TCP CONGESTION WINDOW CONTROL ON AN ISCSI READ ACCESS IN A LONG-LATENCY ENVIRONMENT

Machiko Toyoda † Saneyasu Yamaguchi ‡ Masato Oguchi †
† Ochanomizu University Otsuka 2-1-1, Bunkyo-ku, Tokyo, Japan
‡ Institute of Industrial Science, The University of Tokyo Komaba 4-6-1, Meguro-ku, Tokyo, Japan
machiko@ogl.is.ocha.ac.jp sane@tkl.iis.u-tokyo.ac.jp oguchi@computer.org

ABSTRACT

As the broadband networks are widely used, IP-SAN is expected as the next generation's SAN because of its low cost. The iSCSI protocol, represented as IP-SAN technology, is becoming increasingly important. However, the performance of iSCSI network is lower than that of network based only on TCP/IP, due to its complex protocol processing in accessing storage with iSCSI.

In this paper, we present performance improvement using a dynamic Congestion Window control method to stabilize throughput unevenness. We evaluated iSCSI network performance with a remote storage access on iSCSI protocol using the proposed method. As a result, iSCSI performance improved compare with the case of not using the proposed method in a long-latency environment, and we confirmed the proposed method is effective.

KEY WORDS

iSCSI, Congestion Window Control, Remote Storage

1 Introduction

Because the performance of computer systems has been improved and Gigabit Ethernet is widely used, a large amount of data can be processed in a high speed. Various sorts of applications, which store mass volume of data as multimedia contents, have appeared also. As a result, the volume of data that computer systems process has increased remarkably. In order to consolidate storage, Storage Area Network (SAN) is introduced and becomes popular which reduces storage management cost.

FC-SAN, which is widely used already, connects server and storage with Fibre Channel. Fibre Channel has a light processing protocol, and data transmission load charged to server CPU is relatively low. However, in the case of FC-SAN, the compatibility of interconnection isn't necessarily high if the makers of FC devices are different. Moreover, both an installation cost and a management cost are relatively high because FC products are expensive. For this reason, IP-SAN configured with inexpensive Ethernet and TCP/IP is introduced. As a standard, iSCSI protocol is becoming important in IP-SAN. iSCSI, ratified by the IETF in February 2003, connects between server (Initiator) and storage (Target) with SCSI command. Consequently storage devices at remote place can be accessed as if they are attached directly.

For improving performance of a remote storage access using iSCSI protocol, we have proposed an idea of a storage access with dynamic Congestion Window control[1]. In this paper, iSCSI storage access performance is evaluated using the Congestion Window control method in which remote storage is accessed with iSCSI in network environments of various latencies. iSCSI network throughput is improved about 28% maximum in this experiment. We describe the iSCSI protocol behavior and discuss the effectiveness of our proposed method.

The rest of this paper is organized as follows. Section 2 covers study background, and section 3 introduces dynamic Congestion Window control method. In section 4, we show the experimental result on iSCSI sequential read access using the proposed method in the environment of various latencies, and evaluate performance of the proposed method. Section 5 describes the packet behavior of iSCSI protocol in a long-latency environment, and discusses how it affects the performance. Section 6 covers related works, and section 7 presents the conclusion.

2 Background of Our Research Works

2.1 Linux TCP Implementation

Transmission Control Protocol (TCP) uses a concept of Congestion Window (CWND). CWND is a parameter, which limits the behavior of a data sender, for the purpose of the network congestion control. That is to say, CWND means the number of maximum packets that can be sent consecutively without receiving a reply packet of Acknowledgement (ACK) from a data receiver. CWND increases one whenever a data sender receives one ACK. In Linux OS used in our experiment, CWND increases every time the data sender receives one ACK if the state of communications is judged as normal. However, If TCP implementation detects an error and decides the state of communications as unusual, CWND reduces dramatically. The cases in which CWND reduces are as follows (Figure 1).

1. CWR: Detecting Local Congestion error in which device driver buffer of the data sender overflows.
2. Recovery: Receiving duplicated ACKs or SACK.

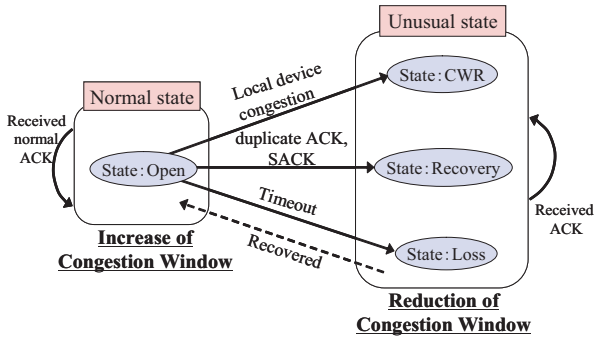

Figure 1. The state transition of Linux TCP implementation

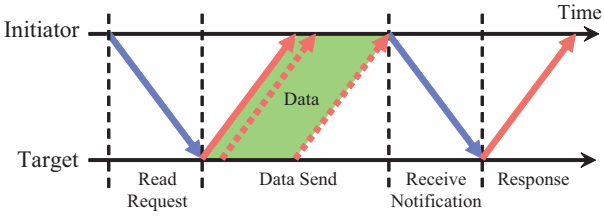

Figure 2. The sequence of iSCSI sequential read access

3. Loss: Detecting timeout.

Linux TCP implementation doesn't change the window size as long as the value of CWND, which has been set during the communication, isn't consumed completely before receiving ACKs. In such a case, we confirmed the throughput remains stable.

2.2 Issues of Accessing Remote Storage using iSCSI

iSCSI encapsulates SCSI command, which is widely used in Direct Attached Storage (DAS), within a TCP/IP packet and transports it on a TCP/IP network[2][3]. In FC-SAN, the limit of the communication distance is about 10km. In iSCSI, on the other hand, communication distance isn't limited because it uses TCP/IP network, and thus iSCSI is expected to be used for long distance communications such as Storage Outsourcing and Data Backup to a data center.

Figure 2 is a packet transport sequence on the iSCSI sequential read access. SCSI Command PDU meaning a Read request in the iSCSI layer is sent ("Read Request" in figure 2) after the read system call in the application has been issued. The block size for the transport request is written in SCSI Command PDU. Therefore, Target sends data of the requested size successively ("Data Send" dotted arrow in figure 2), after Target that has received SCSI Command PDU replies SCSI Data-In PDU ("Data Send" solid arrow in figure 2). Initiator returns ACK for the arrived data in the TCP layer ("Receive Notification" in figure 2). Target sends SCSI Response PDU as a response packet ("Response" in figure 2) when ACK indicating the arrival of the last data at Target, this is the end of one cycle of the Read command.

If we perform communications using a socket in a network based only on TCP/IP (in the rest of this paper, this is referred to as "socket communication"), the TCP self clocking mechanism works as time has passed and the TCP implementation sends data little by little at the reception timing of ACKs. Therefore the sending speed of packets is controlled properly. In iSCSI, on the other hand, the burst of packet transmission doesn't disappear because the data is sent all together after Target has received a Read request. Consequently, CWR error in iSCSI is easy to occur in comparison with the socket communication, and the growth of CWND is also less than that of the socket communication. Figure 3 shows the change of CWND in both socket communication and iSCSI communication cases when the delay time is 16ms. In this figure, all the causes of CWND reduction are due to the CWR error. This figure indicates that CWND of the socket communication grows up to a large value in most cases. However, CWND of the iSCSI communication doesn't grow up. In our paper [4], we have found out that throughput has a close relationship with the size of CWND, and throughput is unstable in the case that CWND increases and decreases repeatedly. When we perform a remote storage access using iSCSI, it is important for the better performance that frequency of CWR error should be reduced and CWND should be kept as a large value.

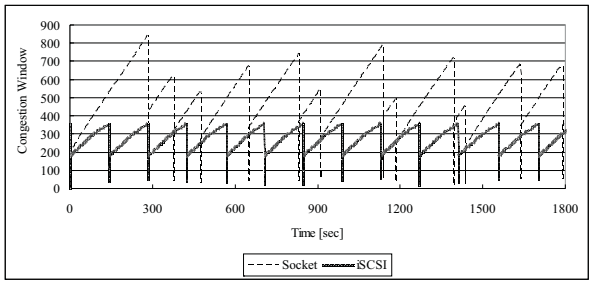

Figure 3. CWND in both socket communication and iSCSI communication cases

3 Dynamic Congestion Window Control Method

In this section, we introduce the dynamic congestion control method, which balances the uneven behavior of throughput observed on iSCSI storage access, proposed for improving the performance.

As described in the previous section, particular care must be taken for iSCSI because the behavior of iSCSI network is different from that of TCP/IP network. Since we have noticed a close relationship between the size of CWND and throughput, we have proposed the method that stabilizes increase and decrease of CWND by controlling the access block size dynamically[1]. Figure 4 shows the outline of our proposed method.

Generally, user programs can't recognize the size of CWND because CWND is controlled in a Kernel space. Therefore we have inserted monitor functions in TCP source code and implemented a recording mechanism of

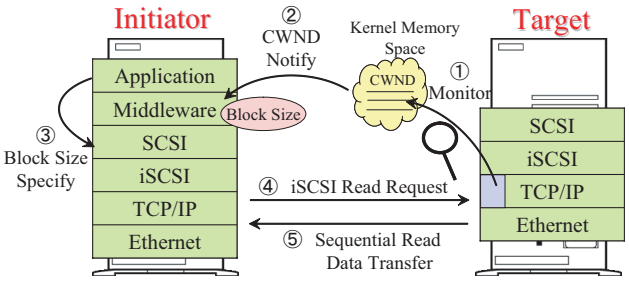

Figure 4. The concept of CWND control method

TCP parameters within a Kernel memory space, so that they are accessible from User space. With this mechanism, we can confirm TCP parameters by reading a special file for accessing Kernel memory space. In the CWND control method, we have implemented this mechanism on Target. An application of Initiator adjusts the block size of storage access by receiving CWND notification from Target. The controlling process of our proposed method is as follows.

1. Target monitors CWND and observes its change.
2. Target notifies the size of CWND when CWR is detected and CWND reduces, and notifies the limit size of CWND (the max size of CWND without CWR error) when CWND remains stable. In addition, Target recodes the size of notified CWND.
3. When Initiator receives the notification, the middleware decides the block size based on CWND and the application on Initiator modifies the block size.
4. Initiator sends a sequential read command for Target.
5. Target transmits the data of requested block size.
6. This process iterates whenever Target detects CWR or decides that CWND is stable.

After applying the proposed method, CWND stays at the value of a limit, in which CWR error doesn't occur. Consequently the block Size is the optimized value calculated by the proposed method. In the proposed method, the specified block size calculated in the middleware is as follows.

Transmission Block Size [byte] = the Size of CWND $\times$ *Maximum Transmission Unit (MTU)*

The size of MTU used in our experiment is 1448Byte except for the TCP/IP header (including the option) from the maximum segment length of Ethernet (1500Byte). The influence of the overhead of monitor on the performance is little because the CWND monitor on Target performs at intervals of a few seconds.

4 Experiment for Performance Evaluation using Congestion Window Control Method

In this section, we have experimented for evaluating iSCSI performance with the Congestion Window control method in network environments of various latencies. We have compared the cases in which the proposed method is used and not used.

Table 1. Experimental machines

CPU	Intel Xeon 2.4GHz
Main Memory	512MB DDR SDRAM
OS	Initiator, Target: Linux2.4.18-3 Dummynet: FreeBSD 4.9 - RELEASE
NIC	Initiator, Target: Intel PRO/1000XT Server Adapter Dummynet: Intel PRO/1000MT Server Adapter

4.1 Experimental Setup

Our experimental system is as follows. We have established TCP/IP connection with Gigabit Ethernet between Initiator and Target. 1000Base-T Switching Hub is inserted in the case of experiment with no-delay and FreeBSD Dummynet[5] as an artificial delay machine is inserted in the case of experiment with delay, between Initiator and Target, respectively. Initiator, Target, and Dummynet are constructed on Personal Computers. We have used Linux as an OS of Initiator and Target, and FreeBSD as an OS of Dummynet. In order to study the network performance of an iSCSI storage access, Target has been set up with the memory mode in which a disc access isn't executed actually. Table 1 shows our experimental machines.

As an iSCSI implementation, we have used UNH IOL reference implementation ver.3 on iSCSI Draft 18[6] offered from the University of New Hampshire InterOperability Laboratory on the Target machine. In this UNH implementation, even if we issue a read command of large block size, the SCSI layer divides the requested block into a small block size. For this reason, the performance of storage access using iSCSI is degraded[7]. In our experiment, instead of using the UNH implementation on the Initiator machine, we have used an Initiator program written by ourselves which has equivalent functions to the UNH implementation of Initiator and can perform data transfer with a large block size. This Initiator program written by ourselves works as an application in the user space and communicates with Target iSCSI protocol on a TCP/IP connection.

4.2 Outline of the Experiments

Our experiment measures the performance of iSCSI sequential read access from Initiator to a raw device of Target in network environments of various latencies. The block size specified in Initiator is 1024KB when the proposed method is not used, and the initial value of the block size specified in Initiator is 1024KB also when the proposed method is used. The total data size reading from Target is 100GB.

4.3 Experimental Result not using the Proposed Method

As an experimental result not using the proposed method, Figure 5 shows CWND, block size, and throughput when one-way delay time is 16ms. In this figure, all the causes of CWND reduction are due to the CWR error.

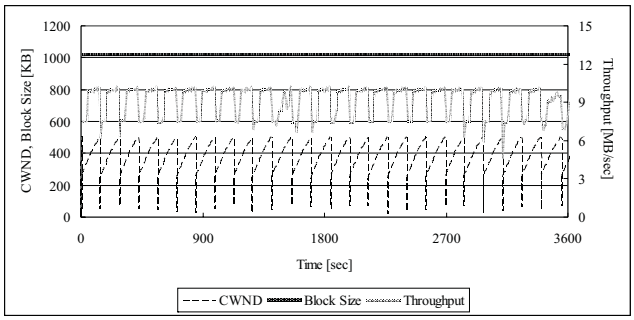

Figure 5. CWND, the block size, and throughput (one-way delay time: 16ms)

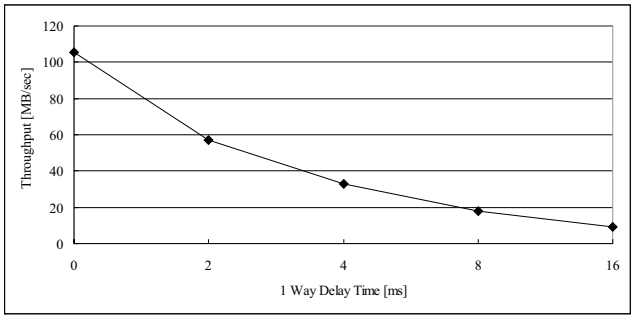

Figure 6. The average throughput

Throughput changes dramatically as CWND increases and decreases. We have compared throughput when CWND achieves the maximum value with throughput after CWND has decreased. As a result, the difference of both cases is about 3MB/sec. This difference is relatively large in a network environment, in which the average total throughput is about 10MB/sec. Figure 6 shows the average throughput when Initiator reads the 100GB data on an environment with various delay time. As delay time increases, the performance of iSCSI access decreases remarkably. In our experiment, because this is a read access without accessing to a disc, the result is only the network performance of iSCSI. This is the performance of a limit, which iSCSI protocol can provide, in a long-latency environment.

4.4 Experimental Result using the Proposed Method

CWND, block size, and throughput when the experiment uses the proposed method is shown in Figure 7 (delay time: 8ms), 8 (delay time: 16ms). When Target detects CWR error, the middleware of Initiator works and controls CWND by changing the block size for accessing. Different from the case of communications not using the proposed method, the block size is set up to be a slightly small value. However, after CWND has been stabilized throughput, has improved greatly compared with the case when CWND has been greatly changing.

Figure 9 shows the average throughput when Initiator reads 100GB data on an environment with various delay time. In addition, Figure 10 shows the graph that compares the average throughput of three cases, when CWND

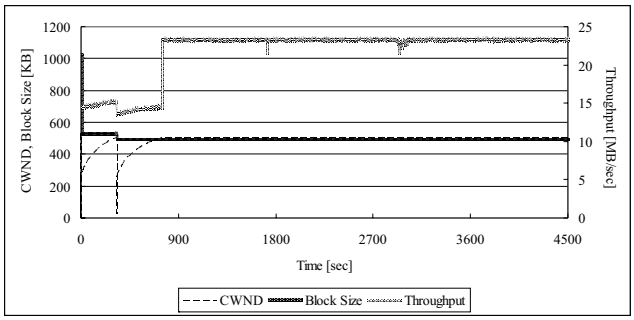

Figure 7. CWND, the block size, and throughput using the proposed method (one-way delay time: 8ms)

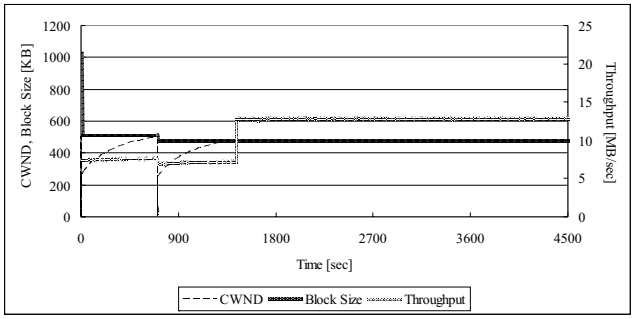

Figure 8. CWND, the block size, and throughput using the proposed method (one-way delay time: 16ms)

is controlled, when CWND isn't controlled, and after controlled CWND is stabilized. In the total performance, as delay time increases, we confirm that the performance decreases greatly from Figure 9. However, when we compare the throughput of using the proposed method with that of not using, we can confirm from Figure 10 that controlling CWND is effective to prevent the reduction of performance. If the total data size of the sequential read access is larger, the difference of performance between the case of controlling CWND and that of not controlling CWND should to be larger. In the communication environment with small latency, we had better transmit data using a large block size. On the other hand, in the environment with long latency, we had better control the increase and decrease of CWND, so as to keep to be stable.

5 Communications Control by iSCSI Protocol Behavior and Its Effect to Performance

In this section, we explain the packets' behavior of iSCSI sequential read access in a long-latency environment, and discuss why CWND control method is effective in such an environment.

According to the experimental result of the previous section, we have obtained that, the performance in a long-latency environment has improved by keeping CWND stabilized even if the block size is slightly smaller. We had thought that a larger block size had better performance because the block size is defined as a data length sent at one time. However, if the block size is too large, CWR error oc-

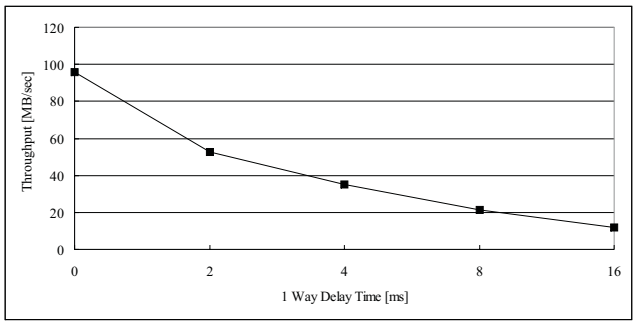

Figure 9. The average throughput using the proposed method

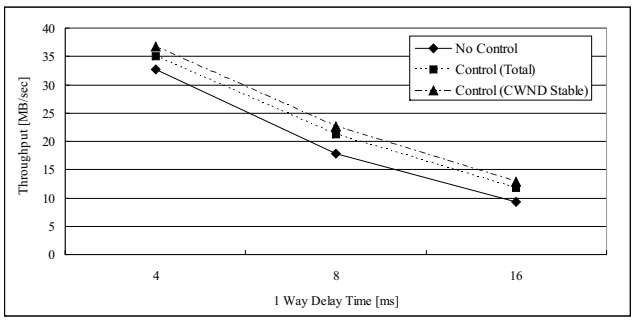

Figure 10. The comparison of average throughput in long-latency environment

curs because a send buffer of a data sender overflows. As a result, CWND decreases greatly since TCP implementation decides that the sent data is too large. In a long-latency environment, if CWND is small, a data sender does nothing but wait for ACK for a long time.

Figure 11 and 12 show the iSCSI packets' behavior when we perform iSCSI sequential read access in a short-latency and a long-latency environments. The iSCSI packets' behavior shown in these figures and Figure 13 focuses on the part of "Data Send" in Figure 2 in detail. Solid arrows from Target to Initiator represent packets including data labeled as "Data", and dotted arrows from Initiator to Target represent ACK. The number of data arrows is equal to CWND. When we perform an iSCSI sequential read access in a short-latency environment, the waiting time for the data transmission in Target is short (Figure 11) because ACK is replied soon. However, the response time becomes long in a long-latency environment. Consequently, the arrival of ACK to the sent data is late and the waiting time for the data transmission in Target increases greatly (Figure 12). Without the proposed method, the number of sent packet is different each time because CWND is changing. On the other hand, Figure 13 shows the iSCSI protocol behavior using the proposed method in a long-latency environment. CWND is stabilized with the proposed method, thus, the number of sent "Data" packets from Target to Initiator is kept to be the same. In this case, Target can keep transmitting the maximum number of packets within the range where CWR error doesn't happen. As a result, Target can

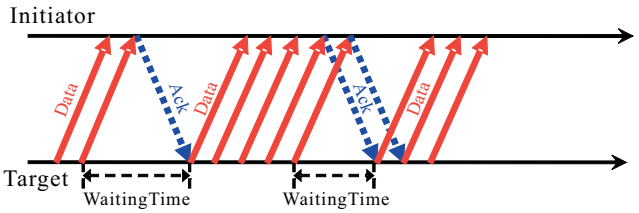

Figure 11. The sequential read access on short-latency environment

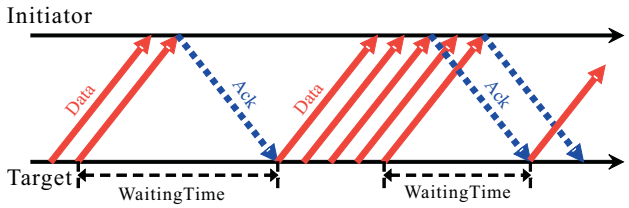

Figure 12. The sequential read access on long-latency environment

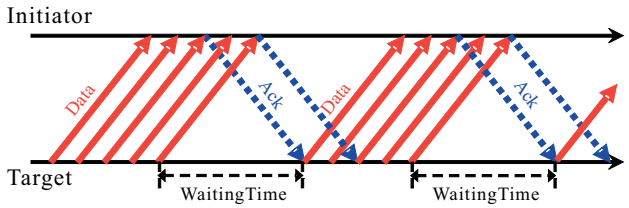

Figure 13. The sequential read access using the proposed method on long-latency environment

transmit the data efficiently, even if the block size is slightly smaller in an iSCSI sequential read access. In the case of not controlling CWND shown in Figure 12, the waiting time is short if the number of sent packets at one time is large. However, CWR error happens regularly so that CWND is reduced drastically. While CWND is reduced, Target is kept waiting for a long time after it transmits a few packets until ACK is received. Due to this inefficient behavior, iSCSI performance decreases dramatically in a long-latency environment. That is to say, it is important to prevent the reduction of CWND as the delay time becomes longer.

Table 2 shows the rate of throughput increase of the experimental result using the proposed method against that not using it. When the delay time is short, the communication performance decreases a little using the proposed method because the block size is slightly small. In contrast, when the delay time is long, the communication performance improves using the proposed method. The rate of improvement achieves about 28% when the delay time is 16ms, which is the case of longest delay in this experiment.

6 Related Work

As related works of iSCSI, P.Sarkar et al.[8] compares iSCSI software implementation with iSCSI hardware implementation. P.Radkov et al.[9] compares iSCSI perfor-

Table 2. Performance improvement of the proposed method

1 Way Delay Time	Throught Improvement
0ms	-9.1%
2ms	-7.9%
4ms	7.0%
8ms	19.7%
16ms	28.3%

mance with NFS performance. However, these studies just measured the overall system performance and didn't analyze the detailed behavior inside the system.

P.Gurumohan et al.[10] indicates that iSCSI performance reduces due to the excessive processing redundancy over several protocol layers, and proposes a method of data handling in the form of fixed data units called "quanta". Although their viewpoint is the same as ours in terms of a cause of iSCSI performance decrease, the approach for improving performance is different.

R.Takano et al.[11] and K.Kumazoe et al.[12] pay attention to the technique proposed as new TCP that modifies the existing implementation, describe the performance evaluation and discuss the CWND behavior for communications. Their standpoint of research works in which throughput has a close relationship with the size of CWND is the same with ours. However, the method described in these papers improves the performance by modifying the existing TCP. Therefore, these studies are different from ours that intends to improve the performance by using existing TCP used generally and widely.

7 Conclusion

In this paper, we have evaluated the performance of iSCSI sequential read access in network environments with various latency using a dynamic Congestion Window control method to stabilize throughput unevenness. Because iSCSI protocol performs the communication that produces burst traffic, the CWND reduction degrades the performance as delay time becomes long. When CWND is kept to be stabilized in a long-latency environment, throughput has improved about 28% maximum. Moreover, we have described a packet behavior of iSCSI protocol in a long-latency environment, and discussed the effectiveness of the proposed method.

As a part of future works, we will evaluate an iSCSI write access and an iSCSI random access using the proposed method. We should evaluate the fairness of our method when multiple TCP sessions are used between Initiator and Target. Also, an evaluation of the packet loss case will be our future work.

Acknowledgment

This project is partly supported by the Ministry of Education, Culture, Sports, Science and Technology, under Grant 13224014 of Grant-in-Aid for Scientific Research on Priority Areas.

References

[1] M. Toyoda, S. Yamaguchi, and M. Oguchi: "A Study of Performance Improvement by Controlling TCP Congestion Window on iSCSI Access," *IEICE Technical Reports, CPSY2004-50*, December 2004, pp.1-6.

[2] iSCSI Specification, http://www.ietf.org/rfc/rfc3720.txt?number=3270/

[3] SCSI Specification, http://www.danbbs.dk/˜dino/SCSI/

[4] M. Toyoda, S. Yamaguchi, and M. Oguchi: "Relationship between TCP Congestion Window and System Performance on iSCSI Storage Access," *Proc. 3rd Forum on Information Technology (FIT2004)*, B-004, September 2004, pp.107-109.

[5] L.Rizzo: "dummynet", http://info.iet.unipi.it/˜luigi/ip_dummynet/

[6] InterOperability Lab: Univ, of New Hampshire, http://www.iol.unh.edu/consortiums/iscsi/

[7] S. Yamaguchi, M. Oguchi, and M. Kitsuregawa: "iSCSI Analysis System and Performance Improvement of Sequential Access in Long-Latency Environment," *IEICE Transaction on Information and Systems, Vol.J87-D-I, No.2*, February 2004, pp.216-231.

[8] P.Sarkar, S.Uttamchandani, and K.Voruganti: "Storage over IP: When Does Hardware Support help?," *Proc. FAST 2003, USENIX Conference on File and Storage Technologies*, January 2003, pp.231-244.

[9] P.Radkov, L.Yin, P.Goyal, P.Sarkar, and P.Shenoy: "Performance Comparison of NFS and iSCSI for IP-Networked Storage," *Proc. FAST 2002, USENIX Conference on File and Storage Technologies*, March 2004, pp.101-114.

[10] P.Gurumohan, S.Narasimhamurthy, and J.Hui: "Quanta Data Storage: A New Storage Paradigm," *Proc. 12th NASA Goddard Conference on Mass Storage Systems and Technologies*, April 2004, pp.101-107.

[11] R. Takano, Y. Ishikawa, T. Kudoh, M. Matsuda, Y. Kodama, and H. Tezuka: "The Analysis of TCP/IP Communication Behavior on Parallel Applications," *Internet Conference 2003*, October 2003.

[12] K. Kumazoe, Y. Hori, M. Tsuru, and Y. OIE: "Transport Protocols for Fast Long-Distance Networks: Comparison of Their Performances in JGN," *IEICE Technical Reports, NS2003-354, IN2003-309*, March 2004, pp303-308.