Design of Firewall in Router using Network Processor

Young-Ho Kim*, Jeong-Nyeo Kim*
*Electronics and Telecommunications Research Institute
*{wtowto,jnkim}@etri.re.kr

***Abstract* — Firewall is a fundamental network security technology that protects inner network from unsafe or malicious network traffic outside. The design suggested in the paper is to put this firewall function into a router to support not only packet routing but also network security. The architecture of those integrated function is based on Intel network processor IXP2400 platform to maximize the performance.**

***Keywords* — Firewall, Router, Network Processor, IXP2400, Intel Exchange Architecture, TCAM.**

1. Introduction

Firewall is a fundamental network security technology that protects inner network from unsafe or malicious network traffic outside[20][21][22]. Now-a-days network security appliances including firewall tend to integrate into one device or embed into legacy network devices such as router. Such integration into one device can have more advantages than separated devices for different functions. First of all, from the industrial view point this integrated appliance enables manufactures to reduce the total cost for product which supports both routing and firewall functionality in a single device. Another benefit from the developer's view point, the implementation of one function can make use of the information from the other function to maximize its performance and utilization. But putting two functions simply into one device itself can lead to high complexity and heavy load to the legacy computing unit. Without new hardware assistance, performance degradation is inevitable. These days, as the advanced hardware technology like NP has been emerging it is possible to make integration on the specialized processor without overload[1][2][4][6][7][8].

The main purpose of our design in the paper is to put firewall function in router to support not only packet routing but also network security. To put firewall in the router based on NP platform which is a kind of specialized hardware, our design of firewall is strongly related with NP architecture to maximize the use of NP characteristics and to make firewall operation's overload which might reduce the performance of routing itself never happen.

In section 2, we discuss in detail the architecture of firewall block designed in IXP2400. Section 3 explains the features of firewall which are optimized for IXP2400. Finally, section 4 ends up with conclusion and future work.

2. Architecture

NP technology is an emerging network chip technology that contains mixture of hardware and software characteristics[5]. It consists of multi-processors and high speed internal bus architecture that enables high speed processing of packet. And its programmability by software makes a developing procedure well-maintained and fast time-to-market possible. For these reasons, many network product vendors make various appliances using this NP technology[3].

The target processor our design's aiming for is Intel IXP2400. Intel IXP2400 is the second generation network processor with processing capabilities at OC-48/2.5 Gbps line rates. IXP2400 uses Intel XScale core and eight 32-bit independent multi-threaded microengines for parallel and pipeline processing of packets[16][17].

Figure 1 depicts the processing steps of IPv4 packet forwarding and firewall location in our design. All packets pass through each block which occupies one or more microengines and they are processed according to the policy in each block.

The security function of firewall is working in the middle of packet processing steps in NP microengines. The firewall block designed is located right before IPv4 forwarding block. After finished processing in firewall block, packet can go through the next IPv4 forwarding block to decide its output port for the destination IP address of packet. During the time packet is processed in these steps, only packet that firewall policy permitted can arrive at IPv4 forwarding block and next following blocks[11][12].

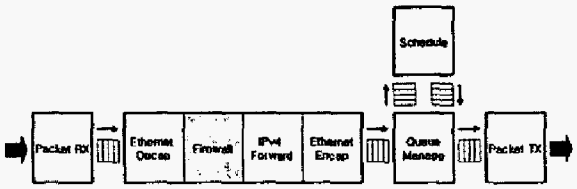

Figure 1. Diagram of processing steps in NP microengines

There are eight microengines inside an IXP2400. With single or cascaded form of IXP2400, a real working platform can be constructed. However, the system assumed in the paper is a reference platform with single IXP2400, Intel IXDP2401 Advanced Development Platform.

There are totally eight microengines in the IXDP2401 because of one IXP2400. All eight microengines of IXP2400 are dispatched to each block shown in Figure 1. To balance the load, more microengines will be allocated for heavy loaded function block. From this rationale, only one microengine is allocated for packet receive(RX) block and three microengines are allocated for IPv4 forwarding block which is the most loaded. Actually the firewall block designed before IPv4 forwarding block does not have separate microengine, but

PAN EX1045
Palo Alto Networks v. Centripetal Networks
IPR2022-00182
Page 00001

shares microengines with IPv4 forwarding block. So total dispatched microengines to firewall block are three and then thirty two threads will be running on these microengines for firewall block because one microengine has eight threads.

Basically, running model of IXP2400 application is same with other NP applications. There are two separated processing paths which are data or fast path and control or slow path. Data or fast path is dedicated for processing packet data which usually requires high speed and repeated computation of limited pattern. On the other hand, control or slow path is for managing control information such as firewall rule update and handling exceptions which never occur quite frequently and it takes long time.

Usually in IXP2400, most processes running on microengines are used for data path while processes running on XScale are for control path.

IXP2400 programming model also uses software framework, so called Intel Internet Exchange Architecture(IXA). This framework is composed of Core Component and Microblock. Core Component is a functional software block running on XScale and implemented as independent module from other Core Component. Microblock is a counterpart of Core Component and resides in microengine. One complete function requires a pair of Core Component and Microblock which means a control path and a data path for processing.

IXA enables developer to create certain application by putting several building blocks of Core Component and Microblock together.

The firewall in the paper is designed according to the philosophy of IXA software framework. Figure 2 shows the Core Component and Microblock of the firewall designed.

Core Component comprises user library and rule management module. In Linux development system, user library is an ordinary user level library while rule management module is a loadable kernel module.

User library gets command from user application and translates it and sends it to rule management module. Then lower level Core Component – rule management module – updates the rules stored in shared memory such as SRAM according to the received command.

Firewall Microblock takes responsibility for processing packet according to the rules specified by firewall Core Component. For instance to filter packet denied, looking up matched rule from the rule DB in the shared memory is done by firewall Microblock. Finally the packet will be disposed by its verdict which might be DROP or ACCEPT.

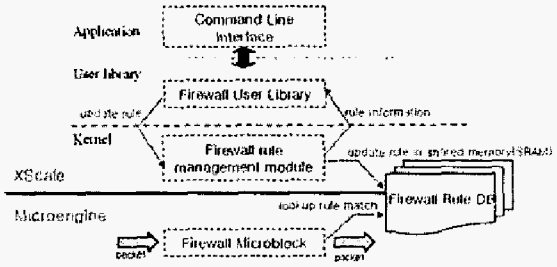

Figure 2. Firewall running on XScale and microengines

Other important operation done in firewall Microblock is stateful packet inspection[24][25]. Stateful inspection is an advanced firewall technique to complement for static packet filtering. To do stateful inspection, all TCP sessions[18] are traced by 5-tuple fields of packet. This operation is also considered as data path.

3. Features

Basically the design of firewall we suggested in this paper has three noticeable features. These features are all related with processor IXP2400 hardware characteristics[9][16][17].

First, the firewall we designed in the paper uses five different types of memory to maximize performance. Table 1 shows the different types of internal memory used in IXP2400. Each memory has different characteristics and limitations. For instance, the available memory with the smallest latency – Local Memory – is only 2560 bytes in size. On the other hand, DRAM takes highest latency and can store massive data. For this reason, DRAM is usually used for temporary packet storage and bulky data buffer.

The firewall designed supports two kinds of access control technique which are static ACL(Access Control List) and stateful packet inspection[24][25]. Static ACL passes or discards packet based on the static firewall policy[22]. Usually firewall policy represents rules stored in memory. If packet comes in, it is compared with these rules and then handled by action of rule hit. So there is a need to look up for a short time whether the packet hits one of firewall rules.

Table 1. properties of Intel IXP2400 memory

Memory	Size in bytes	Approx. latency in clocks
Local Memory	2560(per microengine)	3
Scratchpad	16K(on-chip)	60
SRAM	128M	90
DRAM	1G	120

Among the latest memory technologies, TCAM(Ternary Content Addressable Memory) fits in above case very well. It enables look-up to finish in a constant time and makes a range search possible by using don't care bit. Static ACL of firewall can make use of TCAM for fast look-up operation.

Also stateful inspection of firewall needs state transition table which is necessary for TCP state transition[25]. This transition table requires low latency access because it would be accessed frequently from microengines. Among the internal memories used in IXP2400, Local Memory is optimal for this frequent and low latency access[16].

The second largest memory, SRAM is used for storage of firewall policy which is updated by Core Component running on XScale because it has large space for storage and lower latency than DRAM. Most information shared with Core Component will be implemented in this SRAM.

Likewise, using different types of memory for each purpose makes it possible to optimize the performance and resource.

Second, the firewall uses another hardware component, hash[16]. Originally IXP2400 supports hardware hash unit from 48 bits to 128 bits. Hash unit in IXP2400 creates hash

IPR2022-00182 Page 00002

indices by hard-wired polynomial algorithm and programmable hash multiplier.

As mentioned before, stateful packet inspection traces packet's state information like TCP session. Each session can be identified by 5-tuple packet information which is source IP address, destination IP address, protocol, source port and destination port. We can make 128 bits hash key from 5-tuple fields in the packet and create hash indices by calling primitive 128 bits hash operation.

The last feature is that all operations explained are run by multi-threads. It means that this design of firewall occupies several microengines and uses multithreading technique. It increases overall throughput more than single thread does. Because all threads share same firewall rules and session tables stored in memory, it needs to support a synchronization mechanism to prevent race condition and unpredictable result. Fortunately IXP2400 supports hardware aided atomic test-and-set instruction on Scratchpad memory which can be used to implement mutual exclusion or semaphore. By implementing these synchronization operations, all problems caused by sharing resources can be resolved easily.

4. Conclusion and Future work

The design of firewall suggested in this paper puts security function into router using Intel IXP2400. IXP2400 is a next generation NP technology from Intel to enable such various functions to integrate into one device without sacrificing the performance. The architecture of firewall designed in the paper makes use of the characteristics of IXP2400. Especially according to access pattern of processing, the architecture uses different memory type for different pattern to optimize performance and resources. From the software architecture perspective, the architecture of firewall follows the software framework of Intel IXA to have high portability. This paper also suggests firewall architecture that is composed of three features will help router to be more secure network device without sacrificing performance and eventually contribute to protect information from malicious attackers.

In the near future, we will implement all software designed in the paper and compare the performance with one before integration. Also, finding a bottleneck of processing in the firewall operations will be important future work to do.

REFERENCES

[1] Niraj Shah, "Understanding Network Processors", September 2001.
[2] IBM, "The Network Processor: Enabling Technology for High-Performance Networking", IBM Microelectronics, 1999.
[3] Samuel J. Barnett, Mark R. Fauber, "Network Processors: Uncovering Architectural Approaches for High-Speed Packet Processing", Vitesse, 2001.
[4] Agere Systems, "The Challenge for Next Generation Network Processors", Agere Systems, April 2001.
[5] David Nix, "Using the network processor to mitigate speed vs. programmability tradeoffs", Integrated Communications Design Special Report, Agere Systems, April 2001.
[6] David Husak, "Network Processors: A Definition and Comparison", C-Port, 2000.
[7] David Husak, Robert Gohn, "Network Processor Programming Models: The Key to Achieving Faster Time-to-Market and Extending Product Life", C-Port, May 2001.
[8] Nicholas Cravotta, "Network Processors: The Sky's the Limit", EDN Magazine, November 1999.
[9] Erik J. Johnson, Aaron R. Kunze, IXP2400/2800 Programming: The Complete Microengine Coding Guide, Intel Press, April 2003.
[10] Bill Carlson, Intel Internet Exchange Architecture and Applications: A Practical Guide to Intel's Network Processors, Intel Press, June 2003.
[11] Intel Internet Exchange Architecture Software Building Blocks Developer's Manual, Intel, November 2004.
[12] Intel Internet Exchange Architecture Software Building Blocks Reference Manual, Intel, November 2004.
[13] Intel Internet Exchange Architecture Portability Framework Developer's Manual, Intel, November 2004.
[14] Intel Internet Exchange Architecture Portability Framework Reference Manual, Intel, November 2004.
[15] Intel Internet Exchange Architecture Software Building Blocks Applications Design Guide, Intel, November 2004.
[16] Intel IXP2400 Network Processor Hardware Reference Manual, Intel, October 2004.
[17] Intel IXDP2401 Advanced Development Platform System User's Manual, Intel, July 2004.
[18] Gary R. Wright and W. Richard Stevens, TCP/IP Illustrated, Addison-Wesley, 1995.
[19] Douglas E. Comer, Ralph E. Droms, Computer Networks and Internets, Prentice Hall, July 2003.
[20] Steven M. Bellovin, William R. Cheswick, "Network Firewalls", IEEE Communications Magazine, September 1994.
[21] Wolfgang Weber, "Firewall Basics", TELSIKS'99, October 1999.
[22] Robert Zalenski, "Firewall Technologies", March 2002.
[23] Steven McCanne, Van Jacobson, "The BSD Packet Filter: A New Architecture for User-level Packet Capture", December 1992.
[24] Check Point, "Stateful Inspection Technology", Check Point, 1999.
[25] http://www.netfilter.org