Towards High-performance IPsec on Cavium OCTEON Platform

Jinli Meng[1], Xinming Chen[2,3], Zhen Chen[3], Chuang Lin[3], Beipeng Mu[2], and Lingyun Ruan[2]

[1]Department of Computer Science and Technology, Tsinghua University, Beijing, China
[2]Department of Automation, Tsinghua University, Beijing, China
[3]Research Institute of Information Technology (RIIT), Tsinghua University, Beijing, China
mengjl08@csnet1.cs.tsinghua.edu.cn,chen-xm09@mails.tsinghua.edu.cn, {zhenchen,chlin}@tsinghua.edu.cn,{mubeipeng,rlyswf}@gmail.com,

Abstract. Providing secure, reliable communications is a big challenge to guarantee confidentiality, integrity, and anti-replay protection, especially between endpoints in current Internet. As one of the popular secure communication protocol, IPsec usually limits the throughput and increases the latency due to its heavy encryption/decryption processing. In this paper, we propose a hardware solution to accelerate it. To achieve high performance processing, we have successfully designed and implemented IPsec on Cavium OCTEON 5860 multi-core network processor platform.
We also compare the performance under different processing mechanisms and discover that pipleline works better than run-to-completion for different sizes of packets in our experiments. In order to achieve the best performance, we select different encryption algorithms and core numbers. Experimental results on 5860 processors show that our work achieves 20 Gbps throughput with AES128 encryption, 16 cores for 512-byte packet traffic.

Keywords: network processor, multi-core, network security, IPsec, cryptography.

1 Introduction

Internet's openness brings enormous benefits, as well as many security risks. In order to provide a trusting information exchange environment, many security mechanisms are used across the network. However, only about 10% of the infomation across the Internet are protected because of the encrypting performance and cost [1]. IPsec (IP security) is a protocol suite for securing IP communications, supplying confidentiality, data integrity, anti-replay attack that the IETF delimited for the network layer to provide security services. However, IPsec function greatly increases the load of the network devices [2]. Also there are multiple

parameters in the implementation of IPsec, which provide trade-off between security and efficiency. The design, implementation and optimization of IPsec should be paid special attention to these parameters in order to enhance efficiency.

In this paper, we focus on the IPsec implementation on network processors, because they provide hardware acceleration specialized for packet processing and encryption / decryption. The OCTEON family of Multi-Core MIPS64 processors is one of the industry's most scalable, highest-performance, and lowest-power solution for intelligent networking applications. So our goal is to test and implement IPsec in this hardware platform in an efficient and high-performance way.

We design, implement and test IPsec on OCTEON CN58XX platform. Different encryption algorithms such as AES, DES, 3DES have different impacts on performance. Different packet length from 64 bytes to 1280 bytes and different core numbers from 1 to 16 are also important factors for performance. The most important factor is the system mechanism. Two mechanisms, run-to-completion and pipeline, are implemented and discussed in this paper. Run-to-completion mechanism means to deal with the packets in the same flow in a core from the very beginning to the end, and pipeline mechanism means packets from the same flow can be processed in parallel in multiple cores sometimes. We discuss the reasons and give advices on how to construct high-performance IPsec implementation on multi-core platform.

The rest of the paper is organized as follows: Section 2 gives an overview of Cavium OCTEON network platform and its two kinds of operating mechanisms named run-to-completion and pipeline, introductions about IPsec parameters and about encrypting algorithms are also included. Details of the design of our IPsec implementation are given in Section 3, including the data structures, processing procedure etc. Section 4 illustrates the performance experiments and three groups of performance data, analyses the best parameter combination for a better performance, and discusses the reasons why it should be. As a summary, in Section 5, we state our conclusion.

2 Background

2.1 Cavium OCTEON

The OCTEON CN58XX family of multi-core MIPS64 network processors is released in 2008. It is an industrial solution for intelligent networking applications ranging from 100Mbps to 40Gbps. On OCTEON 58XX network processor, one network processor chip consists of 16 cores. Each core can be programmed separately [3–5], so they can provide solution for scalable networking equipments.

2.2 Run-to-completion

For multi-core processors, there are two ways to perform batch processing: the first one called run-to-completion is to execute the whole processing of a flow

in the same core. The other one, called pipeline, is to divide the processing procedure of packet into several simple executives or stages, and one stage in one core. In this way, multiple cores can deal with packets in different stage from the same flow simultaneously, and the completion of one processing needs multiple cores. Once a packet comes, the de-scheduled module selects the shortest queue of these cores where to put the packet. If the selected queue is blocked, the blocked packet will be put aside, and the subsequent packet in the queue will be scheduled.

2.3 Pipeline

There is a POW (Packet order / Work Unit) also named SSO (Schedule / Synchronization / Order) module in Cavium OCTEON Network Processor[3, 4]. In this mechanism, the processing is divided into several simple executors corresponding to different processing stages; every core has its own queue maintained by POW unit. These simple executors can be tagged with ORDERED, ATOMIC, and NULL. These tags can be switched among them. Multiple packets from the same flow with an ORDERED Tag Type may be processed in parallel by multiple cores. More cores working on packets in the same flow results in higher packet throughput. While ATOMIC Tag Types mean processing should be serialized: one-at-a-time in ingress-order. NULL Tag Types are neither ordered nor serialized: multiple cores can process multiple packets, and no packet order is maintained. Four relationship cases the packet and the waiting queue have:

1. Once a packet arrived, it will enter the queue.
2. Once a simple executive finished, and the tag is not NULL, the packet will be put back to the queue.
3. Once the simple executive is blocked, it will quit and de-schedule.
4. Once a simple executive finished, and the tag is NULL, the packet will run out of the system.
5. This is an endless loop.

2.4 IPsec

IPsec (IP security) is a protocol suite for securing communications in IP networks. The essence of IPsec is to add security fields between IP field and transport layer in original IP packet, which can be unpacked by receiver for verification. In our experiment, we will use ESP and transport mode to simplify the operations.

Fig. 1 illusions the ESP header. SPI (Security Parameter Index) is the data structure to point to SA (Security Association), which records the shared secret key and related information. SN (Sequence Number) stores the counter of all the packets in one SA, in order to resist the replay attack. IV (Initialization Vector) is the initial vector of encryption algorithm, including AES, DES, and etc. Some algorithms need to pad to meet the requirement on packet length.

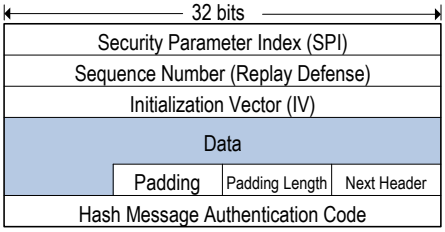

Fig. 1. Data structure of ESP tranport mode in IP payload

In the end, the ESP tail HMAC (Hash Message Authentication Code) provides integrity guarantee.

There are some consideration in our design and implementation of IPsec scheme in Cavium OCTEON CN58XX platform. The IPsec scheme is a multi-service, multi-algorithm framework. Multi-service guarantees kinds of on-demand, on-price services. Multi-algorithm supplies various confidentiality environments. Under our circumstances, the experiments are conducted to choose the best fit encryption algorithm on the consideration the tradeoff of performance and confidentiality.

The number of used cores will also affect the performance. In the scenario of multi-core platform, different parallelization methods of IPsec scheme will result in different performance in throughput. There are run-to-completion and pipeline mechanisms [6–8] should be tested and evaluated for final IPsec scheme in multi-core platform.

2.5 Encrypting Algorithms

There are kinds of encrypting algorithms such as DES(Data Encryption Standard), 3DES(Triple DES), AES(Advanced Encryption Standard). 3DES is almost the same three DES rounds to enhance the safety of DES. AES is faster and safer than DES. For AES, the secret key can be 128 bits, 192 bits, 256 bits. All these algorithms are implemented with hardware.

3 IPsec Implementation

The IPsec processing procedure can be divided into six stages: Defragment, IPsec decrypt, Lookup, Process, IPsec encrypt, and Output. Table 1 shows the stages and corresponding tags (a tag is a special label of different stages). Defragment means to reconstruct IP packet with data fragment. In the IPsec decrypt stage, incoming packets should be decrypted and recovered to the original ones. While forwarding the packet, it needs to check the SPD table and SA table according to the hash value of five-tuple of the packet; that's what Lookup stage does [3]. The Process stage involves the necessary processing of packets before sending them out, such as NAT translation or TCP sequence number adjustment. The

Table 1. IPsec stages

Step	Tag
Defrag	O tag={0,hash(IP src, IP dest)}
IPsec decrypt	A tag={1, IPsec SPI/SA}
Lookup	A tag={2, hash(IP src, IP dest, IP protocol, IP srcport, IP destport)}
Process	O tag={3, hash(IP src, IP dest, IP protocol, IP srcport, IP destport)}
IPsec encrypt	A tag={4, IPsec SPI/SA}
Output queue	A tag={5, Output Queue Index}

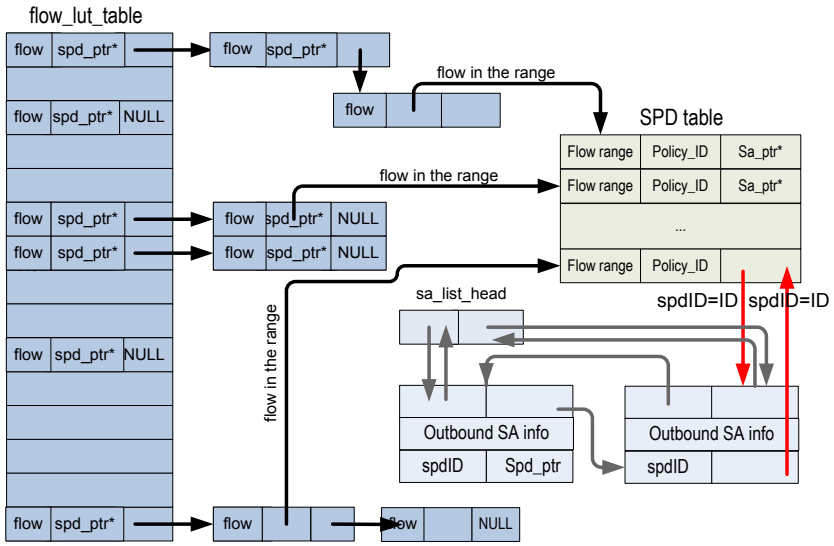

Fig. 2. Data structure and processing flowchart of OUTBOUND

IPsec encrypt stage performs IPsec encryption for packets according to SP. The Output stage places the packet into an output queue and let Tx driver sent it out [9, 10].

The state of a packet is switched from one stage to another. When a stage (except the last one) is finished, it switches its tag, re-enqueued, and de-scheduled. These stages can be grouped into two classes: INBOUND that deals with input flows and OUTBOUND that deals with output ones. They will be detailed in the following parts.

3.1 OUTBOUND design

Fig. 2 illustrates the data structure and flowchart from a five-tuple to a SA, and adding an IPsec header to the original packet.

1. The POW reads a packet and queue it in the flow according to their five-tuple and Tag-tuple.

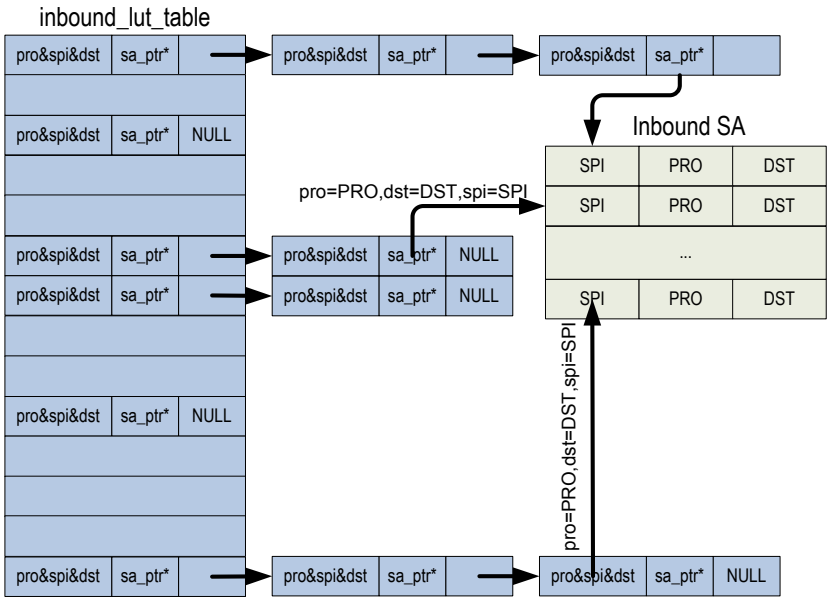

Fig. 3. Data structure and processing flowchart of INBOUND

2. According to the hash(IPsrc, IPdest, IP protocol, srcPort, destPort), the POW looks up the corresponding *flow_lut_table* item, and gets the appropriate spdID in the SPD table. If it does not hit, the POW searches the SPD table to get the correct flow range, and then updates the *flow_lut_table*.
3. According to the spdID, the POW gets the SA information from *SA_list*.
4. Encrypt the packet and append the IPsec header to the packet. Then send the packet out.

3.2 INBOUND design

Fig. 3 illustrates the data structure and flowchart from SPI, destination IP and protocol to a SA, recovering an IPsec packet to the original packet.

1. The POW reads packets and searches the *inbound_lut_table* using the index of SPI&63, destination IP and protocol.
2. According to the tri-tuple arrays <SPI, destination IP, protocol>, the POW gets the SA information in Inbound SA table.

4 Performance Evaluation

4.1 Development Platform and Test Environments

The following experiments are based on a Radisys ATCA-7220 blade, which contains two Cavium OCTEON CN5860 Network Processor[11]. These two processors act as two endpoints of a link, and perform IPsec processing on all the

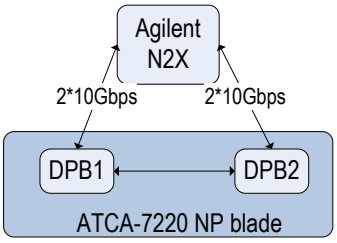

Fig. 4. The testing environment.

packets passing throughput them. They are named DPB1 and DPB2 (Data Processing Block) in the following experiments. In the IPsec processing of DPBs, the plain text is encrypted and signed by the selected SA according to SPD, and the cipher text is decrypted and verified by the selected SA according to the flow table.

An Agilent N2X testing device is used to send out a duplex traffic with 10 Gbps for each direction. The testing topology is shown in Fig. 4.

4.2 Experiment on different encryption algorithms and packet length

In this experiment, a full-duplex traffic of 2×10 Gbps is used. All the 16 cores of each NP are used to achieve the best performance data. The signing algorithm is SHA1, and the schedule mechanism is pipeline. *tag_switch* is performed in order to make the process correctly. Different encryption algorithms including AES128, AES192, AES256, DESCBC and 3DES are tested, and packet lengths from 64 to 1280 are evaluated respectively. The results are shown in Fig. 5.

From this experiment, it can be seen that the longer the packet length is, the better the performance is. This is because there is a fixed work (e.g. five-tuple checking) for processing each packet, which is independent with packet length. So the processing time of each packet is almost the same. But the throughput increase slows down after 512 Bytes, because of the limited port speed.

It seems that various encryption algorithms have almost the same influence on throughput, but AES still has a slightly better throughput value. For different key length of AES, 128 bits seems to have the best performance. And AES is more secure than DES. So we choose AES128 as our encryption algorithm under the condition of using all the 16 cores.

4.3 Experiment on different core numbers

The second experiment also uses full-duplex traffic of 2×10 Gbps. The signing algorithm is SHA1 and the encryption algorithm is AES128. *tag_switch* is performed. The schedule mechanism is pipeline. And in various packet length, the throughput under different core numbers is shown in Fig. 6.

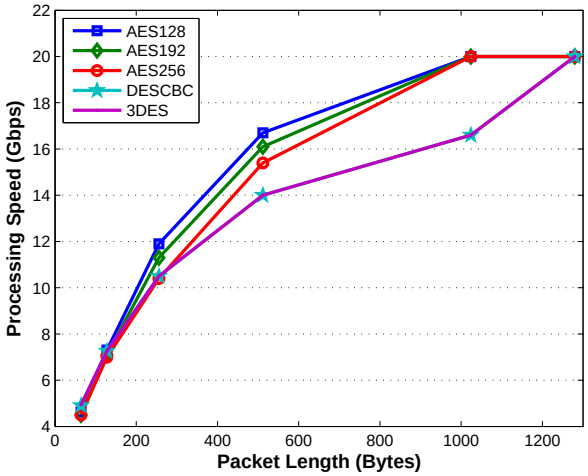

Fig. 5. Throughput of different encryption algorithms and packet length.

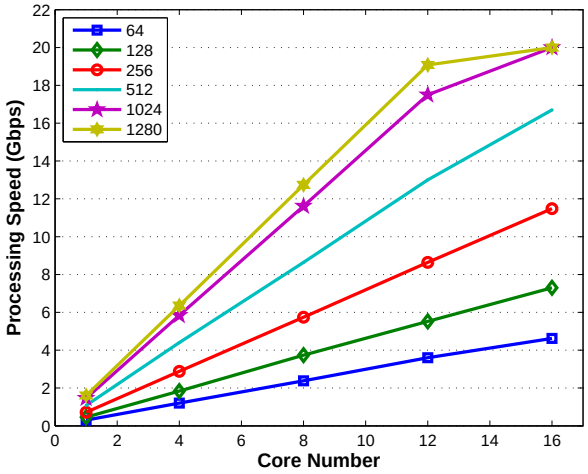

Fig. 6. Throughput of different core numbers.

In Fig. 6, the throughput is almost linear to the core number, indicating a good scalability of this system. This result means there is little correlation between cores, and most of the IPsec steps are parallel, which makes core number not the limit in architecture design. It can be expected that if the core number is further increased, the processing ability can continue to grow, until it reaches a limitation such as memory bus speed or cache speed bottleneck. We hope further experiments to verify this conclusion.

4.4 Experiment on pipeline and run-to-completion mechanism

In this experiment, the core number is set to 16; the traffic is full-duplex with throughput of 2×10 Gbps. The signing algorithm is SHA1 and the encryption algorithm is AES128. The pipeline and run-to-completion schedule schemes are compared with each other under different packet length. The result is shown in Fig. 7.

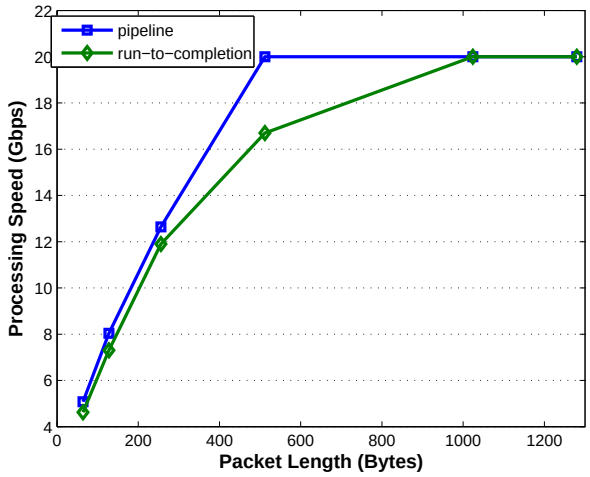

Fig. 7. Throughput of pipeline and run-to-completion mechanism.

This experiment shows that pipeline mechanism is faster than run-to-completion mechanism in performance. The first reason for this phenomenon is that the pipeline can process the packets from the same flow in parallel, so that the utilization of each core can be improved. On the other hand, the run-to-completion mechanism may get stuck while accessing some critical region, leading to a poor utilization of each core, and the flow cannot be processed properly. The second reason is that the instruction data is divided into several small sets in pipeline mechanism. The instruction data may be large during some IPsec operations. Pipeline mechanism divides the instructions and each core only executes some

of the instructions, which increase the cache locality. In contrast, the run-to-completion mechanism can cause a relatively low cache locality and low hit-rate of cache, which leads to a lower throughput. We can come to a conclusion that the distance between pipeline and run-to-completion performance is larger with the growth of packet length. However, the throughput is up to 2×10 Gbps. And it will not increase anymore once the mechanism is no longer bottleneck of the experiments.

5 Conclusion

With one of the industry's most scalable, highest-performance network processor solution for network security applications, Cavium OCTEON CN58XX is used to implement IPsec scheme to achieve 20 Gbps throughput with the packet of length 1024B in full encryption scheme, a record-breaking progress in academic and industrial research. Based on CN58XX, two parallelize mechanisms i.e., run-to-completion and pipeline, are proposed for the implementation of IPsec on Cavium platform, in order to achieve the optimal performance. Extensive experiments are conducted to evaluate the performance of these two mechanisms in real scenarios. The experiment results shows that the pipeline mechanism is the Best Current Practice for IPsec solution in multi-core platform, since it has a better throughput performance than the run-to-completion mechanism. We also give some insightful discussion and analysis of the performance results of these two mechanisms. Besides, the influence of core numbers and algorithms also mentioned in our experiments.

References

1. Michael E. Kounavis, Xiaozhu Kang, Ken Grewal, Mathew Eszenyi, Shay Gueron, David Durham: Encrypting the internet. SIGCOMM 2010: 135-146. (2010)
2. Liu Q.: Study and Implementation on IPsec VPN Gateway Based on Netfilter Mechanism. Master Thesis of Chongqing University. (2009)
3. Cavium Networks: Cavium Networks OCTEON Plus CN58XX Hardware Reference Manual. 221–235.(2008)
4. Cavium Networks: OCTEON Processor Packet Flow. 21–52. (2008)
5. Cavium Networks: OCTEON Technical Presentation. 26. (2007)
6. Intoto Inc.: Virtual Private Network White Paper (2002)
7. RFC 2402: IP Authentication Header (AH) (1998)
8. RFC 2406: IP Encapsulating Security Payload (ESP) (1998)
9. Sang S. L., Sang W. L., Yong S. J. and Ki Y. K.: Implementing High Performance VPN Router using Cavium's CN2560 Security Processor. In: World Academy of Science, Engineering and Technology, vol. 9, pp. 1307–6884. (2005)
10. Cavium Networks: Cavium Networks Announces Industry's First 10Gbps IPsec and SSL PCI-Express Security Accelerators. (2005)
11. Promentum ATCA-7220, `http://www.radisys.com.cn/Products/ATCA/Processing-Modules/Promentum-ATCA-7220.html`