGN [14] and other national research and educational networks (NRENs). Three services, IP, MPLS, and Ethernet over Ethernet (EoE), are set by configuring RPMs. In this experiment, 100 Gbps capacity RPMs are used.

Resource allocation of RPMs is designed to 12.5 Gbps capacity granularity. Total 100 Gbps capacity is allocated to a single RPM. In the experiment, at first, all resources are allocated to an IP service. Then reconfiguring to add an MPLS service and an EoE service sequentially. Experimental scenario is shown in Table 2.

Figure 12 shows a controller’s screen capture image during the step 4 and 5. Expected results are observed in the captured image. Service reconfiguration is successful, but it takes 6 s under the 150 ms round trip time (RTT) en-

vironment. Faster reconfiguration time is desirable for dynamic resource re-allocation according to service usage and network conditions. For example, generally, path protection which changes paths when links and nodes are failed requires 100 ms recovery time [15]. Therefore, if it is assumed that when a failure occurs, an alternative path is provided by reconfiguring another RCP, the reconfiguration time of the RCP should be less than 100 ms.

4.2.2 In-Network Processing and Computing by RSM

In-network processing and computing are an important feature of the S&CC network. S&CC requires that user application and processing functions be provided with collaborated each other. Because each user application has been supported by cloud and computers. In addition, low latency and trustworthy edge computing are needed for especially autonomous driving vehicle (ADV) control. RCP can work as an edge computing platform. For ADV, an agent program of ADV runs at an edge computer, i.e., RSM. The control delay between the ADV and the agent program of the ADV must be kept less than 10 ms to correctly control the ADV [5]. However, the delay exceeds 10 ms when the vehicle moves away from the edge server. One of the solutions is VM migration. The agent runs on a VM, and the VM is migrated to the appropriate edge server where the delay is less than 10 ms.

We constructed cloud and edge computing environment in SC18. Figure 13 shows an experimental setup of the ADV control network. Four RCPs are used in the network. RCP #4 has RSM shown in Fig. 9. Cloud servers are in Japan and ADV control agents in edge servers are located in USA. Linux (Ubuntu 16.04) servers and RSM (Ubuntu 16.04) are connected in the same virtual local area network (VLAN). The agent program for each ADV is installed in its own VM on edge servers including RSM. Several VMs are run in the servers and RSM. OpenStack is applied to VMs’ control. For realizing agent migration, VM live migration technique is applied. A VM where an agent runs is migrated to an appropriate edge server so that the control delay between the ADV and the VM is less than 10 ms. In the live migration, the correct ADV direction (the blue dots on the ADV simulator screen in Fig. 13) was indicated by the agent

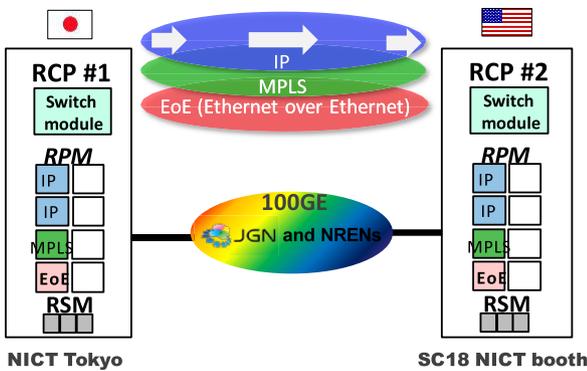


Fig. 11 Set up of the service reconfiguration experiment.

Table 2 Multi-protocols reconfiguration scenario.

Step	Resource Allocation (%)			Input Traffic (Gbps)			Expected Output Traffic (Gbps)		
	IP	MP LS	EoE	IP	MP LS	EoE	IP	MP LS	EoE
1	100	-	-	100	-	-	100	-	-
2	100	-	-	50	50	-	50	0	-
3	50	50	-	50	50	-	50	50	-
4	50	50	-	30	30	-	30	30	-
5	25	25	25	30	30	30	25	25	25

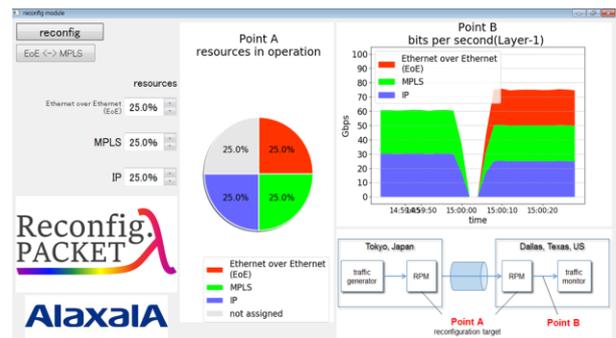


Fig. 12 Screen image of the controller.

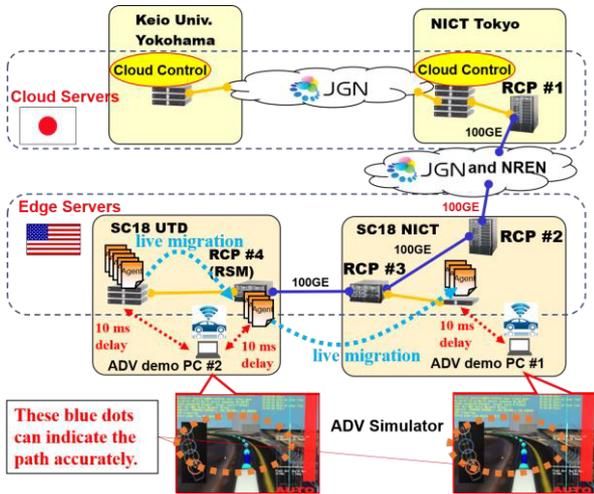


Fig. 13 ADV control network set up.

program.

#### 4.2.3 Resource Pool Feature

Network traffic flow visualization is important for network management. Therefore, realizing wire-rate flow monitoring function is expected. RCP can support this function. As a demonstration, we have implemented IP-based country flow statistics counter. About 240 nations are connected to the Internet, total assigned IPv4 address blocks to 240 nations are about 130,000 and IPv6 address blocks are 40,000. If a flow counter is assigned to each address block, 170,000 flow counters are required. In developed RPM, multiple address blocks can be specified to the matching condition, therefore required a number of counters can be reduced to 480 (240 for IPv4 and 240 for IPv6). In the implementation, because of the restriction of the FPGA resource, 64,000 flow conditions can be set in one RPM. Therefore, three RPMs are required to distinguish all 170,000 flows. In the experiment of SC18, three RCPs are used to demonstrate the network flow visualization. Figure 14 shows the demonstration network structure. A traffic generator generates simulated 100 Gbps 170,000 flows and sends them to RCP #2. RCP #2 makes mirrored traffic and sends it to RCP #3. RCP #3 makes mirrored traffic and sends it to RCP #4. Two counters are implemented to RPMs in RCP #3 and one counter is implemented to one RPM in RCP #4. RCP #3 and RCP #4 are connected by the optical network.

As a resource pool feature, total three RPMs are required to construct the country flow statistics counter, we can find three RPMs for the counter in the RCP network and RPMs are configured to the counter (and also configured to the traffic mirroring function). During the SC18, at first, two counters in RCP #3 are activated. In this case, not all flows are visualized (Fig. 15(a)). Next, the counter in RCP #4 is activated, then all flows are visualized (Fig. 15(b)).

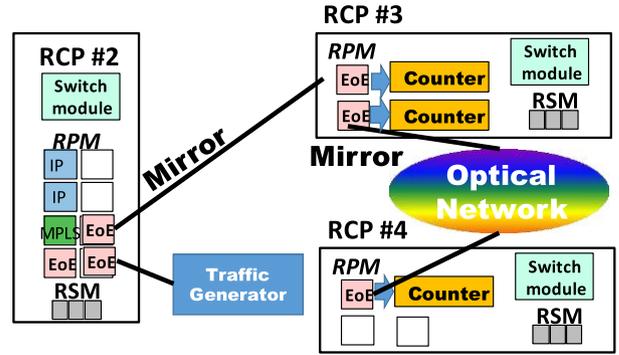


Fig. 14 Country flow statistics demonstration network.



(a)



(b)

Fig. 15 Visualized IP-based nation flow statistics. (a) Using RCP #3 only. Half of flows can be visualized. (b) Using both RCP #3 and #4. All flows can be visualized.

#### 4.3 Networking Demonstration

We construct four RCP nodes with a combination of real RCP prototypes and RCP emulators. A service path control

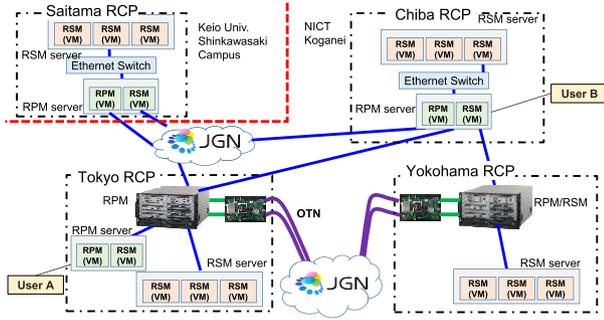


Fig. 16 Networking demonstration network.

server and a resource pool management server are also implemented in the demonstration network. Figure 16 shows the data plane design of the demonstration network. The network has four RCP nodes (Tokyo RCP, Yokohama RCP, Chiba RCP, and Saitama RCP). Tokyo RCP, Yokohama RCP, and Chiba RCP are placed on NICT Koganei site, and Saitama RCP is placed on Keio University Shinkawasaki site. Tokyo RCP node consists of servers (RPM emulator and RSM emulator) and a prototype RPM box. Yokohama RCP consists of a prototype RMP/RSM box and an RSM emulator. Chiba and Saitama RCPs consist of servers and an Ethernet switch. The Tokyo RCP-Yokohama RCP link is composed of two 100GEs and connected by JGN dark-fiber service between NICT Koganei and NICT Otemachi.

100GE signals are converted into optical transport network (OTN) signals and accommodated into JGN. JGN layer 2 service is used to construct links between Tokyo RCP-Saitama RCP and Chiba RCP-Saitama RCP. Gigabit Ethernet (GbE) and 10GE are used in RCP emulators and users.

The user requests a service path between user A and user B. The service path is a service function chain (SFC). Each function is provided by RPM or RSM. Therefore, the resource management server calculates an optimal SFC from the RPM/RSM resource pools. We have developed a quasi-optimal SFC placement algorithm [16], [17] and applied it in this demonstration. To realize the function chain from the resource pool, hardware module addressing is essential. The address format needs three identifiers [18]:

1. Geographic location-based discrimination. This identifier enables to select paths based on the delay constraint of service.
2. Service distinctiveness. It determines which service the hardware module is reserved for.
3. Identifiability of functions. Because RCPs have programmable hardware devices and provide any functions, it is necessary to identify the hardware module's function.

In this demonstration, we have implemented the 128 bit IPv6 address shown in Fig. 17 as a hardware module address. The Location ID field, Service ID field, and Function field satisfy the three identities described above. This addressing makes it possible to monitor, manage, and control the state of the

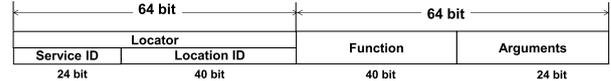


Fig. 17 Address format of a hardware module.

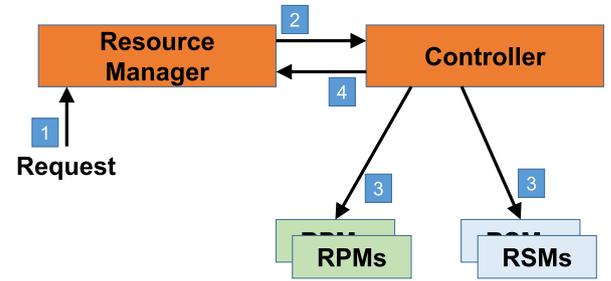


Fig. 18 Procedure of reconfiguring function chains.

nodes only in the address space. Segment routing (SR) is one of the good SFC implementation techniques, IPv6 SR (SRv6) [19]. The RPM and RSM emulators have an IPv6 address that follows the address format in Fig. 17. SRv6 is applied to configure function chains that connect the RPMs and RSMs in this demonstration by enabling IPv6 forwarding and SRv6 with Linux Kernel configuration of RPM and RSM VMs. RPMs and RSMs have SRv6 functions.

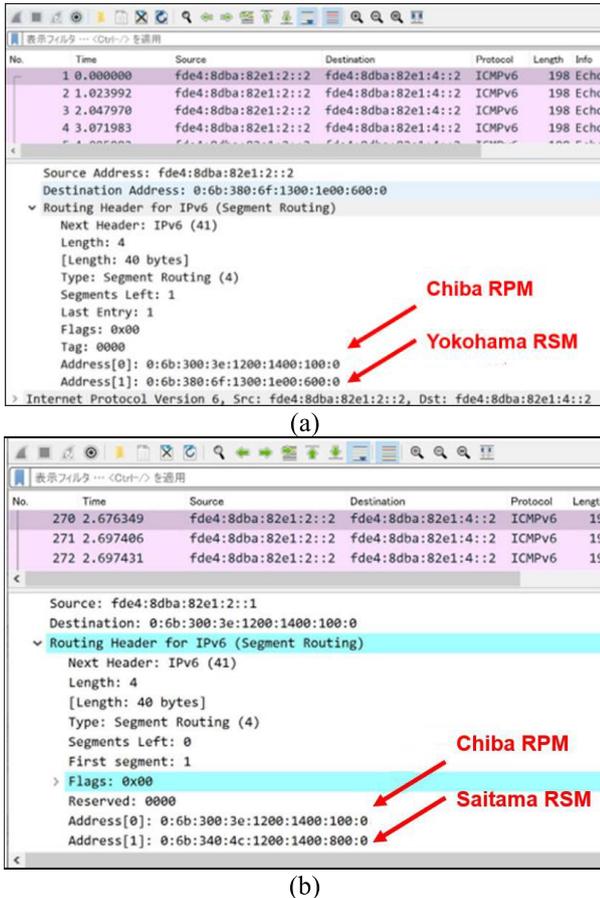
The RPM connected to the source host (user A) has a T.Encap function of SRv6, the other modules have an END function of SRv6, and the RPM connected to the destination host has an END.DX6 function of SRv6. SRv6 routing tables of RPMs are provided by the control server. In the demonstration, reconfiguration of the function chain when a new service request arrives is examined. Figure 18 shows a procedure of reconfiguring function chains. The detailed reconfiguration procedure is as follows.

1. When a new service request arrives, the resource manager calculates a hardware resource allocation problem using the SFC placement algorithm.
2. The resource manager sends the result to the controller.
3. The controller reconfigures SRv6 routing tables of RPMs based on the received result.
4. The controller updates resource information of hardware modules based on the received result.

We set several SFCs from User A to User B. Service request includes parameters of priority, required processing resources and functions, and required bandwidth resources.

The resource manager selects optimal RPMs, RSMs, and links between adjacent RCPs. Figure 19(a) shows captured packet analysis of initial setup of service#1 with parameters of low priority, 2 RPM and 1 RSM resources. The resource manager selected the route of User A-Tokyo RPM-Yokohama RSM-Chiba RPM-User B for the service#1.

Therefore, the SRv6 packet header of the SFC contains the address of Chiba RPM and Yokohama RSM. Next, we set another service with parameters of high priority and 3 RSM resources. As a result, the service#1 is af-



**Fig. 19** Wireshark capture of Tokyo RCP interface connected to User A. (a) Initial SFC setup. SRv6 packet contains the addresses of Chiba RPM and Yokohama RSM. (b) After SFC reconfiguration. SRv6 packet contains the addresses of Chiba RPM and Saitama RSM.

ected. RSM resources of Yokohama are evicted and SFC is rerouted to use Saitama RSM. Figure 19(b) shows captured packet analyses of rerouted service#1. The resource manager selected the recalculated route of User A–Tokyo RPM–Saitama RSM–Chiba RPM–User B. The SRv6 packet contains the addresses of Chiba RPM and Saitama RSM.

We have confirmed that the networking of RCPs is successfully demonstrated.

## 5. Conclusion

The reconfigurable communication processor provides 400 Gbps class reconfigurable data-plane processing and various kinds of in-network processing/computing functions. These functions provide the domain-specific networking platform feature to the RCP network. In this paper, application examples using 100 Gbps class RPMs and RSMs were presented. In the networking experiment, service-specific hardware resources are connected by function chaining. The constructed demonstration network indicated the feasibility of function chaining to create a domain-specific networking platform based on optically interconnected reconfigurable communication processors.

## Acknowledgments

This work is partly supported by the National Institute of Information and Communications Technology (NICT), Japan and JGN-A18002.

## References

- [1] J. Cong, V. Sarkar, G. Reinman, and A. Bui, “Customizable domain-specific computing,” *Trans. IEEE Des. Test Comput.*, vol.28, no.2, pp.6–15, March 2011.
- [2] S. Okamoto, J. Matsumoto, T. Sato, and N. Yamanaka, “Proposal of the photonic programmable node architecture using virtual reconfigurable communication processors,” *IEICE Technical Report*, PN2016-24, Sept. 2016 (in Japanese).
- [3] C. Hara, K. Yarita, S. Okamoto, and N. Yamanaka, “Biological attractor selection and SDN control interworking in the virtual packet optical node network control,” *Proc. 2019 Optical Fiber Communication Conference (OFC2019)*, M3Z.12, March 2019.
- [4] C. Hara, M. Murakami, S. Okamoto, and N. Yamanaka, “Experimental deployment of dynamic resource allocation using biological attractor selection in virtual packet optical node,” *24th Optoelectronics and Communications Conference/International Conference on Photonics in Switching and Computing 2019 (OECC/PSC 2019)*, T3-4, July 2019.
- [5] N. Yamanaka, S. Okamoto, M. Hirono, Y. Imakiire, W. Muro, T. Sato, E. Oki, A. Fumagalli, and M. Veeraraghavan, “Application-triggered automatic distributed cloud/network resource coordination by optically networked inter/intra data-center,” *IEEE/OSA J. Opt. Commun. Netw.*, vol.10, no.7, pp.B15–B24, July 2018.
- [6] K. Kitayama, A. Hiramatsu, M. Fukui, T. Tsuritani, N. Yamanaka, S. Okamoto, M. Jinno, and M. Koga, “Photonic network vision 2020—Toward smart photonic cloud,” *IEEE J. Lightw. Technol.*, vol.32, no.16, pp.2760–2770, Aug. 2014.
- [7] M. Hugo, S. Jose, Q. Josué, Z. Ferad, R. Damian, and N. Mario, “Disaggregated computing. An evaluation of current trends for data-centres,” *Procedia Computer Science*, vol.108, pp.685–694, 2017.
- [8] K. Katrinis, D. Syrivelis, D. Pnevmatikatos, G. Zervas, D. Theodoropoulos, I. Koutsopoulos, K. Hasharoni, D. Raho, C. Pinto, F. Espina, S. Lopez-Buedo, Q. Chen, M. Nemirovsky, D. Roca, H. Klos, and T. Berends, “Rack-scale disaggregated cloud data centers: The dReDBox project vision,” *2016 Design, Automation & Test in Europe Conference & Exhibition*, pp.690–695, 2016.
- [9] K. Keeton, “The machine: An architecture for memory-centric computing,” *Proc. Workshop on Runtime and Operating Systems for Supercomputers (ROSS)*, Portland, Oregon, USA, June 2015.
- [10] Y. Shan, Y. Huang, Y. Chen, and Y. Zhang, “LegoOS: A disseminated, distributed OS for hardware resource disaggregation,” *13th USENIX symposium on Operating Systems Design and Implementation (OSDI’18)*, pp.69–87, Oct. 2018.
- [11] X. Guo, F. Yan, X. Xue, G. Exarchakos, and N. Calabretta, “Performance assessment of a novel rack-scale disaggregated data center with fast optical switch,” *Proc. 2019 Optical Fiber Communication Conference (OFC2019)*, M2C.2, March 2019.
- [12] Alaxala Networks, “AX8600R series,” <https://www.alaxala.com/en/products/AX8600R/index.html> (online available Dec. 8, 2019).
- [13] The International Conference for High Performance Computing, Networking, Storage, and Analysis 2018, <https://sc18.supercomputing.org/>
- [14] NICT, “JGN high speed R&D network testbed,” <https://testbed.nict.go.jp/jgn/english/index.html> (online available Dec. 8, 2019).
- [15] K. Hong, Y. Kyung, T.M. Nguyen, S. Park, and J. Park, “A provisioning scheme for guaranteeing recovery time in WDM mesh networks,” *2015 Seventh International Conference on Ubiquitous and Future Networks*, pp.585–587, 2015.

- [16] C. Hara, K. Yarita, S. Okamoto, and N. Yamanaka, "Biological attractor selection and SDN control interworking in the virtual packet optical node network control," 2019 Optical Fiber Communication Conference (OFC), M3Z.12, March 2019.
- [17] C. Hara, M. Murakami, S. Okamoto, and N. Yamanaka, "Experimental deployment of dynamic resource allocation using biological attractor selection in virtual packet optical node," 24th OptoElectronics and Communications Conference/International Conference on Photonics in Switching and Computing 2019 (OECC/PSC 2019), T3-4, July 2019.
- [18] N. Sumita, M. Murakami, T. Kurimoto, S. Okamoto, and N. Yamanaka, "Proposal of the resource addressing for virtual reconfigurable communication processors," IEICE General Conference, B-12-12, March 2020 (in Japanese).
- [19] RFC8986 "Segment routing over IPv6 (SRv6) network programming," Feb. 2021.



**Masaki Murakami** received the B.E. and M.E. degrees in engineering from Keio University, Japan, in 2018 and 2019, respectively. He is currently working toward the Ph.D. degree in Graduate School of Science and Technology, Keio University, Japan. His research interests include optical data-center interconnection network system and optical parallel networking technologies for beyond 5G. He is a student member of the IEICE.



**Takashi Kurimoto** currently is an associate professor, National Institute of Informatics, Japan and he is involved in the design and implementation of national research and educational network, called Science Information Network (SINET). He has been engaged in researching the switching technology for high-speed computer networks and deployment of the next generation network.



**Satoru Okamoto** is a Project Professor of the Keio University, Kanagawa, Japan. He received his B.E., M.E. and Ph.D. degrees in electronics engineering from Hokkaido University, Hokkaido, Japan, in 1986, 1988 and 1994. In 1988, he joined Nippon Telegraph and Telephone Corporation (NTT), Japan. He is now researching future IP + optical network technologies and their applications over photonic network technologies. He has published over 90 peer-reviewed journal and transaction articles, written over 200 international conference papers, and been awarded 50 patents, including 5 international patents. He received the Young Researchers Award and the Achievement Award in 1995 and 2000, respectively, from IEICE. He also received the IEICE/IEEE HPSR2002 Outstanding Paper Award, the IEICE Communications Society Best Paper Award in 2011 and 2019, the IEEE ISAS2011 Best Paper Award in 2011, and the IEICE ICETC2020 Best Short Paper Award in 2020. He was an associate editor of the IEICE Transactions on Communications (2006–2011) as well as the chair of the IEICE Technical Committee on Photonic Network (PN) (2010–2011), and was an associate editor of Optics Express of The Optical Society (OSA) (2006–2012). He is an IEEE senior member.



**Naoaki Yamanaka** received the B.E., M.E., and Ph.D. degrees in engineering from Keio University, Yokohama, Japan, in 1981, 1983, and 1991, respectively. In 1983, he joined the Nippon Telegraph and Telephone Corporation's (NTT's) Communication Switching Laboratories in Tokyo Japan. He is now researching future optical IP networks and optical MPLS router systems. He is currently a Professor in the Department of Information and Computer Science at Keio University, a Chair of Keio Leading-edge Laboratory of Science and Technology, and a Chair of Photonic Internet Labs. He has published over 130 peer-reviewed journal and transaction articles, written over 220 international conference papers, and been awarded 174 patents, including 17 international patents. He received Best of Conference Awards from the 40th, 44th, and 48th IEEE Electronic Components and Technology Conferences in 1990, 1994, and 1998; the TELECOM System Technology Prize from the Telecommunications Advancement Foundation in 1994; the IEEE CPMT Transactions Part B: Best Transactions Paper Award in 1996; the IEICE Transaction Paper Award in 1999; the IEEE ISAS2011 Best Paper Award in 2011; the IEICE Achievement Award in 2015; and the IEICE ICETC2020 Best Short Paper Award in 2020. He is the technical editor of IEEE Communication Magazine, the broadband network area editor of IEEE Communication Surveys, a former editor of IEICE Transactions, the TAC Chair of the Asia Pacific Board at the IEEE Communications Society, and a board member of the IEEE CPMT Society. He is an IEEE Fellow.



**Takayuki Muranaka** received the B.E. and M.E. degrees in engineering from Tokyo University, Tokyo, Japan, in 1997 and 1999, respectively. In 1999, he joined Hitachi Ltd. in Kanagawa, Japan. He is now developing router and switch products. He is currently a Senior Vice President of Products Development Division at ALAXALA networks Corp. He received IEICE ICETC Best Short Paper Award in 2020.