An Internet Based Data Storage Protocol for ATA devices

Kulveer Singh, Hyotaek Lim
Department of Computer Engineering
Graduate School of Design & IT
Dongseo University, Busan, 617-716, Korea
{kulveer.singh, htlim}@dit.dongseo.ac.kr

Abstract

This paper introduces a data storage transport protocol **internet Advanced Technology Attachment** ***(iATA)*** *for ATA storage devices that can be accessed over (TCP/IP) networks. As the access speed of Advanced Technology Attachment (ATA) type storage is increasing, the importance of sharing the ATA type storages over internet is also increasing. We are proposing a protocol in which we have implemented the basic ATA command set to access the ATA type remote storage devices over the TCP/IP networks .In this paper, we give basic details of the protocol and discuss the existing storage protocols which are mainly used for accessing block storage like internet Small Computer System Interface (iSCSI) Hyper Small Computer System Interface (HyperSCSI) and ATA over Ethernet (AoE).*

Keywords: SAN, Virtual Storage, Home Network, Network Storage.

1. Introduction

With the extensive expansion of internet in conjunction with the wide deployment of IP networks, they are becoming more stable and ubiquitous. Thus there has been growing interest in carrying storage traffic across TCP/IP infrastructure to exploit the ubiquity and cost. Currently, there exist several data transport protocols for storage devices in the market. AT Attachment (ATA) [1] is one of the most well-known storage technologies in the world. However, both ATA and newly developed Serial ATA [2] are designed primarily for inside-the-box storage. There has been lot of research done on SCSI based storage device to transport data on network and iSCSI is extensively used in SAN environment for transporting SCSI storages over TCP/IP networks. The ATA storage device has different segment of users who also need such protocol which can be used to access their personal data over internet without using any hardware or extra device and majority of personal computers , Small office Home office (SOHO) and small enterprises are using ATA type storages so we are proposing a protocol for transporting basic ATA commands over the TCP/IP networks.

The organization of the paper is as follows. In section 2, we briefly describe some of the **Related Work** in this field. In section 3, we give an **Overview** of the features of iATA. Section 4 describes in detail the iATA **Architecture**. In section 5, we describe the iATA protocol **Implementation**. Lastly we will give the **Conclusion** in section 6.

2. Related Work

The Internet SCSI (iSCSI) protocol [4], developed by IETF, defines a means to enable block storage accesses over TCP/IP networks. By carrying SCSI commands and storage data over IP networks, iSCSI is used to facilitate data transfers and storage management over long distances. Because of the ubiquity of TCP/IP networks, iSCSI can be used to transmit data over local area networks (LANs), wide area networks (WANs). HyperSCSI [6] is another transport protocol for SCSI that works on a network.

The protocol has multiple modes of operation. One of them works on a TCP/IP network similar to iSCSI. ATA over Ethernet (AoE) [6] is protocol which uses the basic command set of ATA type storages over the Ethernet. The protocol contracts client-server structure as in figure 1 where the clients are able to access server's shared disk via Ethernet. But iSCSI and HyperSCSI[5] protocols are specific for SCSI based storages and AoE is a protocol for accessing ATA type storages over Ethernet

3. Overview

ATA storage drives keep getting larger and cheaper. Even so, the more disk space there is, the more we use, and soon we run out again. Some kinds of data are huge by nature. Video, for example, always takes up a lot of space.

Businesses often need to store video data, especially with digital surveillance becoming more common. Even at home, we enjoy watching and making movies on our computers. There are protocols which are used for accessing SCSI [3] type storages over TCP/IP networks but there is no protocol to access ATA type data storages over the TCP/IP networks so our protocol is filling that gap for accessing the ATA type storage which is extensively used in personal computers, SOHO small enterprises. In this protocol iATA, it requires only the simple and ubiquitous TCP/IP network to operate. This enables low-cost centralization of storage without all of the usual expense. iATA has some advantages

1. iATA can be used for SOHO, small enterprises, LAN , MAN environments for providing virtual storage .

2. It can be used to increase the virtual storage over the mobile devices like PDA, mobile phone and the device which has TCP/IP protocol stack. These devices cannot support huge memory devices because of space, power consumption, size limitation.

4. Architecture

There are mainly three components in an ATA system: ATA devices, ATA controller, and the software. Software, which is a kernel-mode device driver of the operating system in most cases, communicates with the controller to make requests to devices. Then, the controller conducts the actual communication with the devices. However, it should be noted that ATA specifies the bus interface and the protocol only between ATA controller and ATA devices similarly, iATA architecture includes a kernel-mode device driver, a controller, and actual devices. As shown in the figure 2 the communication stack is shown in which there are two layers of ATAcommands and iATA over TCP/IP layer in protocol stack which is doing the encapsulation (at sender end) of commands and encapsulation of commands over the other end (receiver end) and passing this encapsulated packets to transport layer i.e. TCP and than to IP and Ethernet layer down to the protocol stack.

4.1 Packet Structure

iATA packet structure looks like ATA register set. Each request and reply packet is 80 bits in total. Only the reading reply and writing request packets will be followed by data streams, whose size will be given in the packet. You can see the structure of request packet in figure 3 and the structure of reply packet in figure. 4. Descriptions of fields in request and reply packets are

***Features*:** Parameters for the command

1. *Sector Count*: Number of sectors that will be used
2. *LBA*: Logical Block Address of the starting sector that will be used
3. *Device*: Device selection and additional bits are reserved for future usage.
4. *Command:* Command code
5. *Error*: Bits indicating different errors
6. *Status*: Bits indicating status of the action

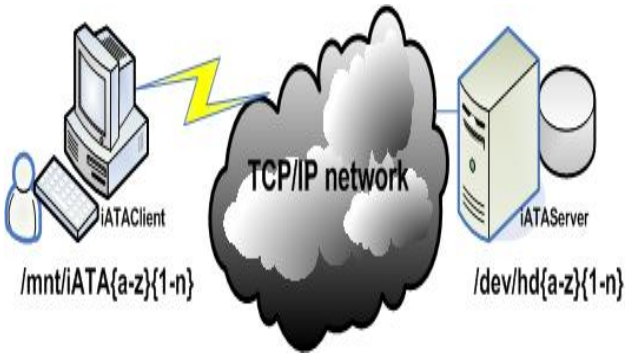

Fig 1 TCP/IP Network

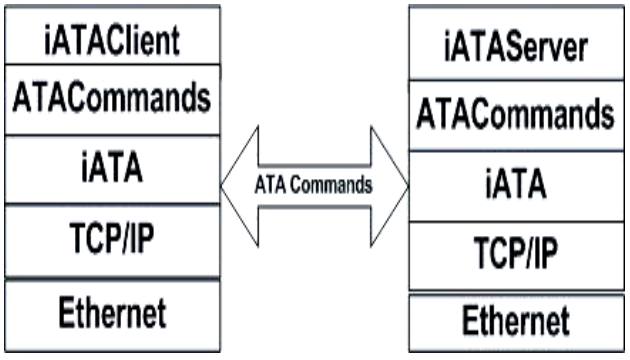

Fig. 2 Protocol Stack

Features, Error, and Status fields can be different for each command or not applicable for some commands.. Device field includes 4 additional bits reserved for future usage. And, LBA fields are 48 bits in total. Device field is used to identify the device selection . Command codes are given for each for command in section 4.2. Sector Count

field is the number of sectors that will be used in this command. For example, a read request must include the number of sectors that should be read in this field. The reply packet will be followed by a data stream composed of (Sector Count * 512) bytes.

4.2 Command Set

The command set is constructed with the help of ATAPI specification. ATAPI[1] specification includes mandatory command set for devices. The command set in iATA actually is a small subset of this command set and includes only the basic

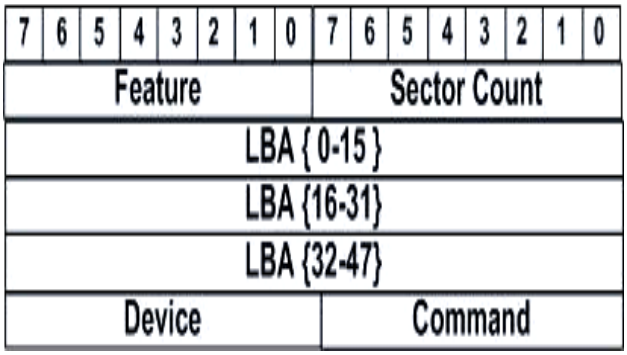

Fig. 3. Request Packet

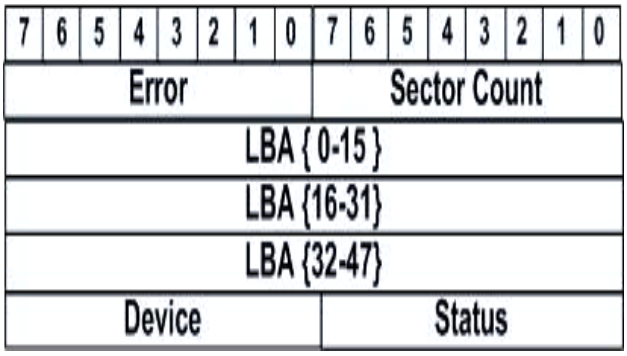

Fig. 4. Reply Packet

required functions for iATA protocol.

Control Commands:

FLUSH-CACHE: This command is used to force the device to flush its cache. If there is any data that needs to be written physically, the device should perform the writing of the data. This cache is nothing to do with the buffering mechanism of iATA protocol. The cache is device specific and thus any device that does not have an inner cache must reply to this command by saying that it is successfully completed.

IDENTIFY-DEVICE: This command is used to gather information from the device. Features field in the request packet is used to specify the type of information requested. If the information cannot be returned within the Status field of reply packet, a string for example, a data stream should be returned after the packet and the size of the data must be given in the Sector Count field of the reply packet.

SET-FEATURES: This command is used to modify device parameters. Features field in the request packet is used to specify the parameter that will be modified. Similarly, Sector Count and LBA fields are used to specify the value of that parameter.

Data Commands:

***READ-SECTORS*:** This command is used to read one or more sectors from the device. After the reply packet, the device must send the data sector by sector as a stream. Size of the data stream can be calculated from the Sector Count field.

***WRITE-SECTORS*:** This command is used to write one or more sectors to the device. After the request packet, the host must send the data sector by sector as a stream. Size of the data stream can be calculated from the Sector Count field.

5. Implementation

As we have implemented it on open source platform i.e. Linux Kernel 2.6 .Our implementation consists of two different modules: iATAClient, iATAServer. The iATAServer is a linux module that is responsibl e for most of the iATA functionality for exporting the ATA storage device over TCP/IP and iATAClient is linux module for importing the block storage from the iATAServer.The following sequence of communication takes places between iATAClient and iATAServer.

1) **Connection establishment:** The iATA connection set-up is a three-step handshaking procedure between a iATA client and server pair. Typically, in a SAN, the host machine (iATA client) is responsible for locating and initiating connections to storage devices (iATA servers).

2) Request for device information like size of device being exported and block size

3) **Data transmission:** When there is a ATA request from the local OS ATA upper layer of the host machine, the iATA client software is responsible for converting the OS-specific ATA command block together with any relevant data (as in a write command) into a platform independent iATA format sends to the ATA server. The iATA server receives the data stream, re-assembles the iATA command block and relevant user data, converts it back to an OS-specific ATA command block and passes it to the relevant hardware for execution. When the result of this ATA request is ready, the iATA server will send the result together with the requested data back to iATA client by issuing the reply packet stream in a similar manner as the request. The iATA client reassembles the iATA command block and converts it back to an OS-specific ATA command block before passing it on to the local OS ATA upper layer.

4) **Connection termination:** The iATA client initiates the connection termination by sending a closing command to the iATA server. The server will then close the connection and remove this client from its connection list

For this protocol, we do not need any type of new hardware to implement the system. As it needs only TCP/IP networks so it works on existing network. Performance is mainly dependent to the physical medium. It is impossible to retrieve disk blocks faster than the physical medium allows. We have not tested the performance of this iATA protocol yet.

6. Conclusion

In this paper we implemented the basic commands out of ATA command set which mainly deals with the basic read/write of data from ATA storage devices on Linux kernel 2.6. This protocol can be used for virtual storage in SAN and WAN. In future work we will measure the performance of this protocol and security requirements and implementation for this protocol.

References

[1] Technical Committee T13: AT Attachment - 7 with Packet Interface, Volume 1 Revision 4b. http://www.t13.org/ (2004).

[2] Serial ATA Working Group: Serial ATA: High Speed Serialized AT Attachment, Revision 1.0a. http://www.serialata.org/ (2003).

[3] Technical Committee T10: SCSI Architecture Model -3 (SAM-3), Revision 13. http://www.t10.org/ (2004).

[4] IP Storage Working Group: Internet Small Computer Systems Interface (iSCSI) (RFC 3720). Technical Committee T10: SCSI Architecture Model -3 (SAM-3), Revision 13.

[5] W. Y. H. Wang and T. C. Chong: An Ethernet Based Data Storage Protocol for Home Network An Ethernet Based Data Storage Protocol for Home Network

[6] http://en.wikipedia.org/wiki/ATA-over- Ethernet.