Design and Implementation of Content Switching Network Processor and Scalable Switch Fabric

You-Sung Chang, Ju-Hwan Yi, Hun-Seung Oh*, Seung-Wang Lee, Moo-Kyung Kang, Jung-Bum Chun, Jun-Hee Lee, Jin-Seok Kim, Sang-Ho Kim, Hee-Jae Jung*, Il-Sung Hong, Yong-Hwan Kim, Yu-Sik Lee, and Chong-Min Kyung

Abstract— **This paper proposes a network processor especially optimized for content switching. With 2Gbps port capability, it integrates packet processor cluster, content-based classification engine and traffic manager on a single chip. A switch fabric architecture is also designed for scale-up of the network processor's capability over hundreds gigabit bandwidth. Applied in real network systems, the network processor shows wire-speed network address translator (NAT) and content-based switching performance.**

Index Terms— **Content Switching, Gigabit Ethernet, Network Processor, Switch Fabric, Wire-line Communication**

I. INTRODUCTION

AS the importance of digital communication networks increases, so do the opportunities and challenges in network component design. Recently, the network processor has emerged to satisfy all of the ever-increasing performance, flexibility and economy requirements, and the networking industry began to opt for the new item to build products of the next generation. The application covers IP routing, voice and data convergence, VPNs, QoS, and all digital communication areas.

In addition to the conventional application, content-based switching has arisen to support quality services, such as server load balancing and security. It needs a completely different type of classification mechanism in which the object field has a variable-length and unfixed location. The characteristic makes the classifiers scan the whole packet payload to search for expected string patterns. Software approaches with the general-purpose processors appear to never meet performance requirements of high-speed network while using external co-processors suffers from cost and efficiency problems.

In this paper, a network processor architecture is defined for real-time content-based switching as well as a companion switch fabric IC. To integrate enough computing resources on a single chip, an area-efficient structure for string matching is developed.

II. CONTENT SWITCHING NETWORK PROCESSOR

The content switching network processor has been designed for the worst-case wire-speed performance while preserving the high programmability of general purpose processors. To achieve the goal, object operations are partitioned into soft-programmed packet processor cluster and hard-wired private processing engines. The hard-wired engines include real-time pattern matching engine and traffic manager supporting

Manuscript received November 5, 2003; revised November 26, 2003.

You-Sung Chang was with Korea Advanced Institute of Science and Technology (KAIST), Taejon, Korea, and is now with Paion Co. Ltd., 4th FL Mirae Asset Venture Tower, 996-1 Daechi-Dong Kangnam-Gu, Seoul, 135-280, Korea

TEL : +82-2-3453-8250(ext.114), FAX : +82-2-3453-8281

E-mail: caviar@ieee.org

Chong-Min Kyung is with Department of Electrical Engineering. Korea Advanced Institute of Science and Technology (KAIST), Taejon, Korea

E-mail: kyung@ee.kaist.ac.kr

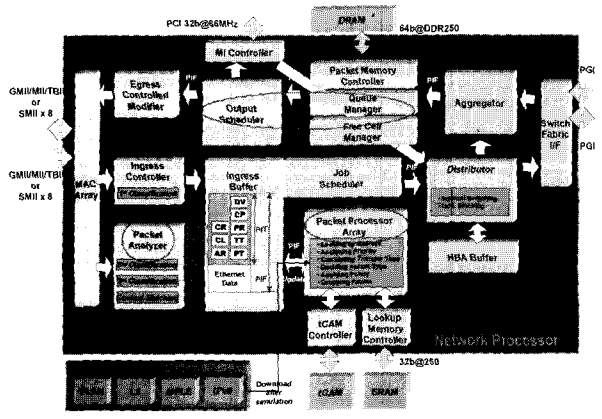


Fig. 1. Chip architecture of the content switching network processor; Dual packet analyzer and packet processor array comprise the main computing resources of the network processor.

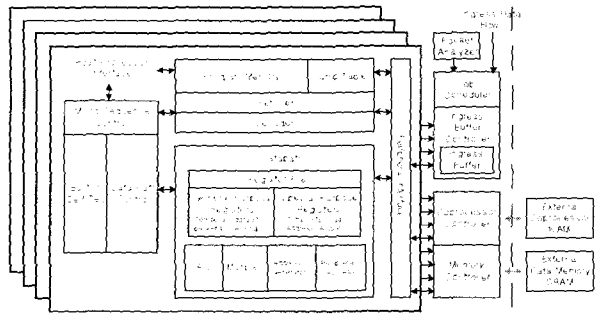


Fig. 2. Block diagram of the packet processor array; It consists of four individual packet processors tightly coupled each other through peripheral controllers.

two Gigabit Ethernet flows. Fig. 1 shows a block diagram and internal data flow of the chip.

Four packet processors are clustered by job scheduler to form the decision center of the chip. Block diagram of the packet processor is shown in Fig. 2. Each packet processor can execute five instructions - control transfer, condition evaluation, address generation, peripheral access and general function commands in a single cycle. As for control transfer, it provides two operational modes of static branch prediction and sequential condition-and-branch. The sequential mode executes condition evaluation and control transfer in a pipelined manner and is effectively applied to series of conditional branches. The control transfer mode can be dynamically switched during program execution. Regarding functional operations, the packet processor implements bit alignment, hash function, 32b immediate constant, packet generation for network data manipulation, 12x8 multiplication for bandwidth calculation, and atomic read-and-write operations for inter-processor communication.

To minimize memory and peripheral access overhead which occupies a large portion of packet processing time, each individual packet processor has the capability to issue multiple memory accesses in a pipelined manner without blocking. Data-path controller in the packet processor intelligently checks whether peripheral access data is ready or not at the point of use. Pipeline stalls if and only if the requested data to be used as an operand is not yet fetched. The memory and peripheral access performance is maximized with parallel execution of address generation, peripheral access and general function.

In interfacing the packet processor cluster to peripherals, two techniques are applied to reduce computing overhead of the packet processors. First, jump table enables direct control transfer to a destination routine according to the classification result from job scheduler. It removes the classification branch overhead of up to 60 cycles. Second, the ingress buffer, which temporarily stores packets under processing, supports unaligned word (2B) or double word (4B) accesses of packet processors. The single-cycle unaligned access leads to 5% computing cycle reduction for IP switching in average.

The number of integrated packet processors came from the performance evaluation reports obtained in the firmware development environment using co-developed protocol library. The firmware images are prepared by maximally binding protocol library routines including

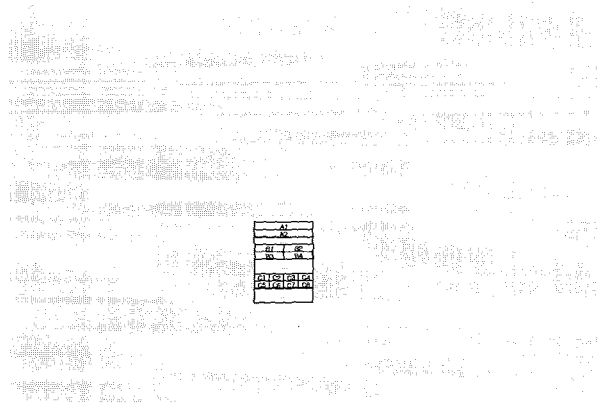


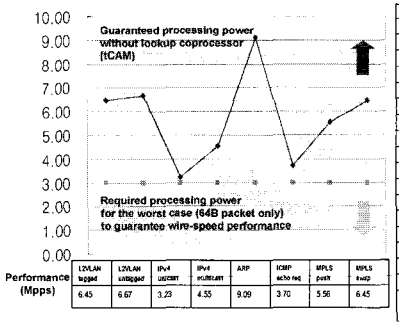
Fig. 3. Simplified control flow in the packet analyzer; Packet analyzer performs content-based classification using the dedicated pattern matching engine in addition to the conventional protocol-based classification engine.

optional context matching, packet mirroring, and header/payload checksum validation routines. The evaluation result is shown in Fig. 4. It presents that the current implementation of four packet processors guarantees at least 3.23 Mpps (test average 5.71 Mpps) while the computing power requirement is 2.98 Mpps for wire-speed guarantee in the worst case.

B. Packet Analyzer

The packet analyzer includes basic packet classification and real-time pattern matching engines to enable wire-speed L4-7 switching and security operations. The packet analyzer and its operations are depicted in Fig. 3. The packet classification engine classifies packets into 256 classes based on the L3-4 header information and the pattern matching engine performs variable-sized pattern matching with about 1K rules (assuming average 10-byte patterns). Classification results are summarized by decision logic. When a packet matches multiple rules, the decision logic selects the rule of the highest priority. The priority is encoded in location. All of the information is also delivered to packet processors and the decision is extended further.

To achieve the wire-speed performance, the pattern matching is based on a finite state machine using embedded SRAM that enables cycle-by-cycle state transition [1]. The pattern matching engine comprises memory blocks to implement PT (prefix table) and FSM (finite state machine). The prefix table stores the prefixes of the rules to reduce the FSM size and generates the destination state address for the matched prefix every cycle. The prefix table is partitioned into a number of



Firmware Images	# of Inst.
LLC_SNAP	103
ARP_SRAM	176
ARP_TCAM	160
ICMPv4	456
L2_SRAM	417
L2_TCAM	416
L2_VLAN_SRAM	726
L2_VLAN_TCAM	780
IPX_SRAM	412
IPX_TCAM	415
MPLS_SRAM	512
MPLS_TCAM	496
IPv4_SRAM	807
IPv4_TCAM	682
IPv4_MPLS_SRAM	896
IPv4_MPLS_TCAM	761
IPv6_TCAM	1319
QoS (flow control, rate limiting)	62

Fig. 4. Packet processing performance for various type of processing that is implemented in firmware and the number of instructions in the reference firmware images

memory blocks for pipelined binary range search. Main FSM contains a comparator block and a state memory. The state memory consists of two memory blocks, a branching state memory and a non-branching state memory. The branching state memory stores the states of multiple branches, while a scheme for word sharing among multiple states with a small number of branches is devised for more efficient memory utilization. PT, FSM, and L3-5 classification results are called upon to make the final classification decision.

Because of the rectangular structure of memories, a memory width is determined by the maximum number of branches, and the mapping of a state to a memory word of the maximum width leads to the waste of memory space. The use of prefix table reduces the area to one sixth. According to the simulation result of Fig. 5, the further area reduction only by increasing the lookup length of prefix is limited to about three fourth of that of using 1B prefix table. However, the combined use of the word sharing scheme enables area reduction to one twentieth, furthermore.

L7 parsing performed by the packet analyzer takes around 40 clock cycles from the time the packet enters the system to the time the associated data is returned. In the L7 parsing, the search of URL, cookie and CGI rules is done in parallel. In terms of area utilization, the result of architectural exploration reports about 3,000 rules can be compiled into a 116 KB SRAM with the applied area reduction technique.

C. Scheduler, Bandwidth Filter, and Peripheral I/Fs

The traffic manager composed of output scheduler and queue manager implements 8-level priority weighted fair queuing (WFQ), weighted random early detection (WRED) and bandwidth filtering with the help of packet processors. Complex operations like flow identification, per-flow bandwidth allocation and transfer time calculation are performed by packet processors. The information is stored in the internal packet header encapsulating the original packet and delivered with the packet data. The traffic manager makes the decision of packet drop by monitoring the bandwidth occupation of the packet flow. For burst traffic compensation, the bandwidth occupation takes the average over a controllable time window. The composite job

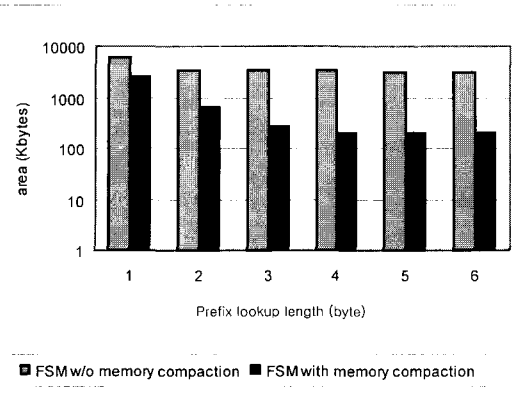


Fig. 5. Required area for context matching; the area comparison shows that the proposed memory compaction scheme, i.e., the word sharing with the prefix table has a significant effect on area reduction.

distribution over functional units successfully works in a real system implementation to be shown in Section V and the controlled bandwidth error is reported to be less than 0.07% for varying input stream bandwidth.

As for an external interface, the network processor engages dedicated switch fabric channels, PGI, for direct connection to companion switch fabrics or network processors. The two 2Gbps channels enable 6-gigabit-port extension with three network processors and support up to 512 Gigabit/Fast Ethernet ports through switch fabric. The extension channel has separate backward signals to back-propagate output port status information to optimize the routing across switch fabric. The information is used to reorder packets to avoid the performance degradation due to head-of-line (HOL) blocking. For management interface, it uses a packet based approach. Through PCI, a control packet of multiple read/write commands can be written and unicasted or multicasted to destination devices. The packet-based management approach significantly reduces the table configuration time.

III. Scalable Switching Fabric Interconnecting Network Processors

To extend the network processors capability over tens and hundreds Gbps bandwidth, a scalable, companion switch fabric is designed. The design effort is focused on the self-configuring extension (supporting up to 512 ports) without any separate supervisors. The separate supervisor monitoring overall packet routing increases the total system complexity and costs. Moreover, it cannot be applied to a flexible topology. Thus, distributed processing system is chosen in which the overall path information is distributed among constituent switch fabric IC's and network processors. In the distributed system, each component needs to know global path information to find the best path for each packet routing. For this purpose, a state back-processing algorithm has been devised in the switch fabric.

The switch fabric performs 16x16 protocol-independent switching. Each port supports 2Gbps data flow. As depicted in Fig. 6, it consists of data-path and control blocks. The input and output buffer are used to align data in front of and behind the shared memory. The fabric router, the queue manager and the free cell router cooperates to control the overall shared-memory switching. The fabric router decides the destination ports of a packet based on the packet and the global system information. The fabric router also implements the weighted random early drop (WRED) for flow control. The fabric router monitors the depth of per-priority output queues and probabilistically drops packets according to the information. The free cell manager allocates a space for a packet in the shared memory and the queue manager manages a linked list of packets for each output port and schedules the packets. The switch fabric supports variable-size packet up to 9KB jumbo packet.

Fig. 6. Chip architecture of the switch fabric that enables flexible routing and continuous back-propagation of global path information across the chip; Management interface are omitted for simplicity.

The fabric router has a programmable switching table inside to give a freedom in the topology of switch fabric extension using multiple chips. The switching table consists of four sub-tables. The first one contains destination port information for each case of unicast and multicast routing, and the second one has loop resolution information to avoid loop in multicast routing. The last two sub-tables store information regarding trunk configuration and packet flows. In the switch fabric, adjacent two, four, or eight ports can be aggregated for a larger bandwidth. Using multiple ports as an aggregated port may cause disordering of sequential packet stream and obviate the link invariance condition of L2 switching. To resolve the problem, packet flows are identified and the fabric router guarantees the same path for the packets in the same flow. The switching table can be updated dynamically during the operation.

Path analyzer plays the key role in the global path information sharing across the switch fabric. In the distributed system, each device gathers global information through conversation with neighbors. Actually, the switch fabric implements one-way backward conversation as shown in Fig. 7. The path analyzer builds path information from the back-propagated path information and the local queue state information. It combines the both information into the path information through the switch fabric itself and propagates it backward, again. The back-propagated path information is ultimately used in network processors to avoid head-of-line (HOL) blocking and also used in the switch fabric, itself, for optimal path selection. Normally, the overall path information is too large to be transmitted to the neighbor devices in a reasonable time. The long propagation time can make the information invalid at the time of switching decision. To avoid the problem, the differences between the previous and the current information are transmitted.

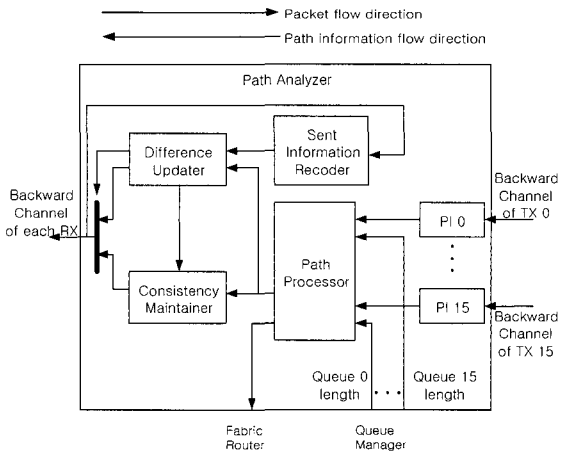


Fig. 7. Path information processing in the switch fabric; Path processor updates the global path information with the queuing information of the current chip.

IV. Chip Implementation

The network processor was implemented in 0.18μm CMOS 1P6M process technology. It integrates 28M transistors in 13 x 13 mm^2. The die-photo is shown in Fig. 8, and its physical specification is summarized in Table I. The companion switch fabric IC was implemented using the same technology. Its die size is 13 x 10 mm^2. Its die-photo and specification are shown in Fig. 9 and Table II.

Both chips are working at 250MHz. For the high frequency operation, signal integrity analysis was performed with the timing analysis. The crosstalk noise was reported up to 54.6% by the Celestry NDC-SI tool. In addition, IR-drop was controlled under the 10% of supply voltage. For fast timing closure, clock skew was intentionally adjusted to fix timing violations less than 5% of the clock period.

V. Reference System Design

Fig. 10 shows the reference system using the two context switching network processors. One network processor is configured with 1x GMII and 8x SMII ports and the other is with 1 x TBI and 8x SMII ports. The two network processors are directly interconnected through two 2Gbps PGI channels. 8MB SRAM for lookup tables and 128MB DRAM for packet buffer are used for each processor, and tCAM is shared by the two network processors. To build a complete working system, the main board is connected to host CPU board through back plane as shown in Fig. 11. The CPU board is engaged for management of the system as a console.

For L2/3 forwarding test, the NetCom's SmartBits-6000 equipment are used for the performance

measurement. 100% throughput and 0% packet loss are observed for the forwarding testing according to the

Table 1. chip Specification of the Content Switching Network Processor

Die Size	13 mm x 13 mm
Transistors	28M
Process	0.18 μm CMOS 1P6M
Supply Voltage	1.8V Core, 3.3V I/O, 2.5V SSTL I/O
Clock Frequency	250MHz
Clock Domains	53
Power Dissipation	8W*
Package	901pin HSBGA

* varies according to system configuration.

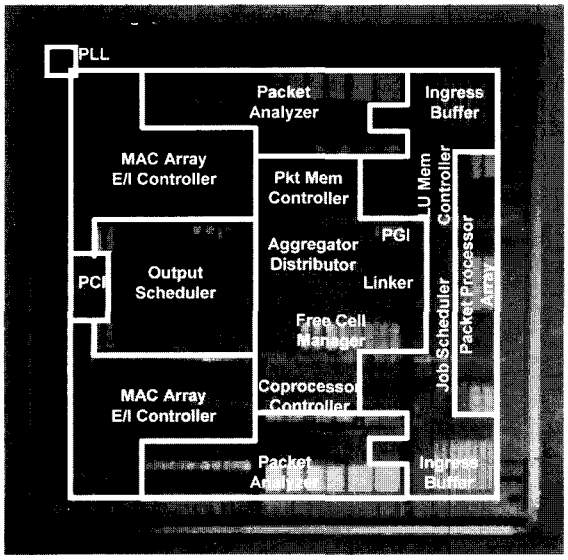


Fig. 8. Die micrograph of the content switching network processor; Capable of two GMII/TBI (Gigabit Ethernet) and two 2Gbps switch fabric ports

Table 2. chip Specification of the 32Gbps Scalable Switch Fabric

Die Size	13 mm x 10 mm
Transistors	14M
Process	0.18 μm CMOS 1P6M
Supply Voltage	1.8V Core, 3.3V I/O, 2.5V SSTL I/O
Clock Frequency	250MHz
Clock Domains	38
Power Dissipation	6W*
Package	901pin HSBGA

* varies according to system configuration.

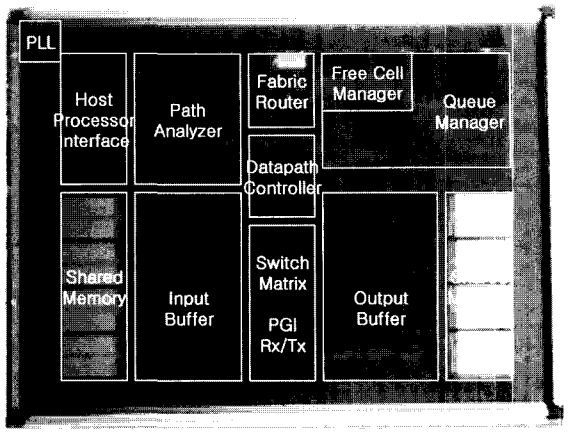


Fig. 9. Die micrograph of the 32Gbps scalable switch fabric; 16 bidirectional channels with 2Gbps data rate

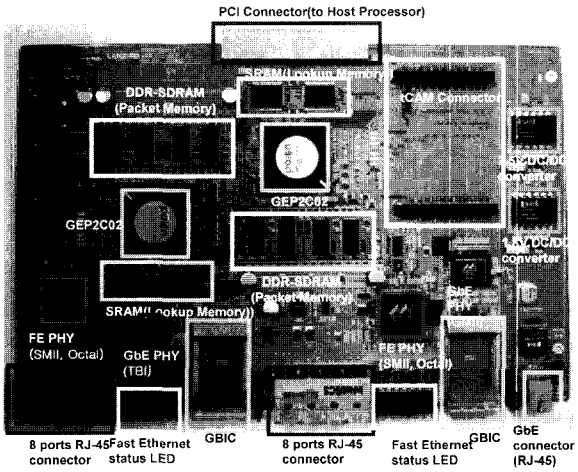


Fig. 10. Reference system board using two content switching network processors (part named by GEP2C02); Two Gigabit Ethernet ports and 16 Fast Ethernet ports are configured for the whole feature testing of the network processor.

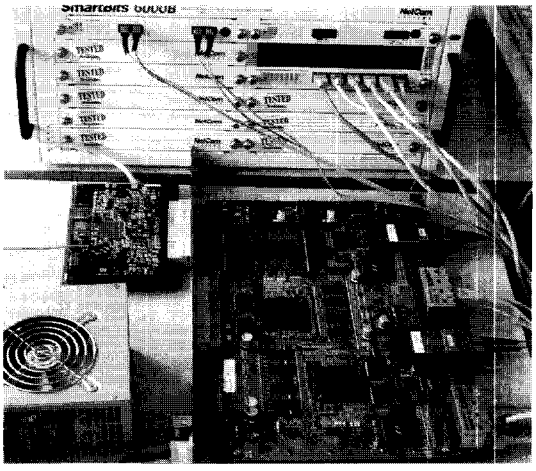


Fig. 11. Performance test environment including the target reference system, the management CPU board, and the test equipment for network system

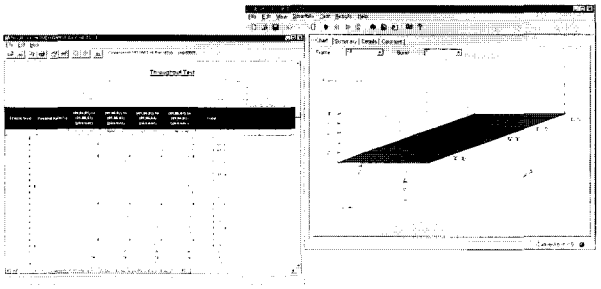


Fig. 12. Results of basic throughput and packet loss test; 100% throughput and 0 packet loss are reported for all packet lengths in L2/3 switching of the reference system.

NETCOM SYSTEMS SmartBits Throughput test results

Vendor Name: Vendor
Product Name: Product
Software Version: SmartApplications V 2.32
Library Version: 3.09 53
Firmware Version: 1.20.20...
Serial Number: 60010125
Throughput test length: 10 seconds
Average of: 1 trial
Port pairs active: 1
Mode: Uni direction
Date: Thu Sep 18 10:32:22 2003

Port Pair Throughput

Frame size	64	128	256	512	1024	1280	1518
1 Gb Max Rate	1488095	844595	452899	234962	119732	96154	81274
Avg % passed	100.00	100.00	100.00	100.00	100.00	100.00	100.00
Avg Tx Time(s)	10.000	10.000	10.000	10.000	10.000	10.000	10.000
(01,01,02) to (01,01,01)	1488095	844595	452899	234962	119732	96154	81274

Maximum Port Pair Throughput with no loss as percent of maximum

Frame size	64	128	256	512	1024	1280	1518
(01,01,02) to (01,01,01)	100.00	100.00	100.00	100.00	100.00	100.00	100.00

reports as captured in Fig 12. In addition to the basic forwarding test, various application systems such as network address translator (NAT), intrusion detection system (IDS), and rate limiting router were implemented using the reference system and proven for wire-speed performance in the application to Gigabit Ethernet.

VI. Conclusion

The content switching network processor integrates real-time pattern matching engine, traffic manager and packet processor cluster in a single chip, supporting 2Gbps input stream. A scalable switch fabric is also designed for extension of the network processor. Through real system implementation, it is proven that the network processor is capable of wire-speed performance for Gigabit Ethernet. Under the NAT configuration, application tests such as FTP, telnet and etc. have been successfully passed, and 100% throughput was measured by the SmartBits-6000 equipment for Gigabit connection.

Appendix

A snapshot of a throughput test report for the Gigabit NAT System which is obtained from SmartBits-6000;

Acknowledgment

Special thanks are given to Dae-Hwan Kim, Sang-Hun Lee, Ki-Yong Ahn, Byung-Woon Kim, for their help in functional verification, and Dr. Dong-Ho Cho for lots of constructive comments on architectural design.

References

[1] H. S. Oh, Y. S. Chang, C. M. Kyung, "An Area-Efficient Architecture for Content-Based, Real-Time Packet Classification", submitted to IEEE Tr. on VLSI.

[2] Iyer, S., Kompella, R., and Shelat, A.: 'ClassiPI: An architecture of fast and flexible packet classification'. IEEE Network, vol. 15, pp. 33-41, Mar./Apr. 2001

[3] F. M. Chiussi and A. Francini, "Scalable electronic packet switches", Selected Areas in Communications, IEEE Journal on, Vol 21, Issue 4, May 2003

[4] GEP2C02 Network Processor Datasheet, June, 2003.

[5] GES0016 Switch Fabric Datasheet, June, 2003.

You-Sung Chang (M'02) received the B.S., M.S., and Ph.D. degrees in electrical engineering from the Korea Advanced Institute of Science and Technology (KAIST), Taejon, Korea, in 1994, 1996, and 2001, respectively. He co-founded PAION, Co., Ltd. in 1999, and currently leading a project of developing network processor and switch fabric IC's. His research interests include microprocessor and DSP architecture, digital communication IC design, and design verification methodology.

Ju-Hwan Yi received the B.S., and M.S. degrees in electrical engineering from the Korea Advanced Institute of Science and Technology (KAIST), Taejon, Korea, in 1994 and 1996, respectively. His research interest covers functional and formal verification of VLSI design.

Hun-Seung Oh received the B.S., and M.S. degrees in electrical engineering from the Korea Advanced Institute of Science and Technology (KAIST), Taejon, Korea, in 1996 and 1998, respectively.

Seung-Wang Lee received the B.S., and M.S. degrees in electrical engineering from the Korea Advanced Institute of Science and Technology (KAIST), Taejon, Korea, in 1997 and 1999, respectively.

Moo-Kyung Kang received the B.S., and M.S. degrees in electrical engineering from the Korea Advanced Institute of Science and Technology (KAIST), Taejon, Korea, in 1997 and 1999, respectively.

Jung-Bum Chun received the B.S. degree in electronic engineering from Sogang University, Korea, in 1998 and M.S. degrees in electrical engineering from the Korea Advanced Institute of Science and Technology (KAIST), Taejon, Korea, in 2000.

Jun-Hee Lee received the B.S. degree in electronic engineering from Kyoungpook National University, Korea, in 1998 and M.S. degrees in electrical engineering from the Korea Advanced Institute of Science and Technology (KAIST), Taejon, Korea, in 2000.

Jin-Seok Kim received the B.S., and M.S. degrees in electrical engineering from the Korea Advanced Institute of Science and Technology (KAIST), Taejon, Korea, in 2000 and 2002, respectively.

Sang-Ho Kim received the B.S. degree in electronic engineering from Dankook University, Korea, in 2000 and M.S. degrees in electrical engineering from the Korea Advanced Institute of Science and Technology (KAIST), Taejon, Korea, in 2003.

Hee-Jae Jung received the B.S. degree in electronic engineering from Yonsei University, Korea, in 2000 and M.S. degrees in electrical engineering from the Korea Advanced Institute of Science and Technology (KAIST), Taejon, Korea, in 2003.

Il-Sung Hong received the B.S., and M.S. degrees in electrical engineering from the Seoul National University, Seoul, Korea, in 1999 and 2001, respectively.

Yong-Hwan Kim received the B.S. degree in electronic engineering from Kyoungpook National University, Korea, in 1994. He worked at LG semiconductor, Seoul, Korea from July 1993 to June 2000 and joined with Paion, October 2000.

Yu-Sik Lee received the B.S. degree in electronic engineering from Kyoungpook National University, Korea, in 1994. He started his career at Hyundai electronics in 1994, and went through LG semiconductor, Hynix, and Vision Telecom. He joined with Paion at 2001.

Chong-Min Kyung (SM'99) received the B.S. degree in electronic engineering from Seoul National University, Korea, in 1975, and the M.S. and Ph.D. degrees in electrical engineering from the Korea Advanced Institute of Science and Technology (KAIST), Korea, in 1977 and 1981, respectively. After graduation from KAIST, he worked at AT\&T Bell Laboratories, Murray Hill, NJ, from April 1981 to January 1983 in the area of semiconductor device and process simulation. In February 1983, he joined the Department of Electrical Engineering at KAIST, where he is now a Professor. During 1993-1994, he served as the Asian Representative in the ICCAD (International Conference on Computer-Aided Design) executive committee. He also served as Vice Chairman of the 1999 COOLChips II held in Kyoto, Japan, and as Cochair of the program committee of ASP-DAC (Asia and South Pacific Design Automation Conference) 2000. He received the Most Excellent Design Award, and Special Feature Award in the University Design Contest in the ASP-DAC 1997 and 1998, respectively. He received the Best Paper Award in the 36th DAC (Design Automation Conference) held in New Orleans, LA, the 10th ICSPAT (International Conference on Signal Processing Application and Technology), Orlando, FL, in September 1999, and the 1999 ICCD (International Conference on Computer Design), Austin, TX.