2005

Iscsi-Based Storage Area Networks for Disaster Recovery Operations

Matthew R. Murphy
iSCSI-based Storage Area Networks for Disaster Recovery Operations

By

MATTHEW R. MURPHY

A Thesis submitted to the Department of Electrical and Computer Engineering in partial fulfillment of the requirements for the degree of Master of Science

Degree Awarded: Spring Semester, 2005
The members of the Committee approve the thesis of Matthew R. Murphy defended on April 5, 2005.

______________________________
Bruce A. Harvey
Professor Directing Thesis

______________________________
Lois Hawkes
Committee Member

______________________________
Simon Foo
Committee Member

______________________________
Michael Frank
Committee Member

Approved:

______________________________
Leonard Tung, Chair, Department of Electrical & Computer Engineering

The Office of Graduate Studies has verified and approved the above named committee members.
TABLE OF CONTENTS

List of Figures ..................................................................................................................... v
List of Tables ..................................................................................................................... vi
ABSTRACT ...................................................................................................................... vii
INTRODUCTION .............................................................................................................. 1

1. DISASTER RECOVERY ............................................................................................... 3
   1.1 Recovery Time and Recovery Point Objectives ....................................................... 3
   1.2 Increasing Assurance using Remote Backup .......................................................... 5
   1.3 The Emergence of Storage Area Networks .............................................................. 6
   1.4 Fibre Channel Protocol ............................................................................................ 7
   1.4.1 The Trouble with Fibre Channel ............................................................................ 8
   1.5 Origins of iSCSI ........................................................................................................ 9
   1.5.1 Solutions Provided by iSCSI ............................................................................... 10
   1.6 iSCSI vs. NFS ......................................................................................................... 13

2. DISASTER RECOVERY USING iSCSI ..................................................................... 16
   2.1 Fundamentals of iSCSI .............................................................................................. 16
   2.1.1 PDU Structure ...................................................................................................... 17
   2.2 Layer Interactions ................................................................................................... 18
   2.3 iSCSI Login ............................................................................................................ 20
   2.3.1 Login Request PDU ............................................................................................. 20
   2.3.2 Login Response PDU ........................................................................................... 22
   2.4 Login Text Parameters ............................................................................................ 23
   2.4.1 Text Request and Response PDUs ....................................................................... 24
   2.5 SCSI Read and Write Operations ........................................................................... 25
   2.6 iSCSI Discovery Sessions ....................................................................................... 27
   2.7 Disaster Recovery using iSCSI ............................................................................... 28
   2.7.1 Small Business Continuity of Operations ............................................................ 29
   2.7.2 Enterprise Continuity of Operations ................................................................... 30
   2.7.3 Remote Backup using iSCSI for Large Organizations ........................................ 32
   2.8 Software and Hardware iSCSI Implementations .................................................... 32
   2.8.1 Software-based iSCSI .......................................................................................... 33
   2.8.2 TCP/IP Offload Engines ...................................................................................... 34
   2.8.3 Hardware-based iSCSI ........................................................................................ 35
   2.8.4 Comparison of Hardware and Software-based iSCSI .......................................... 35
   2.9 Cancer Therapy and Research Center Case Study .................................................. 36
   2.9.1 Background and Requirements ............................................................................ 37
   2.9.2 An IP SAN Solution ............................................................................................. 38
   2.9.3 Results and Performance ..................................................................................... 39
   2.10 iSCSI RAID Configurations ............................................................................... 40
   2.10.1 Distributed iSCSI RAID Configurations ............................................................ 42
   2.11 MC/S and Link Aggregation ............................................................................... 43
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.11.1</td>
<td>Ensuring In-order PDU Delivery over MC/S</td>
<td>45</td>
</tr>
<tr>
<td>2.11.2</td>
<td>Benefits of Using MC/S</td>
<td>46</td>
</tr>
<tr>
<td>2.11.3</td>
<td>MC/S Implementation Results</td>
<td>47</td>
</tr>
<tr>
<td>2.12.1</td>
<td>Limitations of iSCSI</td>
<td>48</td>
</tr>
<tr>
<td>2.12.2</td>
<td>Fault Tolerance</td>
<td>48</td>
</tr>
<tr>
<td>2.12.3</td>
<td>Load Balancing and Manageability</td>
<td>51</td>
</tr>
<tr>
<td>3.</td>
<td>PERFORMANCE TESTING OF iSCSI</td>
<td>53</td>
</tr>
<tr>
<td>3.1</td>
<td>Software-based iSCSI Experiments</td>
<td>53</td>
</tr>
<tr>
<td>3.2</td>
<td>Block Size and Throughput</td>
<td>54</td>
</tr>
<tr>
<td>3.2.1</td>
<td>Fast Ethernet Results</td>
<td>54</td>
</tr>
<tr>
<td>3.2.2</td>
<td>Gigabit Ethernet Results</td>
<td>55</td>
</tr>
<tr>
<td>3.3</td>
<td>iSCSI CPU Utilization</td>
<td>58</td>
</tr>
<tr>
<td>3.4</td>
<td>CPU Loading Effects on iSCSI Throughput</td>
<td>60</td>
</tr>
<tr>
<td>3.5</td>
<td>iSCSI at 10 Gbps</td>
<td>61</td>
</tr>
<tr>
<td>4.</td>
<td>CONCLUSION</td>
<td>64</td>
</tr>
<tr>
<td>APPENDIX</td>
<td></td>
<td>67</td>
</tr>
<tr>
<td>REFERENCES</td>
<td></td>
<td>70</td>
</tr>
<tr>
<td>BIOGRAPHICAL SKETCH</td>
<td></td>
<td>73</td>
</tr>
</tbody>
</table>
## LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure 1</td>
<td>Minimum distance recommendations (in miles)</td>
<td>5</td>
</tr>
<tr>
<td>Figure 2</td>
<td>Generic Storage Area Network</td>
<td>7</td>
</tr>
<tr>
<td>Figure 3</td>
<td>iSCSI Encapsulation</td>
<td>11</td>
</tr>
<tr>
<td>Figure 4</td>
<td>iSCSI as a remote backup mechanism</td>
<td>12</td>
</tr>
<tr>
<td>Figure 5</td>
<td>SCSI device interaction</td>
<td>16</td>
</tr>
<tr>
<td>Figure 6</td>
<td>PDU and BHS Structure</td>
<td>18</td>
</tr>
<tr>
<td>Figure 7</td>
<td>iSCSI Layer Interactions</td>
<td>19</td>
</tr>
<tr>
<td>Figure 8</td>
<td>Login Request PDU</td>
<td>20</td>
</tr>
<tr>
<td>Figure 9</td>
<td>Login Response PDU</td>
<td>22</td>
</tr>
<tr>
<td>Figure 10</td>
<td>Text Request PDU</td>
<td>24</td>
</tr>
<tr>
<td>Figure 11</td>
<td>Text Response PDU</td>
<td>25</td>
</tr>
<tr>
<td>Figure 12</td>
<td>The SCSI Data-In PDU</td>
<td>26</td>
</tr>
<tr>
<td>Figure 13</td>
<td>The SCSI Data-Out PDU</td>
<td>27</td>
</tr>
<tr>
<td>Figure 14</td>
<td>TCP/IP Offload Engine</td>
<td>34</td>
</tr>
<tr>
<td>Figure 15</td>
<td>CTRC Distributed SAN for Backup and Recovery</td>
<td>39</td>
</tr>
<tr>
<td>Figure 16</td>
<td>An S-iRAID and P-iRAID configuration</td>
<td>41</td>
</tr>
<tr>
<td>Figure 17</td>
<td>P-iRAID may allow for disaster tolerant distributed data storage</td>
<td>43</td>
</tr>
<tr>
<td>Figure 18</td>
<td>MC/S allows two physical connections to act as a single session</td>
<td>44</td>
</tr>
<tr>
<td>Figure 19</td>
<td>TCP may deliver PDUs out of order if spanning multiple connections</td>
<td>45</td>
</tr>
<tr>
<td>Figure 20</td>
<td>iSCSI over a small TCP/IP network</td>
<td>49</td>
</tr>
<tr>
<td>Figure 21</td>
<td>TCP routes around congestion or failures</td>
<td>49</td>
</tr>
<tr>
<td>Figure 22</td>
<td>Intelligent iSCSI is aware of alternate replica targets</td>
<td>50</td>
</tr>
<tr>
<td>Figure 23</td>
<td>HCS Lab iSCSI Testbed</td>
<td>53</td>
</tr>
<tr>
<td>Figure 24</td>
<td>Fast Ethernet Block Size vs. Throughput (read operations)</td>
<td>54</td>
</tr>
<tr>
<td>Figure 25</td>
<td>Fast Ethernet Block Size vs. Throughput (write operations)</td>
<td>55</td>
</tr>
<tr>
<td>Figure 26</td>
<td>Gigabit Ethernet Block Size vs. Throughput (write operations)</td>
<td>56</td>
</tr>
<tr>
<td>Figure 27</td>
<td>Gigabit Ethernet Block Size vs. Throughput (read operations)</td>
<td>56</td>
</tr>
<tr>
<td>Figure 28</td>
<td>Link Utilization for writes improved to 80% with 2 MB PDU bursts</td>
<td>57</td>
</tr>
<tr>
<td>Figure 29</td>
<td>Link Utilization for reads improved to 70-80% with 2 MB PDU bursts</td>
<td>57</td>
</tr>
<tr>
<td>Figure 30</td>
<td>Effect of PDU Burst Size on CPU Utilization (Epsilon nodes)</td>
<td>59</td>
</tr>
<tr>
<td>Figure 31</td>
<td>CPU Loading Effects on Write Operations</td>
<td>60</td>
</tr>
<tr>
<td>Figure 32</td>
<td>CPU Loading Effects on Read Operations</td>
<td>60</td>
</tr>
</tbody>
</table>
LIST OF TABLES

Table 1 – Recovery classes ........................................................................................................... 4
Table 2 – Solutions for various recovery classes ......................................................................... 5
Table 3 – Comparison of iSCSI, iFCP, and FCIP........................................................................ 10
Table 4 – Comparison of NFS and iSCSI Transactions ................................................................. 14
Table 5 – Comparison of NFS and iSCSI CPU Utilization ......................................................... 14
Table 6 – NSG and CSG Phases ................................................................................................. 21
ABSTRACT

For a number of years, high performance disaster recovery networks have been unavailable to all but the largest and wealthiest organizations. However, a recently adopted IETF standard known as iSCSI promises to not only provide performance comparable to the best disaster recovery solutions but also extend the availability of these solutions to home and small office environments.

This paper will explore the details of iSCSI and illustrate how it can provide a cost effective and high performance solution to disaster recovery and business continuity operations. A software-based iSCSI implementation is tested to determine whether it can perform at gigabit and higher speeds and what CPU utilization is required to operate the software drivers. If iSCSI can perform well in software then it is reasonable to assume that it can perform as well or better when realized in hardware. It will be shown that iSCSI can perform as well as the best Fibre Channel networks available at a reduced cost. Finally, a number of modifications will be proposed that will enhance the effectiveness of iSCSI as a disaster recovery mechanism.
INTRODUCTION

The last decade has been a time of enormous increases in both the amount and importance of information stored on computers and distributed across networks. Ubiquitous Internet access has accelerated the demand for e-commerce services, media and educational applications, and critical operations involving health care, finance, transportation and law enforcement. As the amount of data has grown exponentially, its importance has also increased significantly. Financial, public health and law enforcement applications require a high degree of security and availability to ensure that potentially life saving information is accessible when it is needed. This is perhaps nowhere more vital than at those agencies that are responsible for national security. In these environments a massive volume of data must be moved for processing and analysis and then safely stored so that it can be accessed if required. National infrastructure protection is therefore dependent on seamless disaster recovery in the event that a primary storage location is damaged or destroyed in a man-made or natural disaster.

As the requirements for availability and performance continue to increase, the cost of protecting and providing that data during a disaster scenario also increases. This cost has long been an obstacle for many companies and resulted in non-existent or substandard disaster recovery plans. In fact, disaster recovery solutions have generally been confined to enterprise networks and large companies with sizable budgets. Few medium sized businesses were able to afford high assurance disaster recovery infrastructure. Small and home office environments have traditionally lacked any data protection aside from infrequent backups to tape and only on occasion were those tapes stored off-site.

This paper will explore disaster recovery operations and the requirements of business continuity in the near future. The need for a robust disaster recovery infrastructure will be outlined and it will be shown that while a number of traditional solutions exist for business continuity, there are no affordable and high-performance solutions available. However, an emerging standard known as Internet Small Computer System Interface (iSCSI) promises to be one such cost effective and high-performance solution. Its use as
A disaster recovery mechanism will be examined not only for high performance environments, but even for small businesses with little financial resources. iSCSI may be the all-purpose disaster recovery solution that has been desired for so long.
CHAPTER 1
DISASTER RECOVERY

1.1 Recovery Time and Recovery Point Objectives

The solution to safeguarding mission critical data is making sure that a replica copy exists at a location other than the primary site. This aims to ensure that if the primary site is damaged or destroyed the data will still be available once functionality is restored. In the past, a company would generally make weekly or nightly tape backups of important information and physically ship the tapes to a remote location where they were stored. Unfortunately, as data has grown in volume and importance, two factors continue to make shipping tape off-site less effective. The first is the decreasing Recovery Time Objective (RTO) [MAYE03].

The RTO is a measure of how fast a company can recover from a disaster. The metric is somewhat ambiguous since it does not specify what constitutes a disaster. For example, the recovery time for a failed disk drive may be a matter of minutes or even seconds. But a site-wide event such as a fire or natural disaster may take many days to recover from. The recovery time objective has decreased to a matter of minutes or hours for some applications regardless of the scope of the disaster. In the most mission critical environments, the RTO may even be a matter of seconds for site-wide disasters. For instance, financial institutions may be able to recover from a site-wide disruption in just a few hours using remote data centers and high-speed networks. However, military and defense applications may require that recovery time be only a few seconds if a primary site is damaged or restored.

The second performance metric is the recovery point objective (RPO). The RPO indicates how up-to-date the recovered data must be in relation to when the disaster occurred. The RPO will tend to decrease as data becomes more time critical. For example, an organization may keep updated personnel files on all employees. These files are rarely updated and may need to be duplicated once a month. If a disaster strikes, the
recovery point may be structured such that the recovered data is no more than a month old. This recovery point would not be sufficient for financial data that may be updated thousands of times an hour. In this case, the RPO would likely be set such that the recovered data is only a few minutes old so that critical financial information would not be lost. In some cases, the RPO may be so low that any loss of data is intolerable and thus the backup copies would have to be up-to-the-second duplicates of the original data at the primary site.

This can be further illustrated by categorizing RTOs and RPOs into different classes. These classes may vary from one organization to another but the general principle still applies. [CNT03] defines these classes as follows. Class 1 is the lowest level, where acceptable recovery times range from 72 hours to 1 week and the most up-to-date data can be from a weekly backup (up to a week old). A Class 4 recovery environment contains the most stringent requirements. Here, the recovery time must be immediate and the data recovered must be less than one second old. Table 1 illustrates these different classes. Any high performance, high assurance system will require a class 3 or class 4 recovery categorization.

<table>
<thead>
<tr>
<th>Class 1</th>
<th>Class 2</th>
<th>Class 3</th>
<th>Class 4</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>RTO</strong></td>
<td>72 hours – 1 week</td>
<td>8 hours – 72 hours</td>
<td>Less than 8 hours</td>
</tr>
<tr>
<td><strong>RPO</strong></td>
<td>Last full backup – less than a week</td>
<td>Last backup – less than 24 hours</td>
<td>Less than 15 minutes before event</td>
</tr>
</tbody>
</table>

Class 4 environments require the most expensive solutions so as to keep up with the high demand of traffic and the high degree of availability required. These demands have traditionally required continuous availability (even in the event of a disaster) as well as fault tolerant hardware to guard against as many failures as possible. Furthermore, the RPOs in these environments tend to be so high that any loss of data is intolerable. Possible solutions for different classes are outlined in Table 2.
Most disaster recovery solutions target one class (as can be seen above). However, it will be shown that iSCSI is appropriate for all of the recovery classes listed above. Moreover, since iSCSI is so affordable, it is possible to give Class 1 and 2 environments the opportunity to achieve RTOs and RPOs of Class 3 and Class 4 environments.

1.2 Increasing Assurance using Remote Backup

In 2002, the US Securities and Exchange Commission issued a draft report in which it was suggested that remote backup sites be located between 200 and 300 miles from the primary facility [SECD02]. It was suggested that increasing the distance between a primary site and backup site would greatly enhance the assurance characteristics of the system. The primary reason for this increase is due to the fact that further separation of primary and backup sites reduces the probability that both sites will be adversely affected by the same disaster. In 2003, the U.S. Chapter of the Association of Contingency Planners (ACP) participated in a survey conducted by
The Disaster Recovery Journal to determine the best distance separation between primary and backup sites. According to the ACP, the average minimum distance was determined to be approximately 105 miles. This is shown in Figure 1 [WEEM03]. It was clear from the survey that this distance recommendation was made in large part to protect against a large natural disaster such as a hurricane. This provides a baseline number so as to understand how high-assurance networks should be formed. There is a performance and assurance tradeoff depending on the separation distance of the primary and backup sites. As the distance increases, the survivability of the system increases at the cost of performance (since latency increasingly degrades performance). On the other hand, as the distance is decreased, the chance of both sites becoming unavailable due to the same event increases, thus decreasing assurance.

1.3 The Emergence of Storage Area Networks

While not built to fulfill only disaster recovery obligations, storage area networks (SANs) have been introduced to allow a dedicated network to provide access to centralized storage resources. In traditional configurations, the SAN is a separate network and isolated from the LAN. This SAN allows for storage expansion without impacting the server resources on the LAN. Similarly, any server upgrades or expansions will not impact the storage resources on the SAN. Since the SAN is isolated from the LAN as shown in Figure 2, it can allocate full bandwidth to storage needs while the LAN only has to be responsible for the communication between servers and other users. In other words, storage access will not impact the performance of the LAN and heavy LAN traffic will not degrade the performance of storage access. It is this characteristic that makes storage area networks extremely valuable for backup and recovery of mission critical data. In the past, tape backups had to be carefully planned so that the LAN was not congested with typical production traffic. As a result, tape backups had to be executed during off-peak hours – generally on the weekends or late evenings when the majority of the workforce had left for the day. This was the only way to avoid network resource contention [MAYE03]. There are a number of problems with this approach. The first is that backups can only be performed, at best, every night. This may not be sufficient in those environments where the RPO is very low. It also means that any recovery operations
will impact the production LAN and that recovery operations will be significantly slowed because of traffic on the LAN during peak hours [BROC01]. Furthermore, there are some applications in which there are no off-peak hours. In these cases, business continuity must be maintained 24 hours a day, 7 days a week and traditional backup methods can not be used. In short, the traditional approach will not be sufficient in low RPO, low RTO settings (class 3 and 4 environments).

Figure 2 - Generic Storage Area Network [HUFF03]

1.4 Fibre Channel Protocol

The Fibre Channel (FC) protocol is an ANSI standard protocol built for storage area networking. SCSI command descriptor blocks (see Section 2) originate from the host but the SCSI bus used for transport has been replaced with Fibre Channel connections. Unlike the SCSI bus, the FC resources can be connected using switches and hubs. Fibre Channel networks can currently achieve throughputs as high as 4 Gbps and it is predicted that 10 Gbps capacity is possible in the near future. Fibre Channel allows consolidated storage to both disk and tape. Storage pools can be shared, management of storage is centralized, and backup and recovery operations are made significantly easier. The most important characteristic is that it provides for LAN-free backup (as discussed in Section 1.3) and is thus very well suited for disaster recovery and replication needs. The serial interface used for Fibre Channel can span a distance of nearly 10 km while a typical SCSI bus only allows separations of a few meters. Fibre Channel also allows far more devices
to be connected. A SCSI implementation is limited to 16 devices but an FC implementation can have 126 nodes per loop and 16 million addresses per fabric [MEGG94].

FC SANs have for years been considered the premier storage solution and they will be found in any high-performance, high availability environment. If data must be available and up-to-date during a disaster, it is a near certainty that Fibre Channel will play some role in the solution.

1.4.1 The Trouble with Fibre Channel

Perhaps the most obvious disadvantage associated with Fibre Channel is its cost. The cause of this dramatic cost increase is two-fold. First, Fibre Channel is only realizable in hardware, with special host bus adapters (HBAs) necessary for each node and FC switches required to interconnect FC components. Essentially, this means that in addition to having a production LAN, a totally separate network must be purchased and maintained in order to use FC. The second cost component is incurred with the management of the FC network. Fibre Channel is not an ordinary networking protocol and requires people with specialized training to operate it. Therefore, it is necessary to hire additional system administrators to oversee the correct operation of the FC network while others may be required to oversee the company’s production LAN. This makes the total cost of ownership (TCO) of FC so high that only government agencies and medium/large corporations could ever afford it [HUFF03]. There is no small business or home office market for FC.

Another disadvantage of Fibre Channel is its inherent distance limitation. Typically, the range of FC is limited to a few hundred meters and if larger distances are required, significant money must be invested in expensive FC extenders. In the past, this was not seen as a critical flaw in Fibre Channel but in the wake of the attacks on September 11, 2001, a renewed interest in disaster recovery has arisen. This interest is especially strong in remote disaster recovery operations where sites are separated by hundreds or even
thousands of miles. Of course, a SAN implemented using FC can not be distributed over remote sites because of its inherent distance limitation. It should be noted that Fibre Channel vendors are attempting to remedy this with Fibre Channel over IP (FCIP) and the Internet Fibre Channel Protocol (iFCP) [IFCP01]. Again, these extensions to the FC protocol are still expensive to implement and assume that the user has already invested in base FC components such as switches and HBAs.

The high prices and difficulty in extending Fibre Channel over large distances have made it not only prohibitively expensive for smaller companies, but even large organizations are reluctant to spend extra money to use it over long distances. Any data that must be protected should be replicated at another site over 100 miles away. Fibre Channel does not allow for remote disaster recovery at a reasonable cost. The need for a high-speed solution, with no inherent distance limitation, at a lower cost is critical in helping businesses (large and small) realize a strong disaster recovery and business continuity infrastructure.

1.5 Origins of iSCSI

The iSCSI protocol was designed as a more cost effective solution to traditional Storage Area Network (SAN) demands. At the same time, a great effort was made to add more functionality without affecting performance. Until recently, any high-performance SAN had to be implemented using Fibre Channel. Fibre Channel was viewed as the best performing protocol for SAN deployment since it was among the first protocols to offer transfer rates of 1 Gbps and can now support 4 Gbps transactions. Also, Fibre Channel is a very “elegant” protocol in that it was designed from the beginning for Storage Area Networking. However, as discussed earlier, there are a number of disadvantages to using the Fibre Channel protocol. Table 3 illustrates how iSCSI compares to FCIP and iFCP [SNIA1].
Both iFCP and iSCSI using native TCP/IP for transport while FCIP uses the proprietary encapsulation provided with Fibre Channel. Thus, iFCP and iSCSI are both equal in terms of distance limitations (of which there are none). Device Integration with Internet Storage Name Service (iSNS) is used for fast discovery of target nodes. This is a completely separate IETF standard but both iSCSI and iFCP can take advantage of it. Finally, FC device support is obviously provided with iFCP and FCIP. However, iSCSI does not provide for this within its protocol. There are switches that can be used for iSCSI to FC bridging so that organizations with existing FC components can deploy iSCSI solutions that will function with their expensive FC infrastructure.

1.5.1 Solutions Provided by iSCSI

The Internet Engineering Task Force (IETF) was keenly aware of the cost and distance drawbacks of FC-based SANs and decided to undertake a project to develop a new SAN protocol that could be affordable while operating at long-distances at high-speeds. Their solution was to develop and adopt the iSCSI protocol. iSCSI is a true SAN protocol that was designed from the beginning to do everything Fibre Channel could do, but also do things that FC could not. In addition, careful steps were taken in order to ensure iSCSI remained an affordable option for smaller businesses.

Table 3 – Comparison of iSCSI, iFCP, and FCIP [SNIA1]

<table>
<thead>
<tr>
<th>Protocol Attributes</th>
<th>iFCP</th>
<th>iSCSI</th>
<th>FCIP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Implementation</td>
<td>Native IP Transport</td>
<td>Native IP Transport</td>
<td>Encapsulation, Tunneling</td>
</tr>
<tr>
<td>SCSI Encapsulation</td>
<td>FCP</td>
<td>iSCSI Layer</td>
<td>FCP</td>
</tr>
<tr>
<td>Device Integration with iSNS</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>End Device Interface</td>
<td>FC/FCP</td>
<td>IP/iSCSI</td>
<td>FC/FCP</td>
</tr>
<tr>
<td>End Device Routing</td>
<td>RIP, OSPF, BGP, others</td>
<td>RIP, OSPF, BGP, others</td>
<td>FSPF</td>
</tr>
<tr>
<td>FC Device Support</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Relative Cost</td>
<td>$$</td>
<td>$</td>
<td>$$</td>
</tr>
</tbody>
</table>
iSCSI provides the same service that Fibre Channel provides, that is, transporting SCSI information over a storage area network. iSCSI differs in that it provides this service using already existing networks by encapsulating SCSI data in TCP/IP packets (as shown in Figure 3). It is this difference that makes iSCSI so versatile and cost-effective. iSCSI generally takes a performance hit because of the extra work required to strip off the TCP/IP layer data. Fibre Channel does not suffer from this problem but has a number of other problems that may make iSCSI a better fit for many users.

Consider the cost associated with iSCSI and Fibre Channel. As illustrated above, FC is not only very expensive from a hardware perspective, but it also requires additional investment in personnel that can maintain the complex network. iSCSI on the other hand can be implemented using significantly cheaper hardware HBAs. Furthermore, iSCSI can operate over standard TCP/IP switches, which means that no additional investment is needed for specialized iSCSI switching equipment [HUFF03]. iSCSI switches do exist however, and provide increased performance and features that standard Ethernet switches cannot provide. Since iSCSI utilizes a standard, well-known networking protocol, the average system administrator has enough expertise to maintain the iSCSI SAN – although a small amount of additional reading may be necessary to fully understand iSCSI. Nevertheless, it is no longer necessary to seek out system administrators with specialized FC training and certification. iSCSI also allows for a full software implementation. This is extremely useful for organizations that can not afford iSCSI HBAs or high-performance switches. The software is simply a driver that can be downloaded and allow for iSCSI transmissions over the existing LAN. In effect, this type of implementation requires no financial investment whatsoever.

---

**Figure 3 - iSCSI Encapsulation [HUFF03]**

TCP/IP layer data. Fibre Channel does not suffer from this problem but has a number of other problems that may make iSCSI a better fit for many users.
iSCSI, unlike FC, has no inherent distance restrictions. Because iSCSI transmits information using TCP/IP infrastructure, it is bounded by the same distance (and performance) limitations of TCP/IP. This means that iSCSI can be used to distribute a Storage Area Network over 100s of miles or interconnect disparate SANs separated by 100s of miles to provide for backup and recovery operations at great distances. An example of this is shown in Figure 4. This increases the assurance metric of the SAN solution by decreasing the chance that a disaster could affect the primary and backup site simultaneously (which would result in loss of data).

Finally, the performance characteristics of iSCSI are very important if the protocol is to be successful. It has been a long held belief that iSCSI could never match the performance delivered by a Fibre Channel network. In fact, in its early form, iSCSI was associated with more affordable, albeit poor performing networks that only had a market for those who could never afford a Fibre Channel SAN. Whenever high-performance was needed, iSCSI was overlooked and FC became the network of choice [HUFF03]. The dependence on TCP/IP for iSCSI transport was originally considered a weakness because iSCSI could never perform better than the TCP/IP fabric in which it was implemented. FC was not bound by the performance of TCP/IP and thus it was thought that FC would always be able to maintain a performance advantage. But what was once considered a flaw in the iSCSI design has turned out to be a blessing in disguise. Certainly, few people thought that Ethernet (or TCP/IP) performance would ever reach Gigabit and 10 Gigabit speeds. As such, the iSCSI protocol has been given the

![Figure 4 - iSCSI as a remote backup mechanism](image-url)
underlying support structure it needs to also perform at multi-gigabit per second throughputs and has now been shown to be just as capable as FC from a performance standpoint. With future projections of 40 Gbps and 100 Gbps Ethernet becoming more widespread, it is reasonable to assume that iSCSI will be able to operate at these speeds as well.

iSCSI also has another benefit that the IETF included to enhance performance. It is known as MC/S (Multiple Connections per Session). MC/S allows iSCSI to achieve higher throughputs by providing a mechanism for link aggregation so that multiple physical connections can be used in parallel to deliver the iSCSI data. iSCSI allows one session to be composed of multiple connections so that a large data block can be sent using all of the connections simultaneously. For instance, an iSCSI node may have access to four separate 1 Gbps Network Interface Cards (NICs) and would be able to provide 4 Gbps iSCSI transfers [HUFF03]. In fact, if each iSCSI node has access to two 10 Gbps NICs, then a sustained throughput of 20 Gbps is possible between the two nodes. Of course, bus speeds and disk read and write speeds must be addressed at these throughputs but there are a number of solutions (such as RAID striping) that will allow for multi-gigabit per second iSCSI implementations.

## 1.6 iSCSI vs. NFS

At first glance, iSCSI appears to offer the same service that the Network File System (NFS) has offered for years. Both allow remote storage devices to be used as if they were local disks and both iSCSI and NFS work over standard TCP/IP networks [ACHA01]. With NFS, the file system resides at the file server. The client issues read and write commands which are sent over the network. These commands are interpreted by the file system on the server and the appropriate actions are taken. iSCSI is different in that the file system resides on the client machine. The commands are interpreted and then translated to block commands which are forwarded to the network. In essence, the commands are already in the correct form once they are received by the target and therefore there is no need to translate the file I/O to block I/O.
In [RADK04] it is shown how the iSCSI protocol compares in performance to NFS. The results show that the two protocols perform similarly when the cache is cold. However, iSCSI takes advantage of aggressive caching techniques and when the cache is filled, the iSCSI performance substantially exceeds that of NFS. When the execution times for the PostMark benchmark are compared (Table 4), iSCSI clearly possesses a performance advantage.

<table>
<thead>
<tr>
<th>Files</th>
<th>Completion Time (s)</th>
<th>Messages</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>NFS v3</td>
<td>iSCSI</td>
</tr>
<tr>
<td>1,000</td>
<td>146</td>
<td>12</td>
</tr>
<tr>
<td>5,000</td>
<td>201</td>
<td>35</td>
</tr>
<tr>
<td>25,000</td>
<td>516</td>
<td>208</td>
</tr>
</tbody>
</table>

Another advantage that iSCSI possesses over NFS is its very low reliance on valuable CPU cycles. The study in [ACHA01] indicated that NFS can task the processor to a large degree and that iSCSI doesn’t require much work from the CPU. While both methods require TCP/IP to perform, the overhead required from iSCSI is significantly lower than that of NFS. The results are summarized in Table 5.

<table>
<thead>
<tr>
<th>Benchmark</th>
<th>Processor Utilization</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>NFS v3</td>
</tr>
<tr>
<td>PostMark</td>
<td>77%</td>
</tr>
<tr>
<td>TPC-C</td>
<td>13%</td>
</tr>
<tr>
<td>TPC-H</td>
<td>20%</td>
</tr>
</tbody>
</table>

The PostMark test is built to mimic the characteristics of small files for Internet applications such as email. It generated random small text files of different sizes and once created PostMark will either create, delete, read, or append the file. These actions are randomly generated. The TPC-C benchmark simulates transactions of small files where 2/3 of the transactions are reads. TPC-H is used to emulate decision support in
complex database applications that use large files. By using these benchmarks, a broad range of application behavior can be measured and a decision made on whether iSCSI or NFS best performs these tasks.

The results suggest that iSCSI and NFS perform similarly for data intensive applications, especially when they are dominated by read operations. However, for write operations, iSCSI is considered to be more capable, yielding better performance in a variety of applications. Thus, the iSCSI network is better equipped to handle large data replication operations common to any high-performance disaster recovery plan.
2 Fundamentals of iSCSI

In any iSCSI implementation, there must be an initiator (sender) and a target (receiver). The initiator is responsible for beginning contact with the target. The target then responds as requested by the initiator. Figure 5 shows a simplified version of the interaction between the application layer, SCSI device driver, iSCSI driver, and network. All SCSI commands are carried inside of command descriptor blocks (CDBs). These SCSI commands are handled by a SCSI driver which translates a device request from an application or file system. This request is then wrapped up in the CDB. iSCSI takes the next step with the CDB. Traditionally, another SCSI device would unwrap this CDB and perform the necessary operations. However, since this CDB is destined for the network

Figure 5 - SCSI device interaction [HUFF03]

(and later a remote SCSI device), it must go through another encapsulation. This can be done in hardware (with an iSCSI HBA) or in software using an inexpensive (or free) iSCSI driver. The iSCSI device driver will pack the CDB into an iSCSI packet and will forward this information to the TCP/IP layers where it will eventually find its way onto the network. At the receiving (target) end, the process is reversed so that the CDB is delivered to the correct SCSI device where it can be interpreted and executed. In many
cases, the SCSI command will consist of a read or write command. In these instances, it is necessary for the CDB to identify the Logic Unit (LU) of the read or write operation. The LU can be viewed as an addressable storage device [HUFF03].

2.1 The Protocol Data Unit

The Command Descriptor Block (CDB) is the basis for SCSI communication, but it is not the basic unit of communication for iSCSI. iSCSI uses a Protocol Data Unit (PDU) in order to send and receive data. The SCSI layer is unaware of the existence of a PDU but the PDU contains SCSI CDBs in which the SCSI layer does understand. In addition to the SCSI CDBs, the iSCSI PDU contains other information as well as the data components from the initiator. The CDB only contains information on what instruction the initiator wants the target to perform. It does not contain information on the address of the SCSI target or the LU to which it is directed. This is all information that the PDU must contain.

2.1.1 PDU Structure

An iSCSI PDU is made up of several segments, the most important of which is the Basic Header Segment (BHS). The BHS is a 48-byte field that contains vital information such as the Logic Unit Number (LUN), the CDB, and other operational information. The other PDU fields, such as the Additional Header Segment (AHS), Header Digest, Data Segment, and Data Digest, are all optional. In fact, the majority of PDUs contain only a BHS. The Data Segment contains the data that is being sent to the target from the initiator. The header digest is an error check for the header of the PDU and the data digest is an error check word responsible for the data. The general structure of the PDU and a generic BHS structure are shown in Figure 6.
The BHS contains iSCSI Opcodes, information about the length of the corresponding data segments, LUN information, SCSI CDBs, and other information necessary for session establishment and management. The structure of the BHS depends on the opcode contained in its first byte. This opcode will correspond to a certain PDU type such as a “Login_Request_PDU” or “Task_Management_Function_PDU”. More information on these PDU types can be found in Section 2.4.1.

2.2 Layer Interactions

Since iSCSI operates between the application layer and the transport layer, it is responsible for only a small subset of the total operations performed at both the application level and the network level. With this in mind, it is essential to understand what iSCSI was designed to handle, and what it is not equipped to handle. It is especially important to understand iSCSI’s limitations at the network level, where its responsibilities end, and TCP’s begin.

iSCSI was not designed as a networking protocol. All of the normal services provided by TCP/IP are still provided by TCP/IP. This is important because it streamlines iSCSI so that certain “assumptions” can be made. The most important assumption is that all iSCSI packets will arrive in order from the initiator node to the target node. This is the sole responsibility of TCP and as such, requires no supervision by iSCSI. Only when iSCSI
spans multiple connections for a single session does this assumption break down. In this case, iSCSI must be able to reconstruct PDUs arriving out of order using command and status sequence numbers.

iSCSI is placed between the SCSI and transport layers as shown in Figure 7. The SCSI Command Descriptor Block (CDB) is treated as a payload by the iSCSI Protocol Data Unit (PDU). This iSCSI PDU is then treated as a data payload by TCP/IP. Since the

![Figure 7 - iSCSI Layer Interactions](HUFF03)

iSCSI PDU is treated as the data payload, the transmission characteristics of TCP/IP tend to dominate the performance of any iSCSI implementation. There are certain iSCSI parameters, such as the burst size, that can enhance performance by ensuring that the TCP overhead is kept to a minimum (see Section 3.2.2).
2.3 iSCSI Login

The login operation of iSCSI is first considered since it is the initial task performed between the initiator and the target. For simplicity, this discussion assumes that a functioning TCP/IP layer is in place to allow for communication between initiator and target nodes. iSCSI requires that a session be established between the initiator and the target before data transfer takes place.

The login has two basic terms – a request and a response. Sending a request and receiving a response is regarded as a single task [HUFF03]. There are three stages of the login process – the security negotiation phase (SNP), login operational negotiation phase (LONP), and full-feature phase (FPP). Data can only be transferred once the connection has entered the full-feature phase.

2.3.1 Login Request PDU

The first step is the leading login request which is sent by the initiator to the target node to inform the target that a session should be established. The leading login is responsible for setting parameter values that will be used for the duration of the session. These values can be negotiated between the initiator and the target. The login request PDU is

<table>
<thead>
<tr>
<th>Byte</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bit</td>
<td>01234567</td>
<td>01234567</td>
<td>01234567</td>
<td>01234567</td>
</tr>
<tr>
<td>0 to 3</td>
<td>.000011</td>
<td>TC000CSGNSG</td>
<td>Version Max</td>
<td>Version Min</td>
</tr>
<tr>
<td>4 to 7</td>
<td>0</td>
<td>DataSegment Length</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8 to 15</td>
<td>ISID</td>
<td></td>
<td>TSIH</td>
<td></td>
</tr>
<tr>
<td>16 to 19</td>
<td></td>
<td>ITT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>20 to 23</td>
<td>CID</td>
<td>Reserved</td>
<td></td>
<td></td>
</tr>
<tr>
<td>24 to 27</td>
<td></td>
<td>CmdSN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>28 to 31</td>
<td></td>
<td>ExpStatSN or Reserved</td>
<td></td>
<td></td>
</tr>
<tr>
<td>32 to 47</td>
<td></td>
<td>Reserved</td>
<td></td>
<td></td>
</tr>
<tr>
<td>48 to n</td>
<td></td>
<td>DataSegment (Login parameters in text request format)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 8 - Login Request PDU [HUFF03]

for setting parameter values that will be used for the duration of the session. These values can be negotiated between the initiator and the target. The login request PDU is
shown in Figure 8. Notice that the Login Request PDU does not have a finite length. In fact, there is no maximum limit to the length of a Login Request PDU. There may be cases where this PDU is too big for a single TCP/IP packet. In such cases, there are mechanisms available to allow single PDUs to span multiple packets. This PDU is sent by the initiator to begin a session with the target. The first byte of the PDU is the opcode, which simply defines the PDU as being a login request. It is at this point that the target must decide how to handle the incoming request. The second byte of the login request PDU contains a transit bit $T$, a continue bit $C$, a CSG and NSG (which are both two bits).

The transit bit indicates that the initiator or target wants to change from one stage (or phase) to another. When set, the $T$ bit informs the target or initiator that the other side wishes to change from the stage indicated in the CSG (current stage) to the stage indicated in the NSG (next stage). In other words, the NSG is only valid if the $T$ bit is set. If the $T$ bit is not set, then the target or initiator must continue negotiation (unless already in the full-feature phase). The stages for the CSG and NSG are encoded using two bits as shown in Table 6.

<table>
<thead>
<tr>
<th>Bit Sequence</th>
<th>Desired NSG or CSG</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>Security Negotiation Phase</td>
</tr>
<tr>
<td>01</td>
<td>Login Operational Negotiation Phase</td>
</tr>
<tr>
<td>10</td>
<td>(Not Used)</td>
</tr>
<tr>
<td>11</td>
<td>Full Feature Phase</td>
</tr>
</tbody>
</table>

The Version-Max field simply indicates the maximum version supported. Likewise, the Version-Min field is used to inform the target and initiator the minimum version supported. Version-Max and Version-Min must not change for any login requests within the login process. The target is required to use the value presented during the leading login request. Next, the DataSegmentLength field is the length of the data payload (text login parameters) of the login request PDU in bytes. It can fall anywhere between 1 and $2^{24}$-1.
The next 6 bytes are reserved for the Initiator Session ID (ISID). This ISID is derived from the Session Identifier (SSID) and follows a formal naming convention. The ISID is set by the initiator on the leading login so that the session is unique within the initiator system. The Target Session Identifying Handle (TSIH) occupies bytes 14 and 15 of the login request PDU. The TSIH is only set by the target during the last target response of the leading login.

The next 4 bytes are used for the Initiator Task Tag (ITT). This is necessary for the target’s response PDU. The ITT is used so that the target responses can identify the command to which they are responding. The initiator will set the ITT and the target responds with the ITT corresponding to the command that it has completed.

### 2.3.2 Login Response PDU

After the login request has been sent from the initiator to the target, the target replies with a login response PDU (as shown in Figure 9). This PDU is sent if the target wishes to

<table>
<thead>
<tr>
<th>Byte</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bit</td>
<td>0 1 2 3 4 5 6 7</td>
<td>0 1 2 3 4 5 6 7</td>
<td>0 1 2 3 4 5 6 7</td>
<td>0 1 2 3 4 5 6 7</td>
</tr>
<tr>
<td>0 to 3</td>
<td>... 1 0 0 0 1 1</td>
<td><strong>T C 0 0 CSGNSG</strong></td>
<td>Version-Max</td>
<td>Version-Act</td>
</tr>
<tr>
<td>4 to 7</td>
<td>0</td>
<td>DataSegmentLength</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8 to 15</td>
<td>ISID</td>
<td>TSIH</td>
<td></td>
<td></td>
</tr>
<tr>
<td>16 to 19</td>
<td>ITT</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20 to 23</td>
<td>Reserved</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>24 to 27</td>
<td>StatSN</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>28 to 31</td>
<td>ExpCmdSN</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>32 to 35</td>
<td>MaxCmdSN</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>36 to 39</td>
<td>Status-Class</td>
<td>Status-Detail</td>
<td>Reserved</td>
<td></td>
</tr>
<tr>
<td>40 to 47</td>
<td>Reserved</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>48 to n</td>
<td>DataSegment</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Figure 9 - Login Response PDU**

accept, reject, or continue receiving the initiator’s login request PDUs. The login response PDU is very similar to the login request PDU. The first difference is the **Version-Act** field, which simply defines the version of the standard that is currently in use. The next difference is the addition of the **StatSN** field. **StatSN** is the status
sequence number and is set on the initial response and incremented whenever a new status is to be returned on a certain connection. **Status-Class** and **Status-Detail** provide critical information on the condition of the login process. The status fields indicate failures because of authentication or authorization errors, too many connections, missing parameters, unavailable targets, and a number of other problems. Like the login request PDU, the login response contains text information on login parameters in its data segment.

### 2.4 Login Text Parameters

The login request and response PDUs (as well as other PDUs) can communicate important parameters by using key=value pairs in the data segment portion of the PDU. These key=value pairs are case sensitive and the key must be entered exactly as defined in the iSCSI standard. The key=value pairs are valid at any stage of the iSCSI process (login, full-feature, etc.). However, certain key=value pairs can only be used at designated times. For instance, the AuthMethod key can only be used during the security negotiation phase of the login process. This key can be used to specify which authentication method is to be used for the iSCSI session. The key can be followed by multiple values separated by commas to form an ordered list. The left-most value would be given the first priority, followed by the next value. The first value that the target and initiator agree on will be used. Target names, maximum connections, session types, and a number of other parameters can be adjusted using the key=value pairs.

It is important to note that some parameters can only be negotiated and set during the login phase. Obviously the Authentication method (with keys), the type of data digest being used (either CRC-32C or none at all), and the type of header digest being used should all be negotiated at login. However, other parameters such as the maximum burst length, default time to wait, maximum connections, and maximum number of outstanding R2Ts (ready to transmit) are all negotiated during the login phase. Thus, these parameters should be optimized to provide the best performance possible depending on the type of environment in which the iSCSI network will operate. For example, increasing the maximum burst size may improve throughput over high bandwidth
networks. Increasing the maximum number of R2Ts may increase the performance of an iSCSI solution in the presence of high latency (for long haul, remote replication operations).

2.4.1 Text Request and Response PDUs

The Text Request and Text Response PDUs allow the initiator and target to communicate information that is not available to the SCSI layer. The initiator will send a text request PDU that contains important text keywords (shown in Figure 10). The opcode for the text request is contained in the first byte of the PDU. In the second byte, the first two bits are used to indicate if there is more information contained in a following PDU (C bit set to 1) or if this is the final text request PDU (F bit is set to 1). In other words, if the PDU is too small for all key=value pairs, the C bit will be set to one. In the case where key=value pairs span more than one PDU, the target will reply with empty text response PDUs until the final PDU appears. At this time, operations can continue as usual. The Initiator task tag is assigned to any new task by the initiator that is unique in the session. This is how requests and responses are coordinated between the initiator and target. The target task tag is useful in the same way since it allows the initiator to obtain information from the target about a certain target task. The target is then charged with replying with a

<table>
<thead>
<tr>
<th>Byte</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bit</td>
<td>0 1 2 3 4 5 6 7</td>
<td>0 1 2 3 4 5 6 7</td>
<td>0 1 2 3 4 5 6 7</td>
<td>0 1 2 3 4 5 6 7</td>
</tr>
<tr>
<td>0 to 3</td>
<td>1 0 0 0 1 0 0</td>
<td>FC</td>
<td>Reserved</td>
<td></td>
</tr>
<tr>
<td>4 to 7</td>
<td>0</td>
<td></td>
<td>DataSegmentLength</td>
<td></td>
</tr>
<tr>
<td>8 to 15</td>
<td></td>
<td>Logic Unit Number (or Reserved)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>16 to 19</td>
<td></td>
<td>Initiator Task Tag</td>
<td></td>
<td></td>
</tr>
<tr>
<td>20 to 23</td>
<td></td>
<td>Target Task Tag (or 0xFFFFFFFF)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>24 to 27</td>
<td></td>
<td>Command Sequence Number</td>
<td></td>
<td></td>
</tr>
<tr>
<td>28 to 31</td>
<td></td>
<td>Expected Status Sequence Number</td>
<td></td>
<td></td>
</tr>
<tr>
<td>32 to 47</td>
<td></td>
<td>Reserved</td>
<td></td>
<td></td>
</tr>
<tr>
<td>48 to 51</td>
<td></td>
<td>Header Digest (Optional)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>52 to x</td>
<td></td>
<td>Data Segment (Text)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>x+1 to x+4</td>
<td></td>
<td>Data Digest (Optional)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*Figure 10 - Text Request PDU [HUFF03]*

In other words, if the PDU is too small for all key=value pairs, the C bit will be set to one. In the case where key=value pairs span more than one PDU, the target will reply with empty text response PDUs until the final PDU appears. At this time, operations can continue as usual. The Initiator task tag is assigned to any new task by the initiator that is unique in the session. This is how requests and responses are coordinated between the initiator and target. The target task tag is useful in the same way since it allows the initiator to obtain information from the target about a certain target task. The target is then charged with replying with a
text response PDU (Figure 11) that contains information on the status of the request. The command and status sequence numbers are used to coordinate the text negotiation. The text request contains a command sequence number and the text response contains a status sequence number. Also, the initiator will send a text request that contains an expected status sequence number. This informs the target the latest status sequence number received by the initiator. Likewise, the target will send a text response with an expected command sequence number. This lets the initiator know the latest command sequence number received by the target. With this information, the initiator and target can retransmit text PDUs as necessary.

<table>
<thead>
<tr>
<th>Byte</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bit</td>
<td>0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0 to 3</td>
<td>. . 1 0 0 1 0 0</td>
<td>FC</td>
<td>Reserved</td>
<td></td>
</tr>
<tr>
<td>4 to 7</td>
<td>0</td>
<td>DataSegmentLength</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8 to 15</td>
<td>Logic Unit Number (or Reserved)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16 to 19</td>
<td>Initiator Task Tag</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
| 20 to 23 | Target Task Tag (or 0xFFFFFC)
| 24 to 27 | Status Sequence Number |
| 28 to 31 | Expected Command Sequence Number |
| 32 to 35 | Max Command Sequence Number |
| 36 to 47 | Reserved |
| 48 to 51 | Header Digest (Optional) |
| 52 to x | Data Segment (Text) |
| x+1 to x+4 | Data Digest (Optional) |

*Figure 11 - Text Response PDU [HUFF03]*

2.5 SCSI Read and Write Operations

The most important operations in an iSCSI system are of course those operations that allow it to access and store information. Without this, the iSCSI protocol is useless. Therefore, there are special PDUs used for iSCSI read and write operations. Once the full-feature phase has been started, the initiator is permitted to send SCSI Data-In (read) and SCSI Data-Out (write) PDUs.
The SCSI Data-In PDU is used for read operations. It contains many of the fields detailed in the Login and Text PDUs. The Initiator Task Tag (ITT) is used to as an identifier to allow the initiator to find the corresponding command that issued the request for the data contained. The buffer offset contains the offset value of the PDU data payload as part of the total data transfer. Therefore, the sum of the length and buffer offset should not be greater than the expected transfer length for the corresponding command. The residual overflow bit O is set to indicate that the residual count specifies the number of bytes not transferred because the expected data transfer length was not large enough. If the U bit (residual underflow) is set, the residual count denotes the number of bytes not transferred out of the number of bytes expected. If neither the O nor U bits are set, then the residual count should be zero. The SCSI Data-In PDU is shown in Figure 12.

<table>
<thead>
<tr>
<th>Byte</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bit</td>
<td>0 1 2 3 4 5 6 7</td>
<td>0 1 2 3 4 5 6 7</td>
<td>0 1 2 3 4 5 6 7</td>
<td>0 1 2 3 4 5 6 7</td>
</tr>
<tr>
<td>0 to 3</td>
<td>. . 1 0 0 1 0 1</td>
<td>FA 0 0 0 0 U S</td>
<td>Reserved</td>
<td>Status or Reserved</td>
</tr>
<tr>
<td>4 to 7</td>
<td>0</td>
<td></td>
<td></td>
<td>DataSegmentLength</td>
</tr>
<tr>
<td>8 to 15</td>
<td>Logic Unit Number (or Reserved)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16 to 19</td>
<td>Initiator Task Tag</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20 to 23</td>
<td>Target Task Tag (or 0xFFFFFFFF)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>24 to 27</td>
<td>Status Sequence Number or Reserved</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>28 to 31</td>
<td>Expected Command Sequence Number</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>32 to 35</td>
<td>Max Command Sequence Number</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>36 to 39</td>
<td>Data Sequence Number</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>40 to 43</td>
<td>Buffer Offset</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>44 to 47</td>
<td>Residual Count</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>48 to 51</td>
<td>Header Digest (Optional)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>52 to x</td>
<td>Data Segment</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>x to x+4</td>
<td>Data Digest (Optional)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Figure 12 - The SCSI Data-In PDU [HUFF03]**

The SCSI Data-Out PDU is used for write operations. Its fields are similar to the Data-In PDU except that there is no need for the O and U bits. Also, there is no residual count as in the read operation. The F bit (final bit) is set for the last PDU of unsolicited data. It is
also set to one at the end of a write operation. The SCSI Data-Out PDU is shown in Figure 13.

<table>
<thead>
<tr>
<th>Byte</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bit</td>
<td>01234567</td>
<td>01234567</td>
<td>01234567</td>
<td>01234567</td>
</tr>
<tr>
<td>0 to 3</td>
<td>.. 0 0 0 1 0 1</td>
<td>F</td>
<td>Reserved</td>
<td></td>
</tr>
<tr>
<td>4 to 7</td>
<td>0</td>
<td>DataSegmentLength</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8 to 15</td>
<td>Logic Unit Number (or Reserved)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16 to 19</td>
<td>Initiator Task Tag</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20 to 23</td>
<td>Target Task Tag (or 0xFFFFFFFF)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>24 to 27</td>
<td>Reserved</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>28 to 31</td>
<td>Expected Status Sequence Number</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>32 to 35</td>
<td>Reserved</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>36 to 39</td>
<td>Data Sequence Number</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>40 to 43</td>
<td>Buffer Offset</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>44 to 47</td>
<td>Reserved</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>48 to 51</td>
<td>Header Digest (Optional)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>52 to x</td>
<td>Data Segment</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>x to x+4</td>
<td>Data Digest (Optional)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 13 - The SCSI Data-Out PDU [HUFF03]

2.6 iSCSI Discovery Sessions

The SessionType key=value pair is sent by the initiator during the leading login, security negotiation, or operational negotiation phase. The purpose of the SessionType key=value pair is to define the session as a normal session or a discovery session. If the session is identified as “normal”, then a corresponding TargetName key=value pair must also be defined in the login. This means that the initiator is already aware of which target it must connect with. However, there are some instances where the initiator is unaware of which targets are available. Here, the data segment of the login PDU will contain SessionType=Discovery as a text parameter. This will inform iSCSI to provide the names and addresses of all known iSCSI target devices. Once a discovery session is started, a SendTargets=All key=value pair may be issued in a text request PDU. The following
text response PDU will contain a **TargetName** key and **TargetAddress** key followed by the names and IP addresses of all other targets known to the discovery target.

One important role of the discovery session is the ability to receive information on other targets the primary target is aware of. Once a full-feature session is entered, a \textit{SendTargets} command may be used to get this information from the target. This is sent by the initiator as a key=value pair in the data segment of the Text Request PDU to the target. For instance, the initiator may send a Text Request PDU which has a command in the data segment \textit{SendTargets}=<\textit{All}\. The resulting Text Response PDU sent by the target will look similar to the following:

\begin{verbatim}
TargetName=<iSCSI Target Name>
TargetAddress=<Ipaddress>:<portnumber>,<PortalGroupNumber>
\end{verbatim}

This information will be repeated for as many target names known by the primary target. Every target address string will contain a 16-bit value that uniquely identifies a group in the iSCSI target node. This is known as the target portal group tag (TPGT). This tag is extremely important in that it allows the initiator to know what IP address to use when trying to execute multiple connections per session on different IP addresses.

It should be noted that during a normal session, the \textit{SendTargets}=<\textit{TargetName}> text request will only report information about the specified target. If the \textit{SendTargets} command is null, the implied target is the target already connected to the initiator.

### 2.7 Disaster Recovery using iSCSI

The “killer app” for iSCSI technology is disaster recovery and business continuity operations. The iSCSI protocol will allow for near real time data backup across vast distances. This is especially important for those organizations requiring the very best protection possible.

It has been possible for many years to facilitate real-time data backup using specially designed software tools. Most of these tools allow the user to specify a network location
in which to replicate data that will be stored on the user’s local hard disk. The tools also evolved to allow users to specify certain files or directories that should be copied to a remote location. However, there are a few reasons why these tools may not be well suited for high-performance environments.

The first problem is related to storage practices for many organizations. In many cases, the data is not stored on the user’s local disk drive, but rather at some centralized storage server. In these cases, replication tools on the user’s personal computer may not be useful since there are no writes to the local hard disk. This poses additional problems when the storage pool is accessed via a distributed SAN. If different files are located at different sites, the backup software may not be able to coordinate the replication operations correctly. The following section will illustrate the importance of iSCSI in disaster recovery operations.

2.7.1 Small Business Continuity of Operations

In a traditional small business (or home office) environment, data backup generally involves storing information on tape and periodically shipping the data to a remote location for storage. These are the quintessential Class 1 environments and generally have the least effective disaster recovery infrastructures. The high cost of mirroring data remotely leaves most small businesses without a method to keep critical data up-to-date and safely stored at another location. As the amount and importance of data increases, the daily tape backup schedules become more difficult to perform (because large amounts of data require more time to backup). Additionally, as the importance of data increases, it may not be acceptable to lose an entire days worth of information in the event of a disaster. Thus, the frequency of data backup must be increased so as to minimize the amount of information lost during a disaster.

Until recently, any variation of remote mirroring required expensive hardware and a considerable investment in communication infrastructure. This high cost prevented many small businesses from acquiring acceptable disaster recovery solutions because of their
limited financial resources. While mirroring will still require an investment in communication links, a great deal of the high cost hardware is no longer required using a software implementation of iSCSI. Software-based iSCSI requires no investment in specialized switches or network interface cards. In fact, acceptable performance may be realized using software drivers. This is perfectly suited for small businesses that require a high degree of protection at a low cost.

Some businesses have opted to invest in software applications that allow incremental or continuous backups over the LAN to a local storage pool. In very few instances, offsite replication can occur over commodity Internet using these same tools. However, this generally assumes an offsite location to store backup copies (which is not usually the case with small businesses). Nevertheless, if there is an alternate site to store data, these tools can be used to execute incremental backups. However, software iSCSI can do the same job over commodity Internet with some tweaking and is generally available as open source (and hence free). Of course, it is simple to provide security by using SSL or IPSec so that information sent over the public Internet is kept confidential. It is quite obvious that iSCSI is perfectly suited for small and home office disaster recovery because it is more cost effective and comparable in performance to software applications that can cost hundreds or thousands of dollars. There are even possibilities to deploy iSCSI as a transport mechanism while still using certain software backup tools (such as Veritas NetBackup and BackupExec, Constant Data, Legato NetWorker or BakBone NetVault. Previous research conducted at HCS-Gainesville has evaluated these software tools in different settings. Further details of these tools are beyond the scope of this paper but can be found in [HADD03], [CONS03], [VERI00], and [LEGA03]. In addition, numerous white papers are available that summarize the features that each software tool provides.

2.7.2 Enterprise Continuity of Operations

At the enterprise level, most organizations will need throughputs of 1 Gbps or higher between primary and secondary storage. In fact, many businesses are requiring throughputs of 2 Gbps to 10 Gbps. iSCSI is limited to about 1 Gbps with current GigE
networks. As such, this high-performance was only considered possible using Fibre Channel, where current throughputs of 4 Gbps can be realized. The Fibre Channel implementation requires components including specialized switches, host bus adapters (HBAs), and special cables. In addition, system administrators with specialized Fibre Channel skills were needed to oversee the system. The total cost of ownership was prohibitively high for all but a few companies and agencies. Because Fibre Channel offers this high performance, it is generally viewed as a better alternative than iSCSI.

For large corporations or government agencies, iSCSI can also provide a cost effective solution to disaster recovery. In environments that require up-to-the-second data protection, iSCSI can allow for any writes to a primary hard disk to be mirrored immediately to a remote disk. This is where iSCSI generally outmatches Fibre Channel. iSCSI was designed from the ground up to allow for cost effective transmission to remote locations hundreds or even thousands of miles away. Fibre Channel on the other hand, was not originally intended to allow for off-site storage. New variations of Fibre Channel, such as Fibre Channel over IP (FCIP) or Internet Fibre Channel Protocol (iFCP) will allow for long distance connectivity – but at a significant cost. Since the SCSI commands (and data) are encapsulated in TCP/IP packets, any TCP/IP based network can be used for transmission.

Because iSCSI is dependent on TCP/IP networks (and the speed at which those networks operate) it will roughly follow the performance characteristics of the underlying network infrastructure. For instance, if an organization has provided all necessary components of Gigabit Ethernet connectivity, then iSCSI will be able to operate at Gigabit speeds. Likewise, in the event of a future move to 10 GigE, iSCSI would be able to function at 10 Gbps. This of course assumes that there are no performance bottlenecks in the network or I/O elements of the system. In other words, iSCSI performance will scale directly with the performance of TCP/IP based network technology. As will be discussed later, this dependence on TCP/IP can also present challenges when networks must maintain high-performance over long distances.
2.7.3 Remote Backup using iSCSI for Large Organizations

The largest organizations and government agencies will require extremely high performance solutions to disaster recovery. Additionally, the disaster recovery solution must be disruption tolerant as well. This high availability is to ensure that during a disaster information will be replicated at more than one physical location. The need for this high availability is further highlighted when the primary and secondary sites fall within the same region affected by a single disaster (as might be the case with a hurricane or nuclear event). If this occurs, it may result in mission critical data loss, which is unacceptable. To guard against this, it is not uncommon to find secondary or tertiary backup sites, especially for data that is deemed so important that its loss can result in severe financial or national security implications. These sites serve as backups to the backup site and are generally separated by a significant distance from the primary and secondary sites. One may inquire as to why primary and secondary sites are not just separated by thousands of miles. First, long distances can introduce problems especially with latency. As throughput requirements increase, the bandwidth-delay product of the network also increases. These high bandwidth-delay product networks can perform very poorly over TCP/IP. The more important reason however, is performance. If a backup facility is located 3000 miles away, then even small problems with the primary site can require data to be fetched from 3000 miles away, introducing significant performance hits. In addition, maintenance and troubleshooting also become much more difficult when the backup site is located so far away. It stands to reason that the closer a backup site is to the primary site, the better the performance (albeit at the expense of assurance).

2.8 Software and Hardware iSCSI Implementations

As stated before, the iSCSI standard allows the protocol to be implemented in both hardware and software. This is one of the many benefits of iSCSI since it allows different organizations to customize their SAN configuration to a particular budget and performance specification. It will be shown however, that in terms of raw performance, a software implementation can match or even exceed the performance of a hardware-based iSCSI system.
2.8.1 Software-based iSCSI

For organizations that do not require the highest performance and cannot afford expensive host-bus adapters and iSCSI enhanced switches, an implementation is available that requires inexpensive (or free) software drivers. In addition, the drivers can be used with existing IP-based infrastructure so that low-cost standard network interface cards and switches that are already deployed can be used for iSCSI transactions. According to [HUFF03], the software implementation is well-suited for home and small office environments where the SAN must be affordable.

A number of iSCSI drivers are already freely available. The Intel iSCSI driver [INTE03] is a simple implementation which allows targets and initiators to be configured in a number of ways. Discovery sessions and multiple targets are supported and certain characteristics such as block size and PDU burst size can be tweaked to increase performance. Another iSCSI driver that is available is provided by the University of New Hampshire (UNH) Interoperability Lab [UNH05]. This implementation is more complex and capable than the Intel driver. In addition to providing the same functionality of the Intel driver, the UNH driver provides support for multiple connections per session, authentication, and authorization procedures. According to [AIKE03], both drivers are comparable in performance and thus the choice of which to use is dependent on the amount of customization required by the user.

In a software-based iSCSI system, all iSCSI processing is done by the processor and (in most cases) TCP/IP operations must also be executed by the CPU. Since the NIC is merely an interface to the network, these implementations generally require a great deal of CPU cycles to function. iSCSI and TCP operations can use up as many as 60% of the CPU cycles. In fact, it will be shown in Section 3.3 that when the throughput demands are increased with iSCSI, the CPU utilization also increases. With this in mind, any system that is heavily loaded can be expected to perform poorly. Likewise, CPUs that are busy with TCP/IP and iSCSI operations will not have the CPU cycles needed to
perform other tasks. As the throughput requirements increase, the number of CPU cycles used for the TCP/IP and iSCSI operations will also increase.

2.8.2 TCP/IP Offload Engines

As network infrastructure has reached Gigabit (and now 10 Gigabit) speeds, the network resources are becoming more abundant and the bottleneck is moving from the network to the processor. Because TCP/IP processing requires a large portion of CPU cycles, some software iSCSI implementations may be coupled with specialized network interface cards with TCP offload engines (TOEs) on board. NICs with TOE integration have hardware built into the card that allows the TCP/IP processing to be done at the interface. By preventing the TCP/IP processing from making it to the CPU, the memory and PCI bus can also be spared for other applications [ATOE03]. This frees the processor and allows it to spend more resources on other applications. In any environment where the CPU may be required to undertake other tasks, having a TOE enabled network interface card is certainly desirable. Networks are beginning to outpace CPUs in performance as evidenced by the emergence of 10 Gigabit Ethernet. Processors are just barely keeping up with the increase in network speeds and as a result, TCP/IP is requiring a larger portion of the CPU resources to sustain these high speeds [ATOE03].
2.8.3 Hardware-based iSCSI

In a hardware-based iSCSI environment, the initiator and target machines employ a host bus adapter (HBA) that is responsible for both TCP/IP and iSCSI processing. This will free the CPU from both TCP/IP and iSCSI functions which can dramatically increase performance in those settings where the CPU may be burdened with other tasks. Any high-performance environment with money to spend will find the hardware (HBA) approach much more satisfactory. Indeed, there are still more benefits to the hardware approach. Some switches being marketed are currently adding iSCSI support for even higher performance. In most cases, iSCSI to Fibre Channel bridging is also being introduced so that organizations can introduce iSCSI to work with their already existing Fibre Channel appliances and infrastructure. It appears as if Fibre Channel and iSCSI will coexist for some time. The hardware-based iSCSI solutions are generally best suited for enterprise applications and environments where CPU loading cannot be tolerated and high-performance is critical.

2.8.4 Comparison of Hardware and Software-based iSCSI

It appears that use of hardware iSCSI HBAs is the most desirable approach when performance requirements are high. However, [AIKE03] shows that there are cases when the software method outperforms hardware implementations. For their experiment, an iSCSI testbed was created that used two Intel network interface cards. The first used a Pro 1000F NIC (1 Gbps) which had no iSCSI hardware built into it. This of course would require a software driver for iSCSI instructions. The other test used an Intel Pro 1000T NIC with on-board iSCSI hardware. The Pro 1000T card does not require iSCSI drivers in software.

When the tests were performed, the Pro 1000F was able to sustain over 60 MB/s for 16KB block sizes. Note that these block sizes are in fact the PDU burst size that can be defined during the login operational negotiation phase. However, the iSCSI enabled Pro 1000T could not exceed 25 MB/s using the same block size. When these blocks were increased to 2 MB, the Pro 1000F performance reached almost 100 MB/s. Again, the
iSCSI enabled 1000T lagged far behind at about 30 MB/s. The important observation is that in this case, the software implementation significantly outperformed the hardware iSCSI platform. This contradicts the previous claim that a hardware system will outperform a system realized only in software. However, this is not as unreasonable as one may first think. The Pro 1000T cards were designed for lower-end systems and used a 200 MHz processor for the iSCSI functions. Since the experiment was run on an 860 MHz Pentium III machine, the host processor was much more powerful than the 80200 processor on the iSCSI card. For this reason, the software implementation running on the host processor could sustain higher throughputs but this performance advantage would likely be severely limited in the event that the host processor was loaded with other tasks.

It should also be noted that in the same experiment, the performance of both 1 Gbps and 2 Gbps Fibre Channel systems were measured. Of course, the 2 Gbps FC system outperformed all other solutions. However, according to [AIKE03], the performance of the software implementation of iSCSI was comparable to that of the 1 Gbps FC system. This illustrates how affordable software-based iSCSI can perform with an equally capable FC implementation. Considering that there is no inherent distance limitation with iSCSI, the benefits of iSCSI begin to become apparent.

As part of this thesis, additional testing was conducted using the same software drivers as those used in [AIKE03]. The experiments provide additional validation of their findings. The performance results of those experiments can be found in Section 3.

**2.9 Cancer Therapy and Research Center Case Study**

In order to illustrate the effectiveness of iSCSI as a disaster recovery mechanism, a case study is provided to show the various ways that iSCSI can be used to improve backup and recovery performance. As demonstrated below, iSCSI provides more than just a cost effective way of performing remote backup and recovery. It is this versatility that may accelerate the adoption of iSCSI as a critical component of business continuity applications.
This case study takes place at the San Antonio, TX based Cancer Therapy and Research Center (CTRC). CTRC provides life-saving treatment for its patients while performing research to discover the causes and cures for various types of cancer. The increasing number of cancer patients requiring life-saving radiation therapy has overburdened the CTRC’s computer storage structure due to an increased volume of MRI, CT, and PET scans which are stored electronically. This increased patient demand requires the adoption of a new storage strategy to address storage needs and enhance the ability of CTRC to recover from planned and unplanned downtime.

2.9.1 Background and Requirements

The Cancer Therapy and Research Center currently uses 8 radiation treatment vaults that are used approximately 200 times per day on average [LUTE03]. Patients are normally on a 4-week cycle and CTRC estimates that each patient requires about 100 megabits (12.5 MB) of data for a treatment plan comprised of MRI, CT, and PET scans stored on the servers [CTRC03]. Other information must also be stored on the servers including radiation treatment details, schedules, medical transcripts, billing records, and patient demographics. Now requiring over 3 terabytes of storage, CTRC estimates that digital imaging techniques will drive server and storage growth even further and as a result, their existing server-attached storage infrastructure will be difficult to scale and will not provide the high availability needed to ensure mission continuity. These restrictions are especially critical given the nature of CTRC’s operation. Because radiation treatment requires a strict scheduling procedure, system downtime cannot be tolerated that causes patients to miss their scheduled therapy appointments.

The current CTRC business model requires that one patient be treated every ten minutes in each of its radiation treatment vaults. To keep operating efficiently, the Radiation Oncology Department uses a number of electronic resources to automatically catalog radiation doses given in each session. Since these electronic resources reside on a server,
a server failure results in disruption of these applications and significantly hinders the treatment process.

Additionally, the problem is further complicated by the physical layout of CTRC. The primary research facility resides on a separate campus from the medical center and the sites are located approximately 22 miles apart. Each facility already made use of its own centralized storage but allowing each site to access resources from the other site would provide many advantages. Moreover, CTRC required a mirroring solution so that the research site could act as a backup location for the medical facility. Avoiding a proprietary solution is also desirable since it is impossible to predict what will become of a company 5 or more years into the future. Employing a standardized solution helps to ensure longevity. CTRC has also invested a substantial amount of money in a 1-Gigabit Metro Ethernet Network connection between the two sites. Any solution that allowed CTRC to use already existing IP network infrastructure would be beneficial [CTRC03].

2.9.2 An IP SAN Solution

With these factors in mind, CTRC decided to install a SAN to accommodate their needs (see Figure 15). The 22-mile separation poses problems for the 10-km distance limitation of Fibre Channel. It would be possible to use Fibre Channel extenders but that option was considered too expensive. Therefore, iSCSI became the protocol of choice since it provided all of the needs that CTRC needed and it took advantage of the already installed IP infrastructure. Distance limitation is not a problem with iSCSI and it was a standardized protocol. CTRC installed an EMC F4700 SAN unit at each location and added redundant Cisco SN5428 storage routers at each site (resulting in two SN5428s at each location). The SN5428 routers offered the iSCSI connectivity as well as bridging for Fibre Channel. CTRC also choose to install Veritas NetBackup Data Center as the managing software [CISC03]. This is an example of software tools and iSCSI being used as complementary technologies.
The iSCSI implementation was realized in software by downloading iSCSI drivers online. The drivers are installed on servers to encapsulate SCSI commands in TCP/IP packets. These packets are sent like any other over the IP network and sent to the Gigabit Ethernet ports on the SN5428 router. Once the packets are transmitted, the traffic is output on the SN5428’s FC ports, which are connected to the EMC F4700 storage array (shown in Figure 15 as Fibre Channel Storage). It should be noted that the storage array is in fact a Fibre Channel device. The SN5428 storage router provides iSCSI to FC bridging so that the FC devices can be used. In this case, CTRC was not required to invest in host bus adapters and could use much cheaper Gigabit Ethernet Network Interface Cards (NICs). In essence, this allows Fibre Channel devices to be used at each of the sites with iSCSI providing the transport between sites [CTRC03].

2.9.3 Results and Performance

The SANs at both facilities were mirrored using synchronous iSCSI over the 1 Gbps Metro Ethernet link already in place. The procedure is very simple in that every time
data is written to an individual logic unit number (LUN) on a local F4700 that data is also encapsulated using iSCSI and transported across the Metro Ethernet link. Once it arrives at the sister-site 22 miles away, it is stripped of its iSCSI header, and written to an identical LUN on the remote SAN. Doing so ensures that an up-to-date copy of data is contained at each site and it does not require a backup window to perform the mirroring operation.

In one instance, a power outage occurred at the research facility and brought down the EMC storage appliance there. Once that happened, the servers on both sides initiated communication across the gigabit MAN link. Since data had been mirrored at the medical facility, it was the data used to continue operations at the research center. Operations continued without any downtime until power was restored to the storage appliance at the research facility [CISC03].

A different set of problems occur when it is not storage that fails, but one of the application servers needed for therapy-related applications. In the event of a failure, the server has to be rebuilt, which requires a boot image and total reconfiguration before it can function again. This process can take anywhere between two and eight hours. Every ten minutes that elapse causes eight patients to miss life-critical therapy. However, the iSCSI remote booting technology (coupled with Cisco Network Boot) allows CTRC to consolidate the system drive or “booting function” from each of its servers onto the EMC F4700 storage appliance [CTRC03]. This capability allows extremely fast system restart times in the event that one of the servers experiences a malfunction. Instead of needing between two and eight hours for the rebuild, the server can be functioning within 10 to 20 minutes.

2.10 iSCSI RAID Configurations

A great deal of work has been done in order to identify the bottlenecks existing in an iSCSI-based SAN. While some believe the performance bottleneck lies on the network, others believe that it is a storage (disk) issue. To find out, a group of researchers from
Tennessee Technological University have developed a novel approach known as iSCSI RAID (iRAID) [HE03]. The main goal was to solve performance and reliability limitations of iSCSI SANs.

Essentially, the iSCSI storage targets were arranged in such a way that they mimicked a RAID configuration where striping and parity techniques could be employed [HE03]. Two methods were examined. The first was a striped configuration, known as S-iRAID. The second approach was a rotating parity methodology, known as P-iRAID. The S-iRAID design will improve the performance since writes are conducted in parallel while the P-iRAID system improves reliability by using the proven advantages of RAID parity. Four iSCSI targets were connected to the initiator with a gigabit enabled switch.

A data set of 1GB was sent to the targets using I/O request sizes from 4 kB to 64 kB. The striping S-iRAID experiment resulted in an average throughput of around 46 MB/s. This was 5.34 times greater than the throughput obtained with the vanilla iSCSI configuration (which maxed out at less than 10 MB/s). The P-iRAID solution improved to about 17 MB/s, 1.71 times faster than the vanilla iSCSI solution. It should be stressed that these

![Figure 16 - An S-iRAID and P-iRAID configuration [HE03]](image)
experiments were actually writing data to a disk and thus did not utilize the entire gigabit link. However, since the striping was done only across 4 drives, it is certainly possible to enhance performance by striping the data across more target nodes. The tests were performed on 10, 100, and 1000 Mbps networks. [HE03] concluded that the network was a bottleneck at the slow transfer rates of 10 Mbps. However, since their experiments only showed 12% utilization of the gigabit link, the authors concluded that with the high capacity networks that the targets will become a performance bottleneck. Experiments conducted at the High Performance Computing and Simulation Research Laboratory (HCS-Tallahassee) will show that even at gigabit speeds, it is possible to utilize over 80% of the network using a software-based iSCSI implementation (see Section 3.2.2). The reliability benefits are extremely useful in that if any target fails, the other three have enough information (through parity) to reconstruct the data. This can dramatically improve the availability of mission critical data.

2.10.1 Distributed iSCSI RAID Configurations

The successful implementation of the S-iRAID and P-iRAID solutions allows a remarkably novel approach to mission continuity in the event of a disaster. When the P-iRAID solution is in place, the overall assurance of the network is increased. When those target nodes are distributed, it is possible to further increase the overall survivability of the network from a single disaster. Recall from Section 1.2 that the agreed upon distance that a backup facility should be from its primary site is 105 miles.

Consider network as shown in Figure 17. Here, the primary facility is located at the optimum distance of 105 km from the backup facility. Furthermore, there is a secondary facility that also performs certain tasks and can also act as part of the distributed iSCSI environment. Using the P-iRAID configuration as described in [HE03], it is possible to allow each site to hold a parity block and act as a distributed RAID-5 system. Thus, if any one site is destroyed or taken offline, there is enough information to rebuild all of the data and avoid any loss of information. These are the types of solutions that may be very beneficial to intelligence and defense agencies where there may be a higher risk of man-
made disasters as well as a large enough budget to afford multiple sites and high-bandwidth network infrastructure.

![Figure 17 - P-iRAID may allow for disaster tolerant distributed data storage [HE03]](image)

This is a very important step in realizing the full potential of iSCSI as a disaster recovery medium. Ensuring that remote disk writes and reads can be done in parallel as if they were part of a local RAID 0 configuration further enhances the performance of the iSCSI system. By implementing the P-iRAID method, it is possible to increase the reliability of the system and thus the fault tolerance. Extending this concept to a distributed network can substantially increase the overall reliability and availability of the system. These developments are absolutely crucial in making iSCSI an attractive technology for remote data replication and recovery.

### 2.11 MC/S and Link Aggregation

The high-performance monopoly of Fibre Channel poses a number of challenges to the eventual adoption of the iSCSI protocol by businesses that require the highest throughput for their backup and recovery networks. The IETF was aware of this and incorporated
mechanisms into iSCSI to allow it to compete with the fastest Fibre Channel hardware. Perhaps the most important addition to iSCSI was the introduction of multiple connections per session (MC/S). Not only does MC/S allow performance enhancements to iSCSI, but it may also be used to guard against single points of failure [SMIT94].

MC/S allows multiple TCP/IP connections to combine in parallel and function as if they were part of the same session as shown in Figure 18. This allows iSCSI data to be sent across several different physical links from the initiator to the target as long as it is supported by the iSCSI drivers. The addition of MC/S does introduce more complexity to the iSCSI protocol. Even though TCP is responsible for delivering data in order to the target, there must be additional information provided by the iSCSI protocol to ensure that data arrives in order across multiple connections. TCP/IP can only guarantee that data arrives error free and in order across each connection – but not necessarily that data will arrive in order across all connections. This means that the iSCSI PDU must contain extra information such as counters and flow control mechanisms. There are a number of command and status numbering fields in the iSCSI PDU which will be discussed shortly.

During the initial login, the initiator will set a new connection ID (CID) so that each connection can be identified. Any connection related command can be managed by the target or the initiator by specifying the CID for the corresponding connection. The iSCSI driver must also keep track of which connections make up individual sessions. The Initiator Session ID (ISID) is responsible for providing a unique name to each session.
within the initiator and target. Therefore, a CID may have multiple ISIDs that are linked to it so that a single connection is comprised of multiple sessions. The ISID is crucial when determining what connections must be torn down or reestablished as part of a single session. An iSCSI initiator port may be made up of more than one iSCSI HBA. This is in contrast to Fibre Channel where each HBA maps one to one with a SCSI port. The result is that iSCSI initiators may actually utilize multiple HBAs. When multiple connections make up a single session, there are a number of issues that must be addressed.

2.11.1 Ensuring In-order PDU Delivery over MC/S

TCP/IP will ensure in order delivery of packets over a single connection. But what happens when a command PDU arrives over one connection (in order) before a preceding command PDU arrives on another connection? This is shown in Figure 19. iSCSI contains sequence numbers in PDUs so that it can reconstruct the PDU ordering at the destination. Commands can be sequenced across all connections within the session [HUFF03]. This creates the need for command sequence numbers (CmdSN) to act as counters so that the target can deliver the commands in order to the SCSI layer even

![Figure 19 - TCP may deliver PDUs out of order if spanning multiple connections](image)
though the commands traveled on different connections and may have been delivered out-of-order by TCP/IP. The CmdSN is the only sequence number preserved across separate connections within a session.

iSCSI must also maintain a corresponding expected command sequence number (ExpCmdSN) for the CmdSN values. The expected command sequence numbers can be used to verify the receipt of command sequence numbers up to the ExpCmdSN value. In other words, the ExpCmdSN is the first command sequence number that has not been received. The benefit of this method is that there is no need to acknowledge the arrival of each PDU. Instead, the target may inform the initiator as to which command sequence number it needs and the initiator knows that all commands have been received up to that point. An additional benefit is that the case is rare when an error arrives to the iSCSI layer that has not been fixed by TCP/IP. Therefore, using this procedure significantly reduces the overhead incurred by sending rarely needed acknowledgments [HUFF03].

### 2.11.2 Benefits of Using MC/S

The importance of MC/S is three-fold. The most obvious benefit to transporting commands in parallel across multiple connections is the ability to provide link aggregation. For example, two 1 Gbps HBAs can be used simultaneously to provide a total throughput of 2 Gbps. This is assuming that the target side has sufficient resources to handle this throughput. In addition, since the transport of iSCSI is dependent on the underlying Ethernet technology, it is possible to combine multiple 10 Gbps links to allow for extremely high-speed communications between initiator and target. This is iSCSI’s answer to high-performance Fibre Channel components. But there are added benefits to spanning several connections in a single session. One such advantage is that it can aid in the overall fault tolerance of the connection. If a session is made up of 4 connections, and one of the connections were to fail, the other 3 could in theory take over for the failed connection in order to continue the data transmission (albeit at degraded performance). At the same time, it also provides a possibility for load balancing. If a connection is beginning to fail (or becomes congested), another connection could be started in order to
stand in for the congested link. This would require that an element of intelligence be added to the iSCSI protocol so that fault tolerance and load balancing could be realized. In a disaster recovery operation however, when mission critical data is being accessed from a remote location, fault tolerance of the system becomes extremely important. This will be discussed further in Section 2.12.1.

2.11.3 MC/S Implementation Results

The ability of an iSCSI session to span multiple connections is used in [XION04] to develop a higher performance solution. An experiment was performed over a metropolitan network and multiple virtual connections were opened over a single physical connection. It was argued that this approach would not only improve performance, but could also provide a better way to handle latency that could become problematic in a metropolitan network. In this experiment, one connection was used for sending SCSI requests from the initiator to the target while the other was used for sending target responses to the initiator.

The results show that multiple connections can allow the gigabit metro network to achieve speeds of 97-107 MB/s [XION04]. It was also discovered that high throughputs result in higher CPU utilizations (a claim that experiments conducted at HCS-Tallahassee can verify). The metro network was able to sustain throughputs of 67 MB/s using 1.5 kB frames (as opposed to the 9 kB sizes used with jumbo frames). This is further indication that an iSCSI software implementation can traverse large distances with considerable latencies and still maintain high-performance. The effect of queue length was also studied to examine how iSCSI performed for different queue sizes. The results were not surprising, since larger queue lengths showed performance gains over smaller queue lengths. For instance, when the queue length was set to 2, the I/O rate was measured at 4000 I/O operations per second (IOPS). However, when the queue length is increased to 16, the I/O rate reaches 15,000 IOPS.
2.12 Limitations of iSCSI

Although iSCSI was designed from the beginning to be more capable than Fibre Channel, there are a number of limitations that must be addressed in order to make iSCSI a viable option for large-scale, high performance, disaster tolerant networks. Some of these limitations are a direct result of iSCSI’s reliance on TCP/IP networks for transport and others are a result of the nature of the protocol itself.

2.12.1 Fault Tolerance

Perhaps the most important limitation in regards to business continuity is fault tolerance. Fault tolerance was never directly addressed by the IETF when the iSCSI standard was being revised. In fact, TCP/IP is responsible for the in-order delivery of the iSCSI PDUs to the target and any link failures during deliver require TCP/IP to find a new route. But this is not necessarily fault tolerant. There are no built-in mechanisms in iSCSI to allow for the connection loss between an initiator and a target login unit [GILL03]. However, there are a number of features that the protocol comprises that may be used to add some degree of fault tolerance. The key components are the target discovery process and temporary redirection functions.

First, it is critical to understand where the responsibility of iSCSI ends and the responsibility of TCP must begin. Consider an initiator connected to an IP network as shown in Figure 20. It is the job of TCP/IP to make sure that the data sent by the initiator makes it to the specified target device. If a switch fails between the initiator and target, then TCP/IP must redirect the data to another switch so that it can arrive at the target device. This is one type of fault tolerance in the iSCSI system in that any infrastructure between the initiator and target that fails can be compensated for by the TCP/IP transport. However, there may be cases where the specified target fails.
Consider the situation in which the initiator at the primary location is replicating data at a remote site (Target A). If any of the switches or links between the initiator and target A fail, TCP/IP can still make sure that the data arrives at Target A by providing a mechanism for finding other routes in the network after a failure as shown in Figure 21.

TCP is not aware that iSCSI even exists. And, in the event of congestion TCP may find another way to traverse a network but in the event of an iSCSI node failure, TCP has no way of finding a replica iSCSI target to write to and read from. In this case, iSCSI must be able to detect such a failure, and then reroute data to an alternative node by changing the target that it will write to as shown in Figure 22. The iSCSI protocol has redirect packet options, as well as target portal group tags that may allow the protocol to be aware of alternate write (and read) targets. As such, an iSCSI failure would go unnoticed to the user or application, granted the performance of the two targets were comparable. And, since iSCSI is encapsulated in TCP packets, adding intelligence to the protocol will not
require any changes to TCP/IP. In short, TCP/IP will maintain congestion control and guard against link failures while iSCSI can address target node failures and congestion. This adds one more degree of fault tolerance to an iSCSI-based distributed storage area network. The effect of link failures in a SAN is highly dependent on the switch architecture and routing algorithms used [MOLE00]. For this reason, more research is necessary to determine exactly how the performance of an iSCSI SAN is degraded during such failures. This may also shed light on how to best organize a fault tolerant iSCSI network.

![Figure 22 - Intelligent iSCSI is aware of alternate replica targets](image)

An important component of the fault tolerance solution is the target discovery process (see Section 2.6). The initiator node can use a PDU with a discovery type session (as a key=value pair) in order to obtain crucial information from the target. This information will give information on the portal configured at the target node. A portal is simply a TCP port that is paired with an IP address. The discovery process will return a set of portals to the initiator that the target is aware of. This set is known as initial target portals. According to [GILL03], the next stage is for the initiator to open a connection to each initial target portal (in order) until it finally succeeds. The initiator then logs in as usual and the target can now direct the initiator to a new portal using the iSCSI temporary redirect function. This is the point where the initiator logs out and then opens a new session with new redirected portal and can now send data. In the event of a connection failure, the initiator can try to connect to one of the initial target portals learned in the discovery session. In most cases, one of the connections would succeed and data could begin flowing once again. If the initiator is unable to reconnect, it can execute another
discovery process, repeat the above steps, connect to a redirect portal, and begin sending data.

Using this methodology, fault tolerance can be achieved if the target is able to instruct the initiator to connect to a secondary network interface or a different target node in a distributed iSCSI target system. It is important to emphasize that the temporary redirect function is critical to this approach. The initiator (by itself) may not know which target systems are part of the same portal (and thus the same storage pool). The target is aware of this because of the target portal group tag (TPGT) and must thus communicate this information to the initiator so that the data can be sent where it is needed.

2.12.2 Load Balancing and Manageability

In the event of a failure, there is no implicit way for iSCSI to balance loads across the “new network”. Moreover, even when the network is healthy, the iSCSI protocol does not manage network loads so as to avoid congesting a specific target node. For example, if a number of initiators happen to connect to a target during the same time frame, the performance realized at each initiator will be severely degraded. In most real-world applications, there will be more than one target that an initiator can connect to in order to access the same data. Thus, the possibility exists that multiple users will unintentionally contend for the same bandwidth at the same target. The users would experience degraded performance but would not have any simple way of reconnecting to another target that holds the same data. Using the same techniques as outlined in Section 2.12.1, it would be possible for some degree of intelligence to be added to the iSCSI protocol. This layer of intelligence would allow iSCSI to identify periods of contention and take appropriate action so as to redirect some initiators to other alternative targets.

When the number of iSCSI initiators and targets increase, so does the complexity of managing the network. This is especially true when considering remote mirroring and replication involving hundreds of nodes spread over hundreds of miles. Moreover, the possibility exists that nodes may not remain static and targets may change over time.
Manually keeping this information would be a heavy burden on any system administrator. The iSCSI discovery options are essential in providing manageable networks in these instances. There has been some discussion on how to best accomplish discovery over large distances with hundreds of nodes while maintaining security and performance.
CHAPTER 3
PERFORMANCE TESTING OF iSCSI

3 Software-based iSCSI Experiments

With the option of using iSCSI in a strictly software-based configuration, a number of questions arise as to what performance can be expected in a software-based implementation. It is critical that the benefits as well as the limitations of software iSCSI are known so that a position can be adopted on its usefulness as a disaster recovery mechanism. This section aims to address various performance issues when utilizing a software-based iSCSI network, especially when considering its usefulness as a disaster recovery mechanism.

3.1 iSCSI Testbed

The following experiments have been executed on a testbed built for the sole purpose of providing a dedicated system for iSCSI experiments. The configurations are changed depending on the nature of the experiments. The initiator is realized using a system with dual Pentium III processors and 256 KB of cache, 256 MB of memory, and one Fast Ethernet network interface card (Epsilon nodes). One of the target machines is the same as the system described above. A second target system (Iota) consists of a 2.4 GHz Intel Pentium 4 processor, 512 KB of cache, 1 GB of memory, Fast Ethernet NIC, and running Linux kernel 2.4.20. This will effectively replace one of the Epsilon targets so that a

Figure 23 - HCS Lab iSCSI Testbed
performance comparison can be made between two different CPUs. Since all experiments will be software-based, it is expected that more powerful CPUs will be able to perform better than lower-end processors.

3.2 Block Size and Throughput

Using the Intel iSCSI driver (version 20), it will be possible to make some performance measurements on the iSCSI testbed. The first test will be to determine if the block size accepted at the target will have any effect on the overall throughput of an unloaded system. Although it is rare that a server would remain unloaded, this will illustrate how iSCSI can perform on a software level. This gives insight as to what performance can be reached by iSCSI. Understanding how the block size affects the performance of the system is essential in understanding how to deploy the optimum solution.

3.2.1 Fast Ethernet Results

The first experiment takes advantage of the already existing Fast Ethernet connections in place on the iSCSI testbed. In any small business environment, this may provide enough throughput to allow for an effective backup and restore operation. On an unloaded system, the processor should have the resources required to manage TCP/IP processing and thus provide a fairly high degree of performance. The block size is configured at the target to be anywhere from 512 to 4096 bytes. After 25 iterations of each block size were
averaged, the results were recorded. It appears as though there is little dependence on the block size except for very small request sizes but overall, the 1024 byte blocks appear to provide the most consistent performance. However, the throughput appears to be quite dependent on the number of bytes per request issued by the initiator. Figure 24 shows how throughput is affected by block size for reads to the disk over the 100 Mbps network. Similarly, when write operations are examined, shown in Figure 25, it is again illustrated that block size has little effect on the throughput. As is the case with write operations, the most important parameter is the bytes/request since the throughput increases as the number of bytes per request is increased. The best throughput obtained using 262,144 bytes/request results in about 10.2 MB/s. This corresponds to a link utilization of approximately 81.6%.

3.2.2 Gigabit Ethernet Results

It is possible that since the Fast Ethernet was essentially saturated by the experiment that the full affect of block size on the throughput could not be realized. In order to find out if this was the case, the same experiment was conducted using a Gigabit Ethernet link between the initiator and target. In this case, no switch was used between the target and initiator. The processor was unloaded during these experiments so that the full performance of this software driver could be realized. The throughput on the 1000 Mbps link increased dramatically as expected. However, it appears that once again the block size had little effect on the performance of the system. The 512 and 1024 byte blocks
showed the best performance for large byte/request sizes during write operations. This was not enough of an increase to make the claim that those block sizes are best suited.

Furthermore, some additional testing would likely reveal that all block sizes resulted in comparable performance. The measured performance of data reads over iSCSI follows a similar trend in that the block sizes do not seem to significantly affect the performance. It is evident by the measured results that the Intel iSCSI driver is not taking advantage of the high speed Gigabit channel. 60 MB/s is about the best this driver can offer in its current form (as illustrated in Figures 26 and 27). This corresponds to a link utilization of between 45-50%. The possibility still exists that the iSCSI PDU data segments are not large enough to fill the pipe and thus performance is not as good as one may expect. The trend in bytes/request gave some
hints as to what parameters may be tweaked in the source code in order to better utilize the network resources.

After carefully examining the driver source code, the bytes/request parameters were observed to correspond directly with the iSCSI PDU burst size. A modification in the source code allowed the default PDU burst size to be increased by a factor of 8. Thus, the PDU burst size was increased from 262144 bytes (256 kB) to 2097152 bytes (2 MB). By packing more information into the data segment of the PDU, it is possible to better fill the Gigabit link while requiring fewer acknowledgements and cutting the amount of overhead incurred by the TCP/IP protocol. The result should be an increase in the throughput. Figures 27 and 28 show the results of this optimization.

**Figure 28 - Link Utilization for writes improved to 80% with 2 MB PDU bursts**

**Figure 29 - Link Utilization for reads improved to 70-80% with 2 MB PDU bursts**

It was necessary to avoid bottlenecks that could be encountered during the disk accesses since the aim of this experiment was to determine the performance of iSCSI over the
network. After all, there are a variety of ways to increase disk write performance with
the deployment of RAID. Thus, a RAM Disk partition was created so that writes and
reads could be executed as fast as possible. Since nearly every high-performance
implementation will incorporate RAID striping to boost disk performance, the higher
read and write rates provided by the RAM disk method is certainly a legitimate
assumption to make. When the experiments were run with the increased PDU burst sizes,
it was clear that this method would result in a substantial boost in the throughput. Once
again, despite using nearly all of the available bandwidth of the Gigabit link, the block
size did not seem to affect the sustained throughput.

It should also be noted that the performance increases were also dependent on the
maximum transmission unit (MTU) size. For standard Fast Ethernet, the MTU size is
generally set at 1500 bytes. However, Gigabit Ethernet includes the option for MTU
sizes as high as 9000 bytes. These are known as jumbo frames. For most 32-bit
processors, jumbo frame support is necessary to guarantee Gigabit transmission rates.
With the MTU size set to 1500 bytes, the best iSCSI link utilizations measured were
about 60%.

3.3 iSCSI CPU Utilization

So far, the software-based iSCSI methodology appears to perform as well as or better
than the hardware configuration that is usually associated with high-performance
systems. However, by themselves, these results can be misleading. The biggest
disadvantage to using a software iSCSI approach is that it puts a heavy burden on the
CPU. In a situation where the host CPU is not required to perform other duties, the
number of CPU cycles wasted on iSCSI and TCP/IP processing can be overlooked.
Unfortunately, this is rarely the case in a real-world environment. To understand just
how much the iSCSI protocol tasks the processor, more experiments were conducted.
The first experiment used the Gigabit link already in place on the iSCSI testbed. The CPU utilization was measured for both reads and writes at PDU Burst Sizes from 8 kB to 2 MB. Since PDU burst sizes directly correspond to the throughput, it is also possible to gauge the effect that sustained throughputs have on CPU utilization. This would give some insight on the performance characteristics achievable by systems with certain specifications. The results in Figure 30 show that the CPU utilization dramatically increases as the PDU burst size increases. It may appear that the PDU burst size increases should decrease the amount of CPU cycles being used since there is less overhead with larger bursts. However, due to the nature of the measurements, this is not the case. Since the utilization measurements were taken as a time average, the smaller bursts were written to memory almost immediately while the next datagram was traveling over the network. The result is that the CPU spends a relatively large amount of time idle, waiting for the next piece of data. With the larger bursts, the data is still being written to memory when the next piece of data arrives. Therefore, the CPU does not spend a lot of time waiting for a new chunk of data but is busy for most of the duration of the transaction.

The overall size of the data being written also has an effect. For instance, when only 8 MB of data is written to the RAM disk, the CPU utilization does not exceed 10% but when 128 MB of data is written to the RAM disk, this utilization can fall anywhere between 20% and 80%. This should not come as a surprise since the larger data sets will require more TCP/IP packets to be sent and as such, more processing on the host CPU.
3.4 CPU Loading Effects on iSCSI Throughput

While we know how the software iSCSI drivers impact CPU utilization, we don’t yet have a clear understanding of how well the Intel iSCSI drivers will perform when the processor is loaded. If the processor has any other tasks to perform, the throughput of the iSCSI backup and restore processes could be adversely affected. Therefore, the next set of experiments was conducted to understand how CPU loading affects iSCSI performance. These tests were executed over the Gigabit Ethernet link already in place between two Epsilon nodes. A variety of programs were run in the background so that CPU loads of 25%, 50%, and 75% could be simulated. After the CPU was properly loaded, an iSCSI connection was formed and throughput measurements were taken for both read and write operations. The results (Figures 31 and 32) were not surprising in that they showed the throughput of the iSCSI transactions drop significantly when the

![Figure 31 - CPU Loading Effects on Write Operations](image)

![Figure 32 - CPU Loading Effects on Read Operations](image)
CPU was loaded. The difference between throughputs at small PDU burst sizes was relatively small but at large PDU burst sizes, the throughput was significantly lower for higher CPU loading conditions. Link utilizations of only 40% were measured when the CPU load was 50%. This is half of the 80% utilization that can be realized when the processor is unloaded. At 75% loading, this link utilization drops to only 32%. It becomes apparent that any software iSCSI implementation will be severely limited in performance if the target processor is busy with other tasks. The possibility exists to decrease the performance hit incurred by CPU loads. If the processor is more powerful, it may be able to do a better job of balancing the demand for iSCSI processing and other CPU intensive functions. Since the Iota cluster at HCS-Tallahassee is significantly more powerful than the Epsilon cluster, a similar experiment could be conducted to measure how the CPU specifications affect the throughput of iSCSI under the same loading conditions.

3.5 iSCSI at 10 Gbps

While Fibre Channel continues to improve in terms of network performance, iSCSI must do the same in order to remain competitive. It is not uncommon for Fibre Channel networks to perform at 4 Gbps and many are beginning to migrate to 10 Gbps already. While iSCSI can match the 4 Gbps networks using MC/S, aggregating 10 GigE connections to form a single 10 Gbps connection becomes a difficult task. iSCSI will benefit considerably from the adoption of 10 GigE technology. In essence, it is 10 GigE that will allow iSCSI to gain a performance advantage over Fibre Channel.

Any 10 Gigabit Ethernet implementation will require the use of high performance ASICs so that SCSI and TCP processing operations can be done without impacting the performance of the CPU. In addition, a direct path to memory is also likely to be needed at 10 Gbps. At these speeds, the throughput from TCP and iSCSI transactions will be far higher than any CPU can handle effectively. Thus, the iSCSI data transfers could be performed directly to and from memory buffers using dedicated hardware [BLAC03]. The buffering of out-of-order packets will reduce network traffic and processing time.
since retransmission can be largely avoided. It should also be noted that a version of TCP that allows selective acknowledgement and explicit congestion notification would be preferable since these features can also decrease the number of overall retransmissions.

With the Initiator Task Tag (ITT) containing information that relates the data to the corresponding commanding, the initiator can start putting the data directly into the host memory as soon as the HBA (in a hardware implementation) receives the TCP/IP segment containing the necessary Data-In PDU header. This is a form of remote direct memory access (RDMA). There are many advantages to this. First, it reduces latency because of the direct route to memory. Also, CPU utilization is reduced since the host processor does not have to perform extra data movement operations. Finally, the data-reassembly buffer is significantly reduced on the HBA. These extra bonuses are extremely important in realizing 10 Gbps iSCSI solutions.

Understandably, 10 GigE is greatly anticipated but at the same time, it appears that it will be difficult to reap the full benefit of the technology with iSCSI without providing the necessary foundations. To find out just how capable standard systems are, some preliminary experiments were run at HCS-Gainesville using two 10 GigE network cards (without a switch). The tests were performed on the new Kappa cluster in Gainesville. Each Kappa node has dual 2.4 GHz Intel Xeon processors, 533 MHz FSB, 1 GB of memory and 100 MHz PCI-X bus slots. The iperf benchmark was executed to gauge how fast transmissions could be completed on the 10 GigE network. The network could not perform any better than 1750 Mb/s (which is a 17.5% link utilization). This was not unexpected since the maximum theoretical bus speed for PCI-X is 400 MB/s (3200 Mb/s). However, from the few tests that were conducted, software iSCSI continued to be impressive. At the 32 kB/request size (burst size), throughput measured 863.04 Mb/s (107.88 MB/s) for reads and 1030.52 Mb/s (128.82 MB/s) for writes. This is compared to the 60 MB/s throughput achieved with the Gigabit network adapters. Although there is much room for improvement, the important result here is to show that a software-based
iSCSI implementation can exceed Gigabit throughputs and it is likely that with some tweaking, software iSCSI could far exceed Gigabit transfer rates.
CHAPTER 4

CONCLUSION

The necessity for disaster recovery infrastructure has grown by a tremendous degree. Unfortunately, the high cost involved with disaster recovery planning and network construction makes adoption all but impossible for most small businesses and home office environments. Traditionally, small and home office (Class 1 and 2) settings have either employed substandard disaster recovery and contingency planning solutions, or avoided them altogether. If backup and recovery mechanisms were put in place by these businesses, they generally involved periodic data backups (perhaps every week) and even less frequent tape shipments to off-site storage warehouses. While the nature of these businesses may allow for this, it could still take days to recover from unplanned downtime resulting in loss of revenue. However, decreasing the RTO and RPO for these organizations has always been too expensive to consider.

At the same time, large organizations and those businesses with considerable budgets (Class 3 and 4 environments) have invested heavily in expensive Fibre Channel hardware, resources, and trained personnel to enable fast data backup and recovery to remote locations. While these solutions perform exceptionally well, they are costly and any upgrades will no doubt be very expensive as well.

This paper has shown how iSCSI can be used as an alternative to any of these other methods. In the Class 1 and 2 settings, iSCSI can be implemented purely in software by using open source drivers that are freely available. The only costs to these organizations are associated with the network infrastructure and servers/workstations. In most cases, this infrastructure already exists in some sense and thus the cost is reduced. Using commodity Internet and some security (IPSec or SSL), these organizations can capture every write operation to a local disk and forward an encapsulated iSCSI packet to a remote location. There are many providers of data warehousing capabilities over the Internet but at a cost. The idea here is that home and small office entities can essentially
become much like Class 3 environments (class 4 requires a large amount of investment in remote boot, cluster failover, and redundant links). Thus, home and small offices can essentially save money, while decreasing their RPOs to near 0, and their RTOs to less than an hour.

The experiments performed at HCS-Tallahassee as well as research carried out at other institutions also shows that iSCSI is very promising as a disaster recovery medium for large organizations with high RTOs and RPOs. In order to maintain high recovery time and recovery point objectives, these organizations generally employed powerful but expensive Fibre Channel hardware. In these cases, performance cannot be sacrificed to save money. With Fibre Channel able to operate at 4 Gbps (and 10 Gbps in the future), it was the only solution that high performance environments could count on. However, iSCSI has been shown to be only bounded to the speed of the network in which it is used. iSCSI can function very well at Gigabit speeds and will be able to take advantage of the large throughput available once 10 GigE networks are available. In addition, iSCSI can already match the performance of Fibre Channel operating at 4 Gbps because of multiple connections per session (MC/S). Since MC/S allows a collection of separate physical links to be used as a single session, it is possible to provide link aggregation thereby matching the performance of 4 Gbps FC implementations.

iSCSI has already been deployed as a transport for disaster recovery and business continuity operations (see Section 2.9.2) over Metro Ethernet networks. Its performance characteristics have been well documented and a number of other strategies such as aggressive caching have been proposed in order to further enhance its effectiveness. Perhaps the biggest challenge iSCSI must overcome is that of fault tolerance. If it is to be accepted as a legitimate contender in the distributed SAN market, it will have to be tolerant to disruption and reliable over long distances. By adding some small degree of intelligence to the protocol (with the use of redirect PDUs and discovery sessions), iSCSI can be made fault tolerant and given highly desirable load balancing characteristics. Furthermore, it would be interesting to see how MC/S could be used to provide for fault tolerance and load balancing. The usual issues with latency and poor link utilizations for
high bandwidth-delay product networks can be mitigated by employing novel new TCP variants such as HS-TCP and Fast TCP. Further study is necessary to understand how iSCSI might perform over such networks and which TCP variants are suitable for certain environments. For 10 GbE settings, a number of new challenges are presented concerning memory and CPU requirements. A thorough investigation of these hardware requirements for 10 Gbps iSCSI must be undertaken to shed light on future solutions that will allow iSCSI to perform at increased throughputs.

The advantages to using iSCSI for disaster recovery are clear. iSCSI outperforms standard software tools and NFS by a significant margin because of the ability to provide for link aggregation using MC/S. Furthermore, the aggressive caching schemes employed by iSCSI allows performance to exceed that of NFS. Meanwhile, the high performance FC SANs such as Fibre Channel will have a great deal of competition with iSCSI. Both can perform equally well but iSCSI is so much more affordable that it has a remarkable potential to capture a good portion of the high performance SAN market. The affordability of iSCSI also allows more investment in redundant components and other features to improve availability during a disaster. Moreover, iSCSI is just as well suited for home and small business as it is for enterprise disaster recovery. Its flexibility in both hardware and software implementations allows it to be adopted in nearly every setting. Home and small businesses will be able to realize much lower RTOs and RPOs than thought possible. Since many small businesses are allowing for Internet transactions, iSCSI provides an affordable way to keep data safe and up-to-date. With its low cost, flexibility, high performance, and scalability, there is little disputing that iSCSI will be crucial in safeguarding information and increasing the availability of data well into the future.
APPENDIX

ACP – Association of Contingency Planners
AHS – Additional Header Segment
ANSI – American National Standards Institute
ASIC – Application Specific Integrated Circuit
BGP – Border Gateway Protocol
BHS – Basic Header Segment
CDB – Command Descriptor Block
CID – Connection ID
CmdSN – Command Sequence Number
CPU – Central Processing Unit
CRC – Cyclic Redundancy Check
CSG – Current Stage
CTRC – Cancer Therapy and Research Center
DR – Disaster Recovery
ExpCmdSN – Expected Command Sequence Number
ExpStatSN – Expected Status Sequence Number
FC – Fibre Channel
FCIP – Fibre Channel over IP
FCP – Fibre Channel Protocol
FFP – Full Feature Phase
FSB – Front Side Bus
FSPF – Fabric Shortest Path First
Gbps (Gb/s) – Gigabits per second
GigE – Gigabit Ethernet
HBA – Host Bus Adapter
HCS – High-Performance Computing and Simulation Research Laboratory
HS-TCP – High Speed Transmission Control Protocol
IETF – Internet Engineering Task Force
iFCP – Internet Fibre Channel Protocol
I/O – Input/Output

67
IOPS – I/O Operations per Second
IP – Internet Protocol
IPSec – Internet Protocol Security
iRAID – iSCSI RAID
iSAN – Internet Storage Area Network
iSCSI – Internet Small Computer System Interface
ISID – Initiator Session ID
iSNS – Internet Storage Name Service
ITT – Initiator Task Tag
kB - Kilobyte
LAN – Local Area Network
LONP – Login Operational Negotiation Phase
LU – Logic Unit
LUN – Logic Unit Number
MAN – Metropolitan Area Network
MaxCmdSN – Maximum Command Sequence Number
Mbps (Mb/s) – Megabits per second
MB/s – Megabytes per second
MC/S – Multiple Connections per Session
MTU – Maximum Transmission Unit
NFS – Network File System
NIC – Network Interface Card
NSG – Next Stage
OSPF – Open Shortest Path First
PCI – Peripheral Component Interconnect
PDU – Protocol Data Unit
P-iRAID – Parity iSCSI RAID
R2T – Ready to Transmit
RAID – Redundant Array of Independent (Inexpensive) Disks
RAM – Random Access Memory
RDMA – Remote Direct Memory Access
RIP – Routing Information Protocol
RPO – Recovery Point Objective
RTO – Recovery Time Objective
SAN – Storage Area Network
S-iRAID – Striped iSCSI RAID
SSID – Session Identifier
SSL – Secure Socket Layer
SNIA – Storage Networking Industry Association
SNP – Security Negotiation Phase
StatSN – Status Sequence Number
TCO – Total Cost of Ownership
TCP – Transmission Control Protocol
TOE – TCP Offload Engine
TPC-C – Transaction Processing Performance Council Benchmark C
TPC-H – Transaction Processing Performance Council Benchmark H
TPGT – Target Portal Group Tag
TSIH – Target Session Identification Handle
UNH – University of New Hampshire
WAN – Wide Area Network
REFERENCES


BIOGRAPHICAL SKETCH

Matt Murphy graduated with a B.S. degree in electrical engineering from Florida State University in April 2003. Since then, he has pursued a Masters Degree in Electrical Engineering while participating in research at the High-Performance Computing and Simulation Research Laboratory in Tallahassee. This research has been sponsored by the U.S. Department of Defense for topics including high-performance networking and computing, computational cryptanalysis, and mission assurance/business continuity operations. He is a member of the Institute of Electrical and Electronics Engineers (IEEE) as well as the IEEE Computer Society. In October 2002, he was also certified as an Engineer Intern by the Florida Board of Professional Engineers after successfully completing the FE examination. Mr. Murphy’s research interests also include computer networks, network security, and wireless security.