Analysis of 10 Gigabit Ethernet using Hardware Engine for Performance Tuning on Long Fat-pipe Network

Takeshi YOSHINO, Junji TAMATSUKURI, Katsushi INAGAMI,
Yutaka SUGAWARA, Mary INABA and Kei HIRAKI
University of Tokyo
7-3-1 Hongo, Bunkyo-ku, Tokyo, 113-8656, Japan
{ysn, junji, inagami, sugawara, mary, hiraki}@is.s.u-tokyo.ac.jp

***Abstract*— With the rapid progress of network technology, several OC-192c lines are settled across the Pacific Ocean and the Atlantic Ocean and now 10Gbps network is ready for round-the-world communication. However, it is difficult to use the bandwidth efficiently by TCP/IP communication.**

We develop programmable 10Gbps wire-rate packet analyzer TAPEE (Traffic Analysis Precise Enhancement Engine), which aims to collect packet header logs with the 10Gbps wire-rate speed adding time stamp. This paper shows our design and implementation of TAPEE, then its experimental observations; distribution of data and ACK packets and its visualization for (a) comparison of real LFN and pseudo LFN with artificial long latency, (b) comparison of with and without hardware support for TCP. Using these observations, we tuned the performance, on the real LFN with 500 ms RTT, we attained 8.8049Gbps of 9.28Gbps WAN PHY bottleneck with TOE (TCP Offload Engine), and 6.96Gbps of 8.6Gbps PCI-X 1.0 bottleneck without TOE.

I. Introduction

With the rapid progress of network technology, such as optical communication and network switches, backbone networks have rapidly expanded and become fatter.

For backbone network infrastructure, new technologies such as WAN PHY and WDM are spreading, which realize very long-distance Ethernet networks. In addition, 10 Gigabit Ethernet (10GbE) Network Interface Card (NIC) became common, and end-to-end 10Gbps communication is ready to be used.

TCP/IP is the standard protocol for end-to-end reliable communication. It is well known that TCP/IP has a limitation on performance for transfer over Long Fat-pipe Network (LFN), and it is difficult to utilize bandwidth efficiently. One reason is congestion control algorithm of TCP, which controls transmission rate by changing window buffer size of each stream according to acknowledgement packets (ACK). When we use TCP/IP on LFN, it needs large window size proportional to Round Trip Time (RTT) and speed of the window size growth is inverse proportional to RTT.

However other than these fundamental problems, there are strange phenomena which cannot be explained by TCP congestion control algorithm. For example, as for the parallel stream data transfer, throughput of each stream with exactly same condition differs a lot and sometimes communication with GbE NIC is slower than with Fast Ethernet [1].

To tackle with this problem, in 2002, we developed a packet analyzer for 1Gbps DR Giga-Analyzer which collects packet log with full wire-rate, adding GPS timestamp with 100ns accuracy [2]. For DR Giga-Analyzer, we used a programmable network processor Comet-NP [3] which consists of Finite State Machine (FSM) engine and State Transition Table (STT). DR Giga-Analyzer suggested us that bursty behavior of GbE induces microscopic congestion which may cause performance decrease.

Now, several OC-192c lines are settled across the Pacific Ocean and the Atlantic Ocean, and 10Gbps networks are ready for round-the-world communication. Here, we again meet strange phenomena. For example, we were troubled by the periodical performance decrease for every 5 minute. For another example, we examined TCP/IP data transfer using Tokyo – Seattle – Tokyo round trip path where data and ACK share the same fiber, and Tokyo – Seattle – Chicago – Tokyo ring path where data and ACK use the fibers of opposite directions exclusively. We have an impression that performance tuning of the round trip path is easier than that of the ring path.

To clarify these phenomena, we designed and

implemented programmable 10Gbps wire-rate packet analyzer TAPEE (Traffic Analysis Precise Enhancement Engine). Since it is not easy to store full dump of the packets from the view point of both storage speed and storage capacity, TAPEE aims to collect packet heads including TCP/IP header information at 10Gbps wire-rate with precise timestamp attached to them. To process network packets at wire-rate and, at the same time, to keep programmability and flexibility, cooperation of hardware and software is essential. TAPEE consists of a hardware engine to collect and forward packet heads, a PC server to store, analyze, and visualize the packet log, and network equipments like Ethernet and an optical tap. The hardware engine is implemented on FPGA-based programmable network testbed TGNLE-1, therefore, TAPEE is applicable to analysis of upper layer protocols such as iSCSI, with which multiple iSCSI packets may be encapsulated into one TCP datagram and some packet may start from arbitrary offset in TCP datagram and single packet may lie over 2 TCP datagrams.

This paper describes TAPEE and its experimental observations; distribution of data and ACK packets and its visualization for (a) comparison of TCP/IP transfer with and without hardware, such as TCP Offload Engine (TOE), and (b) comparison of real LFN and pseudo LFN with artificial long latency. We found that on real LFN, intervals of ACK packets change a lot while passing several intermediate switches, which affects performance especially for TCP without TOE. Using this observation, we tuned the hosts, on the real LFN with 500 ms RTT, and we attained 9Gbps of 9.28Gbps WAN PHY bottleneck with TOE, and 7.1Gbps of 8.6Gbps PCI-X 1.0 bottleneck without TOE.

In chapter II, we describe development platform TGNLE-1 and TAPEE. In chapter III, we practice analysis and problem detection using TAPEE to demonstrate utility of it. In chapter IV, we describe our current achievement on TCP/IP tuning on LFN. In chapter V, we discuss difference and relation to other works, and in chapter VI, we conclude.

II. Implementation of TAPEE

A. 10GbE Testbed TGNLE-1

We used 10GbE testbed TGNLE-1 [4] for implementation of TAPEE's hardware engine. TGNLE-1 is FPGA based reconfigurable 10GbE packet processing equipment, and it has a form of 1U chassis. Fig. 1 shows components of TGNLE-1.

TGNLE-1 has two 10GBASE-LR interfaces consist of 300pin MSA and Intel IXF18104 10 Gigabit Ethernet LAN PHY chip. TGNLE-1 has two FPGAs

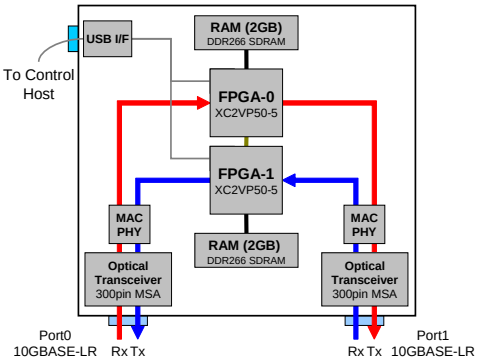

Fig. 1. TGNLE-1 Block Diagram

and each of them is connected to these two 10GbE interfaces. These two FPGAs are Xilinx's Virtex-II Pro XC2VP50. The interface between FPGA and IXF18104 is SPI-4 Phase 2. On FPGA, we used Xilinx's SPI Sink/Source IP Core.

Traffic flow is as follows. Packets received from port0 Rx go to FPGA-0. In FPGA-0, packets are processed by the function user implemented. Packets output from FPGA-0 go to port1 Tx and are transmitted. This flow is shown by red arrow in Fig. 1. In the opposite flow, packets received from port1 Rx are processed by FPGA-1 and transmitted from port0 Tx. This flow is shown by blue arrow.

TGNLE-1 has one USB interface connected to flash ROMs and FPGAs. Functions are downloaded into FPGAs with this USB interface. User can communicate with FPGAs by control host for reading and writing registers to configure function parameters. Each FPGA has one 2GB DDR266 SDRAM which is useful for implementing buffer for memory-based packet logging, delay emulation, packet pacing, and so on. For implementation of TAPEE, we did not use these DRAMs. We programmed a function of packet capturing support on TGNLE-1 which works as a part of TAPEE.

B. TAPEE

TAPEE is a packet capturing system by coordination of commodity PCs and FPGA-based equipment. TAPEE consists of TGNLE-1 programmed with packet capturing support function, commodity PCs for packet logging and analyzing, an optical tap (splits into 50:50 power), and a network switch for media conversion. These components are connected as shown in Fig. 2. An optical tap splits light on the objective line to TGNLE-1 and the original destination.

Fig. 3 shows how TAPEE works conceptually. TGNLE-1 clips heads of received packets, and puts

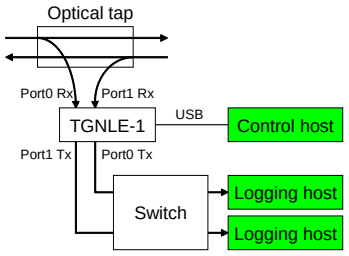

Fig. 2. TAPEE configuration diagram

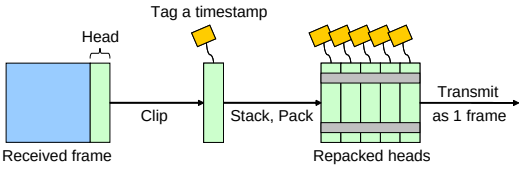

Fig. 3. Conceptual diagram of TAPEE

precise timestamps with 100ns resolution on them. Clipped packet heads are buffered, and several heads are gathered and reconstructed (repacked) as one packet, and transmitted to logging host. This reduces load of logging hosts to logging packets like amount of internal data transfer and number of interruption. This timestamping and repacking enable long-time precise analysis of high speed traffic.

TAPEE enables analysis of TCP streams by information of IP headers and TCP headers in logged packet heads. With header log, we can obtain transition of ACK and SEQ, transmission rate, amount of in-flight data, retransmission, duplicate ack, packet interval, etc.

We used two IBM eServer x345 servers for logging host. Specification is shown on TABLE I. They receive all the repacked packets from TGNLE-1 with Chelsio T110 10GbE NIC, and log them onto Ultra 320 SCSI hard disk drives.

Our system can analyze multistream communications. Captured heads are parsed on logging hosts, tallied by our software, and visualized using Gnuplot. We can see various characteristics of TCP's behavior by plotting transition and statistics of fields in TCP

TABLE I
SPECIFICATION OF LOGGING HOST

Model	IBM eServer x345
CPU	Intel Xeon 2.4GHz Dual
Memory	2GB
NIC	Chelsio T110 Chelsio TOE Network Driver 2.1.4
OS	Linux kernel 2.6.12.6 Red Hat Enterprise Linux WS 4

TABLE II
FPGA RESOURCE CONSUMPTION

Function	Slices	DCM	BlockRAM (RAMB16s)
Pass Through	7364 (31%)	5 (62%)	30 (12%)
TAPEE	7545 (31%)	5 (62%)	30 (12%)

and IP headers of each stream by TAPEE. For example, data like transmission rate are averaged and plotted, and data like packet delay and packet interval are tallied and visualized in histogram form.

The size of 10Gbps header logs is as big as several gigabytes. TAPEE passes processed data to logging host directly over 10GbE. Therefore, no special methods for copying data is necessary, and there is no speed bottleneck like copying method by USB. User can analyze packets freely and flexibly using their own software on logging hosts.

C. FPGA Resource Consumption

We used Xilinx ISE 6.3.03i CAD, and Verilog HDL. We used 133MHz system clock on FPGAs. Xilinx Virtex-II Pro XC2VP50 has 23616 slices. Amount of slices we used for each function is shown in TABLE II. The data of Pass Through is for reference. Pass Through function does nothing but passes packets without processing.

III. PROBLEM DETECTION IN PRACTICE

In this chapter, we show several interesting results of TCP/IP data transfer experiments on LFN obtained by using TAPEE. We analyzed traffic made by various configuration like use of hardware support, IP version, congestion control algorithm in subsection A, and next analyzed traffic on real and pseudo LFN in subsection B. Because of buffering at switches, intervals of short packets like ACK packets are shortened, and transmission became bursty in real LFN. We finally visualized and clarified (A) Burstness of packets output from NICs, and difference of burstness among different configurations of hosts, (B) Difference of ACK packets' behavior between real LFN and pseudo LFN.

A. Comparison of Transmission Burstness

We are performing transfer experiments using two IBM eServer x260 servers which have PCI-X 2.0 slots and Chelsio Communications T310 NIC. Transfer speed of PCI-X 2.0 is fast enough for 10GbE while one of PCI-X 1.0 bus is up to 8.6Gbps that is the bottleneck for 10GbE. We measured TCP transfer between two x260s in microscopic view by using TAPEE. This experiment is performed by using pseudo LFN in order to investigate the effect of

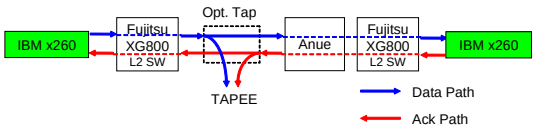

Fig. 4. Configuration of Experiments using T310

TABLE III
SPECIFICATION OF TRANSFER HOST

Model	IBM eServer x260
CPU	Intel Xeon 3.66GHz Quad
Memory	32GB
NIC	Chelsio T310
OS	Linux kernel 2.6.12 RedHat Enterprise Linux 4

only large delay without the effect of multi-hop forwarding. Fig. 4 shows the configuration of TAPEE, Anue network emulator and x260s. Between two x260s, there are 500ms RTT made by Anue H Series network emulator that virtually inserts delay into circuit. Specification of x260 is shown in TABLE III. TCP congestion avoidance algorithm was BIC-TCP.

We pick up two kinds of experiments we performed. One is transfer using IPv4 and hardware support. TCP Offload Engine (TOE) on receiver side, and TCP Segmentation Offload (TSO) on sender side are used. The result is shown in Fig. 5. The next one is transfer using IPv6. TOE and TSO are not available on T310 NIC, therefore, processing of TCP are done by software. The result is shown in Fig. 6. Transfer rate plot in graph is moving average of transfer rate. Thick curve shows 1sec average, and light scatter chart shows 1ms average.

We compared these microscopic throughput data with 1ms resolution of transfer with and without

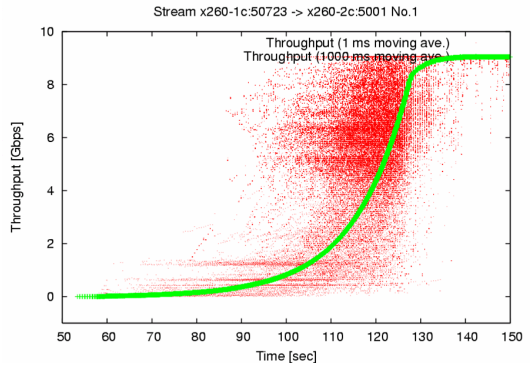

Fig. 5. 1ms and 1s moving average of transfer rate using T310, IPv4

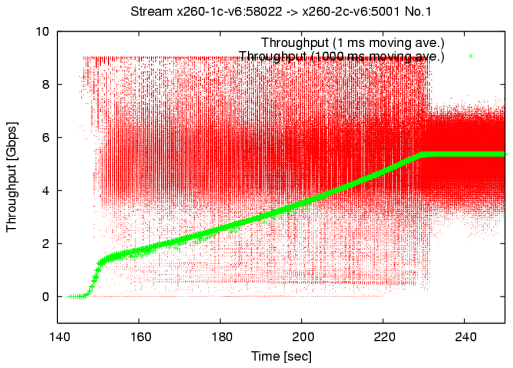

Fig. 6. 1ms and 1s moving average of transfer rate using T310, IPv6

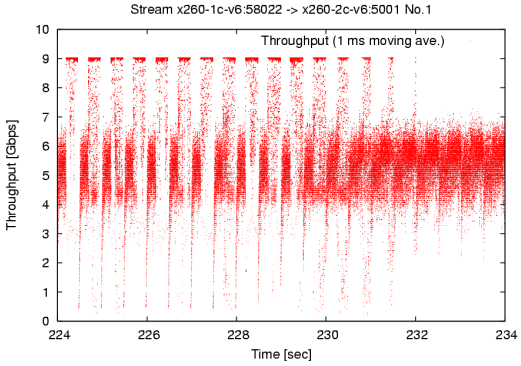

Fig. 7. Magnification of 1ms moving average of transfer rate using T310, IPv6 with shorter time span

hardware support. Transmission rates of both experiments are growing along with TCP window's growth, however 1ms microscopic transmission rate show difference. Using IPv4 with hardware support, packet intervals are put enough, and bursty transfer is corked up while window size have not scaled to large. Using IPv6 without hardware support, microscopic transfer rate is oscillating between 9Gbps and 1Gbps from the time right after connection established. Finally when TCP window scaled up to the product of RTT and maximum throughput, bursty transfer stopped. Fig. 7 shows this process clearly. This is magnification of IPv6 result at the point where scale up finishes. During scale up, a pattern consists of bursty and non-bursty transfer repeats in a period of RTT. At finish of scale up, we can see that oscillation between 1G and 9G stops and transfer becomes stable.

In real LFNs, which are often made with both WAN PHY for intercontinental lines and LAN PHY at edge, these bursty transmissions collide WAN PHY's 9.28Gbps bandwidth limit and lead to packet loss at WAN to LAN conversion points. We cannot

TABLE IV
SPECIFICATION OF TRANSFER HOST

CPU	AMD Opterom 250 Dual
Memory	2GB
NIC	Chelsio N210
OS	Linux kernel 2.6.14 CentOS

avoid this problem with suppressing TCP window because window only controls macroscopic rate. More precise transmission rate control method, for example special pacing device in front of hosts or Chelsio NIC's Inter Packet Gap (IPG) control function, is necessary. We configured T310's parameter which determines lower limit of IPG length to 872 octet, so transmission rate is suppressed to 9Gbps. Hosts are tuned to achieve best performance under this limitation.

B. Comparison of Real LFN and Pseudo LFN

We compared packet behavior on real LFN made of international light paths, and pseudo LFN made of Anue H Series network emulator. Analyzing and clarifying the difference between real LFN and pseudo LFN is important. Because real LFN is precious, the time when it is available for experiments is short. Therefore, we also use pseudo LFN for tuning hosts as large part of whole experiments.

We used AMD Opteron server we assembled for transfer experiments. The specification is described in TABLE IV. TCP congestion avoidance algorithm is BIC-TCP. The experimental configuration is shown in Fig. 9, and real LFN is shown in Fig. 8. This circuit consists of four intercontinental SONET/SDH lines and several WAN PHY switches, and has about 500ms RTT. Packets are sent out from Tokyo, Japan, go through North America, turn at Netherland, go through North America again, and come back to Tokyo. Anue network emulator is configured to make 500ms RTT on pseudo LFN. We focused on ACK packets' behavior and captured them at the point right after receiver and right before sender.

Results of real LFN are shown in Fig. 10, and Fig. 11. Results of pseudo LFN, both incoming ACK packets into sender host and one of outgoing ACK packets from receiver host, are almost the same as Fig. 10. These graph shows behavior of ACK packets. Cut plane at each point on time-axis is histogram. Height shows number of packets which passed observation point. Packet-interval shows the time since the previous packet passed. Because the length of ACK packets are under 100 octet, they are likely to be buffered by switches on the route,

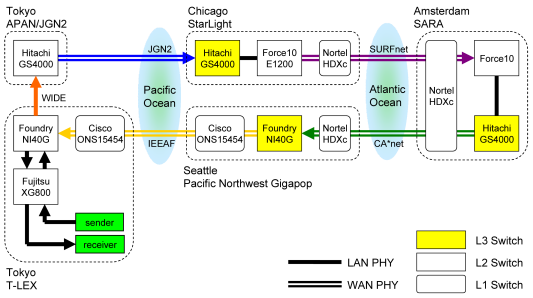

Fig. 8. Real LFN configuration

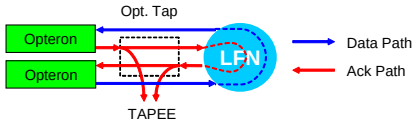

Fig. 9. Configuration of Experiments on LFN

lead to bursty transfer. According to Fig. 10, ACK packets output from receiver have mainly 0, 2 and 4 microsecond intervals. However according to Fig. 11, packets input to sender have only near 0 microsecond interval. It shows that multiple ACK packets were packed together and transfer became bursty on the route.

Bursty transfer of short length packets lead to frequent interruption to the host and finally lead to packet loss. We are tuning coalescing parameter of NIC that moderates interruption invoke, and we use NAPI when we use NICs with it that switches packet receiving scheme from interruption to polling when packet arrival rate become high. With these tuning, we can handle load of bursty transfer.

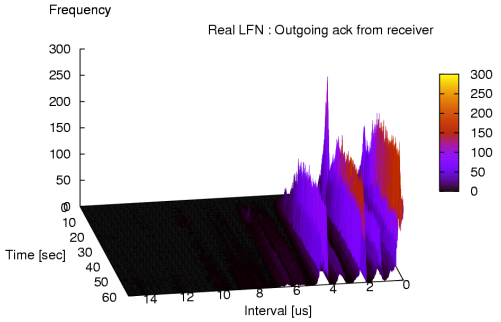

Fig. 10. Packet interval histogram of outgoing ACK packets from receiver host

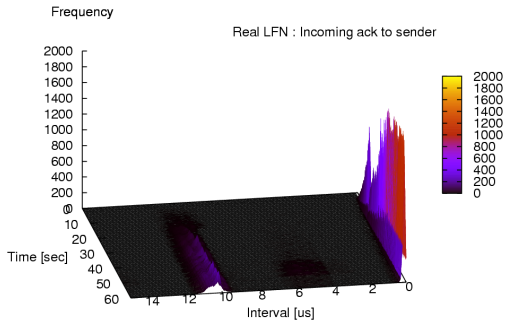

Fig. 11. Packet interval histogram of incoming ACK packets into sender host

IV. Current Achievement on Tuning

Currently, we have achieved 8.8049Gbps on 32372 km light path by IPv4, and 6.96Gbps on 32372 km light path by IPv6. These paths have over 500ms RTT. Precise analysis and tuning like TCP stack parameters and packet coalescing enabled success of our challenge on LFN. We are going to update IPv6 Land Speed Record using PCs with PCI-X 2.0 bus or PCI-Express bus.

V. Related Works

Existing hardware-based commercial analyzer for 10GbE are, for example, IXIA [5], SmartBits [6], Sniffer [7], etc. Large part of processing is done by special hardware in them, therefore, there is little programmability for user. On the other hand, using our system, user gets raw data of packet heads on logging hosts, and user can analyze them freely and flexibly for any purpose.

Other existing researches of programmable packet processors are GtrcNET-10 [8], XGE-ProtoDevel [9], etc. The former has precise rate measuring function. However practical experiments of problem detection are not performed yet, and it does not have long-time packet dump function for 10GbE. The latter has long-time packet analysis capability and error emulation function. It is developed for not TCP/IP on LFN but for protocol development for parallel computing.

VI. Conclusion

We described our data transfer experiments on LFN, and our wire-rate precise packet analyzer TAPEE. Communication on LFN has issues to be solved for utilizing bandwidth. TAPEE can detect and analyze these issues, and obtain data with fine time granularity on speed up to wire-rate. TAPEE enables precise traffic analysis by preprocess using hardware that sotfware method cannot do. Commodity PC hosts log the dump of whole header of objective traffic, therefore, user can analyze data flexibly using his own software for any purpose. Because using hard disk drives we can log for a long time, we do not have to adjust timing to problem occurrence.

We showed our approach to TCP/IP tuning on LFN, and evaluated our system by practical experiments, demonstrating utility of our system for detection of problems on TCP/IP communications on LFN. Our tuning solved several problems on TCP/IP and achieved 8.8049Gbps in IPv4 and 6.96Gbps in IPv6. Microscopic phenomena are invisible in rough statistics by netstat and SNMP. Precise analysis using TAPEE discovered these problems and enabled effective debugging.

Acknowledgment

We acknowledge support by Prof. Akira Kato of University of Tokyo, staffs of JGN II, IEEAF, Pacific Northwest Gigapop and AlaxalA Networks. 10GbE lines were provided from JGN II, SURFnet, IEEAF and CANARIE. This research is partially supported by Special Coordination Fund for Promoting Science and Technology, and Grant-in-Aid for Fundamental Scientific Research B from Ministry of Education, Culture, Sports, Science and Technology Japan and by 21st century COE project of Japan Society for the Promotion of Science.

References

[1] M. Nakamura, M. Inaba, and K. Hiraki, “Fast ethernet is sometimes faster than gigabit ethernet on long fat pipe network environment - observation of congestion control of tcp streams,” in *PDCS2003*, Nov 2003.

[2] S. Nakano and et al., “DR Giga Analyzer,” in *Symposium on Global Dependable Information Infrastructure (in Japanese)*, Feb 2004.

[3] A. Jinzaki, “Stream Processor Comet,” in *JSPP2000, IPSJ Symposium Series Vol. 2000, No.6 (in Japanese)*. IPSJ, 2000, pp. 205–212.

[4] Y. Sugawara, M. Inaba, and K. Hiraki, “Implementation and evaluation of fine-grain packet interval control,” in *IPSJ Technical Report OS-100 (in Japanese)*. IPSJ, Aug 2005, pp. 85–92.

[5] “Ixia’s chassis products,” http://www.ixiacom.com/products/chassis/.

[6] “SPIRENT Communications SmartBits,” http://www.spirentcom.com/.

[7] “Network general,” http://www.networkgeneral.com/.

[8] Y. Kodama, T. Kudoh, and T. Shimizu, “10GbE network testbed GtrcNET-10 - architecture and preliminary evaluation,” in *Summer United Workshops on Parallel, Distributed and Cooperative Processing (SWoPP2005) (in Japanese)*. IPSJ, Aug 2005.

[9] K. Nakashima, S. Sumimoto, M. Sato, K. Kumon, and Y. Ishikawa, “Proposal of communication protocol development support tool to resolve performance issue for 10Gbps class network,” in *Symposium on Advanced Computing Sytems and Infrastructures (SACSIS2005) (in Japanese)*. IPSJ, May 2005, pp. 321–328.