Data Prediction for Response Flows in Packet Processing Cache

Hayato Yamaki¹, Hiroaki Nishi², Shinobu Miwa¹, Hiroki Honda¹

¹Department of Computer and Network Engineering, The University of Electro-Communications, Chofu, Japan

²Department of System Design Engineering, Keio University, Yokohama, Japan

Email: {yamaki, miwa, h.honda}@uec.ac.jp, west@sd.keio.ac.jp

ABSTRACT

We propose a technique to reduce compulsory misses of packet processing cache (PPC), which largely affects both throughput and energy of core routers. Rather than prefetching data, our technique called response prediction cache (RPC) speculatively stores predicted data into PPC without additional access to the low-throughput and power-consuming memory (i.e., TCAM). RPC predicts the data related to a response flow at the arrival of the corresponding request flow, based on the request-response model of internet communications. RPC can improve the cache miss rate, throughput, and energy-efficiency of PPC systems by 15.3%, 17.9%, and 17.8%, respectively.

CCS CONCEPTS

• **Networks** → **Routers** • **Hardware** → Power estimation and optimization; Logic synthesis

KEYWORDS

Packet Processing; Network Cache; Data Prediction

1 INTRODUCTION

The demand for high-throughput and low-power core routers has increased with the rapidly growing internet traffic. The Ministry of Internal Affairs and Communications, Japan, reports that Japanese network equipment will face 190x internet traffic of 2006 and result in 10x power consumption in 2025 [1,2]. This trend in internet traffic growth is not limited to Japan [3-5], and it drives network engineers to develop a new technique of high-throughput and low-power packet processing in core routers.

A key of efficient packet processing is intelligent table lookup. Packet processing requires search operations on several tables such as the routing table, which is usually configured with a ternary content addressable memory (TCAM). TCAM has the capability to quickly search the stored data; however, it does not

Figure 1: Breakdown of cache misses in 32KB and 64KB 4-way PPCs for the real network trace obtained from the Trans-Pacific 1-Gbps link between Japan and the United States in April 02, 2009 [16].

achieve the throughput required for next-generation networks such as a 400-Gbps network. To make matters worse, TCAM consumes a remarkable amount of energy (16x SRAM energy [6]) that accounts for about 40% of the overall router power [7,8]. Considerable effort has been invested to improve the throughput or power-efficiency of TCAM [7,9]; however, no technique has achieved both the 400-Gbps and SRAM-like power, which are required for future core routers.

A promising approach to achieve this goal is to reduce the number of TCAM accesses. Packet processing cache (PPC), which is a high-throughput and low-power memory that retains the results of TCAM lookups, can be used for this purpose [10-15]. Internet traffic has the temporal locality (i.e., many packets with a similar header arrive at a core router in a short period), so that many real packets frequently search the same data from TCAM [10]. This fact means that there are many opportunities to reuse the results of TCAM lookups, and the core routers that employ PPC can therefore reduce the number of TCAM accesses by searching PPC instead of TCAM.

Previous studies have reported that PPC can successfully reduce TCAM access count; however, about 20% of packets still need to access TCAM even if a core router employs the state-of-the-art PPC [11]. Fig. 1 shows the breakdown of cache misses in two PPCs with different configurations (32KB and 64KB 4-way PPC). The figure represents that many compulsory misses (about half of the overall PPC misses) restrict the PPC performance.

In this paper, we present the first technique to reduce the compulsory misses of PPC. Rather than prefetching data from TCAM, our technique called response prediction cache (RPC) speculatively stores data into PPC. RPC predicts the data related to a response flow at the arrival of the corresponding request flow, based on the request-response model of internet communications. Since RPC requires no additional access to TCAM at the prediction, it can remove TCAM accesses if a packet hits the predicted data within PPC. We also propose a

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org.

DAC '18, June 24–29, 2018, San Francisco, CA, USA

© 2018 Association for Computing Machinery.

ACM ISBN 978-1-4503-5700-5/18/06...\$15.00

<https://doi.org/10.1145/3195970.3196021>

technique called adaptive RPC (A-RPC), which selectively uses RPC for the further improvement in the PPC hit rate.

The main contributions of this paper are listed as follows.

- We develop a new technique to reduce compulsory misses in PPC, which have not been reduced by the previous work.
- We build a simulation environment of our RPC, and by using this environment, we show that RPC can reduce PPC misses by up to 64.7% for real network traces.
- We show that our technique is implementable with ASIC. We also show that A-RPC has an area overhead of 5.8%, an increase of 17.9% in throughput, and an energy saving of 17.8%, compared to conventional PPC.

The rest of this paper is organized as follows. We first give a more detailed description of PPC in Section 2. Section 3 introduces our key concept, data prediction for response flows, and Section 4 describes RPC and A-RPC in detail. We present our experimental results in Section 5, and finally conclude this paper in Section 6.

2 PACKET PROCESSING CACHE

We first provide an overview of the PPC architecture and introduce the existing PPC techniques that reduce PPC misses.

2.1 Architecture

A packet that arrives in a core router needs to search the required data from some tables (e.g., a routing table, an address resolution protocol (ARP) table, an access control list (ACL), and a quality of service (QoS) table) configured with TCAM. One or several fields of the packet header are used for this search operation. More specifically, the five-tuple of the header (i.e., source and destination IP addresses, source and destination port numbers, and protocol number), which is defined as a flow, is often used. Therefore, TCAM produces the same output for multiple packets if they have an identical flow.

PPC utilizes the above nature of packet processing to reduce the number of TCAM accesses. Fig. 2 shows the overview of PPC. A typical PPC is accessed with a hash value calculated from the five-tuple. A cache tag consists of the 104-bit flow information (i.e., the five-tuple). A data field contains the results of multiple table lookups. The typical size of the data field is 15 bytes, which includes a 1-byte output interface number in the routing table, a 12-byte destination MAC address in the ARP table, a 1-byte filtering result in ACL, and a 1-byte priority number in the QoS table. Furthermore, PPC can store not only the results of table lookups but also the results of encapsulation, encryption, and packet inspection executed by network intrusion detection systems. PPC can substitute multiple operations executed in a core router for one cache access.

The throughput and energy of core routers with PPC highly depend on the PPC hit rate, because PPC, which is configured with a small SRAM, can serve a larger number of data with lower energy than TCAM. Many efforts to reduce PPC misses have been carried out; however, the miss rate of the state-of-the-art PPC is still high (about 20%) [11].

Figure 2: Packet processing flow with PPC.

2.2 Related Work

One approach to improving PPC is to reduce capacity misses. Some tag-compression techniques have been used to increase the number of cache entries under a given area budget. Chang et al. propose the digest cache, in which the five-tuple is compressed to a 32-bit hash value used as a cache tag [12]. One problem with the digest cache is the hardware cost required to prevent hash conflicts. Ata et al. propose a tuned cache that uses the limited flow information (source/destination IP addresses and lower port numbers) as a cache tag [13], but the tuned cache is unable to conduct fine-grained packet processing such as firewall, because of the lack of the entire flow information.

To reduce conflict misses, some intelligent replacement algorithms have been proposed [11,14,15]. Yamaki et al. proposed a technique to filter flows composed of a single packet (e.g., domain name system (DNS) and network attack packets) from PPC, as an alternative [17]. These methods successfully reduced the number of cache misses in PPC; however, PPC still shows the cache miss rate of 20% [11].

3 DATA PREDICTION FOR RESPONSE FLOWS

Many internet applications communicate between servers and clients based on a request-response model; a client sends a request to a server and the server then replies to the client. This feature involves two types of compulsory misses in PPC. One is a miss caused by a request packet that is first sent from a client to a server, and the other is a miss caused by a response packet that is first sent from the server to the client.

Meanwhile, the five-tuple of a response packet can be predicted from the corresponding request packet. The source and destination IP addresses of the response packet can be computed by swapping those of the request packet. Likewise, the source and destination port numbers of the response packet can be computed with those of the request packet. Moreover, the protocol numbers of the two packets are always the same value. The results of the table lookups by the response packet can also be predicted from the request packet. The result of the routing table lookup by the response packet depends on which interface the request packet comes from. The result of the ARP table

Figure 3: Percentage of symmetrically-routed flows in nine types of networks.

lookup by the response packet is equivalent to the source MAC address of the request packet. Furthermore, the results of the ACL and QoS table lookups by the response packet are usually the same as those by the request packet. A few irregular situations may occur, but they can be detected by adding an asymmetry flag to each table entry. The asymmetry flag is set to an entry at the first write if the table contents of the request and response are asymmetrical.

These insights bring a new opportunity in optimizing PPC; a core router speculatively stores the data for an upcoming first response packet into PPC at the arrival of the corresponding first request packet. The key idea is that the most complete data for the response packet can be created from the request packet without additional TCAM accesses. Therefore, the TCAM lookups can be removed from response-packet processing if the response packet arrives timely at the router.

Since the effectiveness of our approach depends on how much response flows are routed to the same router as corresponding request flows, we first investigated the percentage of symmetrically-routed flows in various network traffic. Fig. 3 shows the traffic breakdown for nine network traces listed in Table 2. Note that *others* shown in Table 2 includes asymmetrically-routed flows and request flows whose corresponding response flow is not returned, such as attack flows. The figure shows that 72% of all flows are symmetrically routed in real networks. Thus, prediction for response flows has a significant potential in reducing compulsory misses in PPC (36% of the overall compulsory misses).

4 RESPONSE PREDICTION CACHE

We first present RPC, an architecture for predicting the PPC data for response flows, and then introduce an adaptive approach of RPC for further improvement. Finally, we discuss the negative impact of RPC.

4.1 Architecture

Fig. 4 shows the overview of RPC, assuming a simple router that has two line cards (LC1 and LC2). This example represents the situation that a request packet sent to LC2 arrives at LC1. PPC, an RPC module, and a hit/miss checker are implemented at each line card. The hit/miss checker detects cache misses in the local PPC, while the RPC module both predicts and writes PPC data. The RPC module consists of two main components: an aligner

Figure 4: RPC overview.

Figure 5: Implementation of RPC. The gray components are additionally required when compared to conventional PPC.

and a queue. The aligner creates the tag and data fields for a first response packet, which wait to be written to PPC in the queue.

The data prediction for the first response packet using RPC is conducted as follows: (a) First, the hit/miss checker in LC1 identifies a first request packet by checking whether an arrived packet misses PPC. The checker sends the results of the table lookups, the packet header and a PPC miss flag to LC2, where the corresponding response packet will arrive. (b) Second, the aligner in LC2 computes the tag and data fields for the corresponding response packet using the information received from LC1 if the received cache miss flag is enabled. The computed data is stored to the queue in LC2. (c) Finally, the data is written to PPC in order when a write port of PPC is free.

RPC has a time lag between receiving the first request packet and writing the predicted data to PPC; however, this time lag is relatively small when compared to the typical server response time. Therefore, the first response packet can hit PPC if it arrives at the router.

Fig. 5 shows the detailed implementation of RPC. A packet arrived in a line card first accesses PPC with a hash value calculated from the five-tuple. If a cache hit occurs, the packet is dealt with the data retrieved from PPC. However, if a cache miss occurs, the packet is forwarded to the cache miss handler (CMH). CMH, which is composed of the cache miss table (CMT) and the cache miss queue (CMQ), handles outstanding misses as well as miss status holding registers (MSHRs). CMT manages flows being retrieved in the TCAM for preventing cache misses caused by subsequent packets that belong to the flows managed in CMT, and the subsequent packets are sent to CMQ. Packets in CMQ are processed after the processing of the precedent packet in the TCAM is completed. The more detailed information about CMH

Figure 6: Training conducted by A-RPC in case that PPC has 256 entries.

is described in [18]. If a CMT miss occurs, the packet is stored to a buffer and then forwarded to a TCAM module, and finally, the cache is updated with the result of the packet processing in the TCAM module.

4.2 Adaptive Approach

RPC can be applied to both communication directions in a core router (i.e., from LC1 to LC2 and from LC2 to LC1 in Fig. 4), but the effectiveness depends on how much downstream traffic flows are within a direction. Conventionally, since downstream includes a larger number of response flows than upstream, applying RPC to downstream allows more opportunities for reducing compulsory misses caused by response packets. Meanwhile, RPC may increase PPC misses for upstream because the data created by the packets that cause conflict and capacity misses pollute PPC.

We propose adaptive RPC (A-RPC) to select the use of RPC in each direction within a core router. Following the normal boot process of the router, each RPC modules do simple training in a short period called a judgment phase, and then enables or disables RPC for the direction based on the training result. The router configuration determined in a judgment phase does not change under operation because many servers and clients keep their topological location for a long time.

Fig. 5 also shows the extension of RPC to A-RPC. PPC is escorted by A-RPC modules before and behind. The A-RPC modules are activated only during the judgment phases and usually turned off for the power-saving purposes.

Fig. 6 shows the processing flow of the training when PPC has 256 entries. In the judgment phase, PPC is split into two areas and RPC is applied to only one area. A hash value is also divided by two in order to access two areas. A packet that arrives at a line card accesses both areas to simulate the impact of RPC on PPC misses. More specifically, A-RPC counts the number of PPC misses in each area using two miss counters. A-RPC will decide to use RPC if the miss counter in the area that uses RPC shows a smaller value than that in the other area.

4.3 Discussion

When network attacks such as a port scan attack or a denial of service (DoS) attack occur, RPC increases PPC misses because it updates PPC with the data for response packets never arrived (i.e., servers do not respond to attack packets). However, this problem can be mitigated by attack-aware cache [17]. Attack-aware cache identifies attack flows, monitoring specific port numbers and the number of flows generated by each source IP

Table 1: Parameters of PPC Systems.

	Value
Cache	28 bytes $\times$ 1,024 entries, 4-way LRU, 0.5 ns/access
TCAM	5 ns/access
CMT	13 bytes $\times$ 16 entries
CMQ	13 bytes $\times$ 16 entries $\times$ 16 queues
Buffer	13 bytes $\times$ 64 entries

address. RPC can prevent the insertion of useless data into PPC by using attack-aware cache before data prediction.

Some packets in real networks are routed by asymmetric routing that delivers request and response on different paths. Asymmetric routing causes two problems in RPC: routing response packets to wrong directions at a branch router that asymmetrically routes packets, and creating unused PPC entries at the routers present between the branch router and a server. These problems can be alleviated by setting an asymmetry flag to each entry in PPC. If a response flow hits PPC and the asymmetry flag in the retrieved entry is set, RPC discards the routing information denoted by the entry and then searches the routing table. Asymmetry flags can be automatically set in branch routers, while they can be set in the other routers by receiving asymmetric signals from branch routers. Our RPC works well under asymmetric routing; however, we do not consider asymmetric routing in this paper because the packets routed by asymmetric routing are small, as shown in Fig. 3.

5 EVALUATION

We first describe our experimental methodology, followed by our experimental results.

5.1 Experimental Methodology

We evaluated our technique using various metrics: PPC miss rate, accuracy of A-RPC, area cost, throughput, and energy. The RTL simulator bundled in Synopsys Design Compiler is used for the experiment in area cost, while our in-house PPC simulator is used for the other metrics. The simulator feeds the packets from a network trace at the appropriate cycle designated by timestamps, and it emulates the cycle-level behavior of PPC such as reading packets, extracting flow information, creating the cache index, and referring to the cache and TCAM. We implemented both RPC and A-RPC on this simulator.

The main parameters of PPC systems are summarized in Table 1. The cache was constructed as an L1-sized cache. We confirmed that the 64 entries buffer present in front of TCAM (Fig. 5) can process the packets without packet loss.

We used various types of real-network traces shown in Table 2 as workloads. The traces were mainly obtained from RIPE Network Coordination Centre [19]. The Trans-Pacific 1-Gbps link [12] and Core 10Gbps-link traces were collected in Japan.

5.2 Experimental Result

5.2.1 PPC Miss Rate. The average miss rates of the conventional PPC, RPC, and A-RPC are summarized in Table 3. The table

Table 2: Network Traces.

	Core	UPCB	UFL	MRA	FRG	WIDE	PSC	BWY	TXG	ODU	MEM
		[19]	[19]	[19]	[19]	[12]	[19]	[19]	[19]	[19]	[19]
Packets per sec.	371,013	57,035	50,043	41,372	31,882	24,657	23,268	19,360	11,882	6,902	1,082

Table 3: Avg. cache miss rates of various PPC systems.

	Conventional PPC	PPC + RPC	PPC + A-RPC
Cache miss rate	17.8%	15.1%	14.6%

Figure 7: Miss rates of RPC for each traffic direction.

shows that both RPC and A-RPC showed good ability to reduce the number of compulsory misses in PPC. More specifically, RPC and A-RPC can reduce cache misses by 15.3% and 17.9%, respectively, compared to conventional PPC.

We present a more detailed analysis of the PPC misses of the conventional PPC and RPC. Fig. 7 shows the cache miss rate per trace and direction. The traces were displayed in the order from the higher bandwidth to the lower bandwidth. As shown in the figure, RPC can reduce the number of cache misses for many types of network traces (by up to 64.7% in FRG). Nevertheless, as we discussed in Section 4.2, RPC showed the different cache performances between LC1 and LC2 due to the difference in the amount of downstream.

5.2.2 Miss Improvement by A-RPC. Next, we evaluated the sensitivities of packet count used for training A-RPC (denoted as N). N was varied in the range of 100 to 100,000. We assumed that the continuous N packets that are randomly picked from each network trace come into a line card during the judgment phase, thereby removing the impact of traffic localities. Fig. 8 shows the average cache miss rate of A-RPC with ten judgments for each trace. We also show the cache miss rates of RPC and ideal judgment for reference. The figure indicates that A-RPC with $N = 1,000+$ can decide the use of RPC well. It can reduce the average cache miss rate by 3.17% compared to plain RPC (85.3% of ideal performance). A-RPC can be trained with a small number of packets and we can therefore implement a miss counter by a 10+ bit register.

5.2.3 Impact of Attacks. We evaluated the attack tolerance of RPC in cooperation with attack-aware cache, using a custom workload including attacks. This workload was based on UFL traffic, and seven attacks captured from WIDE trace were mixed into UFL traffic every 10 seconds. Fig. 9 shows the cache miss

Figure 8: Cache miss rates of four configured A-RPC.

Figure 9: Miss rates of RPC with attack-aware cache.

rates of conventional PPC, RPC, and RPC with attack-aware cache. RPC show the increase in cache misses for the outgoing traffic due to the inefficient prediction in LC1. In contrast, RPC with attack-aware cache shows small increase in cache misses for the outgoing traffic (up to 46.9% and 31.6% on average). This result indicates that attack-aware cache is effective in preventing inefficient predictions caused by attacks.

5.2.4 Hardware Cost. To assess the hardware cost of PPC including A-RPC, we implemented our A-RPC excluding TCAM, with Verilog-HDL and 45-nm Free PDK OSU Library. The area of the combination logic in PPC was computed by logical synthesis with Synopsys Design Compiler X-2005.09. The areas of some memory components, namely the cache memory, the state memory for LRU, CMT, CMQ, and the buffer, as listed in Table 1, were estimated by using CACTI 6.5 [20].

Table 4 shows the area per module in the PPC system. Note that the maximum processing delay of the implemented hardware is very small when compared to the latency of the cache, and therefore, it does not affect the critical path of the entire circuit. Table 4 indicates that the area of the cache is dominant in A-RPC. Moreover, RPC and A-RPC modules are relatively small when compared to the other modules such as CMH that is also needed for the conventional PPC. According to [21], the area of a recent TCAM package used in routers is 729 mm² and is considerably larger than PPC. Thus, our A-PPC can reduce the load of TCAM with small hardware cost.

Table 4: Area of PPC systems.

	Cache	CMH	Buffer	RPC	CRC	Arbiter	A-RPC	Others	Total
Area [mm²]	3.72×10 ⁻¹	1.27×10 ⁻²	3.30×10 ⁻³	1.42×10 ⁻³	5.67×10 ⁻⁴	3.64×10 ⁻⁴	2.29×10 ⁻⁴	2.70×10 ⁻³	3.95×10 ⁻¹
Proportion	94.2%	3.21%	0.836%	0.360%	0.144%	0.0922%	0.0581%	1.08%	100%

5.2.5 *Throughput and Energy.* Finally, we estimated the throughput and energy of various table-lookup techniques using the following two models.

The throughput of the table lookups per packet in PPC, represented as Th_{PPC} , can be calculated as:

$$Th_{PPC} = \frac{1}{t_{Avg}} \cdot 64 \text{ byte} = \begin{cases} \frac{1}{t_{Cache}} \cdot 64 \text{ byte} & (t_{Cache} > t_{TCAM} \cdot m) \\ \frac{1}{t_{TCAM} \cdot m} \cdot 64 \text{ byte} & (t_{Cache} < t_{TCAM} \cdot m). \end{cases} \quad (1)$$

Here, t_{Cache} and t_{TCAM} represent the latencies of the cache and the TCAM (0.5 and 5 ns), respectively; t_{Avg} represents the average lookup latency per packet in PPC; and m represents the cache miss rate of PPC. 64 byte refers to the shortest length of a packet. Meanwhile, the energy consumption of the table lookups per packet in PPC, represented as E_{PPC} , can be calculated as:

$$E_{PPC} = DE_{Cache} + DE_{TCAM} \cdot n \cdot m + (SE_{Cache} + SE_{TCAM}) \cdot t_{Avg} \quad (2)$$

Here, DE_{Cache} and DE_{TCAM} represent the dynamic energy per access of the cache and TCAM (0.0539 and 30 nJ), respectively; SE_{Cache} and SE_{TCAM} represent the static power of the cache and TCAM (0.0159 and 0.85 W), respectively. The energy and power of the cache were estimated with CACTI 6.5, whereas the energy and power of TCAM were computed with a TCAM power and timing model [6]. n represents the number of TCAM accesses needed to process a packet. In this paper, we assumed that a router had the four tables shown in Fig. 2 (i.e., $n = 4$).

Table 5 shows our estimation results. A-RPC can improve the throughput and energy of the table lookups by 17.9% and 17.8%, respectively, compared to the conventional PPC. As shown in the table, our A-RPC can achieve further improvement in throughput and energy efficiency due to the reduction in compulsory misses, which has not been tackled by the previous work.

6 CONCLUSION

This paper presented a novel technique called RPC to reduce compulsory misses in PPC. Our experimental results showed that RPC can reduce the number of cache misses by 15.3% on average and up to 64.7%. In addition, we extended RPC to A-RPC that selectively uses RPC for the further improvement in PPC misses. Our A-RPC can effectively select the use of RPC by using 1,000 or more packets for training. Finally, we showed that A-RPC can improve the throughput and energy consumption of the table lookups by 17.9% and 17.8%, respectively, when compared to conventional PPC.

REFERENCES

- [1] The Ministry, Tabulation and Estimation of Internet Traffic in Japan, 2016, Available: http://www.soumu.go.jp/main_content/000462459.pdf. [Accessed Nov. 16, 2017]

Table 5: Throughput and energy.

	Only TCAM	TCAM + PPC	TCAM + PPC + A-RPC
Throughput [Gbps]	102	441	512
Energy [nJ]	124	22.2	18.2

- [2] METI, Green IT Initiative in Japan, Available: <http://www.meti.go.jp/english/policy/GreenITInitiativeInJapan.pdf>. [Accessed Nov. 16, 2017]
- [3] Cisco Systems, Inc., Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2016–2021 White Paper, Available: <http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/mobile-white-paper-c11-520862.pdf>. [Accessed Nov. 16, 2017]
- [4] J. Fan, C. Hu, K. He, J. Jiang, and B. Liuy, Reducing power of traffic manager in routers via dynamic on/off-chip scheduling, *2012 Proc. IEEE INFOCOM*, Orlando, FL, 2012, pp. 1925–1933.
- [5] X. Zheng, X. Wang, Comparative study of power consumption of a NetFPGA-based forwarding node in publish–subscribe Internet routing, *Computer Communications*, vol. 44, pp. 36–43, May 2014.
- [6] B. Agrawal and T. Sherwood, Ternary CAM power and delay model: Extensions and uses, in *IEEE Trans. on Very Large Scale Integration (VLSI) Systems*, vol. 16, no. 5, pp. 554–564, May 2008.
- [7] M. Nawa et al., Energy-efficient high-speed search engine using a multi-dimensional TCAM architecture with parallel pipelined subdivided structure, *2016 13th IEEE Annual Consumer Communications & Networking Conference (CCNC)*, Las Vegas, NV, 2016, pp. 309–314.
- [8] Hewlett-Packard Development Company, Energy Efficient Networking - Business white paper, 2011, Available: <http://h17007.www1.hp.com/docs/mark/4AA3-3866ENW.pdf>. [Accessed Nov. 16, 2017]
- [9] S. Gamage and A. Pasqual, High performance parallel packet classification architecture with popular rule caching, *2012 18th IEEE Int'l. Conf. on Networks (ICON)*, Singapore, 2012, pp. 52–57.
- [10] G. Chandramohan and G. Ramaswamy, Improving performance of digest caches in network processors, In *Proc. of the 15th Int'l. Conf. on High performance computing (HiPC'08)*, India, 2008, pp.6–17.
- [11] H. Yamaki and H. Nishi, Line replacement algorithm for L1-scale packet processing cache, In *Adjunct Proc. of the 13th Int'l. Conf. on Mobile and Ubiquitous Systems: Computing Networking and Services (MOBIQUITOUS2016)*, Hiroshima, Japan, 2016, pp. 12–17.
- [12] F. Chang, W. C. Feng, and K. Li, Efficient packet classification with digest caches, *Proc. Third Workshop Network Processors and Applications (NP-3)*, 2005.
- [13] S. Ata, M. Murata, and H. Miyahara, Efficient cache structures of IP routers to provide policy-based services, *IEEE Int'l. Conf. on Communications (ICC 2001)*, Helsinki, 2001, vol.5, pp. 1561–1565.
- [14] G. Liao, H. Yu and L. Bhuyan, A new IP lookup cache for high performance IP routers, *Design Automation Conference*, Anaheim, CA, USA, 2010, pp. 338–343.
- [15] N. Kim, S. Jean, J. Kim, and H. Yoon, Cache replacement schemes for data-driven label switching networks, *2001 IEEE Workshop on High Performance Switching and Routing*, Dallas, TX, 2001, pp. 223–227.
- [16] WIDE MAWI WorkingGroup, MAWI Working Group Traffic Archive Available: <http://mawi.wide.ad.jp/mawi/>. [Accesses 17 Nov. 2016]
- [17] H. Yamaki and H. Nishi, An improved cache mechanism for a cache-based network processor, *Proc. of the Int. Conf. on Parallel and Distributed Processing Techniques and Applications (PDPTA'12)*, Las Vegas, NV, 2012, pp. 1–7.
- [18] M. Okuno and H. Nishi, Network-processor acceleration-architecture using header-learning cache and cache-miss handler, *The 8th World Multi-Conference on Systemics, Cybernetics and Informatics (SCI2004)*, pp. 108–113.
- [19] RIPE Network Coordination Centre, Réseaux IP Européens Network Coordination Centre RIPE NCC, Available: <http://www.ripe.net/>. [Accessed Nov. 16, 2017]
- [20] N. Muralimanohar et al., Optimizing NUCA organizations and wiring alternatives for large caches with CACTI 6.0, *Proc. of the 40th Annual IEEE/ACM Int. Symposium on Microarchitecture (MICRO 40)*, Chicago, USA, 2007, pp.3–14.
- [21] RENESAS, 20Mbit Quad-Search Content Addressable Mem., https://www.renesas.com/ja-jp/doc/products/memory/r10cp0001eu0100_tcam.pdf. [Accessed Nov.16, 2017]