Simulation Study of iSCSI-based Storage System*

Yingping Lu, Farrukh Noman, David H.C. Du
Department of Computer Science & Engineering,
University of Minnesota
Minneapolis, MN 55455
Tel: +1-612-625-4002, Fax: +1-612-625-0572
Email: {lu, noman, du }@cs.umn.edu

Abstract

iSCSI is becoming an important protocol to enable remote storage access through the ubiquitous TCP/IP network. Due to the significant shift in its transport mechanism, iSCSI-based storage system may possess different performance characteristics from the traditional storage system. Simulation offers a flexible way to study the iSCSI-based storage system. In this paper, we present a simulation work of iSCSI-based storage system based on ns2. The storage system consists of an iSCSI gateway with multiple targets, a number of storage devices that are connected to the storage router through FC-AL and clients, which access the target through iSCSI protocol. We present the system model, the implementation of the components, the validation of the model and performance evaluation based on the model. Coupled with the rich TCP/IP support from ns2, the simulation components can be effortlessly used for the performance and alternatives study of iSCSI-based storage systems, applications in broad configurations.

1. Introduction

The iSCSI protocol [1][2] has emerged as a transport for carrying SCSI block-level access protocol over the ubiquitous TCP protocol. It enables a client's block-level access to remote storage data over an existing IP infrastructure. This can potentially reduce the cost of storage system greatly, and facilitate the remote backup, mirroring applications, etc. Due to the ubiquity and maturity of TCP/IP networks, iSCSI has gained a lot of momentum since its inception.

On the other hand, the iSCSI-based storage is quite different from a traditional one. A traditional storage system is often physically restricted to a limited environment, e.g. in a data center. It also adopts a transport protocol specially tailored to this environment, e.g. parallel SCSI bus, Fibre Channel, etc. These characteristics make the storage system tend to be more robust, and achieve more predictable performance. It is much easier to estimate the performance and potential bottleneck by observing the workload. While in an iSCSI storage, the transport is no longer restricted to a small area. The initiator and the target can be far apart. The networking technology in between can be diverse and heterogeneous, e.g. ATM, Optical DWDM, Ethernet, Wireless, satellite, etc. The network condition can be congested and dynamically changing. Packets may experience long delay or even loss and retransmission, etc. Thus, the situation facing the iSCSI storage is quite different from the traditional one.

To take advantage of iSCSI protocol to build iSCSI storage systems, we need to better understand the iSCSI characteristics, e.g. the performance characteristics in various

* This work was supported by DISC from DTC of UoM, and gifts from Intel and Cisco.

networking situations, the impact of network transmission error or network component fault to the storage access performance and robustness, the relationship between iSCSI and underlying TCP/IP protocol, etc.

The common way to study the performance characteristics about the iSCSI storage system is through the real performance measurement. Real measurement can be pretty accurate. Papers [4][5][6] represent this endeavor. However, the measurement approach is often restricted to the physical equipments and settings. In a lot of times, the software or hardware of the equipment is not open. Thus a tester cannot adjust parameters, or try an alternative algorithm, etc. in the performance study. In this regard, a simulation approach offers much more flexibility. Once the simulation components have been implemented, it is very easy to construct test configuration, configure parameters of interest, or add an alternative algorithm, etc. to study the iSCSI related issues.

To our best knowledge, no simulation model has been built for iSCSI protocol. Our goal of this work is to establish a simulation model for iSCSI-based storage system for the study of the characteristics of iSCSI-based storage system. In addition, we also study the interactions between the iSCSI and TCP layer to better support the iSCSI access.

We use network simulator NS2 [9] to implement the iSCSI simulation model. NS2 is an event driven simulator widely used in the research of networking arena. It provides substantial support for the simulation of TCP/UDP, routing, and multicast protocols over wired and wireless networks. To validate the simulated model, we also conduct the real performance measurement and compared the simulation results with the real performance data. In addition, we also examine the different TCP parameters and investigate how they affect the iSCSI performance based on the simulation model.

This paper is organized as follows: Section 2 presents the simulation model for the iSCSI-based storage system. Section 3 describes the iSCSI implementation in NS2. Section 4 presents empirical validation of the model. In Section 5, we analyze the performance results of iSCSI model. Section 6 reviews the related work. Finally we conclude this paper in Section 7.

2. Simulation Model

2.1. A Typical iSCSI Storage System Model

Figure 1 shows a typical iSCSI-based storage system model used in the simulation. In this model, an initiator generates SCSI requests, which is encapsulated into iSCSI messages (protocol data unit or PDU). These PDUs are then transmitted over TCP/IP network and routed to an iSCSI storage gateway.

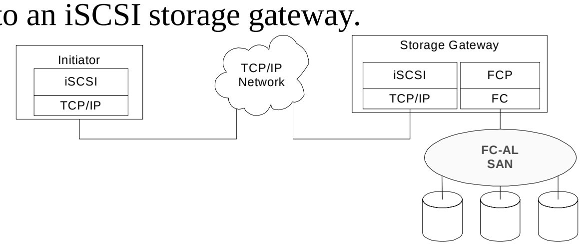


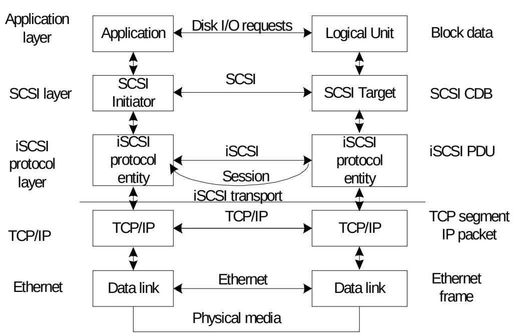
Figure 1 The storage system model

The storage gateway has both TCP/IP networking interface and FC-AL interface. TCP/IP interface provides iSCSI connection for an initiator to access through IP network, while FC-AL interface is used to connect to a SAN storage subsystem. In this SAN environment, the gateway serves as an initiator to the FC-enabled disks. It uses the SCSI over FC encapsulation protocol (FCP) to access the disk devices through Fibre Channel.

2.2. The iSCSI Data Transfer Model

Figure 2 shows the iSCSI architecture model. iSCSI builds on top of TCP transport layer. For an iSCSI initiator to communicate with a target, they need to establish a session between them. Within a session, one or multiple TCP connections are established. The data and commands exchange occurs within the context of the session.

Figure 3 shows iSCSI command execution by illustrating a typical Write command. The execution consists of three phases: Command, Data and Status response. In the Command phase, The SCSI command (in the form of Command Descriptor Block (CDB)) is incorporated in an iSCSI command PDU. The CDB describes the operation and associated parameters, e.g. the logical block address (LBA) and the length of the requested data. The length of the data is bounded by a negotiable parameter “MaxBurstLength”. During the Data phase, the data PDUs are transmitted from an initiator to a target. Normally, the initiator needs to wait for “Ready to Receive (R2T)” message before it can send out data (solicited data). However, both initiator and target can negotiate a parameter “FirstBurstLength” to speed up the data transmission without waiting. FirstBurstLength is used to govern how much data (unsolicited data) can be sent to the target without receiving “Ready to Receive (R2T)”. A R2T PDU specifies the offset and length of the expected data. To further speed up the data transfer, one data PDU can be embedded in the command PDU if “ImmediateData” parameter is enabled during the parameter negotiation. This should be very beneficial for small write operation. The Status PDU is returned once the command is finished.



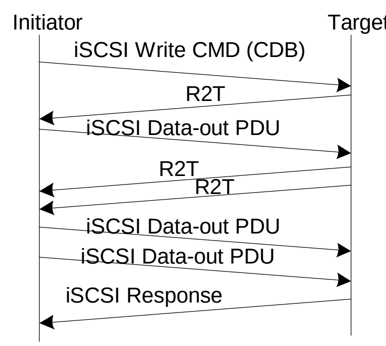


Figure 2. The iSCSI model Figure 3. The command execution sequence

Finally, these messages are encapsulated into TCP/IP packets, where the packet size is bounded by MSS (maximum segment size) in TCP. MSS is determined by the smallest frame size along the path to the destination. Within an Ethernet LAN, the maximum frame size is 1500 bytes (Gigabit Ethernet supports Jumbo frame). The MSS is 1460 bytes (40 bytes for IP and TCP header). When the iSCSI PDU size is greater than the segment size, the PDU is further fragmented into smaller packets.

The size of iSCSI parameters like: MaxBurstLength, FirstBurstLength and PDU size all have certain impact to the iSCSI performance. However, the iSCSI performance is also significantly affected by the underlying TCP flow control, congestion control mechanism. We will also examine the effect of these parameters.

2.3. Disk Model

We use *Seagate ST39102FC Cheetah 9LP* disk as the storage device. The disk access time $T_d = T_{ds}+T_{dr}+T_{dt}$, where T_{ds} is the disk seek time, which is determined by the difference of current cylinder and the target cylinder; T_{dr} is the disk rotation latency, which is determined by the difference of the current sector when the disk head moves to the target cylinder and the first sector of the intended access. Disk transfer time T_{dt} is determined by the number of data blocks transferred. When the data size is large, the requested data may span more than 1 track (cross disk surface) or even 1 cylinder. In that case, we also add the head switch or cylinder switch time into the access time. We consider the disk has enough buffers to hold the requested data.

The disk not only handles the block data access, it also handles the data transmission. The disk has built-in FC-AL logic and interface. As a normal FC node, it has a physical address and needs to participate the arbitration phase to win the channel before transferring data between the gateway and the disk.

2.4. FC-AL Interconnect

Fibre Channel is a popular networking protocol to construct storage area network. It supports switching fabric, loop and point-point construct. FC-AL aims at loop topology where up to 126 FC-AL nodes (hosts and disk devices) are connected to a shared loop. A node obtains the access to the loop through arbitration. The arbitration is determined by the physical addresses of participating nodes. When a node wins arbitration, it opens a connection to its destination node. A node closes a connection and releases the control over a loop when its transfer has finished. In our environment, the disk devices and storage gateways are FC-AL nodes.

On top of the Fibre Channel transport, similar to iSCSI, FCP protocol maps SCSI protocol onto the Fibre Channel protocol. Each SCSI protocol is also performed through three phases, FCP-CMD frame for command transfer, FC-Data frame for data transfer, and FCP-response frame for status transfer. Since the maximum frame size is 2048bytes, a SCSI command may require multiple FCP-Data frames to transfer all requested data. For the SCSI Write operation, FCP-XFER-Ready frame is also used for the flow control between the initiator and disk device.

The speed of FC-AL can be 1Gb/s, up to 2 Gb/s. Since it adopts 8B/10B encoding, the actual bandwidth is 100MB/s (Mega bytes per second) and 200MB/s respectively. In our simulation, we assume the speed to be 1Gb/s. We use a central module FC to handle the channel arbitration. All participating nodes (disk devices, the gateway) are required to register to this module. When a node requires channel, it submits a request to FC. FC module determines who wins the arbitration.

2.5. Storage Gateway Model

The storage gateway works as a bridge between two protocols: iSCSI and FCP. It hosts the targets of the iSCSI and the initiators of the FC storage. In the meantime, it manages the targets, their access control and their related sessions. It also administers the mapping between a target and its constituent disk devices.

In the simulation, the disk devices should be “attached to” (add an pointer in) a target. Each target maintains two interfaces: iSCSI on top of TCP agent and the FCP on top of Fibre Channel interface. An outstanding command queue for the each session glues these two interfaces together. Each command item in the queue contains the CDB and other status information. When a new command arrives from iSCSI interface, it is placed into the command queue. The command is further sent to the disk device through the FCP interface when the number of outstanding commands falls below the threshold in the target disk. When a command completes, it receives an FCP-RSP frame from the disk. Upon receiving this frame, the target sends out an iSCSI Response PDU to the actual initiator, in the mean time, the command is removed from the queue. However, the commands within a session are completed in order.

3. Simulating iSCSI in NS2

3.1. iSCSI Nodes

In our simulation there are three types of nodes: Initiator, Target and gateway node. Figure 4 shows these nodes and their related components. A target node is the peer of an initiator node. The gateway node hosts one or multiple target nodes.
iSCSIInitiator and iSCSITarget are the node applications running on the initiator and target respectively. Within these nodes, iSCSIInitiatorSession and iSCSITargetSession, derived from iSCSISession, perform the session tasks. Similarly, iSCSIInitiatorConnection and iSCSITargetConnection, derived from iSCSIConnection, perform connection tasks. A iSCSISession can open multiple iSCSIConnections.

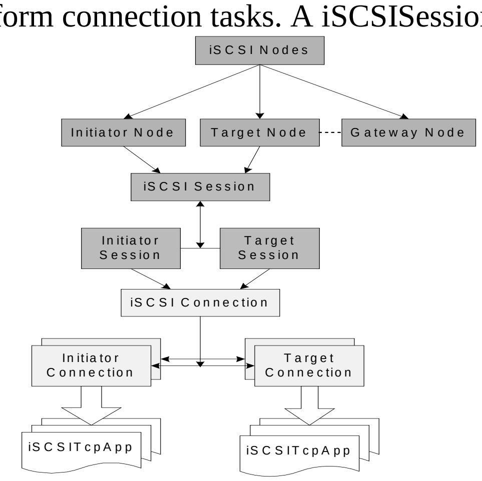


Figure 4. Hierarchy of iSCSI Node

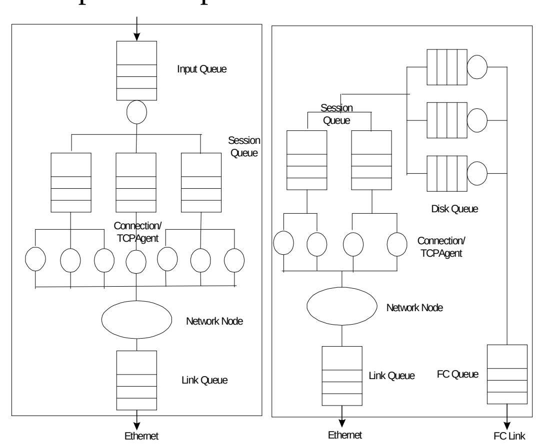


Figure 5 Queuing models (a) Initiator (b) Target

Each iSCSIConnection has its own iSCSITcpApp object through which data is sent and received. We use FullTCP agent for the iSCSIConnection application.

3.2. Queuing Models

Figure 5 shows the queuing models in the simulation. Fig. 5(a) is the model for an initiator. It has an input queue to receive workload (SCSI requests). Under the input queue are several Session queues. Each session possesses a queue for the outstanding SCSI commands. The maximum number of outstanding commands is configurable. The commands in each session are processed by their corresponding connections and then passed to their TCP agents. The link queue at the bottom is the network node's link queue.

Fig. 5(b) shows the Target's queuing model. Four types of queues exist: session queue, link queue, disk queue and FC queue. Similar to the Initiator node, link queue is for the Ethernet link. Per-session queue is for outstanding commands. Each disk has a disk queue, which holds outstanding commands for each disk in the target. Disk queue makes the interaction between the target and disk simpler. FC queue holds the requests to access the FC link. Moreover, each disk itself also has a command queue. The number of outstanding commands is configurable.

3.3. Implementation

Figure 6 shows a typical setting of an iSCSI system implemented in NS2. The system comprises three parts: the initiator, which generates workload; the TCP/IP network, which can be easily configured based on existing NS2 components; and the gateway in conjunction with target disk devices.

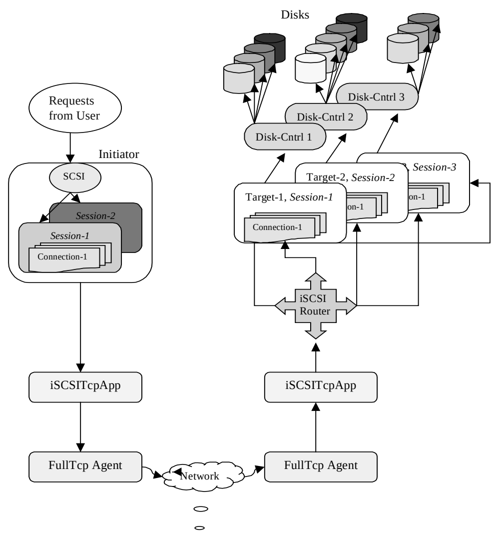


Figure 6. iSCSI system based on NS2

All these components are implemented in C++ to achieve high efficiency. The Otcl script in NS2 is used to setup the simulation environment. It creates network nodes and network topology, creates initiator and gateway nodes, creates targets and disks and attaches disks to target. These C++ components also expose a number of configurable parameters such as iSCSI parameters, disk parameters, to the Otcl. This makes the change of test setting very convenient.

After the test setting is constructed, the script then invokes login method in the initiator to connect to the gateway. The initiator enters Login phase. Eventually it acquires the targets and LUNs in the target from the gateway. The initiator finally creates a session with each target. The number of sessions and the number of connections in a session are also configurable parameters. The SCSI Read or Write requests (workload) are then passed to the initiator to carry out.

The workload for the each test is generated in a separate Otcl script. We have developed a workload generator program that generates requests of even distribution and Poisson distribution. The disk id is also evenly distributed among the specified range of disks. In addition, we also apply the trace file to the initiator to see how the actual application data affect the performance.

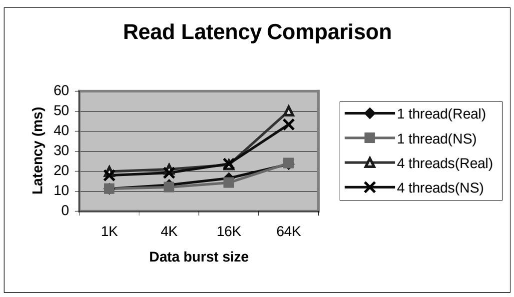
4. Empirical Validation

To verify the simulation model, we compare it with iSCSI performance measurement data obtained from a real iSCSI setting. In this real setup, Initiator communicates with a Cisco SN5420 iSCSI gateway through campus network. The network connection is the 100Mb/s Ethernet. There are 4 Seagate 39102FC disks are accessed. The round trip time in terms of network distance is approximately 2ms.

The test involves reading and writing a burst data of sizes from 1K, 4K, 16K to 64K bytes under light load and heavy load with a PDU size of 8KB.

1) A comparison of the real iSCSI access latency and NS modeled iSCSI access latency is shown in Figure 7. Both the light load (Each time only one thread is sending data request) and heavy load (4 threads is sending requests) are tested. The delay patterns for both figures shown above are approximately similar and follow the similar rate of increase. With small data burst size, the difference is within 2%, for the data burst size of 64K, the difference is within 6%.

2) In another test, the burst sizes vary from 4K to 64K under heavy load conditions. Figure 8 shows close approximation in throughput. The NS model provided a little higher throughput because of ideal environment conditions.



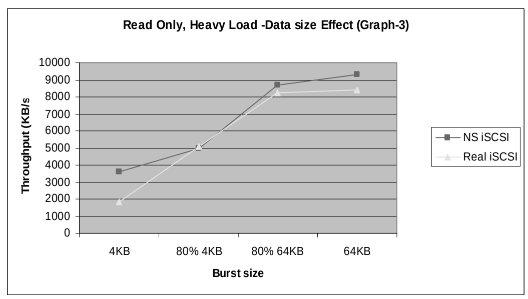


Fig. 7 Response time comparison   Fig. 8 Throughput comparison for heavy load

The results from the test provide reasonable assurance that the NS model can closely approximate a real iSCSI installation. To support the validity of model more thorough analysis can be done with different test scenarios.

5. Performance Analysis

In this section, we present the results of the effect of iSCSI parameters in iSCSI layer and TCP parameters in TCP layer to iSCSI data access performance.

5.1. The Effect of the iSCSI Parameters

We first examine the effect of different iSCSI PDU sizes. Figure 9 shows the read throughput with varying PDU sizes. The parameters involved in this simulation include data PDU sizes from 0.5KB to 8KB and max burst sizes from 1KB to 4MB. It is found out that at larger burst data size, the PDU size makes difference. For a large burst size, e.g. 1MB, with the PDU size of 8K, there will be 128 PDUs, while with PDU size of 1K, then there will be 1024 PDUs. More PDUs cause more R2T messages, and potentially more waiting for the R2T signals. From the figure, we observe the better performance for larger PDU size as data size increases.

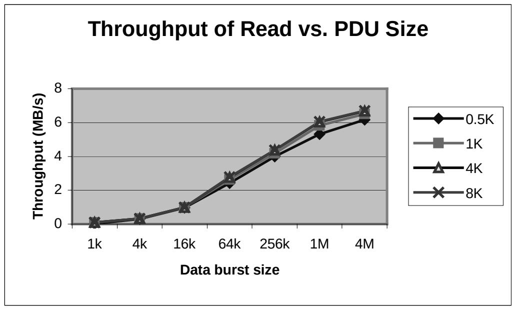


Fig. 9 The effect of iSCSI PDU size

5.2. The Effect of Network Parameters

In this subsection, we investigate how the network parameters like TCP window size, MSS (Maximum Segment Size) and link delays affect the iSCSI performance.

We first examine the effect of MSS size. In the test setting, the TCP window assumes the default value of 20. The link is a Gigabit Ethernet with delay from 10us to 50ms. The MSS sizes are 296B, 576B and 1500 bytes respectively. Figure 10 shows the achieved throughput with varying MSS sizes.

Normally, for a given delay and MSS size, the maximum throughput that can be achieved is approximately one window per round trip time, i.e. (MSS * window)/2*delay, which implies that throughput is inversely proportional to the link delay for the given MSS.

This figure shows that the throughput decreases quickly for smaller MSS sizes, whereas higher MSSs show a gradual decrease even for higher link delays. For link delays less than 1ms, the MSS size does not have much effect on the throughput this is because at short network link delay, bandwidth-delay product is small. The acknowledgement comes back very fast. As the link latency continues to increase, the throughput drops gradually, thus the link utilization is also getting lower. We need more parallelism to take advantage of the link bandwidth. The use of multiple connections may help. Adding more disks will increase the disk I/O bandwidth. The RAID system may also increase the disk access performance.

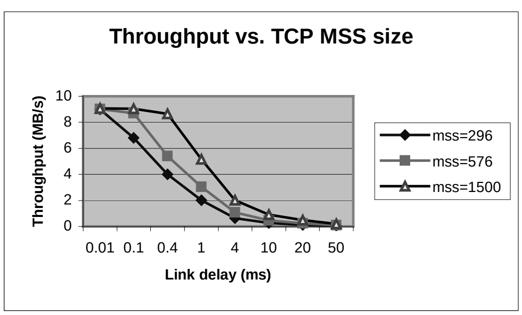


Fig. 10 Throughput vs. MSS size

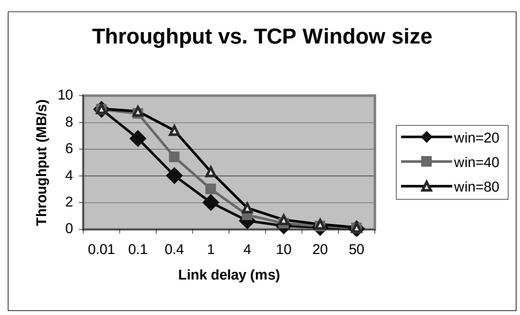


Fig. 11 Throughput vs. Window size

We then examine how the TCP window variation affects the throughput. Three different TCP window sizes (20, 40 and 80) are used. The MSS size is fixed at 296B as shown in the Figure 11. The result shows that the throughput increases with the increase of TCP window. At the short network link latency (e.g. LAN environment) of 0.4ms, the throughput is about 4MB/s for window size 20, but when the window size increases to

80, the throughput reaches to 6.8MB/s. However, with the increase of network latency (long fat network), the window size is too small, and the throughput reduces significantly.

6. Related Work

There have been several simulation work related to disk and storage subsystem. Paper [7] gave an introduction about the disk drive modeling and simulation. It described the principle of a disk drive and present a formula to compute disk seek time. Project DiskSim [8] provides an open source code, which can extract the disk parameters of different disk drives. We benefit a lot from their work in the disk simulation. Paper [10] modeled a disk controller and studied some more advanced features like caching. In paper [11], a Storage Area Network (SAN) is simulated. In this SAN, Myrinet is used to connect the storage devices and servers (initiators).

On the other hand, iSCSI protocol represents a different SAN paradigm, i.e. it uses the ubiquitous TCP protocol as the SCSI command and data transport. There are several studies in iSCSI performance and characteristics. Papers [4][5][6] presented the iSCSI performance measurement and evaluation under different scenarios. However, due to the diversity of network configuration and the impact of the underlying TCP network, it is crucial to build up the simulation model for the iSCSI environment for the iSCSI-related research. Our work incorporates the rich feature of ns2 in supporting the TCP/IP networking, and implements all related components including disk model, iSCSI protocol, FC-AL protocol and iSCSI target. This can be used to easily construct a flexible iSCSI-based storage system to facilitate the iSCSI related research.

7. Conclusion

We have presented a simulation model for the iSCSI-based storage system. The model includes all components for constructing an iSCSI-based storage system. In order to meet the requirements of extensibility and flexibility, we make the components modular and generic. The whole system is composed into several components. These components can be easily replaced or extended.

In order to validate the implementation, we also conducted real measurement and compared the simulation results with the real measurement results under the same settings. It turns out that the performance results in our model are close to that of real measurement.

With the simulation model, we further conducted the performance characteristics study to examine how the iSCSI parameters and the underlying TCP parameters affect the iSCSI performance. Our results show that with larger burst data size, the larger PDU size will help the throughput. Increasing TCP window size and MSS also affects the end-to-end performance. But this effect is more pronounced for higher link delays.

In the future, we'll further study the iSCSI storage system in a diverse network such as fat network (long latency, high bandwidth), wireless network, we'll also examine the impact of the underlying TCP protocol on the upper-level SCSI access in terms of performance and resilience based on the simulation model.

Acknowledgement

The authors thank to Avinashreddy Bathula who helped the initial work of this project. We are also grateful for the help offered by our shepherd Randal Burns.

References

[1] Julian Satran, et al. iSCSI Specification, Internet Draft, http://www.ietf.org/internet-drafts/draft-ietf-ips-iscsi-20.txt, Jan. 2003

[2] Kalman Z. Meth, Julian Satran, Design of the iSCSI Protocol, IEEE/NASA MSST 2003, Apr. 2003.

[3] K. Voruganti; P. Sarkar, An Analysis of Three Gigabit Networking Protocols for Storage Area Networks, 20th IEEE International Performance, Computing, and Communications Conference", April 2001

[4] S. Aiken, D. Grunwald, A. Pleszkun, Performance Analysis of iSCSI protocol, IEEE/NASA MSST 2003, Apr. 2003.

[5] Y. Lu, D. Du, Performance Evaluation of iSCSI-based Storage Subsystem, IEEE Communication Magazine, Aug. 2003

[6] S. Tang, Y. Lu, D. H.C. Du: Performance Study of Software-Based iSCSI Security. IEEE Security in Storage Workshop 2002: 70-79

[7] C. Ruemmler and J. Wilkes. An Introduction to Disk Drive Modeling. IEEE Computer, Vol. 27, No. 3, 1993, pp 17-28

[8] G. Ganger, B. Worthington, Y. Patt, The DiskSim Simulation Environment, http://www.pdl.cmu.edu/diskSim/index.html

[9] The network simulator ns2, http://www.isi.edu/nsnam/ns/

[10] M. Uysal, G. A. Alvarez and A. Mechant, A Modular, Analytical Throughput Model for Modern Disk Arrays, MASCOTS-2001, Aug. 2001

[11] X. Molero, F. Silla, V. Santonja, Jose Duato, Modeling and Simulation of Storage Area Networks, MASCOTS-2000, Sep. 2000, pp. 307

[12] D. Anderson, J. Dykes and E. Riedel, More Than An Interface – SCSI vs. ATA, Proc. of the 2nd Annual Conference on file and Storage (FAST), Mar. 2003