Experimental Results of TCP/IP data transfer On 10Gbps IPv6 Network

Junji Tamatsukuri, Katsushi Inagami, Mary Inaba and Kei Hiraki
the University of Tokyo
7-3-1 Hongo, Bunkyo-ku,
Tokyo Japan, 113-0033
{ junji, inagami, mary, hiraki}@is.s.u-tokyo.ac.jp

***Abstract*— It is well known that TCP/IP data transfer on LFN (Long Fat pipe Network) is difficult, and this problem has become more serious as the bandwidth grows. Especially single stream TCP is the base of all the internet high speed communication and the key of scientific data transfer system such as Data Grid. We show the performance and behavior of single stream TCP at LFN, both on pseudo and real 10Gbps network. Pseudo network by network emulator makes a long latency almost around the world. We analyzed and compared real network with pseudo LFN network, and we tried LSR (Internet2 Land Speed Record) of single stream IPv6. This paper describes the detail of experimental results of single stream TCP data transfer on LFN with 10Gbps speed, using both IPv4 and IPv6 by standard/jumbo frame 10Gbps Ethernet.**

I. INTRODUCTION

Current important problem to realize large scientific data transformation is to know how much performance we can exploit from single high performance host and single TCP/IP streams, both IPv4 and IPv6. It is well known that TCP has a difficulty on getting performance when it is used for data transfer over long distance and high bandwidth network, called "Long Fat pipe Network (LFN)". For reliability, TCP/IP uses ACK, and data, which is sent but not ACKed, is called "inflight" data, whose maximum size is called "window size". Data transfer rate of TCP is roughly defined $windowsize/RTT$, where RTT is Round Trip Time. Hence, for LFN, large window size is needed, but, at the same time, in Linux TCP/IP stack (it uses New Reno and Big algorithms), at congestion avoidance AIMD phase, the growth of the window size is proportional to RTT, which means the growth is slow for LFN. Many research groups proposed new congestion avoidance algorithms for improving its growth of window size. But low performance on LFN is not only caused by the slow growth window size. In [2], we observed that Fast Ethernet interface at both ends attains better and more stable performance than Gigabit Ethernet interface for the circuit whose bottleneck is OC-12, about 622Mbps. Current network equipment and 1Gbps interface card can handle real 1Gbps at wire rate. We guess this is caused by the difference of the speed of link layer; where the speed is network interface card speed, such as 100Mbps, or 1Gbps for all streams in total, and transport layer; where the total speed is roughly $windowsize/RTT \times numberofstream$. As far as TCP stack has sufficient sending data in window, sender stack puts data to network interface with the maximum speed of the interface, for Gigabit Ethernet 1Gbps. When no data left on buffer, sender stops sending, then it results in the performance speed limits to the average transfer rate of sender. It shows macroscopically bandwidth = window size / RTT. But peak transfer rate is the maximum speed of network interface physical layer. This burst behavior brings self-congestion; that is, although macroscopically there exists no congestion, microscopically there exist buffer overflow in intermediate routers, which occasionally induces unfortunate packet loss, which results in unnecessary dispersion of the performance. To tackle with this problem, we proposed " Transmission Rate Controlled TCP (TRC - TCP)"[7].

This kind of phenomena is standing when the bottleneck of the circuit is narrower than the network interface speed. But, we also face to the similar phenomena, when the network interface speed is slower than the network or there is some bottleneck in the end to end connection. One of the interesting points of difficulties of data transfer over 1Gbps is the varieties of the bottlenecks. For example, in [2], the bottleneck was OC-12 (about 622Mbps) across the Pacific Ocean and we control the transmission speed of the sender. In [7], hard disk I/O speed is the bottleneck and we striped the data, and in [3], the PCI interface between network adaptor and CPU and memory is the bottleneck. As the result, although network adaptor can handle 10Gbps stream, as far as main CPU of the host computer is used to make the sending data, the end-node can accept data only with the PCI-X speed, which is currently, 133MHz $\times$ 64bit; that is, about 8.5Gbps. But microscopically, sender side is pushing to data at the rate of maximum rate of interface, 10Gbps. This means, the bottleneck is not on the circuit, but is inside the receiving node I/O limitation. In such cases, it is true that pacing the sender is effective but TCP/IP congestion control. However, no matter how carefully we pace the sending packet interval, some burst packets are observed which is probably caused by the intermediate routers, controlling receiver is rather effective. This kind of problem has become more serious as the bandwidth and distance grows. Now very high speed link such as OC-192, 10Gbps Ethernet is used, and 40 and 100Gbps network is coming to connect all over the world, so it becomes more important to tackle this problem.

This paper describes this bottleneck congestion phenomenon outside network by the experimental results of data transfer on pseudo and real LFN, 10Gbps Ethernet both LANPHY(10.31Gbps) and WANPHY (9.95Gbps), using both IPv4 and IPv6, with both standard and jumbo Ethernet frame. We observed that the performance of real LFN is much worse than pseudo LFN, and controlling the receiver is effective. This paper is organized as follows. In Section 2, we describe the experimental settings. In Section 3 and 4, we present pseudo and real LFN experimental results. In Section 5, we conclude our experiments.

II. Pseudo Network Environment

Mostly RTT (end to end latency) influences TCP/IP performance on LFN. Many TCP/IP stacks adopt AIMD algorithm on stream control. It decides window size from RTT of end to end ACKed packets and space of receiver buffer memory. Sender side on TCP stream decides amount of transmitting data from ACKed packets. This TCP's end to end control causes the difficulty of effective use on large RTT LFN. For packet drops decrease the window size and performance, there are many researches in TCP congestion control to recover from packet drop quickly. But for getting high performance from single stream TCP, it is important to get rid of the influence of long latency ACKed packet and to make use of large congestion window and write/receive buffer on TCP stack. Therefore how to get high performance is how to avoid going into congestion control phase of TCP/IP protocol. Pseudo LFN, it has only pure latency, show the most primitive performance of TCP/IP protocol and behavior. LFN on real network have burst, reorder, and jitter of packet arrival. These are behavior of physical circuit and L3 routers. They are effect on the worse of TCP/IP performance. We supposed Ideal pseudo LFN behavior is same as only long RTT LFN. If TCP/IP stream on only long latency environment shows good performance, TCP/IP can utilize for LFN.

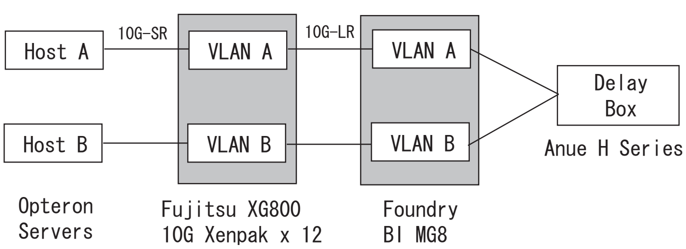


Fig. 1. Pseudo Network Environment

A. Configuration

Recently 10Gbps network equipments are in general at the world wide network backbone. This network constructs LFN among inter-ocean scientific researches. Pseudo network equipments appear to emulate such vast bandwidth, long latency. It can generate the latency, jitter, packet loss, and so on.

We used Anue H series Network Emulator which add from 0 to 400msec artificial delay on full wire rate at single direction.

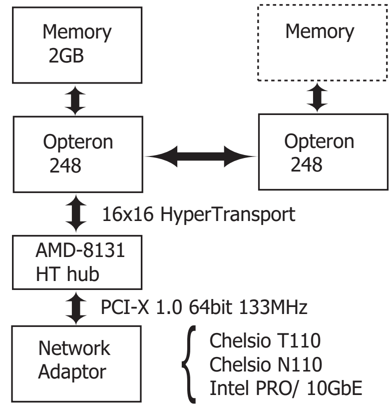


Fig. 2. Dual Opteron Server

It supports LAN-phy and WAN-phy for physical layer. We used LAN-phy and functions for inserting latency.

We used Fujitsu XG1200 and Foundry BigIron MG8 for network connection. (Figure 1) MG8 is a substitute of LFN L3 router. We used only L2 forwarding because the result of one hop routing and L2 direct connection is almost same. Fujitsu XG1200 is our edge 10Gbps switch. It converts from 10G-LR/LW to 10G-SR. All terminals have same L2 VLAN. We measure traffic by SNMP of MG8. We used white Linux 2.6 kernel. It doesn't contain any change. Experiment parameters are Window size, TX Queue, and Buffer Size. These are general parameter of Linux TCP/IP stack.

We use dual Opteron 248 2.2GHz CPU with Rioworks HDAMA mother board, with DDR3200 CL2 2GB (512MB*4 on the memory slot of main CPU) (Figure. 2). Operating System is Linux 2.6.6 and 2.6.12. Network adaptors are Chelsio T110 Protocol Engine with cxgbtoe ver 2.1.1 and chtoe-t1 ver.1.1.4, Chelsio NI110 Server Adapter with cxgb ver 2.1.1, and Intel PRO/ 10GbE SR Server Adapter (IXGB) with ixgb ver 1.0.110.

In this experimental configuration, the most influence limitation is the PCI-X I/O bandwidth between Hyper Transport and network adaptor. It is limited to about 8.5Gbps (//64 bit $\times$ 133MHz). This bottleneck happen packet loss rather than on the network. When packet loss occurred, TCP control goes into congestion control phase. Small amount packet losses are recovered by SACK function. But I/O limit occurs with large packet loss. TCP congestion control decreases inflight packet window size and grows slowly again. Once the stream is in the growing again phase, it is difficult to sustain the high throughput.

To avoid packet loss by I/O limitation, we use two methods. The one is the transmission rate control at the sender side network adaptor. Transmission control is effective for receiving packet performance. 10G network adaptor can transmit packet data at almost 10Gbps at a moment. So receiver side must get it at 10Gbps. Most of 10G network adaptor itself can get at that high rate. But host I/O cannot tolerate such high rate. Therefore we must limit transfer rate at the sender side.

The latter is the flow control at both edge sides. When I/O saturation occurred between network adapter and PCI-X bus, MAC(Media Access Controller) can generate PAUSE packet to switch direct connected. LFN routers have large size packet buffer on each port. If the buffer cannot happen overflow, this flow control effects for backpressure from receiver side. This backpressure makes a little bit of bigger RTT. This bigger RTT ACKed packet stops growing the inflight packet window. This TCP stream can sustain high throughput almost the maximum rate of sender/receiver connection. This behavior observed on pseudo network result.

B. experiment results

We measured the data transfer rate from memory to memory, using Iperf 2.0.2. We mainly compare 1500bytes standard frame and 9198 bytes of jumbo frame, which is the maximum MTU of the Foundry MG8. Other parameters are memory size for TCP window, socket buffer size for application, and TXQUEUE length. These have default value for suitable for 1Gbps network adaptor.

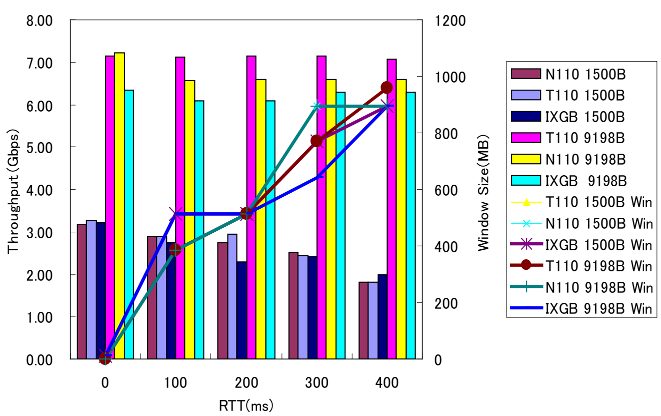


Fig. 3. IPv6 Transfer Rate and Window Buffer Size

Fig.3 shows IPv6 performance on pseudo network. We changed RTT on pseudo LFN from 0ms to 400ms stepping 100ms. The peak traffic rate is variable according to window buffer size of TCP stack and application buffer of iperf. Linux TCP stack defines cWnd size from space of read/write buffer memory. But larger memory size is getting lower the performance because of efficiency of memory management on TCP stack. Iperf buffer size defines the ratio of system/application CPU usage. The load of iperf is very small, but the producing data push into TCP stack and make higher its memory usage. The shortage of space of read/write buffer memory goes lower the cWnd. It decreases the traffic rate of the stream. The ratio highly influences the inflight data size and stream performance. When we set proper value to these parameters, the peak result is as same as local communication at jumbo frame (9198Byte). Chelsio adaptors have better interrupt response than Intel PRO/10GbE. Therefore, the base performance of Chelsio adaptors is better than Intel adapter. This result shows the potential performance of TCP on LFN is enough to exploit 10Gbps network. But in standard frame (1500Byte); same host cannot transfer only up to almost 3 Gbps. The increase of packet on the same rate increases CPU usage for TCP packet production and interrupt from network adaptor. This saturation on CPU and interrupts happen to lower standard frame performance than jumbo frame.

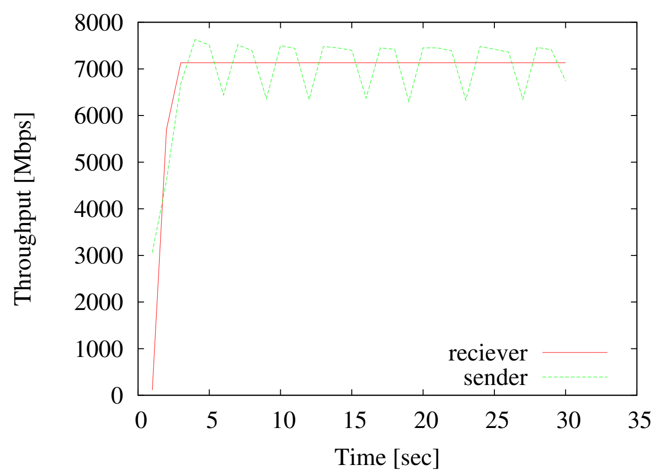


Fig. 4. IPv6 Traffic Rate (RTT=100ms), Iperf

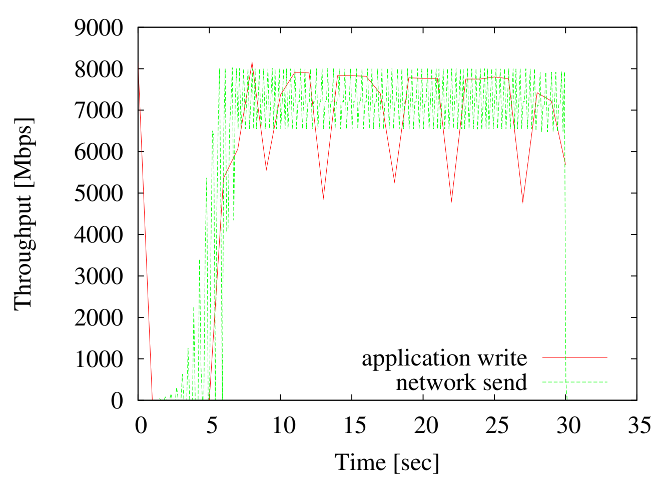


Fig. 5. IPv6 Traffic Rate (RTT=400ms), SMNP

Fig.4,5 show the behavior of TCP startup phase on 100ms and 400ms. Fig.4 is from iperf result both sender/receiver side. Fig.5 is from MG8 SNMP octets value on receiver and iperf result from sender side. The difference between these graphs is the time of initial growing of the cWnd. 100ms RTT spends almost 2 second to the maximum size. 400ms RTT spends 6 second. Both sides are varied within 1Gbps. But receiver side is stable on the peak performance. Both streams can sustain over 7Gbps without any other influence.

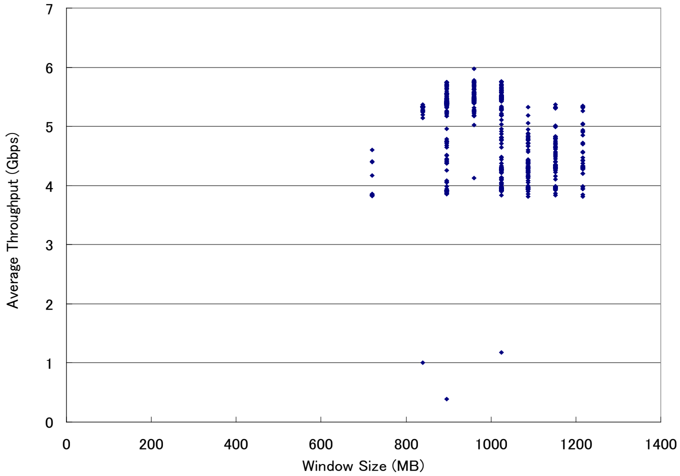


Fig. 6. Window Size

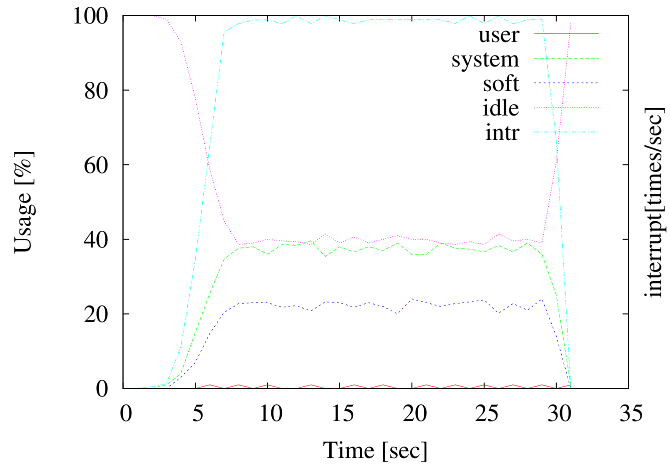


Fig. 8. CPU Usage of receiver side

Theoretical TCP windows size is defined from $RTT \times Bandwidth$. Actual cWnd is this size. But current Linux TCP stack advertises space of read memory buffer by ACKed packet. Linux TCP stack decides available space of TCP stack by (read buffer memory - current used memory)/2. This divided part is for communication/application buffer. So Software TCP stack needs 3 times lager than theoretical value. Fig.3, 6 show this requirement. We tried to confirm by required window size. Fig.6 shows precise 3 times larger window size can get the best performance.

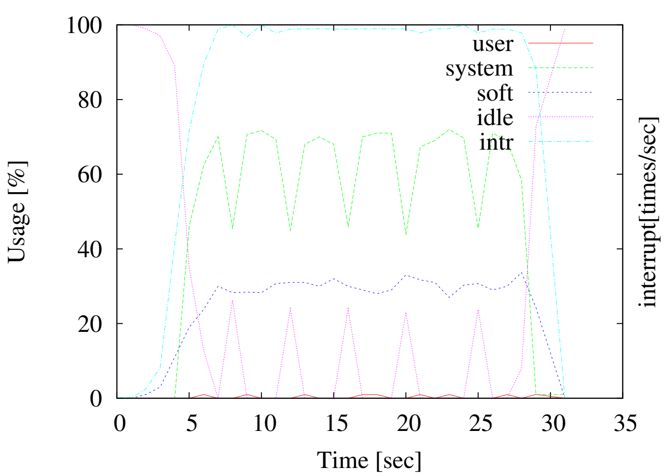


Fig. 7. CPU Usage of sender side

Fig.7,8 is CPU usage at 400ms communication. CPU of Sender side is busy for system and exhausted. Iperf application usage is within 1%. 60% of CPU used for TCP stack and 40% used for softirqs. On the receiver side, CPU isn't exhausted. Receiver side performance is limited by I/O. So this side usage is stable on long RTT and high performance situation.

Fig.9 shows IPv4 performance on the same host, network

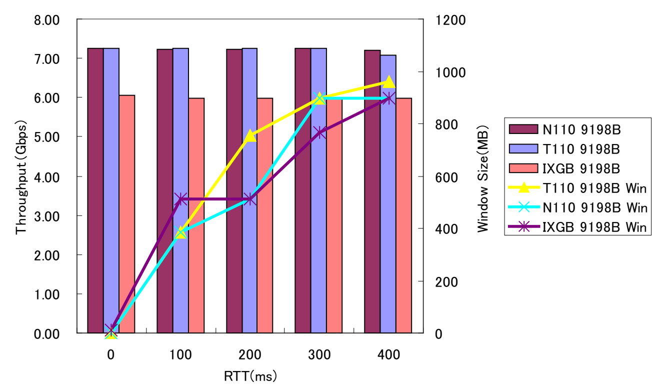


Fig. 9. IPv4 Performance

environment. There is no difference between IPv4 and IPv6. Disadvantage of IPv6 is the increased packet header. But this result could not show this disadvantage.

III. JAPAN U.S. EXPERIMENT

On real LFN, a lot of factors from constructing network influence TCP performance. These are expressed by packet loss, jitter among packets and low throughput on TCP stream. According to our pseudo LFN experiment, large RTT doesn't degrade the TCP performance. But it difficult to acquire large stream performance on real LFN. We tried to get IPv6 performance on real LFN as same as over 7Gbps, IPv4 LSR record. We used the sender side transfer limitation and both side flow control method.

We examine real LFN experiment based on the result of pseudo LFN. We used two inter-pacific networks for real LFN experiments, IEEAF Tokyo-Seattle line, and JGN Tokyo-Chicago line.

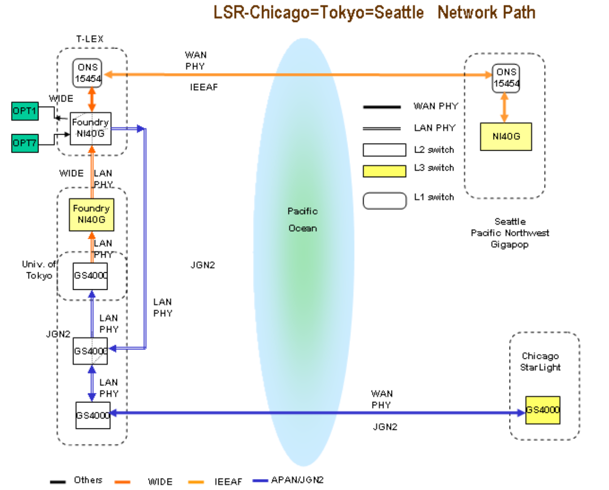


Fig. 10. Network Configuration

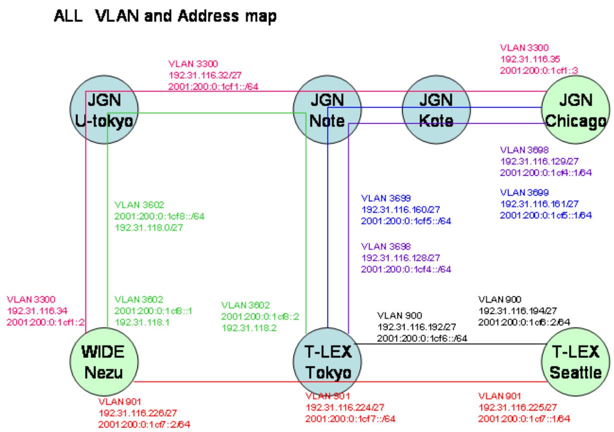


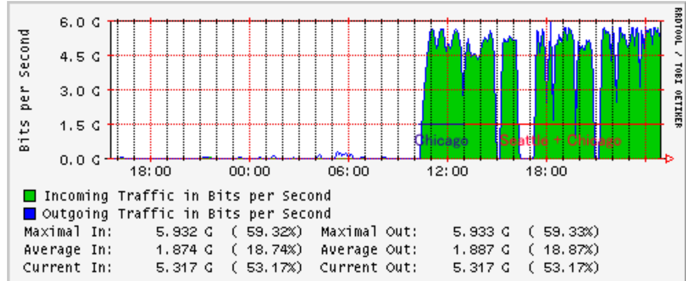
Fig. 11. VLAN Configuration

Fig.10 shows the circuit configuration of the network.

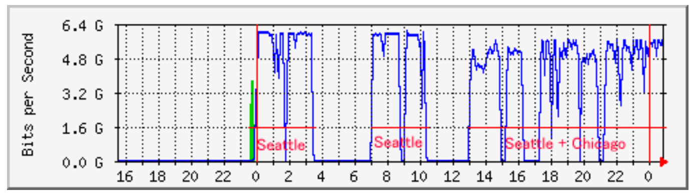
We sited all hosts for experiments on T-LEX Tokyo (it places in NTT Otemachi building.). These routes made of JGN (Japan Gigabit Network) section from University of Tokyo site (U-Tokyo) via Note, KDD Otemachi (Kote) to Chicago starlight, and IEEAF-WIDE section from U-Tokyo via T-LEX Tokyo to T-LEX Seattle. All the routers in JGN section are Hitachi GS4000, and in IEEAF-WIDE section are Foundry NetIron 40G. VLANs were configured on same physical line of production network. We could use multiple routes by changing the edge port setting on T-LEX NI40G.

We have examined three routes using two circuits

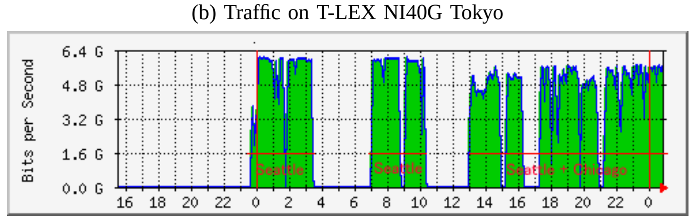
a. Tokyo-Seattle-Tokyo, RTT=178ms, 15,461km,
b. Tokyo-Chicago-Tokyo, RTT=322ms, 20,294km
c. Tokyo-Seattle-Tokyo-Chicago-Tokyo, RTT=500ms, 35,755km



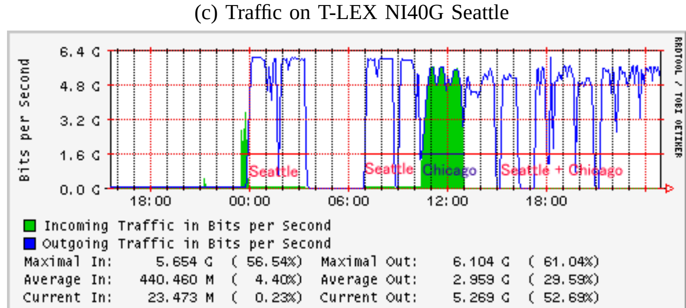
(a) Traffic on JGN GS4000 Chicago



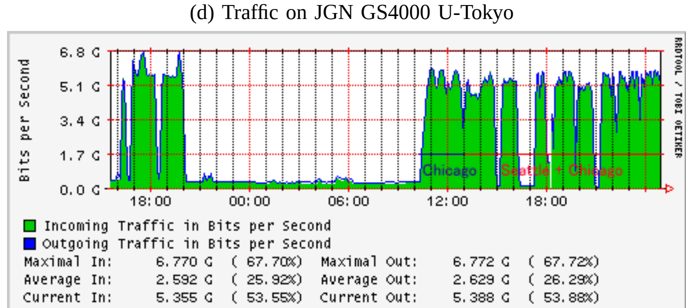
(b) Traffic on T-LEX NI40G Tokyo



(c) Traffic on T-LEX NI40G Seattle



(d) Traffic on JGN GS4000 U-Tokyo



(e) Traffic on JGN GS4000 NTT - KDD

Fig. 12. Network Traffic (Oct 28, 2005)

These distances are sum of straight course distance of host and all L3 routing points. All routes crossed from T-LEX Tokyo to JGN U-Tokyo, JGN Note, and T-LEX Tokyo for L3 routing. We combined this loop with inter-pacific roundtrip circuits to make above three routes.

To control sender side transmission rate, we changed the clock speed of network adaptor; we set a sender network interface card as slow speed so that the packet sending rate becomes slower, and a receiver has the highest clock. Other parameters were settled by scaling to RTT from pseudo LFN experiment. We used same hosts on pseudo experiment but used newer kernel version 2.6.12 for host stability for real network. Network adaptor is Chelsio T110 in all the hosts without TOE (TCP Offload Engine is disabled). The communication is mainly performed by IPv6 TCP. But for checking circuit condition, some communication is by IPv4 UDP, TCP.

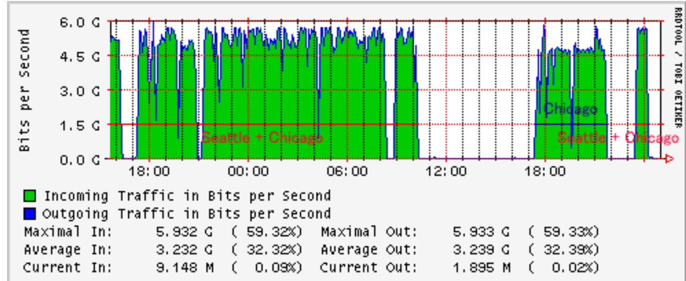
We tried the experiments on continuous two days Oct. 28, 29 2005. First, we tried to examine route (a) and route (b) independently, and got the result (a) 5.94Gbps, and (b) 5.60Gbps. After that, we examined (c) as the longer route, whose result was surprisingly, 5.60Gbps, which was exactly same as (b) Chicago route, although the RTT becomes 1.5 times longer. And, this result is worse then pseudo LFN environment with 400msec delay. In addition, a strange phenomenon was observed. As for route (a), we could only see 5.92 to 5.96Gbps as the output result of Iperf; hence, the performance was stable, but, as for route (b), there seems to exist in a periodical up and down of the performance.

Fig.12(a) to Fig.12(e) show the statistics of the traffic which each switch reports. At first, from 0:00am to 3:00am, 7:00am to 10:00am, we performed communication on route (a), then, from 10:00am to 13:00pm we tried route (b). (All time is JST.) And finally, from 13:00, route (c), 2 roundtrips from Tokyo to U.S. As is described in Fig.11, U-Tokyo - Note - T-LEX is the crossing point, and, we could see all traffics in Fig.12(d). Since JGN is a public network, there existed some other traffic, and Fig.12(e) shows the constant several hundreds Mega bytes back-ground traffic, such as IPv6 multicast.

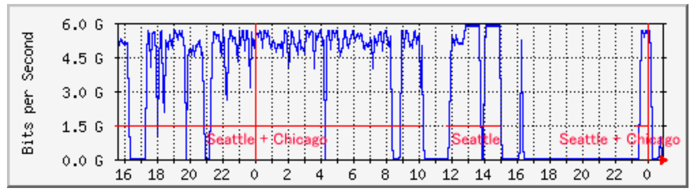
Oct.29, we tried to achieve real LFN experiment to use Tokyo - Seattle - Tokyo - Chicago circuit (Seattle - Chicago roundtrip). (Fig.13(a) to Fig.13(d)) Still 10:00am, we tuned for Seattle-Chicago roundtrip. We observed to have been continued a periodical problem. After 12:00pm, we changed route to Tokyo - Seattle - Tokyo circuit (Seattle roundtrip) to check the problem caused by physical circuits. End hosts could communicate by same rate without configuration change from Chicago Roundtrip route. Then we reduce transmission rate to 6Gbps but circuit rate is almost only 4.5Gbps average. From 23:00, we tried Seattle - Chicago roundtrip experiment again. It showed better performance than only Chicago roundtrip experiments. We suppose that the good circuit condition on Seattle roundtrip effects on the better result.

A. Tokyo - Seattle - Tokyo Roundtrip Experiment

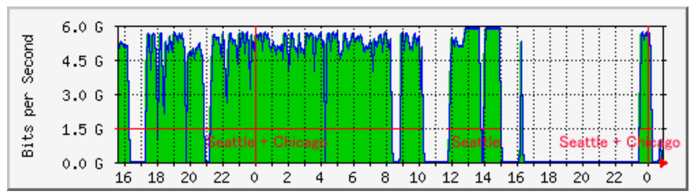
The route is from T-LEX to Seattle to U-Tokyo via IEEAF/WIDE, U-Tokyo to T-LEX via JGN. This route has



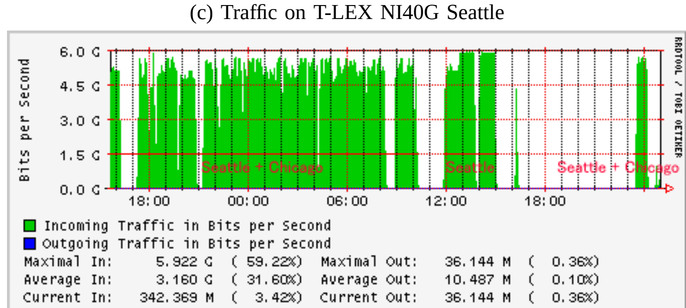
(a) Traffic on JGN GS4000 Chicago



(b) Traffic on T-LEX NI40G Tokyo



(c) Traffic on T-LEX NI40G Seattle



(d) Traffic on JGN GS4000 U-Tokyo

Fig. 13. Network Traffic (Oct 29, 2005)

small 172ms RTT. We tried the parameters as same as 100ms and 200ms RTT pseudo LFN.

At first, we observed some packet drop on the network. We reduced transmission rate toward 6Gbps (to minimum clock speed of the Chelsio)to be stable the communication. Other parameter is set same as 200ms pseudo LFN experiment. Window buffer size is 200MB on sender side, and 768MB on receiver side.

Then we got success for continues over 20 minutes run sustaining over 6Gbps. The starting up of the TCP stream is Fig.14. We compared this result with Fig.4. Fig.14 is different from the behavior after reach to the maximum rate. For reducing transmission rate, sender side transmission rate is not varied relatively. In this condition both sender/receiver hosts have room about CPU usage and I/O performance. The aimed rate of the route was over 6Gbps. This stream records average 5.94Gbps.

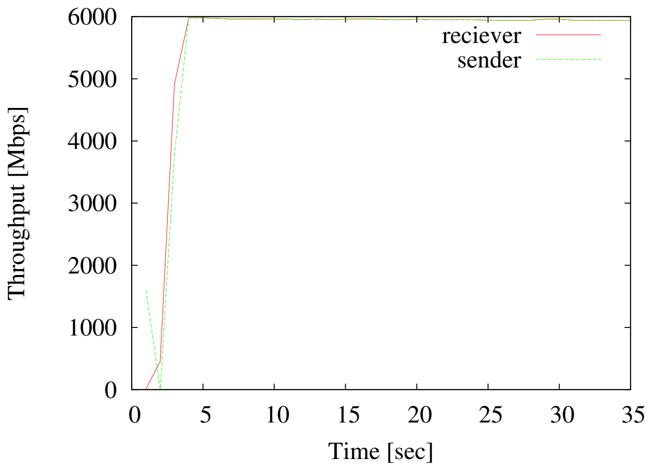


Fig. 14. Seattle roundtrip Performance

B. Tokyo - Chicago - Tokyo Roundtrip Experiment

Next, we tried Chicago Roundtrip route. This route had unstable condition. We observed periodically up/down phenomenon. (Fig.15) It is 900 second communication stream. First several 10secs and 100sec after 400 second performs stable high throughput. But from 30 second to 100 second and from 500 second to 650 second shows relative stable low performance, and the other shows unstable low performance. This behavior is not from TCP/IP congestion control or host performance. The trigger of the changing status is from varied condition of the circuit. Window Buffer size is 256MB on sender side and 896MB on receiver side. This line throughput is 5.6Gbps, lower than Seattle roundtrip.

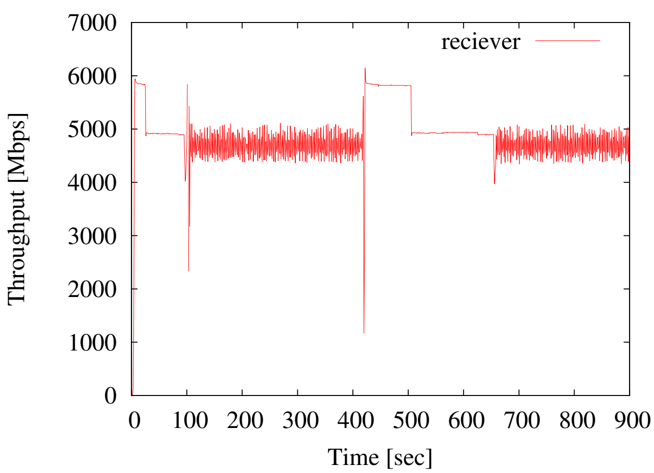


Fig. 15. Periodical UP/DOWN behavior

C. Tokyo - Seattle - Tokyo - Chicago - Tokyo Roundtrip Experiment

The route had the periodical problem above. But we could measure the performance in short time among the problem. The latency was larger than we have examined 400msec on pseudo LFN. We tried to enlarge the 400msec RTT parameter to fit the 500ms RTT Seattle - Chicago roundtrip circuit. Window size is 544MB on sender side, and 1072MB on receiver side. The stream spent 8 seconds for standing up to maximum rate. Its starting up time is larger than the result of 400ms, 5 seconds. 500msec RTT needs so large memory that sender side cannot send the maximum rate of network adapter. So we could not reduce the clock speed till minimum rate. The stream (Fig.16 shows the same behavior corresponding to pseudo 400ms RTT LFN macroscopically (Fig.15). The difference between result on pseudo 400ms and real 500ms is that the sender side shows pointed shape pushing to the network. It shows an influence of transmission rate control. The receiver side is not stable rather than Seattle roundtrip. This stream records average 5.58Gbps.

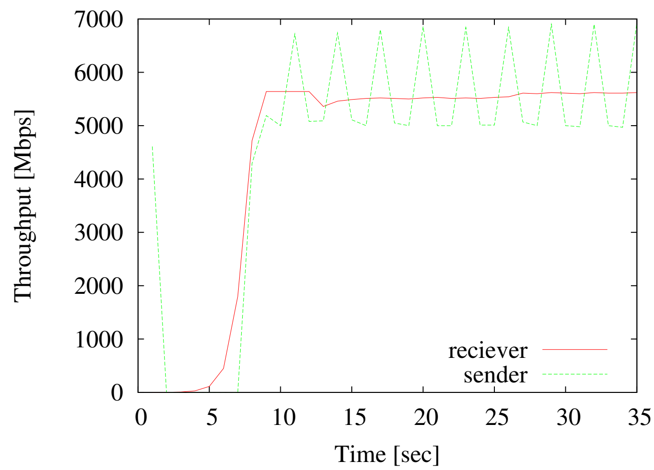


Fig. 16. Seattle - Chicago roundtrip Performance

Whole results on real LFN are lower than the results on pseudo LFN. We may need to take into their debugging status. The main causes come from worse condition on Chicago circuit than Seattle circuit. For this reason, we must limit the transmission rate down to 6Gbps.

But in limited condition, our optimization is effective on real LFN. We got stable TCP stream from limiting sender performance and flow control on both side.

We performed these experiments with the debugging network. For much detail comparison, it is necessary to be stable the network condition. We used inter-pacific network for roundtrip. The influence of roundtrip circuit may effect on the results. We need to compare the result on roundtrip LFN with the result on non-roundtrip circuits.

IV. CONCLUDING REMARKS

In this paper, we show the experimental results for data transfer on LFN. We attained 5.60Gbps for IPv6 TCP data transfer using Jumbo Ethernet Frame, which is the Internet2 Land Speed Record, IPv6 category. This result is lower than pseudo network. Sender side burst transmission and characteristic of real network influence on the result. The result shows that the real LFN is more difficult than pseudo LFN with artificial delay. One of the biggest differences is the packet loss, but, at the same time, burst of the packets is also the problem, which may be caused while transferring by the intermediate routers. This experiment also tells us that as for the data transfer on 10Gbps network using hosts server with PCI-X, a buffer for receiver may be useful, which is our future work.

ACKNOWLEDGEMENT

Special thanks to Prof. Akira Kato of University of Tokyo for useful discussions and networking coordination. We also thank Dr. Felix Marti and Dr. Wael Noureddine of Chelsio Communications Ltd., for controlling Network Interface Card. We also thank Dr. Itaru Mimura Mr. Hayashi and Mr. Yazaki of Alaxala Networks and Dr. Tomohiro Baba, Dr. Yasushi Fukuda and Dr. Yoichi Tsukioka of Hitachi Co. Ltd., and Mr. Hattori, Mr. Watanabe and Mr. Kurokawa of APAN. We also thank organizations and people StarLight, Tyco Telecommunications, IEEAF, Pacific Northwest Gigapop, CA*net4 networks JGN and WIDE project. This research is partially supported by the Special Coordination Fund for Promoting Science and Technology, and Grant-in-Aid for Fundamental Scientific Research B(2) #13480077 from Ministry of Education, Culture, Sports, Science and Technology Japan, Semiconductor Technology Academic Research Center (STARC) Japan, CREST project of Japan Science and Technology Corporation and by 21st century COE project of Japan Society for the Promotion of Science. We acknowledge the support of Global Crossing, Industry Canada, NTT Communications, and ITC of the University of Tokyo.

REFERENCES

[1] K. Hiraki, M. Inaba, J. Tamatsukuri, R. Kurusu, Y. Ikuta, H. Koga, and A. Zinzaki, "Data Reservoir: Utilization of Multi-Gigabit Backbone Network for Data-Intensive Research", SC2002, Nov. 2002. http://www.sc-2002.org/paperpdfs/pap.pap327.pdf

[2] M. Nakamura, M. Inaba, and K. Hiraki, "Fast Ethernet is sometimes faster than Gigabit Ethernet on LFN–Observation of congestion control of TCP streams" Proc. Int. Conf. on Parallel and Distributed Computing And Systems(PDCS2003) Nov. 2003 pp.854-859

[3] K. Hiraki, M. Nakamura, J. Senbon, Y. Sugawara, T. Itoh and M. Inaba, "End-node transmission rate control kind to intermediate routers - towards 10Gbps era" Second International Workshop on Protocols for Fast Long-Distance Networks(PFLDnet 2004) Feb. 2004

[4] H. Kamezawa, M. Nakamura, M. Inaba, K. Hiraki. "Coordination between parallel TCP streams on Long Fat Pipe Network" 1st International Workshop on Data Processing and Storage Networking: Towards Grid Computing" (DPSN04) May14th 2004 Athens, Greece pp.41-48

[5] Tsuyoshi Ito, Mary Inaba, "Theoretical Analysis of Performances of TCP/IP Congestion Control Algorithm with Different Distances", Lecture Notes in Computer Science, Springer-Verlag Heidelberg Volume 3042 / 2004 NETWORKING 2004, Third International IFIP-TC6 Networking Conference, pp962-973

[6] Yutaka Sugawara, Mary Inaba, and Kei Hiraki, "Over 10Gbps String Matching Mechanism for Multi-Stream Packet Scanning Systems" Lecture Notes in Computer Science, Springer-Verlag Heidelberg Volume 3203 Field-Programmable Logic and Applications, 14th International Conference, FPL 2004, Antwerp, Belgium, Aug29-Sept1,2004 pp484-493

[7] H. Kamezawa, M. Nakamura, Junji Tamatsukuri, Nao Aoshima, Mary Inaba, Kei Hiraki, Junichiro Shitami, Akira Jinzaki, Ryutaro Kurusu, Masakazu Sakamoto, and Yukichi Ikuta "Inter-layer coordination for parallel TCP streams on Long Fat pipe Networks" Super Computing 2004, High Performance Networking and Computing, SC2004 Pittsburgh

[8] M. Nakamura, R. Kurusu, Felix Marti, Wael Noureddine, M. Sakamoto, Y. Ikuta, Junji Tamatsukuri, Yutaka Sugawara, Nao Aoshima, Mary Inaba, and Kei Hiraki, "Experimental Results of inter-layer cooperative hardware for TRC-TCP on 10Gbps Ethernet WANPHY 18,500km Network", Third International Workshop on Protocols for Fast Long-Distance Networks(PFLDnet 2005) Feb. 2005