Architecture of RENA-CHIP

Yukikuni Nishida[†]

Abstract

The architecture of RENA-CHIP, which is designed for use in network adapters, is based on an application-specific integrated circuit (ASIC) and has an inline IPsec (Internet protocol security) engine. This article explains the reasons for these choices by considering personal computer and network processor performance estimation results.

1. Introduction

The performance of processors for embedded devices in network adapters has improved dramatically in recent years as conversion of the network to broadband speeds has progressed. Compared with processors of the 100-Mbit/s era, however, an even higher level of performance will be needed to process packets at 1 Gbit/s. To meet this need, chip makers are developing processors that can support Gigabit Ethernet (GbE).

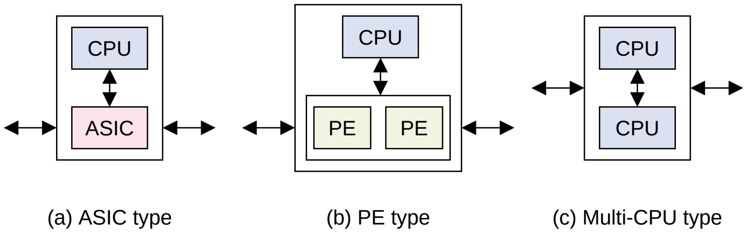
RENA-CHIP was designed to enable the construction of a network adapter that can communicate at the full wire rate of 2 Gbit/s (1 Gbit/s each in the upstream and downstream directions) at a low cost. When the adapter receives a burst of packets that exceeds the network adapter’s performance, but does not exceed the capacity of the receive packet buffer, there is no problem. However, when the capacity of the receive packet buffer is exceeded, and packets are dropped, the quality of service (QoS) may be degraded. This is particularly true for realtime services such as VoIP (voice over Internet protocol). If the processing can be performed at 2 Gbit/s, then the arriving packets should never exceed the packet buffer [1]. The next-generation network adapter will feature not only higher speeds but also upgraded functions. For instance, a QoS function for triple play (voice, video, and data) has been added and IPv6 (Internet protocol version 6) is supported as well as IPv4. Therefore, the chip must have not only sufficient performance, but also a QoS function, a classifier function that can filter and classify packets, and an IPsec (Internet security) function to handle IPv6. These functions generally create a heavy load [2]. To meet these requirements, we chose an ASIC (application-specific integrated circuit) architecture for RENA-CHIP, as explained below. Moreover, since the CPU controlling RENA-CHIP is not embedded in RENA-CHIP but is separate from it, our design does not limit the CPU used.

In this article, I introduce network processors and compare the specifications of RENA-CHIP and a network processor. I examined the packet filtering performance and QoS performance of this chip and network processors in terms of catalog values and estimates for several personal computers (PCs) because boards for estimating network processors were not readily available for purchase at that time. I also discuss the IPsec engine and power consumption.

2. Communication LSIs

In general, a communication LSI (large-scale integration) chip rather than a general purpose processor is used to achieve the high-speed processing described in the introduction. There are currently several types of communications LSIs [3]. Recently, the network processor has attracted attention as a communication LSI suitable for a network with high performance and high functionality. Communication LSIs can be classified roughly into three types, as

† NTT Cyber Solutions Laboratories
Yokosuka-shi, 239-0847 Japan
E-mail: nishida.yukikuni@lab.ntt.co.jp



(a) ASIC type (b) PE type (c) Multi-CPU type

Fig. 1. Various types of communication LSI.

Table 1. Features of RENA-CHIP, IXP2325, and CN3120.

Device name	RENA-CHIP	IXP2325	OCTEON CN3120
CPU core	external	Xscale	MIPS64 × 2
Operation clock	133 MHz	900 MHz	550 MHz
Forwarding engine	specific hardware	Microengine × 2 (600 MHz)	—
Performance	2.8 Mp/s	1 Mp/s	202 kp/s (estimated)
IPsec performance	2 Gbit/s	200 Mbit/s	>1 Gbit/s
QoS mechanism	yes (hardware)	yes (microcode needs to be developed.)	yes (hardware)
Power consumption	2 W (RENA-CHIP) + 0.37 W (CPU)	9.5 W	7 W
Package size	456 balls 27 × 27 mm^2	1752 balls 42.5 × 42.5 mm^2	868 balls 40 × 40 mm^2
Price	Low	High	High

shown in **Fig. 1**. The ASIC type (a) performs packet processing in hardware and the CPU handles exceptions and performs hardware control. The PE-type network processor (b) is composed of a processor and several programmable elements (PEs); it works at high speed for the hardware processing part of the ASIC type though the instruction set is small. The multi-CPU-type network processor (c) has two or more processors. For comparison with our RENA-CHIP, which has the ASIC architecture, we chose the IXP2325 (Intel) as a PE-type network processor and OCTEON CN3120 (Clavium Networks) as a multi-CPU-type network processor. Their features are summarized in **Table 1**.

RENA-CHIP has a packet processing performance of about 3 million packets per second (3 Mp/s) to achieve the full wire-rate throughput of 2 Gbit/s. The processing performance of IXP2325 was derived from the performance of its PE, which the IXP2325's product catalog says is 2.5 billion instructions per second. The catalog also says that IXP2325 can handle virtual local area networking (VLAN), forwarding, and QoS at a rate in excess of 1 million packets per second. The information we obtained about CN3120 indicates that it has two MIPS64 cores, can be driven at 550 MHz, and can achieve a performance of several gigabits per second [4]. We consider that the VAX MIPS* value of CN3120 at an operating frequency of 550 MHz is 1543 DMIPS (Dhrystone million instructions per second) because the VAX MIPS value of the MIPS64 architecture at 310 MHz is 455 DMIPS [5]. The estimated performance of CN3120 was calculated for 202 kp/s using Eq. (1) in Sec. 3.1. When the packet size is 1500 bytes, the throughput is 2.4 Gbit/s. We chose to use 202 kp/s because this performance corresponds to the performance of several gigabits per second mentioned in the press release [4]. Although the other types have high flexibility, unlike the ASIC type, there is less need for flexibility when the specifications of the network in question have been decided.

To compare the prices of RENA-CHIP, IXP2325, and CN3120, we added the price of the external CPU to the price of RENA-CHIP because an external CPU is necessary. This CPU need not be an expensive one:

* VAX MIPS: A widely used performance benchmark.

a low-price one (MIPS32 architecture and 200-MHz operation) is adequate. As a result, the combined price of RENA-CHIP plus CPU is less than about half the price of the other network processors. Therefore, a network adapter can be made at a low cost by using RENA-CHIP plus CPU.

In terms of programming, it is easy to use CN3120, which is a multi-CPU type. If the operating system is designed for multiple CPU cores, the user can use existing software. The next easiest is RENA-CHIP, which is the ASIC type. Software is necessary for packet processing and for managing the tables in RENA-CHIP. However, the ASIC type is easier to use than the PE type because we do not need to make any microcode. The PE type is generally said to be difficult to program, especially for a symmetric PE type such as IXP2325 [6]. There is no problem if the library has been enhanced, but if it has not, PE programming will take a lot of time.

As shown in Table 1, only RENA-CHIP meets the performance requirements. Moreover, this chip has a price advantage over other communication processors. Its effectiveness including its future outlook is described in the next section.

3. Performance estimation

3.1 Packet filtering performance

For a packet length of 1500 bytes and unidirectional packet transfer, processors that can achieve a throughput of 1 Gbit/s are currently available on the market. In many cases, however, the performance described in CPU/processor catalogs refers only to the transfer of IP packets without filtering, and there is no way to determine what kind of performance a processor can achieve if packet filtering is included. To overcome this problem, we measured the packet-filtering performance of routers constructed from PCs and predicted the performance required for packet filtering at a bidirectional transfer rate of 2 Gbit/s based on the results obtained.

To ensure a common level of performance for the network interface cards (NICs) and drivers used in the measurements, we used the same type of NIC throughout, Linux was chosen as the common operating system, and netfilter was used as the packet filtering mechanism [7]. Four types of PCs with CPU operating frequencies of 700 MHz, 1.5 GHz, 2.4 GHz, and 3.06 GHz were used in the measurements, as listed in **Table 2**. Of these, PC3 had a PCI-X bus, while the rest had only a PCI (peripheral component interconnect) bus. The number of filtering rules at the time of measurement was varied among 0, 64, 128, and 256, and throughput and number of transferred packets were determined for the case of bidirectional communication. Filtering rules for passing the packets used in the measurements were established using the last two entries with the source and destination IPv6 addresses used as search conditions.

As an example of the measurement results, the throughput when PC2 was used is shown in **Fig. 2**. The throughput peaked at about 200 Mbit/s. This is because the NIC was connected to a PCI bus. For 32-bit/33-MHz operation, a PCI bus has an effective bandwidth of 800–900 Mbit/s. However, one must keep in mind that the measurements performed here assumed bidirectional communications, and that in one direction, a frame was transferred in the manner: NIC → CPU → NIC. As a result, the PCI bus was used twice in one direction and another two times in the opposite direction, making a total of four times. This explains the peak transmission speed of about

Table 2. Main specifications of PCs used in the measurements.

	PC1	PC2	PC3	PC4
CPU	Celeron 700 MHz	Pentium 4 1500 MHz	Xeon 2.40 GHz	Pentium 4 3.06 GHz
Memory (memory bandwidth)	SDR SDRAM (0.53 GB/s)	PC800 RDRAM (1.6 GB/s)	PC2100 DDR266 (2.1 GB/s)	PC2700 DDR SDRAM (2.67 GB/s)
VAX MIPS value	764.588	1191.991	2021.28	2539.61
Bus connecting to NIC	PCI	PCI	PCI-X	PCI

SDR SDRAM (single data rate synchronous dynamic random access memory) is designed to operate in synch with an external clock. Popular memory modules such as PC66, PC100, and PC133 are of this type.

DDR SDRAM (double data rate SDRAM) can read/write at double the speed of the SDR SDRAM. PC1600, PC2100, and PC2700 are of this type.

RDRAM is a high-speed DRAM with a bus architecture developed by Rambus Co. RIMM 1600, etc. are of this type.

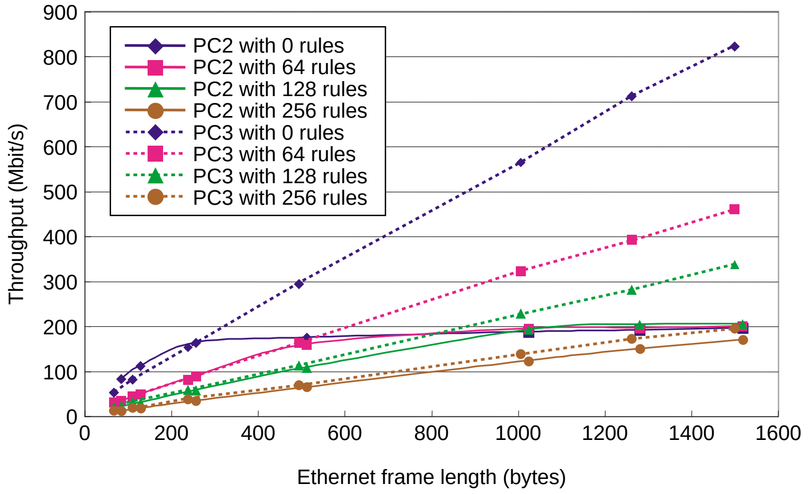


Fig. 2. PC2/PC3 throughput measurement results when numbers of packet filter rules were 0, 64, 128, and 256.

200 Mbit/s. To verify this value, we reduced the CPU operating speed of PC3 to 1.6 GHz so that a comparison with PC2 could be made at about the same operating frequency. Measurement results using PC3 in this way are also shown in Fig. 2. They show that throughput increased as the Ethernet frame length increased for both PC2 and PC3 up to a transfer rate of about 200 Mbit/s, but that PC3's throughput continued to increase past 200 Mbit/s up to the limit of the packet-transfer performance in question. This is because the PCI-X bus has a bandwidth of about 6 Gbit/s (for 64-bit/100-MHz operation). Consequently, to achieve 2-Gbit/s transfer in both directions combined, it is not sufficient to consider just CPU performance—the bandwidth of the data bus must also be taken into account.

Next, as a preliminary step, a graph was drawn with the horizontal axis representing the number of filtering rules and the vertical axis showing the packet processing time. The measurement data lined up along straight lines whose slopes depended on CPU performance. The values obtained from this experiment were then used to perform regression analysis using Eq (1) to predict packet processing time.

$$time = \frac{C_0 + C_1 R}{DMIPS} \quad (1)$$

Here, *time* denotes the time for processing one packet, C_0 is the amount of processing unrelated to the number of filtering rules, C_1 is the increase in amount of processing related to an increase in filtering rules, R is the number of packet filtering rule, and *DMIPS* is the CPU performance indicated as a VAX MIPS value obtained from a Dhrystone benchmark program. The values of C_0 and C_1 obtained from the regression analysis were 7610.874 and 144.8462, respectively.

Now, to ensure that the above packet processing time can be applied to other CPU architectures, we compared actual throughput measured for a 200-MHz, MIPS32-architecture CPU (VAX MIPS:117) and the throughput computed from processing time obtained from Eq. (1). A maximum difference of about 20% was found between these values. Nonetheless, we thought that the above expression is still suitable for predicting packet processing time.

Here, the VAX MIPS value of IXP2325 was assumed to be 2500 based on the 2.5 billion instructions per second that is the processing speed of its PE (microengine). It was assumed to be 1543 DMIPS through a similar calculation to that for CN3120 in the previous section. In the case of no filtering, achieving 2 Gbit/s (2.8 Mp/s) requires 22,650 DMIPS. Consequently, IXP2325 and CN3120 need performance improvements of about 9 and 15 times, respectively.

If we assume that CPU performance will improve according to Moore's Law, it will take from four to five years for an IXP2325 capable of 2-Gbit/s, simultaneous bidirectional throughput to become available on the market. In short, to achieve such throughput, not only should packet filtering be performed by hardware, packet transfer should be as well.

We also considered a change in the number of packet filtering rules. Obtaining a performance of 2 Gbit/s with 256 packet filtering rules requires about 125,000 DMIPS, according to our estimate using Eq. (1). This value is unrealistic. Therefore, the packet filter was made by wired logic in RENA-CHIP to achieve 2-Gbit/s performance.

3.2 QoS processing performance

QoS processing is necessary if services are to be provided in a smooth and acceptable manner. This technology controls packets according to their quality requirements as dictated by the service being provided such as VoIP, video delivery, or connection to an Internet service provider. RENA-CHIP and CN3120 both achieve this with specific hardware while IXP2325 does it with its microengine. In determining the extent to which QoS can be controlled with a CPU, we decided to measure the performance of the packet shaping function, which controls the packet transfer rate and smoothes out the packet transmission interval.

For this set of measurements, PC1 having a performance level similar to that of an embedded CPU was used, and tc-htb, which is one of the packet shapers for Linux, was used as the QoS mechanism [8]. The shaping rate at the PC was varied among 1, 3, 6, 12, and 25 Mbit/s. Packets having an Ethernet frame length of 242 bytes, which simulated VoIP packets, were sent from a measurement device to the PC at different transmission rates, and the packet output rates from the PC for each input rate were measured. The measurement results are shown in **Fig. 3**. For shaping rates up to 3 Mbit/s, packets were output from the PC at the set shaping rate. But for shaping rates of 6 Mbit/s and above, the CPU could no longer perform shaping at the set rate. In fact, for shaping rates of 12 and 25 Mbit/s, the best that the CPU could achieve was an output packet rate of about 7 Mbit/s. On the other hand, for an Ethernet frame length of 1500 bytes (a small number of packets per unit time), we found by experiment that shaping could be performed well even for a set shaping rate of 25 Mbit/s. In these experimental measurements, the kernel was recompiled at a maximum Linux timer accuracy of 1/2048 s using a CPU with an x86 architecture. Accordingly, the maximum shaping rate for an Ethernet frame length of 242 bytes turned out to be $242 \times 8 \times 2048 \approx 3.96$ Mbit/s. If the shaping rate is set higher than this maximum value, the CPU will output packets consecutively to achieve the set shaping rate as much as possible, as demonstrated by the experiment.

Timer accuracy of 672 ns is required for shaping 64-byte packets at 1 Gbit/s. The QoS mechanism of the operating system does not achieve this because the timer interrupt period of the operating system is up to about 1 ms. We concluded that to meet this requirement it is necessary to achieve the QoS mechanism separately from the operating system. However, it is generally difficult to make a QoS processing program separately from the operating system, so it will be difficult to implement the QoS mechanism in IXP2325.

The above results indicate that QoS processing should also be performed in hardware to achieve QoS control at a maximum rate of 1 Gbit/s. RENA-CHIP can flexibly adapt the packet scheduling mechanism to suit the network service. Even though it is hardware, RENA-CHIP has the flexibility of a QoS switch that changes the connections of packet scheduling modules. [1]

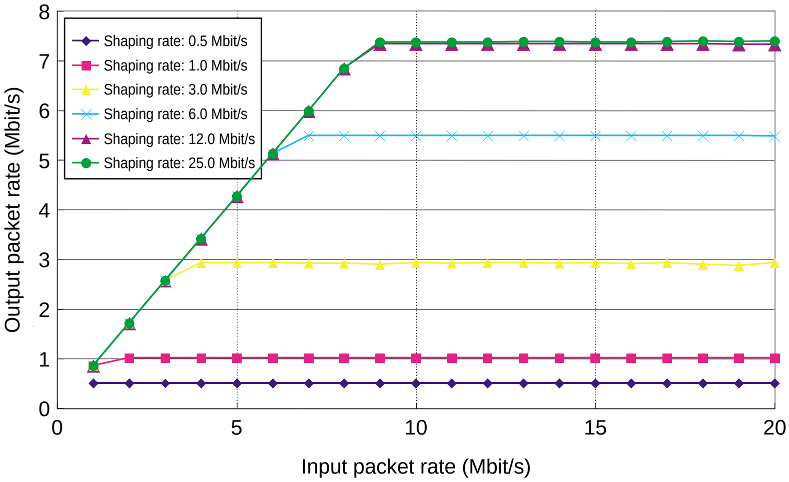


Fig. 3. Packet shaping performance.

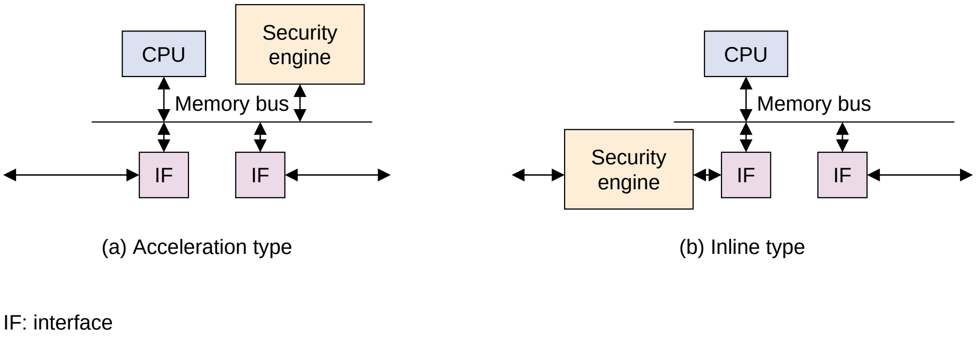
3.3 IPsec processing performance

To investigate IPsec processing performance, we profiled the IPsec-based communications (by calculating how many times program functions were called and the duration of their use). The specific objectives of this profiling were to find out how much of the packet transfer time is taken up by IPsec processing when IPv4-over-IPv6 tunnel mode is used and to determine the processing time required per packet. For this, we used PC4 and measured the processing time per packet for various encryption and authentication schemes and for Ethernet frame lengths of 66 and 1000 bytes. Encapsulated Security Protocol (ESP) with authentication was used for data transmission, while NULL, DES, 3DES, and AES128 were used as coding/decoding schemes and NULL, HMAC-MD5-96, and HMAC-SHA1-96 were used as authentication schemes [9]-[15]. For the operating system, FreeBSD was used because it can create IPv4-over-IPv6 tunnels.

The processing times per packet for an Ethernet frame length of 1000 bytes when PC4 was used were 33.72, 65.361, 14.101, 9.30, and 23. 05 µs for DES, 3DES, AES, MD5, SHA-1. Thus, these schemes shown can be ordered as 3DES>DES>SHA-1>AES> MD5 in terms of required processing time. Furthermore, for the combination of 3DES and SHA-1, the coding and decoding processes each required 88 µs, which means that the encryption process took up 77%, that is, the majority, of the packet transfer processing time. To raise the performance for AES to 2 Gbit/s, for instance, it would be necessary to improve the performance of PC4 41 times (to 2539.61 DMIPS), which is unrealistic. Therefore, a security engine is installed not only in RENA-CHIP, but also in IXP2325 and CN3120.

Two main systems can be considered for performing IPsec processing by hardware, as shown in **Fig. 4**. These are the accelerator system and the inline system (in which IPsec processing on a packet is performed without CPU intervention, i.e., before the CPU receives the packet and after the CPU outputs the packet). In the accelerator system, the CPU performs a direct memory access (DMA) transfer of frame data in main memory to the IPsec accelerator, and on completion of processing at the accelerator, performs another DMA transfer to write that frame data again in main memory. Consequently, the processing of one frame of data uses the memory bus four times, i.e., from the NIC to main memory, from main memory to the IPsec accelerator, from the IPsec accelerator to main memory, and from main memory to the NIC. This requires a memory-bus bandwidth of at least 500 Mbyte/s (4 Gbit/s) for unidirectional communications and 1 Gbyte/s (8 Gbit/s) for bidirectional communications. The latter is nearly the same as the 1.064-Gbyte/s bandwidth of a 32-bit, 133-MHz DDR-SDRAM (double data rate synchronous dynamic random access memory). If we assume a memory-bus usage rate of 50% for frame transfer, the DDR-SDRAM would have to be operated at 266 MHz, which is twice the operating frequency. Since the memory-bus usage rate fluctuates when applications are running, the inline system is preferable to the above accelerator system.

The inline type cannot be used for purposes other than IPsec. In the inline type, there is no bottleneck in the memory bus because the bottleneck is in the Ethernet communication line. Therefore, the inline type can be operated at high speed. Moreover, the function for inserting/deleting the outer IPv6 header was placed in the security engine in RENA-CHIP, and this simplifies the chip's forwarding engine and frame generation block. Therefore, the inline type was chosen for RENA-CHIP.



(a) Acceleration type

(b) Inline type

Fig. 4. Types of IPsec processing.

4. Power consumption

It is difficult to limit how the user sets up a network adapter. He or she may put it behind a desk and stand it upright or lay it flat, so the designer should consider various situations. If the LSI could be operated without a heatsink, this would lower the cost and improve the reliability by simplifying the design, and it would also increase the setup flexibility. We considered power consumption. The processor will not need a heatsink if its package satisfies the following criterion.

$$\theta jc < \frac{Tj - Ta}{Q}, \qquad (2)$$

where θjc is the thermal resistance of the package (°C/W), Q is the power consumption, Tj is the maximum temperature of chip, and Ta is the maximum air temperature. The relationship between package size and thermal resistance is shown in **Fig. 5**, assuming the maximum chip temperature to be 120°C and the maximum air temperature to be 50°C. Because the number of samples was small, lines have been drawn as guides for the eye. IXP2325 and CN3120 both require heatsinks because they fall under the line and do not satisfy condition (2). While the two lines in Fig. 5 indicate the thermal resistance of packages, each mark for RENA-CHIP, IXP2325, and CN3120 in the figure indicates the upper allowable limit of the package thermal resistance where each processor (chip) can operate normally. IXP2325 and CN3120, therefore, require heatsinks or cooling fans that lower their thermal resistances because they require even lower thermal resistances than those of the packages themselves.

In section 3, we showed that performance improvements are needed for a network processor to achieve 2-Gbit/s performance. However, to satisfy this requirement, the network processor requires more PEs or CPUs, and/or a higher operating speed, which will increase the power consumption. Therefore, the goal of operating the LSI without a heatsink cannot be met using a network processor.

In contrast, the RENA-CHIP can be operated without a heatsink because it stays above the lines (Fig. 5) and the RENA-CHIP can be operated even if its package thermal resistance is higher than that of the package itself. Thus, a network adapter that uses RENA-CHIP is easy to design and should have high reliability.

5. Conclusion

In designing RENA-CHIP, the ASIC-type architecture was chosen to achieve the full wire rate of 2 Gbit/s. For IPsec processing, the inline type was chosen. We compared our chip with two network processors in terms of catalog specifications and estimated performance. The ASIC type was chosen because it is expected to have better performance and lower power consumption. Current network processors need performance improvements ranging from about nine to fifteen times, which gives RENA-CHIP a lead of four

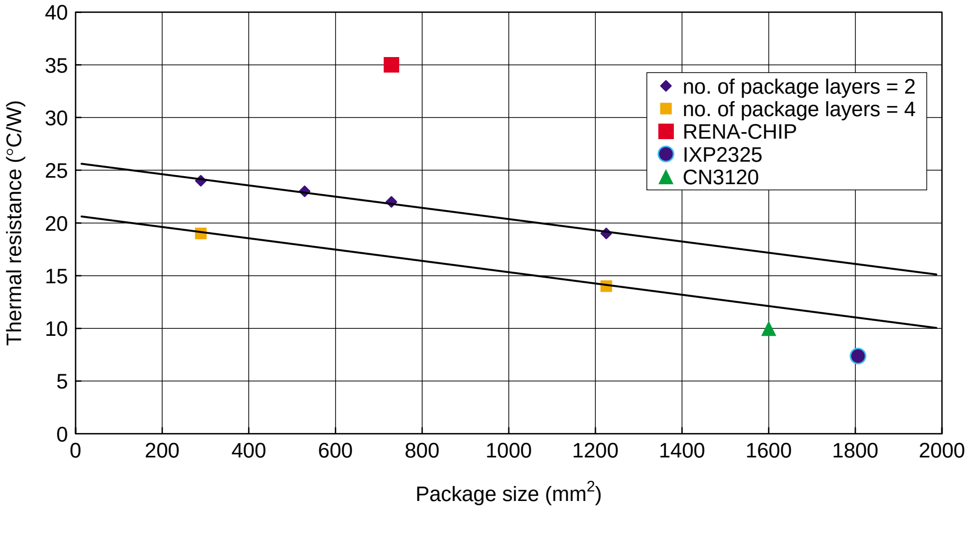


Fig. 5. Thermal resistances of packages are shown by lines, and thermal resistances of packages which processors require are shown by dots.

to six years. Our chip is advantageous in terms of both cost and performance compared with other network processors.

References

[1] K. Kawai, "RENA-CHIP Hardware Configuration Technologies," NTT Technical Review, Vol. 4, No. 9, pp. 30-34, 2006 (this issue).

[2] O. Shimokuni, J. Kawai, A. Jinzaki, M. Yamasawa, O. Nakamura, and J. Murai, "Implementation and Evaluation of Low Power IPsec ESP by Security Network Processor," IC2003, Tokyo, Oct. 2003.

[3] E. Kawai, Y. Kadobayashi, and S. Yamaguchi, "A Survey on Network Processor Technologies and R&D Trends," IPSJ Trans on Advanced Computing System, Vol. 45, No. SIG 1 (ACS 4), pp. 54-64, 2004.

[4] http://www.cavium.com/newsevents_OCTEONMIPS64.html

[5] http://www.us.design-reuse.com/news/news29.html

[6] N. Shah and K. Keutzer, "Network Processors: Origin of Species," Proc, ISCIS Int. Computer and Information Sciences, 2002.

[7] http://www.netfilter.org/

[8] http://luxik.cdi.cz/~devik/qos/htb/

[9] S. Kent and R. Atkinson, "IP Encapsulating Security Payload (ESP)," RFC2406, 1998.

[10] R. Glenn and S. Kent, "The NULL Encryption Algorithm and Its Use With IPsec," RFC2410, 1998.

[11] P. Karn, P. Metzger, and W. Simpson, "The ESP DES-CBC Transform," RFC1829, 1995.

[12] P. Karn, P. Metzger, and W. Simpson, "The ESP Triple DES Transform," RFC1851, 1995.

[13] S. Frankel, R. Glenn, and S. Kelly, "The AES-CBC Cipher Algorithm and Its Use with IPsec," RFC3602, 2003.

[14] C. Madson and R. Glenn, "The Use of HMAC-MD5-96 within ESP and AH," RFC2403, 1998.

[15] C. Madson and R. Glenn, "The Use of HMAC-SHA-1-96 within ESP and AH," RFC2404, 1998.

Yukikuni Nishida

Research Engineer, First Promotion Project, NTT Cyber Solutions Laboratories.

He received the B.E. and M.E. degrees in electrical and computer engineering from Nagoya Institute of Technology, Aichi, in 1992 and 1994, respectively. He joined NTT LSI Laboratories, Kanagawa, in 1994 and has been engaged in research on the design of low-power-consumption DSPs, automatic speech recognition systems, and network appliances. He is a member of the Institute of Electronics, Information and Communication Engineers of Japan.