Evaluation of Gigabit Ethernet with Quality of Service for event builder

Y.Yasu[1], Y.Nagasaka[2], A.Manabe[1],
M.Nomachi[3], H.Fujii[1], Y.Watase[1], Y.Igarashi[1], E.Inoue[1] and H.Kodama[1]
[1] High Energy Accelerator Research Organization (KEK), Oho 1-1 Tsukuba, Ibaraki 305-0801, Japan
[2] Nagasaki Institute of Applied Science (NIAS), 536 Aba-machi, Nagasaki, 851-0193, Japan
[3] Research Center for Nuclear Physics (RCNP), Osaka University, 10-1 Mihogaoka, Ibaraki 567-0047, Japan

Abstract

A Gigabit Ethernet (GbE) is one of the candidates to be integrated to an event builder for next generation experiments on high-energy physics. This paper discusses the feasibility of applying GbE to an event builder.

Congestion avoidance of event data flow in a switching network is crucial. Traffic management of the data flow is an essential point to avoid the congestion. Global traffic control is one of the mechanisms to manage the traffic. It uses global information of the switching network. While, traffic shaping technique is another mechanism not to use them. This paper studied traffic shaping of GbE with a mechanism of Quality of Service (QoS). The performance of GbE with large frame called jumbo frame was also studied.

I. INTRODUCTION

In an event builder, event fragments from all sources are concentrated coherently into one destination via a switching network. However, commercial-available switching networks are designed for random traffic like Tele-communication data. They may not be able to handle coherent traffic like event builder. Congestion avoidance is indispensable for switch type event builder.

A way to avoid the congestion is to establish a global traffic control. A circuit switch was applied to an event builder at Fermilab[1] and KEK[2]. The data traffic is controlled by a global traffic signal. Another way is to design data flow by preserving the bandwidth for each node in a switching network. RD31 group at CERN established the way to shape traffic [3,4]. The other way to solve the problem is over-provisioning to the switching network.

Then, we investigated the feasibility of GbE with a QoS, which is a way to design data flow by reserving bandwidth. We also analysed coherent data flow over GbE[5].

A. Requirements and Gigabit Ethernet

Packet routing latency should be low enough to allow for the routing of several hundreds of packets for the second level trigger while it is not so low for the third level trigger. This means high level protocol such as TCP/IP with QoS may be used for the third level. On the other hand, the packet size for the third level may be larger than that for the second level. GbE technology with jumbo frame was investigated [6]. It showed that the jumbo frame improved the throughput twice. It is expected that the jumbo frame works well for the third level trigger.

Ethernet provides a best-effort service to all of its applications because of Carrier Sense Multiple Access/ Collision Detection (CSMA/CD). However, full duplex Ethernet allows simultaneous flow of traffic from one station to another without collision. So, Ethernet in full duplex mode does not require collision detection when only one port station is attached to each port. Recent GbE companies guarantee "wire speed" at each port, but the possibility of the congestion at burst data flow still remains. It is not clear to guarantee the speed on a coherent traffic.

In order to reduce the cost of development and ease maintenance, the use of industrial standard equipment is strongly recommended. Great advantage of GbE is that it is Ethernet. Ethernet is not only International standard but also de facto standard. GbE is fully compatible with existing Ethernet installation. It is also attractive from viewpoint of cost/ performance.

The event builder system should be scalable to accommodate future upgrades. The GbE switches with a few ten GbE ports and high speed back-plane are already available. For an example, Alteon Company supplies a switch, which has a back-plane with a bandwidth of 180 Gbit/s and 32 GbE ports with non-blocking[7]. It supports jumbo frame. Some companies announce switches with over hundred ports of GbE.

Next section describes congestion control and bandwidth allocation because continuous multimedia application has similar problem of congestion in the network. Third section describes the performance measurement of GbE with QoS on PC/Linux.

II. CONGESTION CONTROL AND BANDWIDTH ALLOCATION

Researches and developments of QoS are recently done for Internet, LAN and WAN. Many companies for the switching network supply IEEE 802.1Q/p, namely, VLAN/Traffic prioritization. VLAN provides greater segmentation and organizational flexibility while it makes the physical boundaries free. Internet Engineering Task Force (IETF) is working on terminology and architecture of QoS guarantees and management, and made Resource ReSerVation Protocol (RSVP)[8] for multimedia application such as Voice over IP and Video on demand in Internet. On the other hand, packet queuing disciplines, which were studied on technology related to ATM in computer science, are Class-Based Queuing (CBQ), Weighted-Fair Queuing (WFQ) and so on.

A. Terminology of QoS

We will introduce element technologies of QoS first. There are admission control, packet classification, packet scheduling

and traffic shaping. The admission control is a way to control reserving resources in a session such as RSVP. A setup protocol makes signaling on the path. The packet classification is to classify incoming packets in groups by using Type of Service (TOS) field in IP packet, for an example.

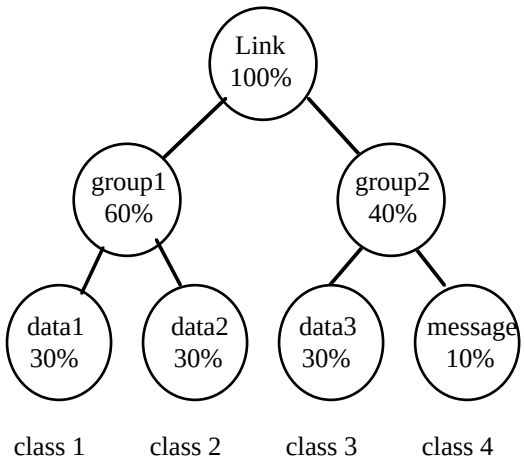

Figure 1: Class Configuration

Fig.1 explains the packet classification. The Link has 100 % of the bandwidth, for example, 1000Mbit/s. The grouop1 has 60 % of the bandwidth while group2 has 40 % of that. In this case, there are four classes. 300Mbit/s of the bandwidth are guaranteed for the class 1.

The packet scheduler is to arrange the scheduling for outgoing packets. There are many ways according to the queuing method and the buffer management. Fig.2 shows a packet-scheduling algorithm. The outgoing packet will be sent according to the size of the token buffer and the rate. The traffic shaper is a technology to make the burst flat.

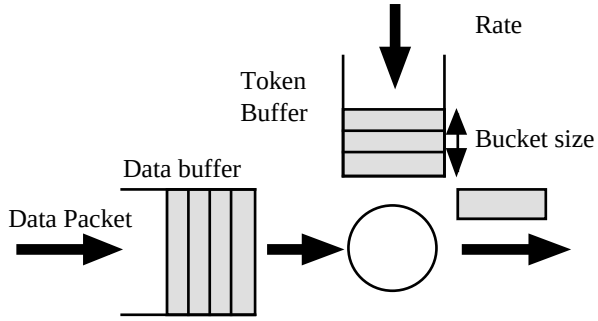

Figure 2: Packet scheduler

B. RSVP

Today's Internet provides a best-effort service. It does not make any promises about the QoS that an application will receive. ATM is suited for applications such as Video on demand, because ATM has QoS and different services such as Constant Bit Rate (CBR).

Recently, QoS on Internet such as RSVP and Diff-serv is intensively studied. Fig.3 shows RSVP architecture. It can manage bandwidth allocation in a switching network including hosts and routers. RSVP itself is a signaling protocol and does not include packet classifier, packet scheduler and/or packet shaper. Usually those modules are provided in kernel of operating system while RSVP daemon and RSVP application program can communicate with the modules in the kernel. RSVP can reserve bandwidth between RSVP application programs on source (sender) and destination (receiver) in a switching network. After establishing the bandwidth reservation, bandwidth between any sender and any receiver is reserved and then guaranteed. RSVP sets up unidirectional reservation and receiver makes the reservation request. For the event builder, the receivers on destination nodes can make the reservation request dynamically.

Big companies such as Cisco Systems, Intel and 3Com announced support for RSVP.

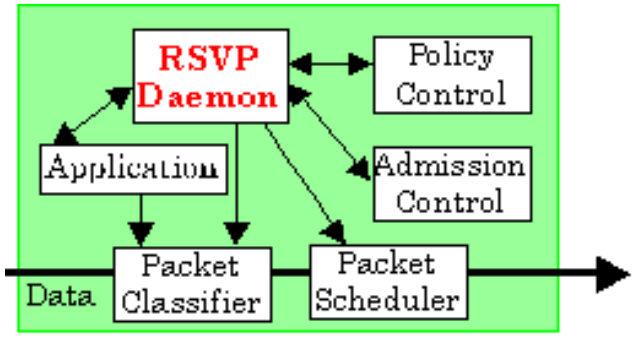

Figure 3: RSVP architecture

C. Traffic Management

Traffic management is one of topics in recent packet network. The essential point of the manager is packet scheduling known as queuing. PC-based scheduling mechanism is studied in computer science [9]. The implementation is called ALTQ and the preliminary result is also reported. A queuing algorithm CBQ was investigated and the performance was measured on FreeBSD operating system. The result shows the throughput overhead was 0.4 % in comparison with that of original FIFO queuing on 100Mbit Ethernet while the latency was 10 µsec when using request/reply style transaction with UDP on 10Mbit Ethernet/ PentiumPro 200MHz.

D. Linux Traffic Control

The implementation of QoS in Linux kernel and QoS application interface (QoS API) is also in progress [10, 11]. One of QoS API is RSVP API called RAPI. Linux traffic control functions are implemented in Linux kernel [11]. We can manage the scheduler via RAPI or a tool for setting the parameters in the kernel. The tool is called tc command [12]. Linux could not control the traffic of packets because old Linux kernel had only a simple queue so far. Now new Linux kernel supports more complicated queuing discipline, which may use filter to distinguish among different classes of packets and process each class in a special way. A Token Bucket Filter (TBF) queue is one of useful queues.

E. Congestion control in TCP/IP

We investigated GbE with TCP/IP device driver, not special driver. From viewpoint of flow control, UDP is simple, but TCP is complicated. We checked the flow control of TCP. For event builder, it is not clear whether this algorithm is useful or not.

Delayed ACK

If TCP were to generate a separate ACK for every packet it receives, the network would quickly become overly congested. Delayed acknowledgements (Delayed Ack) is a way to reduce unnecessary packets, by acknowledging multiple packets with a single ACK.

Nagle's algorithm

Nagle's algorithm states that when there is data that has been transmitted but not yet acknowledged from the receiver, the sender must not transmit any small segments. It is possible for a user process to disable it using the TCP_NODELAY socket option.

TCP sliding windows

TCP has a special kind of buffer called a Sliding Window. This window has a maximum size indicating how many buffers are allocated to queuing incoming packets. When the buffer is full, additional packets cannot be read and therefore must be dropped. To avoid having to drop packets, TCP uses Window Size Advertisements as a way of informing hosts communicating with it as to how much buffer space is left. By controlling the window size, a host can control the rate at which other machines communicate with it. When a host is congested, it can advertise a window size of 0 to force other hosts to stop receiving until future advertisements of available buffer space.

III. PERFORMANCE EVALUATION

A. Setup

We used two PCs and a GbE switch for the evaluation. The configuration of the PCs and the switch is shown in Table 1. The GbE switch we used was Alteon AceSW180, which has 9 ports of GbE with a bandwidth of 8Gbit/s in the back-plane. Linux driver for AceNIC was used [13]. Th receiver was PC1 and the sender was PC2.

Table 1. Configuration of PC1 and PC2

CPU	PentiumII-Xeon 450MHz with 512KB cache x 2	PentiumII-333MHz with 512KB cache
Memory	256 MB	160 MB
Gigabit Ethernet	AceNIC with 1MB memory	
OS	RedHat5.2 with kernel 2.2.6	RedHat6.0 with kernel 2.2.5
C compiler	gcc version 2.7.2.3	gcc version egcs-2.91.66

B. TCP buffer size

First, we checked TCP buffer size. When the buffer size was 8192 bytes and the message size was 8000 bytes, the transfer speed was 4MB/s. But, the speed became 35MB/s when the TCP buffer size was 65535 bytes and the message size was 65535 bytes. Large TCP buffer size makes the speed fast.

C. Jumbo frame

Alteon NIC and the Switch support large MTU called Jumbo frame. We measured the transfer speed with normal frame (MTU=1500) and Jumbo frame (MTU=1500-9000). When MTU enlarges, data size per transfer on Ethernet frame enlarges. When the message size is greater than around 1.5 KB, the transfer speed saturated at around 35 MB/s for normal frame and 57 MB/s for jumbo frame. Fig.4 shows that jumbo frame improved 60 % of the network performance around 3KB of the message size. The TCP buffer size was 65535 bytes and a network tool called Netperf[16] was used.

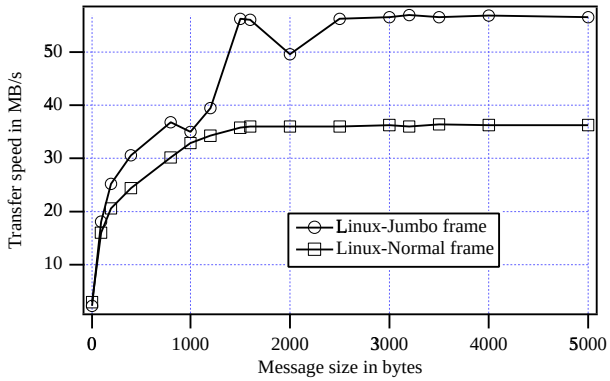

Fig4: Performance comparison with TCP/IP

D. CPU usage with TCP

We checked CPU usage at data transfer. We used a tool called TTCP [17]. It was assumed that the TCP buffer size was 65535 bytes and the message size was 10000 bytes. At normal frame, a sender consumed 75 % of CPU time while a receiver consumed over 90 % of the time. On the other hand, the sender took 50% while the receiver took 75 % at jumbo frame. Jumbo frame reduces CPU time.

E. Packet loss of data transfer with UDP

When UDP was used on the TTCP test, over 50 % of packets were lost. The reason of the packet loss causes the processing power of receiver, not that of switch because the switch guarantees "wire speed" at the condition.

F. Latency and Round Trip Time

When using a network tool called Netperf, the round trip time with TCP between PC1 and PC2 via the GbE was 4.8k packets/sec with a byte length of packet. When using a ping command with a 64-byte message, the round trip time with ICMP took 190μsec.

G. Overhead of Traffic Control on Linux

We installed Linux QoS, which includes class-based queuing discipline (sch_cbq), token bucket filter queue (sch_tbf) and universal 32-bit key packet classifier (cls_u32). The configuration was same in Table 1 except the NIC and the device driver. The NIC was G-NIC [14] and the driver was "yellowfin"[15]. A tool called tc command for managing the parameters related to QoS such as CBQ and TBF was used and then the overhead of the traffic control function was measured. Netperf was used for measuring the overhead. The sender was PC1 and the receiver was PC2. The QoS was installed only on the PC1. The TCP buffer size and message size were 65535 bytes. The result is shown in Fig.5. The X-axis is bandwidth assigned by the tc command in Mbit/s. The Y-axis is measured bandwidth in Mbit/s. The maximum transfer speed without QoS in the configuration was 185Mbit/s, which was measured by Netperf, too. The result showed that the overhead was negligible.

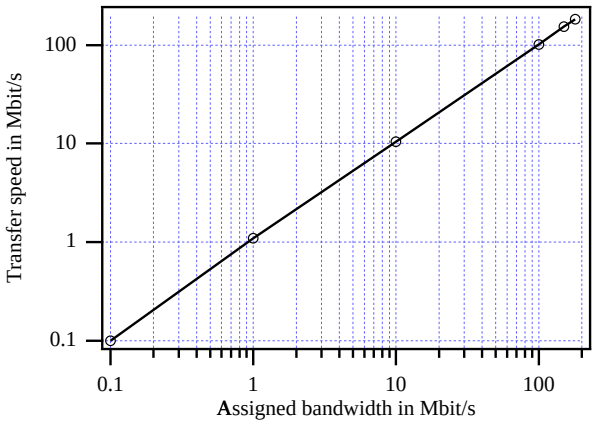

Fig.5: Transfer speed of TCP/IP with CBQ packet scheduler

IV. CONCLUSION

The traffic management of event data flow is necessary for the event builder. We investigated another solution of the traffic management instead of ATM-based traffic management. Then, Quality of Service for LAN, WAN and Internet is surveyed. From the result of ALTQ project, the overhead of the traffic modules was very small and the bandwidth guarantee was done successfully. The Linux distribution kit is already available. We installed the kit and tested on PC/Linux. From the result, the traffic control function could limit the bandwidth of the source node without the overhead. Therefore, the method is useful for congestion control of the event builder.

On the other hand, jumbo frame worked well. The jumbo frame is also useful for the event builder.

V. ACKNOWLEDGMENTS

The authors wish to thank Prof. Kondo at KEK for his support and encouragement, and people of Netone Systems Corporation and Alteon Web Systems in Japan for their help.

V. REFERENCE

[1] Ed. Basotti et al., A Proposed Scalable Parallel Open Architecture Data Acquisition System for Low to High Rate Experiments, Test Beam and All SSC Detectors, IEEE Trans. NS, Vol.37, No.3 (1990)

[2] Y.Nagasaka et al., Performance Analysis of a Switch-type Event Builder with Global Traffic Control System, IEEE Trans. NS, Vol.43, No.1(1996)

[3] D.Calvet et al., Evaluation of a Congestion Avoidance Scheme and Implementation on ATM Network based Event Builders, Proc. Second International Data Acquisition Workshop on Networked Data Acquisition Systems, World Scientific Publishing 1997, pp.96-107.

[4] D.Calvet et al., Operation and Performance of an ATM based Demonstrator for the Sequential Option of the ATLAS Trigger, IEEE Trans. NS, Vol.45, No.4(1998)

[5] Y.Nagasaka at al., Analysis of coherent data flow over Gigabit Ethernet, these proceedings.

[6] Y.Yasu at al., Evaluation of Gigabit Ethernet with Java/HORB, Contributed to the International Conference on Computing in High Energy Physics, CHEP98, Chicago, August 31 - September 4, 1998

[7] Alteon Web Systems, http://www.alteon.com/

[8] R.Braden, L.Zhang, S.Berson, S.Herzog and S.Jamin, Resource ReSerVation Protocol (RSVP) – Version1 Functional Specification, RFC2205.

[9] K.Cho, A Framework for Alternate Queueing: Towards Traffic Management by PC-UNIX Based Routers, Proceedings of USENIX 1998 Annual Technical Conference, New Orleans LA, June 1998

[10] P. Y. Wang, RSVP distribution kit for Linux, http://www.cs.columbia.edu/~yhwang/ftp/qos/rsvp/

[11] Alexey Kuznetsov, Implementation of queuing disciplines in Linux kernel. They are available in Linux kernel 2.2.x. http://www.linuxhq.com/

[12] Alexey Kuznetsov, tc command is available. ftp://ftp.inr.ac.ru/ip-routing/

[13] Jes Sorensen, Linux device driver for AceNIC, http://home.cern.ch/~jes/gige/acenic.html

[14] Alcatel Internetworking, Inc., Packet Engine G-NIC, http://www.ind.alcatel.com/pe.html

[15] Donald Becker, G-NIC Ethernet driver for Linux, http://cesdis.gsfc.nasa.gov/linux/drivers/yellowfin.html

[16] Netperf Home page, http://www.netperf.org/netperf/NetperfPage.html

[17] ftp site of TTCP, http://ftp.arl.mil/ftp/pub/ttcp/