AN ISCSI DESIGN OVER WIRELESS NETWORK

Yan Gao, Yao-long Zhu, Hui Xiong, Renuga Kanagavelu, Jie Yan, Zhe-jie Liu
Data Storage Institute, DSI building, 5 Engineering Drive 1, Singapore 117608
{gao_yan, zhu_yaolong}@dsi.a-star.edu.sg
Tel: +65-68746824, Fax: +65-68732745

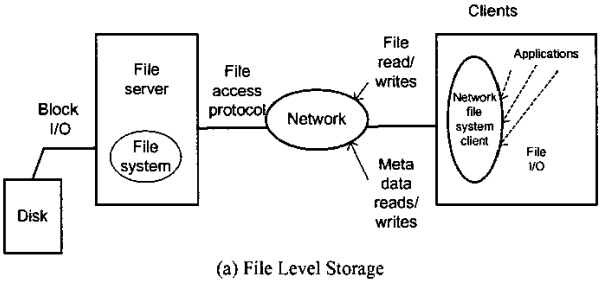
Abstract - With the trend that wireless network and mobile devices have become more and more prevalent, there is an increasing need to build a wireless storage system that can access information efficiently and correctly. iSCSI (internet SCSI) is an important protocol to enable remote storage access through the TCP/IP network. The performance of iSCSI over wireless network is an interesting research topic due to the impacts of the low bandwidth, unreliability and long latency of the wireless network. This paper presents a new iSCSI design with the concepts of multiple virtual TCP connections in an iSCSI session and parallel working mechanism in iSCSI layer over wireless LAN 802.11b. Test results show that for small I/O request, 2K for example, multiple connection iSCSI design can achieve 112% throughput improvement compared to normal iSCSI model. The maximum throughput can reach 0.62 MB/s for big I/O (128K), which is closed to the theoretical analysis result.

1. INTRODUCTION

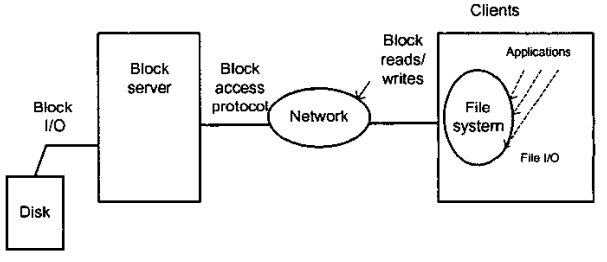
The end user market for portable and mobile devices keeps increasing every year. The mobile devices are playing more and more important role in people's life. Compared to their desktop counterparts in wired environment, mobile devices have created new challenges for data accessibility due to the low bandwidth, unreliability and limited storage capacity. In order to solve above challenges, many researchers have proposed a variety of solutions to deal with mobile data access in file level, especially focusing on disconnected operation [1][2][3]. However few researches have achieved high storage performance and network utilization in the wireless network's limited bandwidth.

1.1 File Level and Block Level Wireless Storage

There are currently two methods of wireless network storage. The first one is to simply employ a network file system such as NFS (Network File System). In this approach, the server makes a subset of its local namespace available to clients. Clients access metadata and files on the server using RPC (Remote Procedure Call). This storage architecture is commonly named NAS (Network Attached Storage). In contrast to this approach, the other approach for accessing remote data is to use an IP based SAN (Storage Area Network) protocol such as iSCSI. In this approach, a remote disk exports a portion of storage space to a client. Rather than accessing blocks from a local disk, the I/O operations are carried out over a network using a block access protocol. The above two approaches are shown in Fig. 1. Advantages of file level data access provide high security and cross platform data sharing. Advantages for direct block level data access provide high storage performance. In order to achieve high storage performance and sufficiently utilize the limited bandwidth in wireless storage, the block level data access approach such as iSCSI is the better choice. But so far, there are very few researches that focus on the block level data storage for wireless network.



(a) File Level Storage



(b) Block Level Storage

Figure 1. File Level and Block Level Wireless Storage

1.2 Related Work in iSCSI

iSCSI is a storage network protocol carrying SCSI command over TCP/IP. In addition to the reduced costs, iSCSI also provides good scalability and easy way to implement remote backup and disaster recovery. Many researches and projects have been carried out to analyze and implement iSCSI. University of Minnesota carried out the research on performance analysis of iSCSI [4]. University of Colorado also did some tests and performance studies in both hardware and software iSCSI [5]. Some solutions have also been proposed to handle the performance issue. A research group proposed a solution to use memory of iSCSI initiator to cache iSCSI data [6]. Other solutions included using iSCSI adapter

0-7803-8783-X/04/$20.00 © 2004 IEEE

[7] and a TCP/IP offload Engine (TOE) [8] to reduce the burden of host CPU.

Most iSCSI researches are based on wired environment. There are few researches that focus on iSCSI in wireless environment. Due to the limited bandwidth of wireless network, all other system resources are powerful enough compared to the limited bandwidth to make it possible to optimize the storage performance in wireless environment for iSCSI software solutions.

This paper presents a multiple connection iSCSI design with the multiple virtually TCP connections in an iSCSI session over one physical wireless connection and parallel working mechanism in iSCSI layer in order to achieve high utilization of the limited bandwidth of wireless network.

The rest of the paper is organized as follows. The next section details the software design of multiple connection iSCSI and section 3 demonstrates the experiment environment, system setup and methodology. Section 4 discusses the experimental and theoretical results of the multiple connection iSCSI. Finally, the conclusion of the paper is drawn in section 5.

2. Software Design of Multiple Connection iSCSI

2.1 iSCSI Storage Model

Fig. 2 shows the iSCSI storage model including an initiator and a target. iSCSI builds on top of TCP/IP layer. For iSCSI communication between an initiator and a target, it needs to establish a session. The data and command exchanges occur within the context of the session. In initiator, the application issues file requests. The file system converts file requests to block requests from application layer to SCSI layer. The SCSI command execution consists of three phases: Command, Data and Status Response which are shown in Fig. 3. The initiator iSCSI driver encapsulates SCSI commands into iSCSI Protocol Data Units (PDUs) and sends them to the network via TCP/IP layer. After receiving iSCSI PDUs from TCP/IP layer, the target iSCSI driver de-capsulates them. Then SCSI commands are mapped to the storage device. The target driver then sends response data and status back via TCP/IP layer. iSCSI parameters, such as PDU size, FirstBurstLength, MaxBurstLength, and the underlying TCP flow control algorithm, maximum frame size and MAC mechanism significantly affect iSCSI performance. There are two data copy and one DMA operation in the initiator during one I/O access and there are one data copy and one DMA operation for RAM I/O and two data copy and one DMA operation for disk I/O in the target side.

2.2 Multiple Connection iSCSI Design for Wireless Network

Standard iSCSI implementation is based on single TCP connection, which may not sufficiently utilize the network bandwidth especially for small I/O because of the TCP window size and the round trip time of TCP ACK over low speed wireless network. Furthermore, in low speed and unreliable wireless environment, it may face serious performance problems caused by packet failure and long latency.

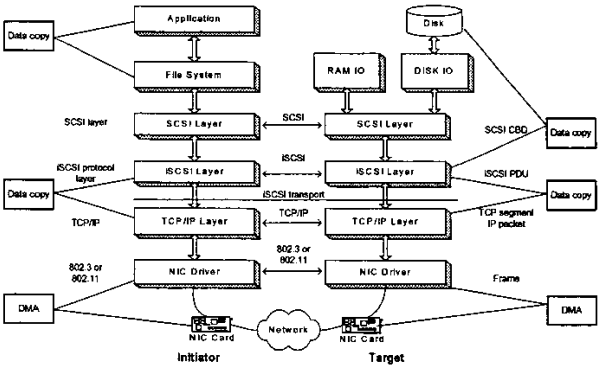


Figure 2. General iSCSI Model

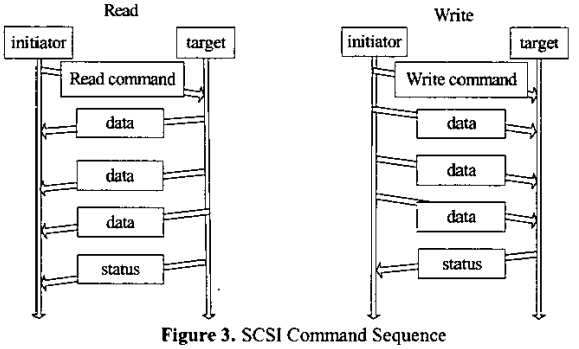


Figure 3. SCSI Command Sequence

A new iSCSI design is proposed which introduces the multiple virtual TCP connections in an iSCSI session and parallel working mechanism in iSCSI layer. The new design actually employs multiple TCP connections over one physical wireless connection, which is totally different from general idea of multiple physical connections according to iSCSI draft [13]. The multiple virtual connection design is supposed not only to improve the iSCSI performance by increasing the utilization of limited wireless network bandwidth, but also to provide a better mechanism to handle the long latency issue in multi-hop wireless network environment.

The detailed working principle is shown in Fig. 4. Multiple virtual TCP connections are built on one physical connection. Half of the connections are used for sending SCSI requests from initiator to target, the other half of the connections are used for sending responses from target to initiator. One pair of transmitting thread (Tx_thread) and receiving thread (Rx_thread), which locate in initiator and target separately, are responsible for data and command communications within one connection.

The detailed communication procedure is explained by illustrating a typical read operation. The SCSI middle layer is

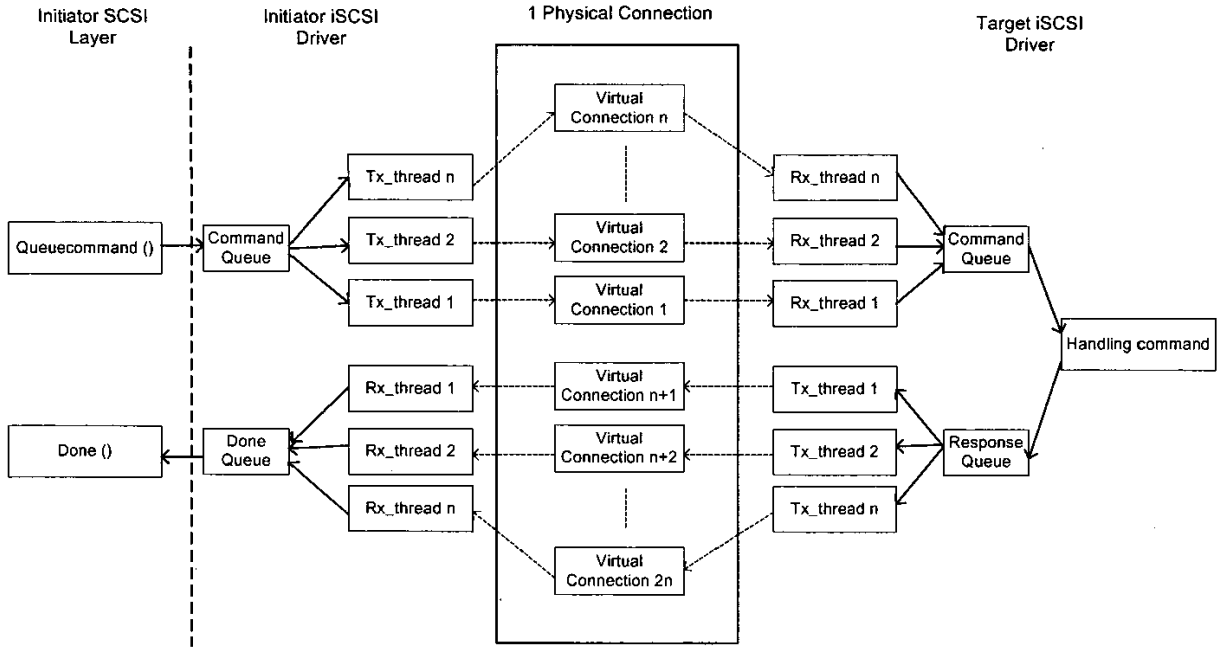


Figure 4. Multiple Virtual TCP Connection Design

a standard interface for SCSI layer to communicate with iSCSI device driver. The two main functions of SCSI middle layer are queuecommand ( ) which issues SCSI command to the iSCSI driver and done ( ) which informs SCSI middle layer that the command is finished by iSCSI driver. The iSCSI initiator driver gets read commands from SCSI middle layer. These commands are encapsulated in iSCSI request PDUs and queued in the initiator command queue.

As shown in Fig. 4, in initiator the Tx_threads (1, 2,….,n) of connections (1, 2,….,n) send these PDUs from command queue to the target. The target driver receives these PDUs by the Rx_threads (1, 2,….,n) of their corresponding connections. The target driver then de-encapsulates iSCSI request PDUs and map SCSI commands to the storage device (RAM or disk). After getting data from storage device and forming iSCSI response PDUs (including data and status), the target driver queues these iSCSI response PDUs in response queue. Tx_thread (1, 2,….,n) in the target send these PDUs by connection (n+1, n+2,….,2n). The initiator driver receives response through the Rx_threads (1, 2….,n) of their corresponding connections. Then the initiator driver put it to done queue to call the done ( ) function to finish the SCSI exchange.

3. Experiment Setup and Methodology

3.1 System Setup

Fig. 5 shows the configuration of the iSCSI implementation and experiment. The target is a PC with 866MHz PIII processor, 256M RAM and a Cisco Aironet 350 series PCMCIA 802.11b Wireless LAN Adaptor. The Adaptec 39160 SCSI card is used to connect an IBM 18.2G SCSI hard disk. The initiator is a 733MHz PIII PC with 256M RAM and a Cisco Aironet 350 series PCMCIA 802.11b Wireless LAN Adaptor. Both machines run Redhat 8.0 with the Linux kernel version 2.4.18-14.

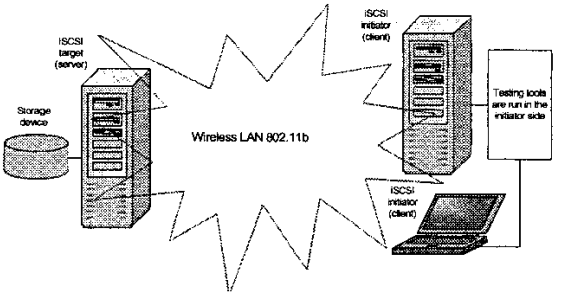


Figure 5. iSCSI Experiment Setup

3.2 Experiment Methodology

First, the TCP network performance is tested by using Netperf in Linux and Chariot 5.0 in Windows. The objective of this test is to verify the TCP layer throughput that can provide for the iSCSI layer and compare it with theoretical analysis result of wireless 802.11b. Next the performance of normal single connection iSCSI is compared with NFS by dd command. IOMeter, which is an industry standard I/O benchmark tool, is

used to test the performance of the multiple virtual connection iSCSI with respect to different number of connections, network latency, queue length for small I/O. The experiment is also conducted to show the effect of lower layer parameters on iSCSI data access performance.

4. Experiment Result and Performance Analysis

4.1 TCP Stream Analysis and Test

The effective net throughput of IEEE 802.11 [9] depends on the data rate, but there is a lot of overhead, for example, the transmission time of the physical preamble, the transmission time of the PHY head, the MAC header, transmission time of ACK frames, transmission protocol overhead, DIFS (Distributed Interframe Space), SIFS (Short Interframe Space) and processing delay in local and remote computer. The net throughput of 802.11b [10] is far less than the nominal data rate, which is 11 Mbps. According to the 802.11 specification, 11 Mbps is defined as physical layer raw data rate. The MAC throughput of 802.11b with default packet size of 1,500 bytes without considering the impact of transport layer can be calculated approximately as 6.1 Mbps[1].

The total net throughput with TCP is about 15% to 20% lower than MAC layer throughput [11]. TCP layer net throughput of 802.11b is around 4.9 Mbps (0.61 MB/s) to 5.2 Mbps (0.65 MB/s) (45% to 47% of 11 Mbps) with 1,500 bytes packet size. The TCP performance is tested in Linux by Netperf and in Windows by Chariot 5.0 respectively. The results show that throughput is 5.07 Mbps and 4.93 Mbps respectively.

4.2 Comparison of iSCSI and NFS

Fig. 6 shows the read throughput comparison of normal single connection iSCSI and NFS. The throughput of iSCSI always outweighs NFS. This is because to access the raw device from block level can achieve higher throughput than access storage device from file level. The testing result shows that in order to achieve higher throughput, the iSCSI gain advantages over NFS.

4.3 Performance Analysis of Multiple Connection iSCSI

Fig. 7 shows the throughput comparison of multiple connection iSCSI with disk I/O mode to normal iSCSI by using 802.11b. The I/O request size ranges from 2K to 128K and the frame size is set as default size of 1,500 bytes. From Fig. 7, it can be seen that for small I/O (2K ~ 8K), the normal single connection iSCSI's throughput is far less than the maximum throughput in theory and it is very low compared to big I/O request. However for small I/O request (2K ~ 8K), the

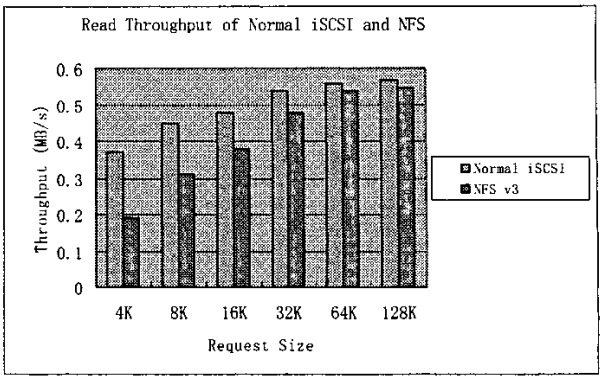


Figure 6. Throughput of iSCSI and NFS

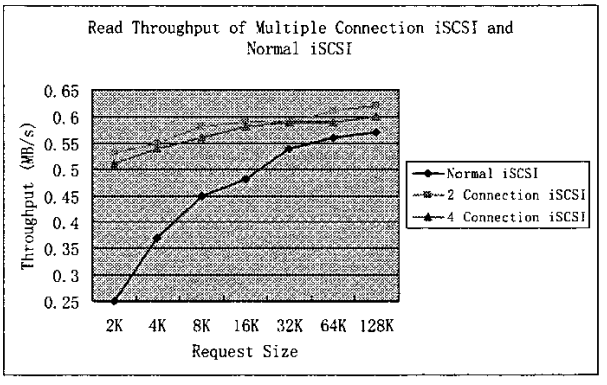


Figure 7. Throughput Comparison of Normal iSCSI and Multiple Connection iSCSI

multiple connection iSCSI can achieve a significant throughput improvement compared to normal single connection iSCSI. For 2K request size, for example, the throughput is improved from 0.25 MB/s to 0.53 MB/s, which is about 112% improvement. The wireless environment is very unreliable. In normal iSCSI, the initiator needs to wait for a long time before sending the next I/O request because of the packet retransmission for the ACK status. Especially for small I/O, the requests are sent frequently compared to big I/O, so much time is wasted when waiting for the status. And for small I/O request, if the request size can not be divided exactly by the maximum frame size in the lower layer, there are a lot of 802.11 frames, which do not sufficiently utilize each frame size. In multiple TCP connection iSCSI design, the above mentioned problems can be solved. By using the multiple virtual TCP connections in an iSCSI session and the parallel working mechanism, the iSCSI driver can utilize the wireless channel more efficiently, which means the iSCSI driver can continuously send I/O requests to low layer via different connections and does not need to wait for the ACK status before sending the next I/O request. Also due to the continuous requests from the iSCSI layer, each 802.11 frame can be sufficiently used to carry the data. In wireless storage, since the demand for data is not as large as that in wired network, multiple connection iSCSI design that significantly

[1]The MAC net throughput of 802.11b can be expressed as:

$$\frac{8L_{DATA}}{T_p + T_{PHY} + \frac{8L_{MAC} + 8L_{DATA\_TCP}}{R_{DATA}} + T_p + T_{PHY} + \frac{8L_{MAC} + 8L_{ACK}}{R_{ACK}} + 2\tau + T_{DIFS} + T_{SIFS} + \overline{CW}}$$

The parameters of 802.11b and notations can be found in [11][12].

improve throughput for small I/O is very valuable for some application scenarios in wireless environment.

For big I/O request (128K), there are also some improvements. The maximum throughput can reach 0.62 MB/s, which is closed to the theoretical analysis result. For the system, by using 2 connections, it can achieve the maximum throughput. Adding more connection, the test results show that the system performance reduces due to imposing more overheads on the system.

Further analysis show that for normal iSCSI and multiple connection iSCSI, the CPU utilizations in initiator are 40% and 49% respectively. This means that although the multiple connection iSCSI design can further utilize the system resource, the system bottleneck is still on the low wireless bandwidth.

4.4 The Impact of Queue Length on I/O Rate

Because the queue depth is an important parameter that affects the iSCSI performance for small I/O, the multiple connection iSCSI is designed with the function of supporting different queue length. Fig. 8 shows the impact of queue length on multiple connection iSCSI performance. The test is done with the request size of 4 KB and the MTU (Maximum Transmission Unit) size of 1.5 KB. From Fig. 8, it can be seen that the I/O rate increases with the queue length until the queue length equals to 8. The I/O rate peak value is around 140 IOPS. Further increasing the queue length can not increase the I/O rate anymore. This is because when queue length is 8, the bandwidth has already been sufficiently used by iSCSI.

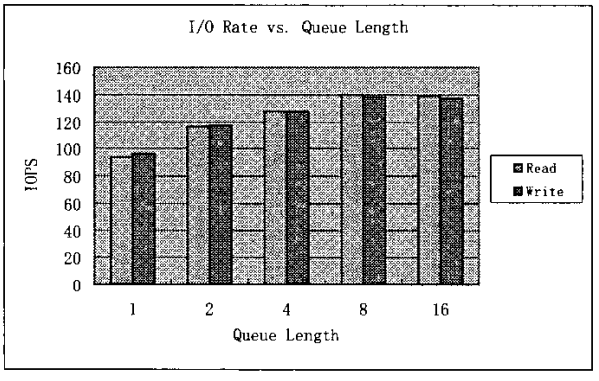


Figure 8. I/O Rate vs. Queue Length

4.5 Throughput with Different Network Latency

Fig. 9 shows that throughput with different network latency. The 1.2 ms delay refers to peer to peer ad-hoc network while the 2.5 ms delay and 3.8 ms delay refer to the communication between initiator and target with one hop and two hops respectively. From Fig. 9, it can be seen that although the network latency is significantly increased, the throughput of the multiple virtual connection iSCSI can still achieve good storage performance. For big I/O request, e.g. 128K, although the delay is doubled (2.5 ms) or tripled (3.8 ms), the throughput is only 8% or 11% less than peer to peer network scenario (1.2 ms). For small I/O request, e.g. 2K, the decrease is only 2% and 5% for 2.5 ms delay and 3.8 ms delay respectively.

The above mentioned achievement for long latency wireless network is due to the multiple virtual TCP connection design and the architecture that can support long queue. In wireless environment, the multi-hop channel and the unreliable signal strength can delay the data transmission and make the network latency quite long. The experiment verifies that the multiple connection iSCSI design is effective enough to achieve high throughput in such scenario.

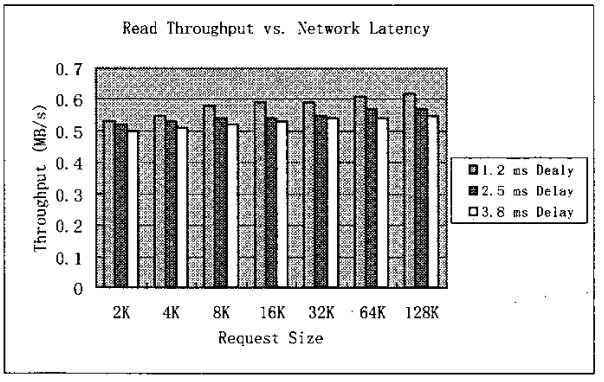


Figure 9. Result of Throughput vs. Network Latency

4.6 The Impact of MTU on Multiple Connection iSCSI

Fig. 10 shows the achieved throughput of the multiple connection iSCSI (2 connections are used) with different MTU (Maximum Transmission Unit) size. The maximum packet size used in the experiment is 2,260 bytes. From Fig. 10, it can be seen that for large MTU size, e.g. 2,260 bytes, the throughput for 128K request is 0.72 MB/s, while for default 1,500 bytes MTU size, the throughput for 128K is only 0.62 MB/s. A conclusion can be drawn that the iSCSI performance is increased as the MTU size is increased all the time. This is mainly because the big frame can carry more data, which make the payload size greater in proportion to TCP header overhead than that of small frame size. The big frame can also decrease the frequency of interruption of wireless adapter.

5. CONCLUSION

This paper presents a multiple virtual TCP connection iSCSI design for wireless storage which can achieve high storage performance. Experiment results show that the multiple connection iSCSI design can achieve significant throughput improvement compared to normal single connection iSCSI model for small I/O. For 2K request, for example, the throughput improvement is 112%. For big I/O (128K), the maximum throughput can reach 0.62 MB/s, which is closed to the theoretical analysis result that the TCP layer can achieve.

Experiment results also prove that the design can achieve high storage performance in multi-hop, unreliable and long latency wireless network.

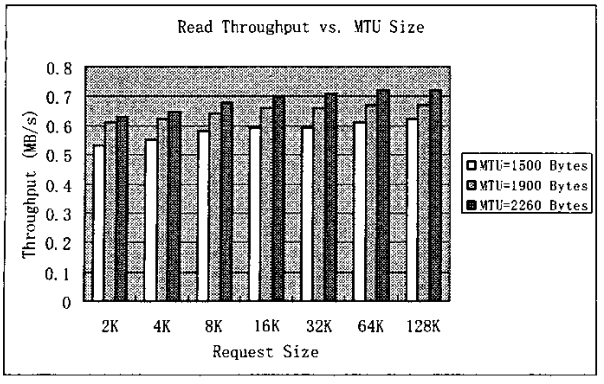


Figure 10. Throughput vs. MTU size

6. REFERENCES

[1] I.R. Chen and N.A. Phan, "Update propagation algorithms for supporting disconnected operations in mobile wireless systems with data broadcasting," *IEEE 23rd International Conference on Distributed Computing Systems*, pp. 784-789, 2003.

[2] K. Yasuda, "Cache cooperation for clustered disconnected Computers," *IEEE 9th International Conference on Parallel and Distributed Systems,* pp. 457-464, 2002.

[3] J.C.S. Lui, O.K.Y. So and T.S. Tam, "NFS/M: an open platform mobile file system," *IEEE 18th International Conference on Distributed Computing Systems*, pp. 488-495, 1998.

[4] Y.P. Lu and D.H.C. Du, "Performance study of iSCSI-based storage subsystems," *IEEE Communications Magazine*, vol. 41, issue. 8, pp. 76-82, 2003.

[5] S. Aiken, D. Grunwald, A.R. Pleszkun and J. Willeke, "A performance analysis of the iSCSI protocol," *20th IEEE/11th NASA Goddard Conference on Mass Storage Systems and Technologies*, pp. 123-134, 2003.

[6] X.B. He, Q. Yang and M. Zhang, "A caching strategy to improve iSCSI performance," *Proceedings of the 27th Annual IEEE conference on Local Computer Networks*, pp. 278-285, 2002.

[7] Adaptec white paper, "Building SANs with iSCSI, Ethernet and Adaptec," http://www.graphics.adaptec.com/pdfs/buildingsanwithiscsi-21.pdf

[8] Alacritech white paper, "Delivering High Performance Storage Networking," http://www.alacritech.com/html/storagewhitepaper.html

[9] "Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification," *IEEE 802.11 WG*, 1999.

[10] "Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification: High-Speed Physical Layer Extension in the 2.4 GHz Band," *IEEE 802.11b WG,* 1999.

[11] A. Kamerman and G. Aben, "Net throughput with IEEE 802.11 wireless LANs," *IEEE Wireless Communications and Networking Conference,* vol. 2, pp. 747-752, 2000.

[12] Y. Xiao and J. Rosdahl, "Throughput and delay limits of IEEE 802.11," *IEEE Communications Letters,* vol. 6, issue. 8, 2002.

[13] J. Satran et al., "iSCSI (Internet SCSI) Specification, Internet Draft," http://www.ietf.org/internet-drafts/draft-ietf-ips-iscsi-20.txt, 2003.