1998 International Computer Symposium
Workshop on Software Engineering and Database Systems
December 17-19, 1998, N.C.K.U., Tainan, Taiwan, R.O.C.

Server-based Maintaining Software Integrity for Computer Classroom Workstations

Chiung-San Lee
Computer Center, the Dept. of General Education
National Taipei College of Nursing, Taipei, Taiwan, R.O.C.
Email: csl@ntcn.ntcn.edu.tw

ABSTRACT

This paper presents a server-based approach to maintain all workstation's software system in a computer classroom. The approach is applicable to the FAT (file allocation table) and NTFS file systems, which are widely installed in computer classrooms. Moreover, the maintenance programs can automatically modify computer name and internet protocol (IP) address to an appointed one for each workstation. According to our measurement, maintaining an abnormal workstation (being able to run whole Windows NT Workstation system and applications about 613 Mbytes) will take about 21 minutes without locking/unlocking computer shell.

1. INTRODUCTION

Computer classrooms have been prevalently established in many education organizations for promoting teaching efficiency. These classrooms play an important role for teachers to conduct courses and for students to do homework. In the other words, a computer classroom should contribute to more efficient teaching and learning [1], [2], [3]. Truly, computer classroom can enable active "learning by doing" to supplement the more passive "learning by seeing and hearing", nearly doubling the learning effectiveness [1]. To provide a well teaching and learning, a computer classroom is need to meet the following several requirements. (1) All workstations must be workable and normal. (2) Desktops of all workstations should be identical. (3) It must be easy and fast for adding/removing software to/from all workstations. (4) Each workstation has to be assigned a unique computer name and internet protocol (IP) address.

In general, a computer classroom consists of several workstations for students and one teacher's console with a workstation and a projector. The projector is an apparatus for projecting the desktop of teacher's workstation by rays of light on to a screen; and thus students can closely follow the teacher's instructions to operate their workstations. Typically, there are 50 or more workstations in a computer classroom with each running the Windows 95, 98, or NT systems and providing an enriched and friendly environment for obtaining effective learning by doing.

The Windows operating system is now widespread because of its powerful features [4], [5], [10]. These features include object-oriented concept, user-centered interface, real multitasking, long filename, plug and play (automatic detecting hardware and setting its driver), supporting network, and auto-run. The object-oriented design provides a natural and intuitive way for users to interact with information on a workstation. The user-centered design supports little conscious thought from the user. The long filename feature will enable users to name their files with meaningful names of up to 255 characters rather than the old 8.3 file naming convention that has been used for over a decade. The plug and play automatically changes how adapter boards and devices are configured; rather than requiring the user to do configuration settings on the new hardware. The Windows network architecture enables applications to work with several different network types, and it includes a set of interfaces designed to enable the coexistence of several network components. On the whole, teachers and students find that these features compel enough to encourage them to install the Windows systems, such as 95, NT, or 98, on computer classroom workstations.

This paper presents a server-based maintenance approach to keeping the software integrity of the Windows system and applications for all workstations in a computer classroom. This approach takes several advantages. (1) The system manager does not need to unlock/lock computer shell. (2) Computer viruses will be thoroughly cleaned. (3) The system manager will take a very short time to add new software or remove old one. (4) All workstations can be automatically set an appointed computer name and IP address by the server-based recovery program. (5) Both FAT (File Allocation Table) and NTFS file systems can be easily maintained. The basic concept of the proposed approach is to install a workstation with the whole operating system and applications and to make one image copy of the workstation's hard disk onto the network server [11]. While a something wrong occurred to a workstation, the classroom manager can only perform the server-based recovery operation to repair the abnormal workstation's software. In the case of recovering 613 Mbytes data (including the whole Windows NT system and some applications, such as Office 97, Visual Basic), it only takes about 21 minutes.

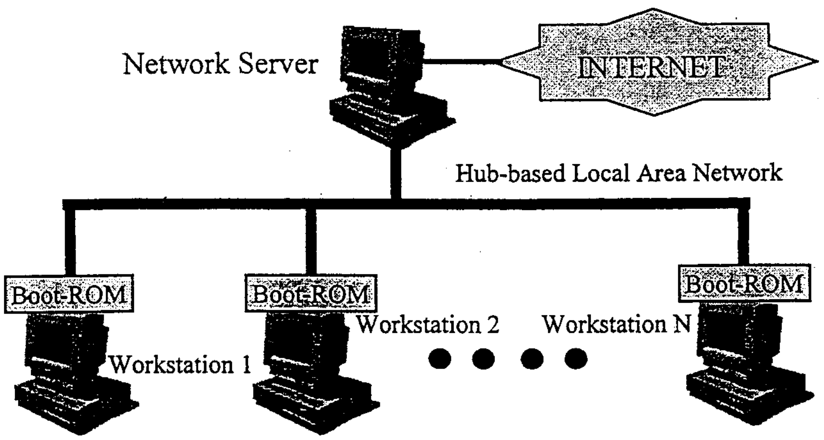

Fig. 1: The Client/server architecture for computer classroom workstations

2. SERVER-BASED MAINTENANCE APPROACH

The server-based maintenance approach is based on client-server architecture, which is a popular network structure used in a computer classroom [6], [7], [8]. A client (i.e., workstation) is a consumer that requests information, and a server is a provider that provides information.

2.1 Structuring workstations in client/server architecture

In client/server architecture as shown in Fig. 1, a network adapter connects the communication channel with a host station to send or receive data [7]. In a sending case, the host combines the data with transport and network protocols to form a packet. Then the data link layer copies the packet into a network adapter and accesses the communication channel to transfer the packet. In a receiving case, the process follows the inverse protocols. From the host's viewpoint, the network adapter has a buffer; outgoing packets will be written into the buffer and be transmitted by the adapter hardware; the incoming packets have to be copied from the buffer to the host [8]. Furthermore, the network adapter may be equipped with a boot-ROM chip for communication connection. When a workstation's hard disk is fail to boot, the workstation may connect to server via the boot-ROM for maintaining its software. Therefore, the server-based maintenance approach is developed under the client/server architecture.

We consider a client/server system, consisting of several clients and one server interconnected by hub-based local area network. Hub is a connectivity component that provides a common connection among computer hosts in a star-configured network. The hub-based local area network is an abstract representation of intermediate and header hubs. The header hub performs two major functions: to signal regeneration/repeating and collision detection/notification. When a host transmits, the header hub repeats the signal and broadcasts it to all hosts. When more than one host transmits, the hub detects a collision and then generates a collision presence signal to broadcast. An intermediate hub has an upstream and a downstream signal process unit and is connected to each subordinate host or hub by one upstream and downstream line. In other words, we can connect all workstations to a local area network by employing the hub-based structure.

The network server shown in Fig. 1 plays the information provider and the internet router roles. When a workstation's software system is abnormal, the classroom manager may issue a command access the maintenance data files that were stored on the server for repairing the abnormal workstation. Moreover, teachers may distribute their teaching material via the server to all students. In the router role, the server has to provide internet protocol (IP) switching that passes IP packets from one network to another.

2.2 Maintaining concept

In general, all workstations in a computer classroom are equipped with identical hardware devices and software systems for teachers conducting courses and students doing homework. To reduce installation process and setup time for all workstations, we may install an entire set of software systems on a workstation hard disk. Then we use the hard disk's image as a source to install the remaining workstations. In other words, we take three major steps to setup all workstations with no the classroom manager input required, as the following:

I. To deploy operating system and applications to a workstation, namely ***normal workstation***.
II. To make one copy of the normal workstation's

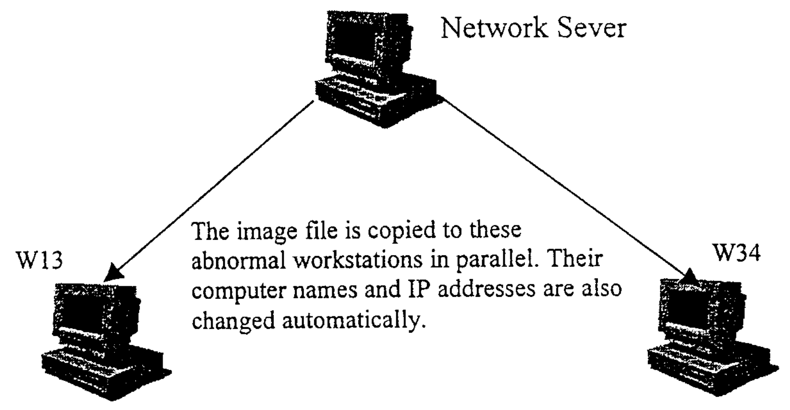

Fig. 2: One step to repair the abnormal workstations W13 and W34

hard disk as an image file that will be stored on the network server, (i.e., backup).

III. To copy the image file from server to remaining workstations and change computer name and IP address simultaneously, (i.e., recovery).

After the step III is complete, all workstations can be directly opened to use and each workstation holds a unique computer name and an IP address without any more settings. However, after we open a computer classroom for a long time, some workstations may be unable to work normally or will run into some hitches. For examples, some files may be deleted by students or inflected by viruses. Furthermore, configurations are possibly changed so that hardware is unable to work. Therefore, the repairs are needed before we can reuse these workstations. In this case, we can restore the hard disk's content of these abnormal workstations to integrity by performing server-based recovery copy according to the image file, as shown in Fig. 2. The recovery copy will repeatedly write a cylinder size per time to the abnormal workstation based on the image file. When the recovery copy operation is complete, these workstations will also hold the whole software system.

Each workstation in a classroom has to be assigned a unique computer name and IP address. We apply the Boyer-Moore algorithm [9] to do string matching for finding the default computer name and IP address, while the recovery copy is proceeding. If the default computer name or default IP address in the image file is found, then the image file with the assigned name and IP address is written into the target hard disk, as shown in Fig. 2. For example, the W13 and W34 workstations with defective software system will be recovered by using the image file. While the copy is complete, the two workstations can be normally reused without any extra settings.

Generally speaking, as all workstations in a computer classroom were bought at same time, they should be equipped with identical hardware, such as the hard disk and network adapter. Thus, we can only back up one image copy of the normal hard disk and store it on server. Note that it is necessary to back up again if we add new software to all workstations.

2.3 Minimizing the image file size

The image file size is in direct ratio to the hard disk capacity of the normal workstation. If the hard disk is a large one, then the image file size is also large. Thus, it will reside a large space in the network server. Therefore, it is necessary to reduce the image file size, as the following two policies.

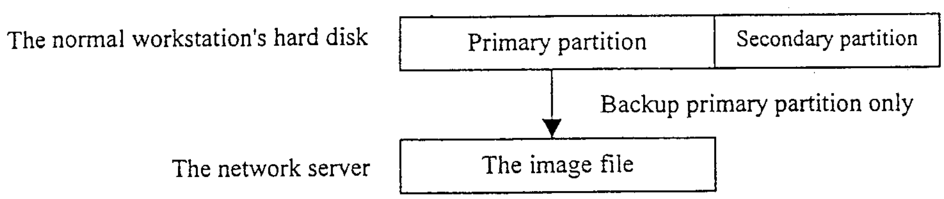

Fig. 3: Backing up the primary partition

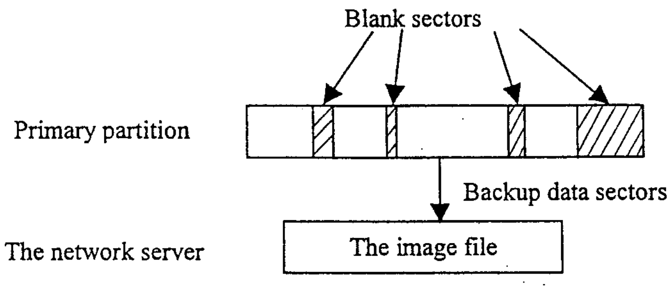

Fig. 4: Taking blank sectors out of the image file

(1) Backing up the primary partition only, as shown in Fig. 3. We can break the normal workstation's hard disk into two partitions for storing system software (primary partition) and user data (secondary partition). The operating system and applications will be installed in the primary partition. In order to reducing the image file size, we can specify an adequate size for the primary partition to store the software system.

(2) Taking blank sectors out of the image file, as shown in Fig. 4. Before a new blank hard disk can be used for data storage, it must be formatted, including low-level formatting, creating partitions or logical drives in an extended partition, and doing a logical formatting. The logical formatting operation will check if each sector is a reliable storing medium, banish bad sectors from free space, and write a default value into the storing medium. (In general, the default value in both FAT and NTFS are 0F6h and 00, respectively). In other words, a sector recording the default value is called a blank sector. After we installed a hard disk with operating system and applications, many blank sectors are still left in it. These blank sectors do not need to be backed up and thus the image file size will be reduced. Therefore, the step II in Section 2.2 has to identify blank sectors and take out them for reducing the image file size.

3. SERVER-BASED BACKUP AND RECOVERY ALGORITHMS

In this section, we will explain the algorithms of our proposed approach: server-based backup and recovery. The algorithms will be described by using a hard disk representation.

3.1 Hard disk representation

In general, each hard disk structure is divided into heads, cylinders, tracks, and sectors. The smallest component is the sector, which holds 512 bytes of data. A number of sectors are grouped together to form a track. A cylinder is the vertical group of tracks, usually twice the number of platters. Some of the highest capacity drives also use techniques to have more sectors per track on the outer cylinders, but fewer sectors per track on the inner cylinders. All of these mapping are made invisible to software accessing the drive. From the software point view, software always thinks of the drive as having a fixed number of sectors per track, head, and cylinder. In this paper, the hard disk will be introduced as a fixed number of sectors per track.

We assume that a hard disk is equipped with K sectors per track (1, …, K), J heads (0, …, J-1), and I cylinders (0, …, I-1). Its content can be represented in cylinder format as $[C_0, C_1, \ldots, C_i, \ldots, C_{I-1}]$ where $0 \leqq i \leqq I-1$, or in head format as $[H_{0,0}, H_{0,1}, \ldots, H_{i,j}, .., H_{I-1,J-1}]$ where $0 \leqq j \leqq J-1$, or in sector format as $[S_{0,0,1}, S_{0,0,2}, \ldots, S_{i,j,k}, .., S_{I-1,J-1,K}]$ where $1 \leqq k \leqq K$. In other words, C_i represents the data in the i^{th} cylinder, $H_{i,j}$ represents the data in the j^{th} head of the i^{th} cylinder, and $S_{i,j,k}$ represents the data in the k^{th} sector of the j^{th} head of the i^{th} cylinder, respectively. A head data is consisted of K sectors; i.e., $H_{i,j} = [S_{i,j,1}, S_{i,j,2}, \ldots, S_{i,j,k}, .., S_{i,j,K}]$. A cylinder data is consisted of J heads; i.e., $C_i = [H_{i,0}, H_{i,1}, \ldots, H_{i,j}, .., H_{i,J-1}]$.

A byte in a hard disk of the i^{th} cylinder, the j^{th} head, the k^{th} sector, and the m^{th} byte will be represented as $B_{i,j,k,m}$. In a general personal computer, a sector stores 512 byte data; therefore m=1, 2, …, 512. For a presentation example, a string with five bytes in a certain sector is shown as $B_{i,j,k,m}$, $B_{i,j,k,m+1}$, $B_{i,j,k,m+2}$, $B_{i,j,k,m+3}$, and $B_{i,j,k,m+4}$, sequentially.

In fact, disk access is a data transferring operation between hard disk and memory. Reading the i^{th} cylinder data can be represented as $M \leftarrow C_i$. In writing case, one cylinder data in M writing into a hard disk will be represented as $C_i \leftarrow M$. Here, the memory size, M, is used as one cylinder.

The image file of the normal workstation's hard disk can be created by using DOS-level appending operation. During the creating process, a set of cylinders is appended to the image file sequentially. Let us append a cylinder to the image file, F, in which the appending operation can be represented as $F \leftarrow F+C_i$. Appending a head data to the image file can be represented as $F \leftarrow F+H_{i,j}$. Reading one cylinder data from the image file to memory may be represented as $M \leftarrow F(C_i)$.

3.2 Server-based backup algorithm

This server-based backup algorithm will sequentially read one cylinder data per time from the normal workstation's hard disk and to write the read data onto sever to form an image file. The basic approach is to employ DOS function calls with specifying the numbers of cylinder and head. Thus, the image file includes the information of partition table, boot sector, file allocation table, directory, and data. The backup algorithm is shown in Fig. 5. Note that the algorithm does not eliminate blank sectors.

1. Create a file F, namely image file, in the network server
2. Get the normal hard disk drive parameters: I (the number of cylinders), J (the number of heads), and K (the number of sectors)
3. Append I, J, and K to the image file
4. Allocate a temporary memory space M for one cylinder data
5. FOR i=0 to I-1
6. Read one cylinder data from the normal hard disk; i.e., $M \leftarrow C_i$
7. Append the data to the image file, F; i.e., $F \leftarrow F+M$
8. NEXT i
9. Close the image file F

Fig. 5: Server-based backup algorithm

The image file stored the entire operating system and applications will inhabit a large disk space in the network server. Therefore, it is necessary to minimize the image file size. One approach is to take all blank sectors out of the image file. In general, there are many blank sectors in a new installed hard disk. The server-based backup algorithm has to identify all blank sectors and exclude them. For this purpose, the cylinder number, head number, and the sector number have to be recorded with a sector data.

3.3 Server-based recovery algorithm

The server-based recovery algorithm is similar to the server-based backup, except that it is to perform a disk-write operation on a workstation. This algorithm has to detect if the hard disk type recorded in the image file is the same with the hard disk equipped in the target workstation. The server-based recovery algorithm is shown in Fig. 6, in which it does not take care of blank sectors. The 'Input' is used to specify the target workstation number.

Input: workstation number

1. Open the image file, F, from the network server
2. Read three hard disk parameters I, J, and K from the image file
3. Get the target workstation's hard disk drive type: cylinder number, head number, and sector number
4. Compare the target hard disk type with I, J, and K. If the both types are same, then go to Step (5); otherwise stop
5. Allocate a temporary memory M for one cylinder data
6. FOR i=0 to I-1
7. Read one cylinder data from the image file F; i.e., $M \leftarrow F(C_i)$
8. Search the default computer name and IP address from M
9. IF found THEN Modify them to the input of workstation number
10. Write the i^{th} cylinder data to the target hard disk; i.e., $C_i \leftarrow M$
11. Clear M
12. NEXT i
13. Close the image file F

Fig. 6: The server-based recovery algorithm

4. IMPLEMENTATION AND MEASUREMENT

This section presents the implementation of the server-based maintenance approach and measures its performance for maintaining two computer classrooms.

4.1 Storing format of computer name and IP address

The server-based recovery program will set an appointed computer name and an IP address to each workstation. For this purpose, the recovery program will search the image file for particular strings and modify them to new ones. The particular strings are the default computer name and IP address. Therefore, we have to understand the storing format of them in the Windows 95 and 98 (FAT), and NT (NTFS) file systems.

In the Windows 95 and 98 systems, each character occupies one byte, as the following examples,

1. We assume that the default computer name is 'W00' which occupies three bytes and is stored in ASCII code as "57h 30h 30h".
2. The default IP address '140.131.89.100' occupies 14 bytes and is stored as "31h 34h 30h 2eh 31h 33h 31h 2eh 38h 39h 2eh 31h 30h 30h".

In the Windows NT system, each character is stored in two bytes, as the following examples,

1. The default computer name 'B00' occupies six bytes and is stored as "42h 00h 30h 00h 31h 00h" format.
2. The default IP address '140.131.90.100' will occupy 28 bytes and is stored as "31h 00h 34h 00h 30h 00h 2eh 00h 31h 00h 33h 00h 31h 00h 2eh 00h 39h 00h 30h 00h 2eh 00h 31h 00h 30h 00h 30h 00h".

4.2 Renaming computer name and changing IP address

The server-based recovery program plays the modifying computer name and IP address role. For example, we will maintain the 13th workstation, namely 'W13', in the 'W' computer classroom. First the W13 workstation will be connected to server via boot-ROM (or drive A booting) on DOS environment. Then, we executes the recovery program (called ROOMWHD.EXE) on the W13 workstation to maintain its hard disk content; the command is

A:\>ROOMWHD 13 <Enter>.

We assume that the default computer name resided in the image file is the 'W00'. The ROOMWHD program will search the 'W00' string, change to 'W13', and write into the W13 workstation's hard disk. Therefore, the classroom manager does not need to modify the workstation's name after it was recovered. In the changing IP address case, we assume that the default IP is '140.131.89.100'. The program will also search the default IP address string at the same time of searching computer name. If the program found the default IP address, then the new IP address '140.131.89.113' will be set and written into the target workstation.

4.3 Measuring Maintenance Time for Two Computer Classrooms

For the first case, we apply the server-based maintenance approach to manage a computer classroom, in which there are fifty workstations, as shown in Fig. 1. It takes about 90 minutes to set up the Windows 95 system and some applications on a workstation, namely normal workstation; the size of the software is about 340 Mbytes. By using the hard-disk to hard-disk directly copy, it will take 10 minutes to make one copy. Since each workstation's shell is securely locked on the computer desk, moving its hard disk to the normal workstation will take about 20 minutes, including the time to lock/unlock the shell. By using the server-based recovery program, the recovering time for one workstation is about 14 minutes. According to our measurement, using the proposed approach to recovering 49 workstations will take 102 minutes.

If we want to install new software to 50 workstations, the server-based maintenance approach is also the fastest. We assume that installing the new software 120 Mbytes, such as Visual Basic, to the normal workstation will take about 20 minutes and the server-based backup time for the 460 Mbytes is about 17 minutes. Applying the individual setup and hard-disk to hard-disk copy will take 1000 (=20*50) minutes and 1637 minutes, respectively. According to our measurement, using the server-based maintenance approach totally takes 165 minutes.

For second case to maintain a computer classroom with 53 workstations, each workstation is equipped with 1.08 Gbytes hard disk storing the Windows NT Workstation system, OFFICE 97, and other applications about 613 Mbytes. Table I presents the maintenance results in NT classroom. We take about 390 minutes to setup these software systems onto the normal workstation. Applying hard-disk to hard-disk copy approach takes 38(=20+10+8) minutes to duplicate one copy and 1996 minutes for 52 workstations, By using server-based maintenance approach, it takes 21 minutes of backup time and 172 minutes to recover 52 workstations.

5. CONCLUSIONS

This paper presents an efficient, fast, and cheap approach to keeping data integrity of hard disk's content for all workstations in a computer classroom. The approach can adapt to the FAT (file allocation table) and NTFS file systems, which are widely installed in computer classrooms. The basic concept of the approach is to generate one image copy of the whole hard disk's content; then the image file will be stored on a server that can be accessed by each workstation in the computer classroom. When a workstation is unable to boot or run into some hitches because of its software, the classroom's manager may take a short time to restore the hard disk's content by executing a maintain program that performs a cylinder-based recovery work according to the image file. By using this approach, maintaining 52 NT workstations will only take 172 minutes. Moreover, the approach can automatically clean all viruses and assign a different computer name and a different IP address to each workstation. Thus, we may provide a proper computer classroom for teachers conducting courses and students doing homework.

Table I : A comparison of three approaches to maintaining the Windows NT system

Testing item \ Three approaches	Individual Setup	Hard-disk to Hard-disk Copy	Server-based Maintenance
Setup one workstation	390 minutes	390 minutes	390 minutes
Backup time	0 minute	10 minutes	21 minutes
52 Workstations	390 minutes *52	10 minutes *52	172 minutes
Time to move hard disk (including the time to lock/unlock shell)	0 minute	20 minutes *53	0 minute
Time to modify computer name	0 minute	8 minutes *52	0 minute
Total	20670 minutes	2396 minutes	583 minutes

ACKNOWLEDGEMENT

This work was supported in part by the National Science Council of R. O. C. under Grant NSC87-2218-E-227-001.

REFERENCES

[1] R.O. Harger, "Teaching in a computer classroom with a hyper-linked, interactive book," *IEEE Trans. on Education*, Vol. 39, No. 3, pp. 327-335, Aug. 1996.

[2] D. Wolff, "Computers in classroom research," *Computers and Education*, Vol. 23, No. 1/2, pp. 133-142, 1994.

[3] A.R. Garrod and M. Maziar, "Development of Classroom Management skills," *IEEE Trans. on Education*, Vol. 31, No. 2, pp. 128-132, May. 1988.

[4] R.A. Tamura, P. Belew, J.M. Blakely, O. Eman, E. Grantham, R. Griswold, W.S. Hall, L. Harries, P. Hipson, B. Kenner, J.J. Kottler, R. Laeremans, P.L. Lujan, B. Montemer, T. Parker, D. Pietrocarlo, J. Thayer, and V. Toth, "Programming Windows 95 Unleashed," *Sams Publishing*, 1995.

[5] N. Behrmann, C. Chambers, S. Fuller, F. Houlett, K. Pagan, "Windows 95 Network Administration," *Simon & Schuster Pte, Ltd.*, 1996.

[6] K.K. Pamakrishnan, "Performance Considerations in Designing Network Interfaces," *IEEE J. Selected Areas Comm.*, Vol. 11, No. 2, pp. 203-219, Feb. 1993.

[7] P.A. Steenkiste, "A Systematic approach to host interface design for high-speed networks," *IEEE Trans. on Computer*, March 1994.

[8] O.C. Ibe, H. Chow, and K.S. Trivedi, "Performance evaluation of client-server systems," *IEEE Trans. Parallel and Distributed Systems*, Vol. 4, No. 11, pp. 1217-1229, Nov. 1993.

[9] A. Apostolico and R. Giancarlo, "The Boyer-Moore-Galil string searching strategies revisited," SIAM Journal on Computing, vol. 15, no. 1, pp. 98-105, Feb. 1986.

[10] M. Minast, C. Anderson, E. Creegan, "Mastering Windows NT Server 4.0," *Network Press*, 1996.

[11] C.S. Lee, "Using Server to Maintain Workstations Software for Improving Computer Classroom Teaching," Proceedings of ICCE'98, Vol. 2, pp. 590-594, Beijing, China, Oct. 1998.