모바일 기기를 위한 가상 스토리지 프로토콜(iATA)의 설계 및 파라메터 최적화

양즈민* · 임효택**

Design and Parameter Optimization of Virtual Storage Protocol (iATA) for Mobile Devices

Chee-Min Yeoh* · Hyotaek Lim**

요 약

최근들어 다양한 인터넷 서비스들이 개인 컴퓨터뿐만 아니라 무선 네트워크 환경에서의 모바일 기기에서도 자주 사용되고 있으며 서비스 컨텐츠의 양이 점점 증가함에 따라 모바일 기기의 스토리지 한계가 큰 문제가 되고 있다. 본 논문에서는 이러한 문제를 해결하기 위해 iATA라는 새로운 블록-레벨 스토리지 네트워크 프로토콜을 제안하며 iATA 프로토콜의 개념, 설계 및 고려사항들을 포함한다. iATA는 TCP/IP 네트워크상에서 ATA 블록-레벨 데이터와 명령어를 전달하게 되며 이를 통해 모바일 기기는 원격의 ATA 저장장치를 마치 모바일 기기 자신의 스토리지로 이용할 수 있게 된다. 벤치마킹 실험과 시험을 통해 제안된 iATA 프로토콜은 기존의 제안한 스토리지를 가지고 있는 모바일 기기상에서 효율적이며 경량의 가상 스토리지 프로토콜로서 사용될 수 있음을 보여주고 있다.

ABSTRACT

Nowadays, numerous of valuable internet services are available not only for personal computer but also for mobile appliances in wireless network environment. Therefore, as the amount of contents is increased for those services, the storage limitation on mobile devices has became a significant issue. In this paper, we present a new block-level storage network protocol, iATA (Internet Advanced Technology Attachment) as a solution to the above problem. iATA is designed to transport ATA block-level data and command over the ubiquitous TCP/IP network. With iATA, a mobile appliance is able to access and control the ATA storage devices natively through network from anywhere and at anytime as if the storage devices is attached locally. We describe the concepts, design and diverse consideration of iATA protocol. Based on the benchmark experiments and application exploitation, we strongly believe that iATA as a light-weight protocol is efficient and cost-effective to be used as a storage network protocol on a resource limited device that utilizes common-off-the-shelf storage hardware and existing IP infrastructure.

키워드

Virtual Storage, iATA, Mobile Computing, Ubiquitous

* 동서대학교 디자인&IT전문대학원 석사과정
** 동서대학교 컴퓨터정보공학부 부교수(교신저자)

접수일자 2008. 11. 28

Ⅰ. Introduction

With the rapid development of technology in wireless communication, mobile computing and ubiquitous computing, mobile appliances including cell phones, Personal Data Assistants (PDAs) and smart phone are now able to access digital information from anywhere and at anytime [1-8]. Due to their convenience and ubiquity, it is widely accepted that such mobile appliances has become people's best companion in their daily life. There are many efforts of deploying conventional wired network environment services such as video streaming, database, commerce and so on to mobile appliances in wireless network environment [9-12]. As the amount of contents is increased for the services, there will be a growing demand for larger storage. Due to mobility, mobile devices should be small and only few Gigabytes of flash memory will be equipped to store the data. However, it is still difficult to store multimedia data such mp3, mpeg, etc and install large software such as database engines in mobile device [13]. As a result, the limited storage capacity of mobile device will be obstacle for the adaptation of usable services of wired environment to mobile environment.

Up to now, there are several protocols that are being developed to transport block-level access protocol over network. Internet Small Computer System Interface (iSCSI) is a protocol that developed by Internet Engineering Task Force (IETF). It enables SCSI devices to be accessed over an existing IP infrastructure [14]. Unlike iSCSI that providing storage device access over TCP/IP network, there are two network layer protocols that offering device access via Ethernet connection such as ATA over Ethernet (AoE) and HyperSCSI for ATA and SCSI device respectively [15][16]. A number of papers have proposed to deploy some of those technologies for mobile appliance to counter the storage limitation issue [17-21].

To date, ATA hard disk is still the most popular storage device in SoHo (Small Office and Home Office) and home users environment in term of devices shipped. This is because of its economical price and simplicity. ATA storage device can be found effortlessly in any home personal computer. Due to the cost-effective and ubiquity of ATA storage standard, we are motivated to fill the gap between the routable iSCSI protocol and non-routed AoE protocol by designing a new block-level storage network protocol called iATA (Internet Advanced Technology Attachment). iATA is aimed to transport ATA commands over existing TCP/IP network for mobile devices. iATA enables user to access and control ATA device natively over the network by encapsulating ATA standard storage interface protocol[22].

The remainder of this paper is organized as follows. In the next section, we present the main iATA protocol architecture and operations. Section III describes a benchmark experiment where several tests were executed for iATA parameter optimization in wireless environment and followed by performance result analysis. Then, application of iATA is demonstrated in Section IV. Finally, Section V concludes the paper.

Ⅱ. iATA Protocol Description

Internet Advanced Technology Attachment (iATA) is a new block-level network storage protocol which is designed to transport Advanced Technology Attachment (ATA) commands, status and data over the existing TCP/IP network. In our design, the protocol consideration of both iSCSI and AoE in [23][24] were used as our reference model.

2.1. iATA Storage Model

Viewing from TCP/IP layering model, iATA protocol is positioned on the application layer that lies on top of transport layer as shown in Fig. 1. This layer is responsible for translation from ATA command to iATA request and mapping the iATA responses from server to the corresponding ATA requests.

In the design of iATA protocol, client-server communication architecture is taking into consideration. iATA server exports its ATA devices by making them a vailable over TCP/IP network. While iATA clients are able to use those exported ATA devices by sending request commands to the server. With the deployment of this architecture, iATA server is able to serve multiple client connections simultaneously. In the case where iATA server exporting several ATA devices over the network, each exported device can be accessed by different iATA clients concurrently. As a result, this model established a many-to-many relationship between server entity and client entities that greatly improved the flexibility of this storage model.

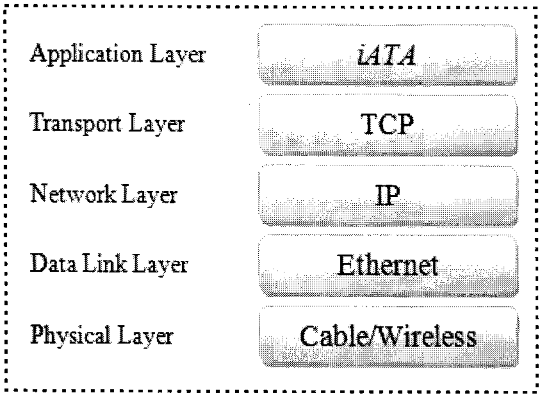


그림 1. iATA 프로토콜 스택
Fig. 1. iATA's Protocol Stack

2.2. iATA Protocol Data Unit (PDU) Format

The communication between iATA server and iATA client involved the exchange of several pieces of message. A single piece of iATA message is referred as a Protocol Data Unit (PDU). In order to enable the iATA PDU to be routed over TCP/IP network, the entire iATA PDU needs to be encapsulated by the TCP header, IP header and Ethernet header as illustrated in Fig. 2. iATA PDU is consists of Common Header Segment (CHS) and follows by either configuration message or ATA command block.

Ethernet	IP	TCP	iATA PDU

그림 2. iATA PDU 캡슐화
Fig. 2. iATA PDU Encapsulation

The first fragment of all iATA PDUs is always a 16-byte Common Header Segment (CHS) which has the template format shown in Fig. 3. The main purpose of this header is to carry the payload length of the subsequent variable-size payload. Operation Type (Op Type) field is the first byte in this header. This is used to indicate the type of the operation message it carries. There are two bitwise flags defined in flags field which are response flag and error flag. Response flag is set to value of 1 if the message is response from server, otherwise 0. If the associating command message generated an error, the error flag will be set to value of 1 and the 1-byte error field will contain the error code in the response message. The device is a 16-bit integer indicating the device number of the device that has been exported by the iATA server. Session identity and task tag are used for session management and ATA request-response mapping.

Op Type	Reserved	Flags	Error
Session ID		Device ID	
Payload Length			
Task Tag			

그림 3. 공통 헤더 세그먼트
Fig. 3. Common Header Segment

There are two special types of message that will be attached accordingly after CHS which are Configuration/ Query message and ATA Command Block. Fig. 4 depicts the format of 8-byte configuration/query message.

Command / Status	String Length
Configuration / Query String Pairs	

그림 4. 구성/질의 구조
Fig. 4. Configuration/Query structure

This message is used to exchange out-of-band iATA I/O message likes configuration parameters, system details and so on. Besides that, the most important message during the iATA I/O operation is ATA command block. As shown in Fig. 5, this message consists of 48-bit logical block address (LBA), ATA command and data that to be read or write on the disk. These command blocks are similar to in-band ATA I/O messages and can be used directly in the ATA device.

Flags	Error /Feature	Sector Count	Command /Status
LBA0	LBA1	LBA2	LBA3
LBA4	LBA5	Reserved	
Data			

그림 5. ATA 명령어 블록
Fig. 5. ATA Command Block

2.3. iATA Operation Phases

In one complete iATA session, iATA client needs to go through four operation phases which includes login phase, discovery phase, full feature phase and termination phase as illustrated in Fig. 6.

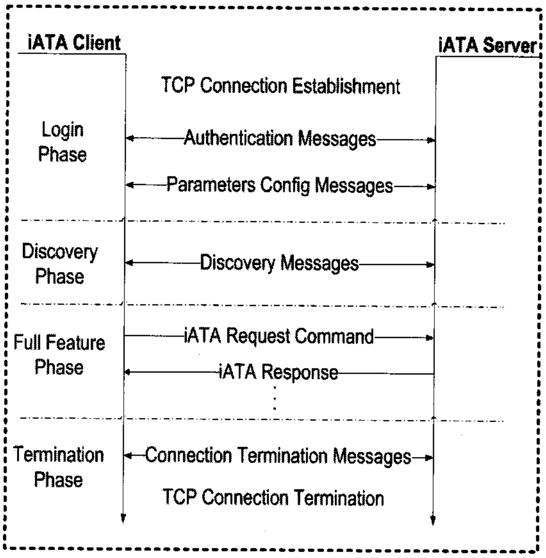


그림 6. iATA 프로토콜 동작단계
Fig. 6. Operation phases in iATA protocol

After the establishment of TCP connection, iATA client will enter into the first phrase which is the login phase. In this phase, client will authenticate with iATA server. There are several authentication protocols that can be usedfor client-server authentication in this phase such as Secure Remote Password (SRP), digital certificate, Access Control List (ACL) and so on. These protocols could provide certain degree of security depending on the client requirement. Following the authentication, the parameters of the session are negotiated between client and server with session's parameters configuration request and response messages.

Subsequently, iATA client sends device discovery message to iATA server in discovery phase. The server replies client with device discovery response which is enclosed with the geometry information of the exported ATA device at the server side.

During the full feature phase, if there is an ATA request from local OS ATA upper layer of iATA client, the iATA client driver will encapsulate the ATA command block together with any relevant data into iATA packets and send to iATA server. Then, the iATA server receives the data stream, re-assembles the iATA command blocks, de-capsulate it back to ATA command block and passes it to relevant hardware for execution. When the result of this ATA request is ready, the iATA server will encapsulate the result together with requested data and send back to iATA client. Next, the iATA client will reassemble and de-capsulate the data stream and pass it to the local OS ATA upper layer. This cycle is continuing until there is no request from the ATA upper layer.

In termination phase, the iATA client ends the session by issuing exit command to the iATA server. The server clears the session records of client and removes client from connection list. Lastly, iATA client could safely terminate the TCP connection.

2.4 Design of iATA Client

The iATA client is implemented as an I/O device driver in Windows CE. There are two main modules in iATA client system design which are modified ATA driver module

(ATA.dll) and iATA driver module (iATA.dll) as illustrated in Figure 7. Both modules are designed to execute for different important tasks.

Instead of letting ATA device driver passes the I/O commands to the hardware interface directly, the modified ATA driver module redirect all ATA I/O request to the iATAdriver. This module is responsible to transform I/O request into iATA PDUs and send them to the iATA server over TCP/IP network using WinCE built-in Winsock module.

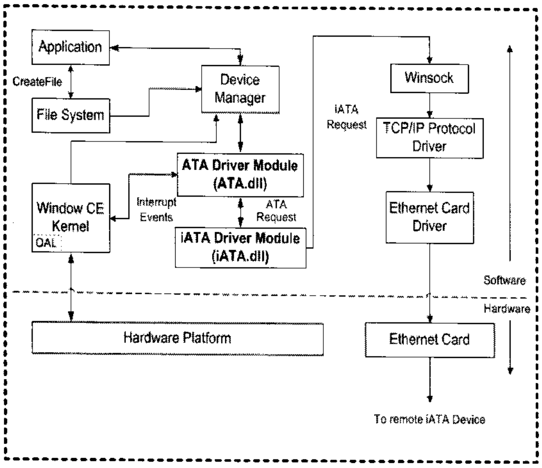


그림 7. iATA 클라이언트 드라이버 구조
Fig. 7. iATA client driver architecture

Once the requested command has been completely executed at the server, the iATA client module will receive the iATA response message which contains the status and the result of command execution from the server. The iATA response message is then decapsulated and the corresponding ATA response is passed to the local OS ATA upper layer.

2.5. Design of iATA Server

The iATA server component is an entity that listens and accepts client's connections, process the received iATA requests and give responses to the clients after the commands have been executed. It is implemented on a Linux system.

Referring to Figure 8, the iATA server component acts as a middle man between the network interface and storage devices. This component is liable to extract the ATA request from the network packets and redirect them to the storage devices using both network I/O and device I/O facilities provided by Linux system calls. The connection and session module handles each client session independently and maintains the state of each connection. The decapsulated ATA commands and hard disk data are passed to the appropriate ATA device by device management module. Each device can only handle a single request at a time. Thus, locking mechanism is applied in this module to prevent concurrency access of a single device.

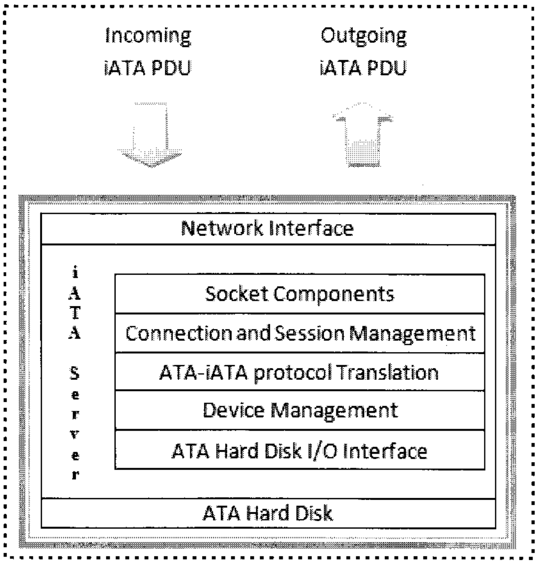


그림 8. iATA 서버 프로그램 구조
Fig. 8. Structure of iATA server program.

Ⅲ. Performance Evaluation

As a storage network protocol, iATA provides simplicity and efficiency in transferring data. Based on our design, we implemented iATA protocol over IEEE 802.11b WLAN to verify the performance of iATA protocol over wireless environment. As such, we executed several tests on iATA parameter optimization and benchmarks in order to obtain the best performance. The results so far prove to be most encouraging.

3.1. iATA Parameter

Simplicity is one of the beautiful features in iATA protocol. Unlike other storage network protocols like ISCSI, there is only one parameter need to be configured for performance optimization in iATA protocol which is Number of Sectors per Command (NSPC). NSPC affects the performance of iATA read and write operations. Most of the ATA disks define a total of 512 bytes as a sector size and require multiple of sectors in one I/O operation. Thus, iATA client is required to declare a limit on the number of sectors in a single ATA I/O operation. This parameter limits the request size of a single iATA read command and limits the number of bytes of data to be written in a single iATA write command. In general, if increase in NSPC means fewer commands are required to transfer a fixed amount of data.

3.2.Experiment environment setup

In our experiment, there are four main hardware devices that are needed such as router, wireless access point, Personal Data Assistant (PDA) and Personal Computer (PC).

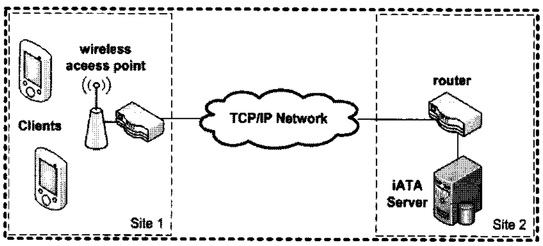


그림 9. 환경설정
Fig. 9. Environment setup

Referring to Fig. 9, this experiment spans two sites connected through the TCP/IP network. Site 1 is the iATA client site which consists of 802.11b wireless access point, router and a PDA. The PDA has 300MHz S3C2440 Samsung processor with 64 MB of RAM and equipped with wireless capability. This PDA runs Microsoft Mobile Pocket PC 2003 powered by Windows CE 4.21 and is installed with iATA client driver. On the other site is iATA server site which contains router and a 1GHz Intel Pentium III Coppermine PC with 512MB RAM.This machine runs Fedora Core 6 Linux Kernel 2.6.18 and installed with iATA server driver which exported 2GB of ATA disk over the network. The utilization of two different operating system platforms between iATA server and client demonstrated that iATA protocol is platform independent and works well in two non-identical OS platforms.

3.3. Experiment methodology

The performance metrics considered during our studies was average throughput which is the total number of application-level bytes carried over an iATA connection divided by the total elapsed time taken by the application and expressed in Megabytes per second (MB/sec). Total elapsed time is calculated as the time interval from the initial time when the benchmark tool generating the first byte of data to the time when the last byte of data was confirmed to haven been sent/received. Thus, the elapsed time includes all data transfers and all read and write command as well as responses at all levels of the protocol stack.

In our experiment, we employed the awards winning Windows Mobile maintenance utility, Pocket Mechanic Professional version 2.90 as our benchmark tool. Pocket Mechanic Professional provides a storage card benchmark module which comprised of comprehensive set of storage card benchmarks to evaluate the file system performance. For read operation, the tool generates and reads from a set of test files with various file size from 4KB to 2MB. While for write operation, the tool creates and writes a set of different size test files to the server. For each different size of test files, a total of 10 files were created and average throughput was obtained as data point plotted in a graph in this paper. Besides that, in order to observe the effect of

I/O operations without the use of buffer cache, the tool uses specific algorithm to read/write from different clusters and therefore not allowing operating system to buffer the I/O operations. In our experiment, NSPC was increased from 8 sectors to 4096 sectors which in turn mean increases in expected command size from 4KB to 2048KB.

3.4. Experiment results

3.4.1. iATA read experiment results

From Figure 10, there is not much improvement on throughput from 8 of sectors per command until 128 of sectors per command. A tremendous growth can be seen when the number sectors per command is greater than 256 for both buffered ans un-buffered operation. Beside that, we could clearly see that the performance of buffered read operation beyond 256 of sectors per command is outperformed if compared with un-buffered file system read. In term of un-buffered read operation, there is a slightly decrement in throughput when NSPC is beyond 2048.

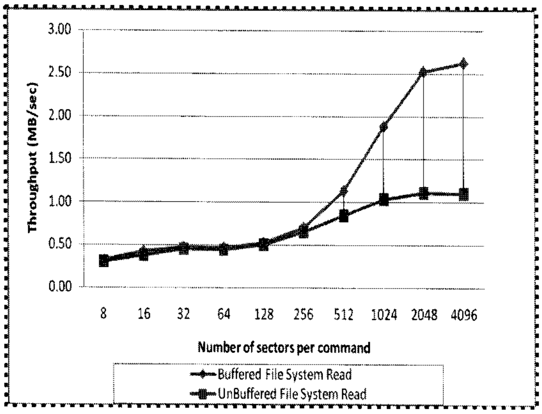


그림 10. 버퍼 및 비-버퍼 read 동작에서의 NSPC효과

Fig. 10. Effect of NSPC on buffered and un-buffered read operation

In a nut shell, the results illustrate that the throughput does not increase when the NSPC is greater than 2048. Thus, we suggest that 2048 of NSPC should be used in the iATA read operation in order to obtain an optimum performance.

3.4.2. iATA write experiment results

From Figure 11, a significant increment is noticeable when NSPC beyond 16 as benchmark tool effectively bypassed the buffering. In general, as NSPC increases, the throughput of write operation is increase as shown in Figure 13. Nevertheless, the throughput of write operation slips slightly when NSPC reaches 2048.

In short, the results show that there is a decrease in throughput when NSPC beyond 2048. Thus, we suggest that the iATA client should declare a 2048 limit on the NSPC in a single iATA write operation.

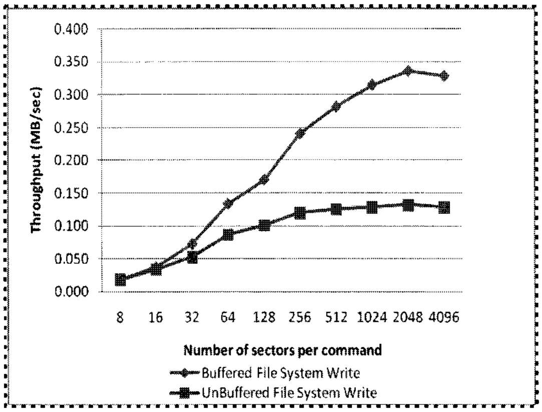


그림 11. 버퍼 및 비-버퍼 write 동작에서의 NSPC효과

Fig. 11. Effect of NSPC on buffered and un-buffered write operation

IV. Application Exploitation

In order to provide a user-friendliness interface for end user when using iATA storage system, we developed the iATADisk Explorer which incorporated iATA protocol into a file explorer. Figure 12 demonstrates iATADisk Explorer is used to browse the content of ATA disk. There is no physical connection between ATA disk and PDA. The connection is carried out by the wireless connection and the existing TCP/IP network. With a single click, PDA is able to connect to iATA server and mount the exported ATA disk effortlessly from the iATADisk menu. In addition, several important parameters for iATA connection such as server IP address and so on can also be configured through iATADisk wizard.

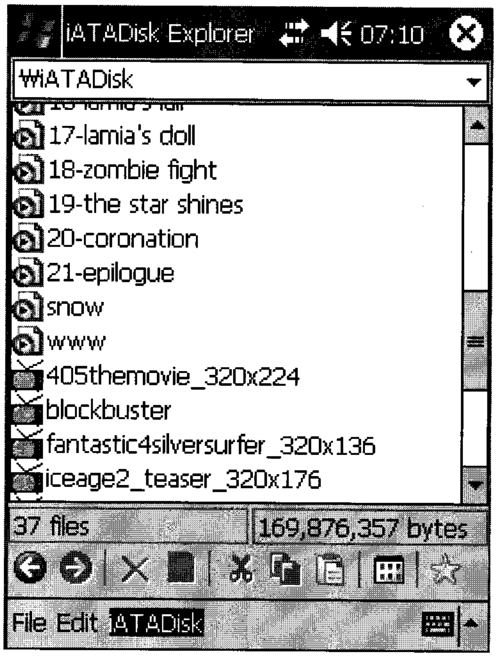


Fig. 12. iATADisk Explorer
그림 12. iATADisk 익스플로러

V. Conclusions

In this paper, we have presented a new network storage protocol, Internet Advanced Technology Attachment (iATA) which can be used to access ATA devices over the TCP/IP network. It is aimed to utilize common-off-the-shelf storage hardware and well-deployed network structure yet provide users with simplicity and flexibility. In our experiments, we discovered the optimal parameter setting for iATA read and write operations and proposed as the default value in iATA. Based on the benchmark experiments and application exploitation, we strongly believe that iATA as a light-weight protocol is efficient and cost-effective to be used as a storage network protocol on a resource limited device.

References

[1] M. Weiser, "The Computer for the 21st Century," Sci. Amer., Sept., 1991.

[2] G.D. Abowd, E.D. Mynatt, "Charting Past, Present, and Future Research in Ubiquitous Computing". ACM Transactions on Computer-Human Interaction, 2000.

[3] James Weatherall and Alan Jones, "Ubiquitous Networks and their Applications", In IEEE Wireless Communications 9(1) (2002) 18 - 29.

[4] Debashis Saha, Amitava Mukherjee, "Pervasive Computing: A Paradigm for the 21st Century," Computer, vol.no.pp. 25-31,Mar.,2003

[5] Satyanarayanan, M. 1996. Fundamental challenges in mobile computing. In Proceedings of the Fifteenth Annual ACM Symposium on Principles of Distributed Computing(Philadelphia, Pennsylvania, United States, May 23 - 26, 1996).

[6] Flinn, J. and Satyanarayanan, M. 1999. Energy-aware adaptation for mobile applications. In Proceedings of the Seventeenth ACM Symposium on Operating Systems Principles(Charleston, South Carolina, United States, December 12 - 15, 1999). SOSP '99. ACM, New York, NY, 48-63.

[7] Alicherry, M., Bhatia, R., and Li, L. 2005. Joint channel assignment and routing for throughput optimization in multi-radio wireless mesh networks. In Proceedings of the 11th Annual international Conference on Mobile Computing and Networking (Cologne, Germany, August 28 - September 02, 2005). MobiCom '05. ACM, New York, NY, 58-72.

[8] Varshney, U. and Vetter, R. 2000. Emerging mobile and wireless networks. Commun. ACM 43, 6 (Jun. 2000), 73-81.

[9] Rafael Andrade, Aldo von Wangenheim, Mariana Kessler Bortoluzzi, Wireless and PDA: a novel strategy to access DICOM-compliant medical data on mobile devices, International Journal of Medical Informatics Volume 71, Issues 2-3, , September 2003, Pages 157-163.

[10] Jia-Lang Seng, Sally Lin, A mobility and knowledge-centric e-learning application design method, International Journal of Innovation and Learning 2004 - Vol. 1, No.3 - 311

[11] Varshney, U. & Vetter, R. (2002) Mobile Commerce: Framework, Applications and Networking Support. Mobile Networks and Applications. Vol. 7, no. 3, 185 - 198

[12] Goto, K. and Kambayashi, Y. 2003. Integration of electronic tickets and personal guide system for public transport using mobile terminals. In Proceedings of the 2003 ACM SIGMOD international Conference on Management of Data (San Diego, California, June 09 - 12, 2003). SIGMOD '03. ACM, New York, NY, 642-646.

[13] S.J.Shepard: Embedded Databases Can Power Emerging World of Information to Go. IEEE Distributed Systems Online.

[14] K. Z. Meth and J. Satran. Features of the iscsi protocol .IEEE Communications Magazine, 41(8):232-249, August 2003.

[15] Ed L. Cashin, "ATA over ethernet: Putting Hard Drives on the LAN Linux Journal, Specialized Systems Consultants, Inc., June 2005.

[16] Wang, W.Y.H., Yeo, H.N., Zhu, Y.L., Chong, T.C., "Design and development of Ethernet-based storage area network protocol", Proceedings. 12th IEEE International Conference on Networks, IEEE, Singapore, November 2004, pp. 48-52.

[17] D. Kim, M. Ok, M.-s. Park. An Intermediate Target for Quick-Relay of Remote Storage to Mobile Devices. Proc. of ICCSA, May (2005).

[18] Sura Park, Bo-Suk Moon, Myong-Soon Park: Design, Implement and Performance Analysis of the Remote Storage System in Mobile Environment, Proc. ICITA 2004

[19] M.Ok, D.Kim, M-s. Park, UbiqStor: A Remote Storage Service for Mobile Devices, The Fifth International Conference on Parallel and Distributed Computing Application and Technology (PDCAT04) Singapore, December (2004).

[20] Hyotaek Lim, Saebom Choi, "Design and implementation of iSCSI-based virtual storage system for mobile health care", HEALTHCOM 2005, Jun. 2005, pp37-42

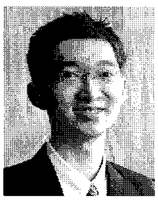
[21] Wang, W.Y.H., Tow Chong Chong, An Ethernet based data storage protocol for home network, IEEE Transactions on Consumer Electronics, IEEE, May 2004, pp. 543-551.

[22] Yu-Shu They, Chee-Min Yeoh, Hoon-Jae Lee, Hyotaek Lim, "Design and Implementation of ATA-Based Virtual Storage System for Mobile Device," mue, pp. 490-495, 2008 International Conference on Multimedia and Ubiquitous Engineering (mue 2008), 2008

[23] Brantley Coile and Sam Hopkins, The ATA over Ethernet Protocol. Coraid Inc,2005

[24] Joglekar, A., Kounavis, M. E., and Berry, F. L. 2005. A scalable and high performance software iSCSI implementation. In Proceedings of the 4th Conference on USENIX Conference on File and Storage Technologies - Volume 4 (San Francisco, CA, December 13 - 16, 2005).

저자소개



양 즈 민 (Yeoh Chee Min)

2002년 Multimedia University (이학사)

2007년~현재 동서대학교 디자인& IT 전문대학원 석사과정

※관심분야:IP Storage Network, Wireless Sensor Network, IPv6, Mobile Application

임 효 택 (Hyo Taek Lim)

1988년 홍익대학교 전자계산학과 졸업(이학사)
1992년 포항공과 대학원 전자계산학과 졸업(공학석사)
1997년 연세 대학교 컴퓨터과학과 졸업(공학박사)
1988년~1994년 한국전자통신연구소 연구원
2000년~2002년 Univ. of Minnesota(미) 컴퓨터공학과 연구교수
1994년~현재 동서대학교 컴퓨터공학과 부교수
※관심분야 : Computer Network, Protocol Engineering, Storage Networking, IPv6, Mobile Application