Network File Server Performance in a University Environment: A Case Study

A Thesis
Submitted in partial fufillment of the requirements
for the degree of

Master of Science in Computer Science

University of Southern Maine
School of Applied Science
Depatment of Computer Science

by

Stephen A. R. Houser

August 1996

The University of Southern Maine

Department of Computer Science

August 7, 1996

We hereby recommend that the thesis of *Stephen A. R. Houser* entitled *Network File Server Performance in a University Environment: A Case Study* be accepted as partial fulfillment of the requirements for the Degree of *Master of Science in Computer Science*.

Committee:

John R. Heath (Chairperson)

Charles Welty

Robert Boothe

Accepted:

Brain Hodgkin,
Dean, School of Applies Science

Abstract

Workloads of network file server disk IO subsystems have very different characteristics than observed in timesharing or local IO systems described in the literature. In this study, we provide a detailed analysis of both disk and network workload traces collected from Novell NetWare network file servers. We characterize file server disk and network traffic and give insights into access patterns; we also consider the relationship between network and disk throughput. Measurements and statistics presented will aid designers and managers in designing and tuning network file servers and their disk subsystems. Our results can be used by analysts to parameterize synthetic models for file server and server subsystem studies.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Networks and Network File Servers

Local area network (LAN) file servers provide global file systems that are shared by client workstations. Novell NetWare is the most popular file server operating system for personal computer LANs. Workstations utilize NetWare's global file system by submitting IO requests to a file server using simple network protocols. The file server processes these requests and replies to the workstations over the LAN.

Workstations use the global file system provided by the file server as they would use a local file system, to store data and application programs. Workstations may or may not have their own file system resources. In the case of *diskless* workstations, the file server provides all of a workstations' file services, including its operating system, application storage, data storage, and temporary (swap) space. In the case of *diskfull* workstations, the file server provides a shared disk resource among many workstations which have their own local file systems.

For the global file system to be useful, the file server must provide the performance and reliability of a local file system. For data sharing, the server must provide adequate locking and synchronization mechanisms to allow sharing of data between workstations.

The performance of the file server, in providing shared access to its global file system, can be optimized in many ways. Using caching, the file server retains, in volatile memory, file data recently accessed by workstations. Read requests serviced from the file cache do not require disk accesses, substantially reducing response time. Disk accesses for write requests are postponed by using *lazy-write* caching. This reduces the client request response time. File system performance can be increased by increasing the size of this file cache and tuning cache parameters. File servers dedicate a large portion of their memory for file cache use.

Caches, found in disk subsystem adapters and disk controllers, also improve the performance of the disk subsystem. Even with these improvements, disk accesses remain an order of magnitude slower than server file cache accesses. To obtain acceptable performance, most client IO requests should be serviced from the server's file cache. Read-ahead, write-behind, and replacement algorithms have all been implemented to realize this goal of increased performance.

1.2 Purpose of This Study

Tuning and designing network file servers and server disk systems requires thorough knowledge of both disk and network access patterns. Knowledge of workstation file access patterns, and network activity, allows us to characterize the performance of the file server at the client interface. Disk access patterns allow us to characterize cache and disk subsystem performance. By analyzing both network and disk workloads together we are able to provide insight into file server and file cache performance.

In this study, we characterize disk access patterns, with detailed information from trace data collected on *live* NetWare file servers. This characterization is useful in tuning and optimizing disk subsystems for use with network file servers. It also provides insight into disk access patterns on a network file server. The results are useful for developing server disk workload models, which can be used to design and parameterize analytical models. Trace data are used in trace-driven simulations and comparative analysis.

We also characterize network access patterns with network trace data collected on the same file servers, during the same time periods as disk trace data. This characterization is useful in tuning overall file server performance and for future capacity planning. It provides insight into workstation access patterns, and allows us to compare diskless and diskfull systems to assess their effects on file server performance. As with disk trace data, these data are useful in analytical modeling and simulations.

Finally, we compare both disk and network trace data statistics. This comparison allows us to determine the correlation between network and disk throughput and to determine if we can predict disk workload characteristics based on network workload characteristics. This comparative analysis is useful in tuning the file server as it provides us with complete knowledge of the entire system's behavior.

1.3 Review of Related Studies

Disk workload analysis is discussed in the literature, but it is not used in combination with network workload analysis nor does it address file server workloads specifically. Most studies focus on time-sharing or local file systems. VAX/VMS trace data were collected in [27, 3], BSD UNIX trace data were collected and applied in [8], and IBM mainframe systems were traced in [28]. All of these traces were used to drive simulation models, and to evaluate cache performance. Simulations to evaluate non-volatile write caches [4] were conducted using trace data from [27, 3]. Cache replacement algorithms were also investigated in [26] using these traces.

Network workload analyses are numerous in the literature, but they are not studied in combination with disk workload analyses. NetWare network workloads are examined in [17, 16, 18]. Chappell [6] discusses several methods of capturing and characterizing network workloads. Traces and characterizations of Sun Microsystems' network file system (NFS) are described in [5], and are used to describe important aspects of trace collection. Diskless UNIX workstation traffic is analyzed in [10] to optimize a university computer network using the network file system (NFS).

Chapter 2

Experimental Environment

A network file server is a very complex part of a LAN. To perform an accurate analysis, we need to have a clear understanding of its configuration. We also need to understand how the file server connects to and interacts with the LAN. In this chapter we begin by describing networks and network file servers in general. We then describe the specific network environment in which we conducted our study. Lastly, the specifics of the two network file servers measured are discussed.

In any empirical study, it is useful to understand the environment in which the study occurs. Here we describe the network and file server independently. We begin by looking at the user workstations (file server clients) on the network. The workstation descriptions are followed by a detail of the network layout and connectivity. Then the file server is examined, beginning with its hardware configuration. The internal cache and disk subsystem parameters are described as they are major focal points of our study.

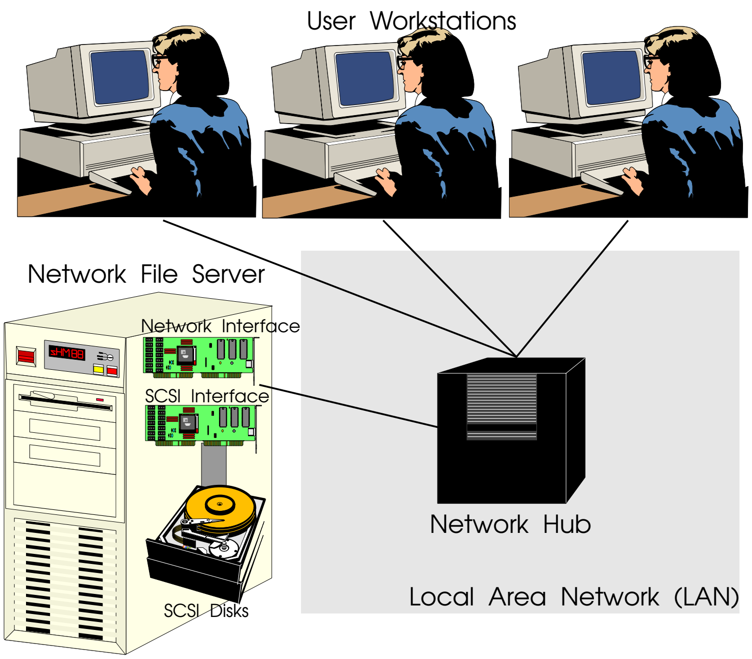

Figure 2.1: Overview of Network Components

2.1 Introduction Networks and Network File Servers

For the purposes of discussion, we partition the LAN into five sections: the user workstation(s), the physical network, the file server (software, hardware, and cache), the file server's IO bus, and the file servers disk drive(s). Figure 2.1 diagrams these components, and their interaction with each other. For simplicity we have identified parts of the file server, namely its IO bus and disk drive(s), as separate network components. These are integral parts of the file server. We have separated them from the file server for further discussion and analysis.

2.1.1 Network Workstations

Workstations are connected to a LAN through a network interface. The network interface is a specialized hardware device that allows the workstation to access the physical network. Each workstation loads driver software that is specific to its operating system and network interface. The driver software is responsible for processing network and file server requests. In addition, the driver may be required to handle unsolicited requests for diagnostic and management information [18]. The network driver software consists of several layered programs, corresponding loosely to standard network architecture models.

2.1.2 Physical Network Media

At the lowest level the network interface card connects to the physical network media. The physical network is responsible for transporting bits between workstations and file servers. This transport is done via a communications channel. Typically, it is high speed, on the order of 10 megabits per second for a LAN. Low level data link LAN protocols [30, 12] determine how the communications link is accessed.

2.1.3 Network File Server

The file server is a focal point of LAN activity. The server allows multiple workstations to share files that are stored on its disk drives. Sharing any files (both data and program) requires that access to the shared files be synchronized. Synchronization is needed to prevent data corruption and to ensure workstations receive the most current data when requested. The file servers primary responsibility is providing and synchronizing access to the files it stores [7].

It is important to note that a server makes no distinction between data files and program files, it is the workstation's responsibility to use the file in the correct context. A strict file server does not interpret the files it stores. Servers only act as a storage medium for files, managing concurrent access as needed. In practice file servers often use some storage for operating system files and data storage.

2.1.4 File Caching

Rather than forward all data accesses directly to disk drives, file servers use caching to improve request response time. The configuration of the cache, maintained in the server's main memory, can greatly affect a file server's performance. Thus, many file servers allow these parameters to be adjusted to tune performance. In addition, servers may use a combination of read-ahead and write-behind (or lazy-write) caching [6, 11, 25].

2.1.5 Disk Access

When a server cannot service a request from its file cache, a disk request is generated to retrieve or store data. First, the request is marked as pending. Then the server's disk drives are signaled via an internal IO bus (SCSI [1] in our study). Through this IO bus, requests are passed to the attached disk drives. Data, or result codes, from the disk drives, are returned via the IO bus to the file cache. After this internal processing completes, the pending request is serviced from data stored in the server's file cache.

2.1.6 Other Network Services

In addition to managing access to files, file servers often provide various levels of file security. Security allows authorized users to access data, while prohibiting other, unauthorized, users access. File servers can also provide varying levels of access to different users. The security aspects of a file server can be very complex [7]; such a discussion is beyond the scope of this paper.

Other resources, such as printers, modems, and fax machines, are often shared on a LAN. These resources may be managed by the file server or by other similar *resource servers* on the LAN.

A file server is not the only method of sharing access to data resources. Other systems, called *disk* *service* systems, were in use on personal computer networks before the file server idea was fully developed. These systems normally involved block level access to disks. Often these systems were prone to problems due to MS-DOS's single-user nature [7]. Access was based on actual disk blocks, as compared to the file level access that we see in today's file servers.

2.2 The University Network Environment

The University network in which we conduct our analysis is made up of several interconnected Ethernet LANs. The LANs are interconnected via fiber optic cabling to a central network gateway. The gateway acts as a filter on the network, routing between administrative networks, student lab networks, off-campus networks, and the Internet. The gateway only passes inter-network traffic between LANs, ignoring local network traffic. Thus, traffic on the LANs is comprised of mostly local communications.

The majority of the campus LANs are Ethernet based, using either 10BaseT (twisted pair) or 10Base2 (coaxial) wiring. A small number of AppleTalk based LANs are used on campus; they are not routed by the central gateway, nor are they active in our experimental LANs. LAN connections in each building are concentrated in wiring closets. In the wiring closets, 10Base2 segments and 10BaseT lines are connected in a multi-media access center (MMAC). In addition, the MMAC connects via a fiber optic port to the central gateway.

Beyond the physical separation of LANs, the network is logically divided into two more general networks: a student network and an administrative network. Both of these logical networks span several LANs. The student network consists of workstations and file servers used to support the University's general access computer labs. The administrative network is composed of many diverse workstations and servers that support the general administrative operation of the University. This division is largely a managerial division, although the central gateway does provide physical separation.

The Novell NetWare file server that serves the student lab is located on the student network (student LAN). Similarly, the administrative file server resides on the administrative network (administrative LAN). Again, a central gateway prevents unnecessary inter-network traffic.

To allow workstation communications with LAN file servers and with hosts on the wide area Internet, both IPX and TCP/IP protocols are used. MS-DOS workstations load *packet driver* software that allows both protocols to run concurrently over a single network interface. Thus, IEEE 802.3 and Ethernet II frame types are available to the workstation and are present on the network. NetWare is configured to use IEEE 802.3 type frames and TCP/IP is configured to use Ethernet II type frames.

2.3 Student Lab Environment

The University's general access student computer lab is used by students of all disciplines for a variety of applications. Measurements on use of the student computer labs are gathered by a software license monitoring program that was developed, at the computer lab, to monitor application use. Measurements from past semesters show the most heavily used application is word processing, accounting for nearly 60% of all computer lab use, by time. A distant second is electronic spreadsheets which accounts for 13% of total use. Communications is third, at 10%, then miscellaneous MS-DOS use, at 9%, programming, at 4%. Mathematics, database, and miscellaneous courseware account for the remaining application use.

2.3.1 Workstation Configuration

The student LAN considered in this study contains 70 workstations. Forty-five workstations are Intel 80486-based MS-DOS computers with 16 megabytes (MB) of RAM and two 3 1/2-inch 1.44 MB floppy disk drives. Other configurations include 20 Intel 80286-based, and 5 Intel 8086-based

MS-DOS computers with 1 MB of RAM and a single 3 1/2-inch 1.44 MB floppy disk drive. None of the workstations have internal hard disks, they rely entirely on the file server for all operating system and application files. Most workstations run MS-DOS version 5.00; the 8086-based machines run MS-DOS version 3.21 to conserve memory. All of the workstations use Cabletron 10BaseT Ethernet adapters to access the network.

Each workstation is initialized with a floppy boot diskette. The boot diskette loads the operating system and network drivers. When the network driver is operational, the operating system is directed to the file server for additional file services.

During operation, each workstation is allocated temporary storage space on the file server. This space may be used by applications to create temporary files. In general, permanent storage of student data is on floppy diskettes, not on the file server. To improve processing speed, students will often copy large data files to the temporary space while using an application. All files created in the temporary storage space are deleted when an application terminates (i.e., between work sessions). Each station is restricted, using NetWare accounting services, to a maximum of 8 MB of data in this temporary space. This restriction prevents users from acquiring excessive file server storage.

In addition to workstations, three shared printers are attached to the file server analyzed in this study. Print files are stored on the file server's system disk while they are waiting to be printed. This queuing mechanism creates additional temporary files on the server while the print jobs wait to be serviced. Temporary print files are not included as part of the space restrictions for each station as mentioned above. They are deleted when the printer has completed servicing the file.

Here is a summary of the workstation characteristics on the student LAN that access the student file server:

- 45 MS-DOS (v5.0) 80486/33 MHz computers
- 20 MS-DOS (v5.0) 80286/12 MHz computers
- 5 MS-DOS (v3.21) 8086/8 MHz computers

- Cabletron 10BaseT Ethernet network interface
- Client operating system files on file server
- Applications stored on file server
- Permanent storage on floppy diskettes
- Limited temporary storage on file server
- Print queue data stored on file server

2.3.2 File Server Configuration

The student access file server analyzed in this study is an IBM PS/2 model 95 running Novell NetWare version 3.11. The processor is an Intel 80486 running at a clock speed of 33 MHz. The server has 16 MB of RAM, a single 3 1/2-inch floppy diskette, and two SCSI hard disk drives. The first drive is a 320 MB, IBM SCSI disk, ID number 0. This drive contains system files, printer queue directories, and standard NetWare utility programs. The second drive is a 1.2 gigabyte (GB), Fujitsu SCSI disk, ID number 1. This drive contains applications and temporary storage areas used by workstations in the student computer lab.

The file server is connected to the LAN via a Cabletron 10BaseT Ethernet interface card. This LAN connection is to the same 10BaseT hub that services workstations in the student computer lab. The hub is also connected to the rest of the campus network through a router via fiber optic links.

During our analysis, and in typical use, the student file server runs several server processes, known as NetWare Loadable Modules (NLMs). The following NLMs were running on the student file server during our analysis:

- PS2SCSI.DSK – SCSI disk driver,
- N386LCE.NET – Cabletron network interface driver,
- PATCHES.NLM – patch management module,
- RSPX.NLM – remote LAN transport module,

- REMOTE.NLM – remote management module,
- MONITOR.NLM – server management module.

Here is a summary of the student file server hardware and software characteristics:

- IBM PS/2 Model 95 (80486/33 MHz)
- Novell NetWare v3.11
- 16 MB RAM
- Cabletron E31xx 10BaseT Ethernet Adapter
- Adaptec/IBM SCSI Interface (SCSI id 7)
- IBM 320 MB SCSI disk (SCSI id 0)
- Fujitsu M2266A 1.2 GB SCSI disk (SCSI id 1)

And finally, here is a summary of memory use on the student file server, as reported by the MONITOR.NLM program:

- Permanent 3.52 MB, 3.15 MB in use
- Alloc Short Term 689 KB, 117 KB in use
- Cache 6.75 MB
- Cache Moveable 3.94 MB
- Cache Non-Moveable 957,940
- Total Server Work Memory 15.49 MB
- (MS-DOS partition takes the rest)

2.4 Administrative Environment

The second file server analyzed is on the same campus, but is used for University administrative functions. Faculty and staff use the administrative LAN and file server for many applications. In this environment, word processing is also believed to be the most heavily used application. Unlike the student access server, no detailed application tracking is performed on the administrative server. Our estimation of usage is based on the number of workstations that have access to each particular application. Word processing is available on more workstations than any other software program. Spreadsheet, telecommunication, and database programs are also available to administrative users. These applications, however, are not enabled on all workstations.

2.4.1 Workstation Configuration

Administrative workstations vary more in type and configuration than those in the student environment. There are many different types of workstations, including Intel 8086 (XT), Intel 80286 (AT), Intel 80386, Intel 80486, and Intel Pentium MS-DOS computers. There are also several Apple Macintosh computers. The Macintosh workstations use the file server strictly for data file storage and electronic mail. There are no Macintosh applications available on the file server. Overall, there is an upper limit of 255 simultaneous file server connections.

We estimate that 50 to 75 workstations use files located on the file server at any given time. This is based on the average number of users logged on during our analysis periods.

The majority of administrative workstations have local hard disks. These local disks store the workstations operating system, some specialized application programs, and most of the workstations data files. Generally available applications, such as word processing programs, are stored on the file server. A number of shared configuration and data files are stored on the file server. Outside of these shared files, most activity on the file server is sharing of applications and print queue management, as in the student environment.

Here is a summary of the workstations on the administrative LAN that access the administrative file server:

- MS-DOS 80486 computers
- MS-DOS 80286 computers

- MS-DOS 8086 computers
- Apple Macintosh computers
- Mixture of Ethernet adapters (mostly Cabletron)
- Client operating system files on workstation
- Applications stored on file server and workstation
- Permanent storage on file server and workstation
- Temporary storage on workstation
- Print queue data stored on file server

2.4.2 File Server Configuration

The administrative file server is an IBM PS/2 Model 80 also running Novell NetWare v3.11. The processor is an Intel 80486DX running at a clock speed of 25 Mhz. This server is configured with 16 MB of RAM, and two SCSI hard disks, a 320 MB IBM disk, ID number 0, and 1.2 GB Fujitsu disk ID number 1. The drive configuration is similar to the student file server with system files on the 320 MB disk and application files on the 1.2 GB disk. The 1.2 GB drive, on this server, also contains directories for shared file storage. These shared directories are used for long-term storage of data files.

The file server is connected to the LAN via a Cabletron 10BaseT Ethernet interface card. This LAN connection is to a 10BaseT hub that services the University's network center. The hub is connected to other administrative LANs on the campus network via fiber optic links. In addition, a fiber optic link connects the network center's LAN to the central gateway, which connects to the rest of the campus network and the Internet.

As on the student file server, the administrative file server runs several server processes, NLMs. The following NLMs were running on the administrative file server during our analysis:

- PS2SCSI.DSK – SCSI disk driver,

- LSLENH.NET – link support module,
- E31n4x.NET – Cabletron network interface driver,
- PATCHES.NLM – patch management module,
- PATCHMAN.NLM – patch management module,
- XDMFIX.NLM, SPXDDFIX.NLM, SPXFSFIX.NLM, SPXLISFX.NLM, SPXMSFIX.NLM, SPXNSFIX.NLM – operating system patches,
- APPLETALK.NET – AppleTalk network driver,
- AFP.NLM – Apple Filing Protocol support,
- RSPX.NLM – remote SPX transport module,
- REMOTE.NLM – remote management module,
- MONITOR.NLM – server management module.

Here is a summary of the hardware and software characteristics of the administrative file server:

- IBM PS/2 Model 80 (80486/25 MHz)
- Novell NetWare v3.11
- 16 MB RAM
- Cabletron E31xx 10BaseT Ethernet Adapter
- Adaptec/IBM SCSI Interface (SCSI id 7)
- IBM 320 MB SCSI disk (SCSI id 0)
- Fujitsu M2266A 1.2 GB SCSI disk (SCSI id 1)

Chapter 3

Novell NetWare

The Novell NetWare file server software is a collection of asynchronous processes that interact to provide file access services to workstations connected to a local area network (LAN). In this chapter, we describe some of the NetWare processes and communication protocols. Specifically, we describe workstation-file server communications using the NetWare Core Protocol (NCP), NetWare file caching in the file server, and file server to disk communications.

3.1 NetWare Server to Workstation Communications

The basis of all NetWare network communication is the NetWare Core Protocol. The NCP defines a session layer protocol in the ISO networking model. For performance reasons, NCP specifies a *best-effort* transport protocol, namely the Internet Packet eXchange (IPX). To handle error detection and packet sequencing problems, NCP implements a stop-and-wait protocol [21]. Each request packet is also assigned a sequence number, which is used in reply packets. Congestion problems on servers are handled with special *wait* acknowledgements, which reset timeout parameters on waiting workstations.

3.1.1 The NetWare Core Protocol

The NCP specifies over 300 operations with associated data structures for communication between network entities. NCP supports print servers, queue management, and network management to name a few. NCP is designed by Novell, primarily for use with their networking products. Much of the specification is undocumented or difficult to obtain. The workstation and file server primitives used by the NCP are however available through some third party sources, such as network analyzer vendors [20, 31, 6]. Alternatively some trade magazines have published limited information on NCP and NetWare file servers [29].

As mentioned above, NCP operates as a session layer protocol as defined by the ISO networking model. NCP handles error correction and packet sequencing internally. Thus, only a best effort transport protocol is needed in lower network layers. The most common transport layer, for NCP, is the Internet Packet eXchange (IPX) protocol. The IPX protocol is based on another similarly named protocol, designed by Xerox.

Because NetWare is commonly used on local area networks, NCP is defined as a stop-and-wait protocol. The use of this type of flow control, at this high layer in the network model, allows lower network layers to ignore error detection and correction with the hope of increasing performance. Each outgoing NCP request is acknowledged by the destination host with an NCP acknowledgement. To reduce traffic generated by numerous acknowledgements, the NCP allows hosts to *piggyback* an acknowledgement onto a packet with other data.

The NCP assigns each packet a sequence number. These numbers serve two purposes; sequencing packets at the receiver and identification of packet loss situations. Sequencing ensures that receiving hosts process packets in the order they were intended. A benefit of using sequencing is that NCP entities can determine when a packet has been lost at some lower level. Because NCP is connection-oriented, it provides for error recovery in these situations.

Congestion problems on a central file server are handled by the NCP as well. The NCP protocol provides a special *wait* acknowledgment packet. This reply is sent to workstations requesting operations that the server cannot immediately service.The wait signifies that the requesting station should continue waiting for the request to be serviced.

In NCP each request is acknowledged in one of three ways:

- Positive acknowledgement. The requested operation succeeded; any requested data is attached to the acknowledgement packet.
- Wait acknowledgement. The server is busy; the request is queued to be performed on the server, and another acknowledgement packet should follow.
- Negative acknowledgement. The requested operation failed; a result code is attached to the acknowledgement packet.

3.1.2 NCP File Operations

When analyzing file access patterns, we need to inspect NCP requests and replies, extracting important file operations. The four major file operations that we are concerned with, read, write, open, and close, are readily available by examining NCP request packets [6]. Each operation is represented by a unique NCP request code. Unfortunately, NCP reply packets do not specify which NCP request they are associated with. Using the stop-and-wait nature of NCP combined with the packet sequence numbers we are able to relate request packets to their corresponding reply packets. A reply packet from a server will have the same connection and sequence information as the original request packet.

The four requests that we are most concerned with in our analysis are:

- Read request. Requests data to be read from an open file, starting at a byte offset (from the beginning of the file). Data is returned to workstation with acknowledgement packet.
- Write request. Requests data to be written to an open file, starting at a byte offset (from the beginning of the file). Data is sent to server with request packet.

- Open file request. Requests that a named file be opened. A *file handle* is returned with acknowledgement.
- Close file request. Requests that a *file handle* be closed.

3.1.3 NCP and Lower Layers

At the lower layers of the ISO network model, the Internet Packet eXchange (IPX) protocol is used to transport NCP requests. IPX provides a connection-less transmission between nodes across multiple networks and it provides an abstract layer on top of the physical network to accommodate multiple hardware implementations.

A connectionless transport protocol is used to provide transmission rather than a connection-oriented protocol because of the additional overhead a connection-oriented protocol would incur at this layer. The NCP, as defined, is already a connection-oriented protocol, negating the need for this service at lower layers, assuming the lower layers are somewhat reliable. Use of a connection-oriented transport protocol would duplicate many features of NCP and make its stop-and-wait nature redundant.

IPX provides a transport layer (ISO model) on which communications can occur. Forwarding of packets to remote networks is handled through IPX routers. Because IPX is similar to IP packets, many routers that support IP forwarding also handle IPX packets.

3.2 SCSI Disk Channel Communication

The small computer system interface (SCSI) standard [1] defines the connection and communication protocols between a computer and peripheral devices, such as disk drives. In many ways the SCSI protocol is similar to network protocols. A message is dispatched from an initiator (host) with a target address (disk). The target services the request and then sends any requested data back to the host. In the time between the request and response the SCSI bus is available for use by other devices.

Like Novell's NCP, the SCSI standard specifies many commands that may be sent to a SCSI target. These commands differ depending on the type of target (fixed disk, CD-ROM, optical scanner, etc.). All SCSI devices handle a base set of commands for management and basic data transfer. Each device type then extends the commands to facilitate the devices' special needs. In our analysis we only need to look at commands used to communicate with fixed disks. Our analyzed systems have only fixed disks on the SCSI bus. On the SCSI bus, as in file server to workstation communication, we are primarily concerned with read and write operations. The disk subsystem has no notion of a file, therefore we can not track file open and close operations at the SCSI level.

The SCSI protocol is different than the stop-and-wait network protocol. Multiple requests can be sent to the disk and responses are not necessarily received in the same order. The SCSI protocol has the following characteristics:

- Read request. Requests *blocks* of data to be read, starting at a numbered block on the disk. Requested data is returned with the acknowledgement.
- Write request. Requests that *blocks* of data be written to a numbered block on the disk. Data to be stored is sent with the request.

SCSI commands address and transfer data in units of disk blocks. A disk block is the smallest individually addressable data unit that can be read or written to a fixed disk device. When a read or write request is put together it specifies how many blocks to read or write, and which block to start with. The disk itself is mapped out into consecutive blocks, starting on the first sector of the first disk platter of the first cylinder. It is important to remember that the disk has no notion of files, only blocks. The operating system (NetWare) is responsible for mapping files and directories onto the disks block structure. In our systems the SCSI disk block size is set at 512 bytes.

There are numerous other SCSI commands, many that relate directly to fixed disk drives. For our analysis however we will not consider them. We can easily discard them knowing they make up only a small portion of all used commands.

3.3 File Server Memory Use and Disk Caching

To service requests from workstations efficiently NetWare incorporates a disk cache. In fact most of a file server's memory is devoted to the disk cache. The NetWare cache serves several purposes and is partitioned into several different *pools*[25].

3.3.1 NetWare Memory Pools

When NetWare loads, it divides the servers memory into several *pools* that are allocated by various processes. These memory pools are *file cache buffers* (movable and non-movable), *permanent* memory, *semi-permanent* memory, and *alloc short term* memory. The file cache buffers pool is the most important (and largest). This is the pool that is used to cache disk requests, both read and write. NetWare loadable Modules (NLM's) may allocate space from this pool when they load, hence the fewer NLM programs loaded provides more cache memory for file accesses.

3.3.2 Basic Cache Architecture

NetWare's cache is used for all disk requests. Both read and write requests are submitted to the cache for resolution. If a disk operation cannot be resolved by the cache, it is passed to the disk driver. When data is requested from the disk driver, it is requested in cache blocks of 4096 bytes, on our analyzed systems. Thus, an entire cache line is read from disk. Write requests are optimized, requiring that only modified disk blocks be written to disk. The disk driver is responsible for converting the cache block into disk blocks and servicing the disk drive.

3.3.3 Read Data Resolution

A read request can follow two paths through the caching mechanism. In the first instance, a cache hit, the read request is serviced almost immediately. When the request is received by the server a lookup on previously read blocks is performed, as well as searching the queue of blocks waiting to be written out to disk. If the request is found, the required data are sent back to the requesting station or process.

A cache miss on a read request falls through the caching mechanism and calls the disk driver module to read data from the disk drive. The data is read in cache blocks, which may constitute multiple disk blocks. In our case, a read of a single byte causes NetWare to cache a 4 kilobyte (KB) block, which translates to eight 512 byte disk blocks read from the disk drive. The data is kept in the cache until the data space is needed for more recent data, or until it is invalidated by a write data request.

3.3.4 Write Data Resolution

Writing data through the cache can also follow two paths, similar to the two paths for reading data. If the data to be written is already in the server's cache, it is first invalidated to prevent other process from reading expired or outdated data. A new request is placed on the disk drivers' queue to write the newly received data to the disk when it is available. The data may be written immediately, or can be delayed (in write intensive environments) to make multiple writes to the same block more efficient. In addition, written data remains in the file cache.

The second case involves data that has not been previously read into the cache. This data is held in the cache, waiting for a full cache block (4 KB) to accumulate, or the write-back timer to expire. In either case the data will be written, perhaps in small increments (512 bytes at a time) until an entire cache block is written. Thus, a single NetWare cache block write may appear as several disk block writes.

3.3.5 Read After Write Verification

NetWare also has a facility for read-after-write verification. The SCSI drives used in the systems being analyzed handle error correction internally and NetWare's read-after-write feature is disabled to enhance performance of the disk drives. Other drive types, MFM, RLL, ESDI, and IDE may not have this feature internally and require NetWare to read each block after is is written to verify its integrity.

Chapter 4

Methodology

To obtain a comprehensive understanding of network file server performance, we measured three distinct aspects of system behavior. First, we describe general patterns of the systems uses. That is, we determine applications that are loaded from the file server and executed on the users workstation. Next, to characterize file server disk activity, we monitor activity on the file server's SCSI bus. The SCSI bus traffic consists of requests that are not satisfied by the file server's disk cache. Finally, we characterize network traffic. Traffic between workstations and the file server. This traffic includes all file operations necessary to read the applications from the file server and all the file operations used to retrieve and store any application data on the file server.

The times during which we collected our measurements were determined by analysis of previous system usage patterns. We predicted the heaviest traffic periods and chose those periods to collect trace data.

4.1 Measurement Tools

In the student computer lab, we used a combination of hardware and software monitoring tools to gather statistics on application use, network activity, and disk activity on the file server. In the administrative environment, we used the same hardware and software tools with the exception of monitoring application use.

To get an overall view of application use, in the student computer lab, a local license monitoring program was used [13]. Its primary function is to track application use (by time) and help in determining where support efforts of lab staff should be directed. We use the data collected from the monitor to measure the amount of traffic generated by various application. Application usage on the administrative server is estimated from software licenses purchased and manual tracking of installed copies.

Traffic on the network destined for the file server is captured using a *network sniffer* [31] located on the same LAN. A fast sniffer machine is used to eliminate possible packet loss. A large amount of storage space is used to capture adequate trace lengths. The sniffer operates in the *promiscuous* mode on an Ethernet network, monitoring at all network traffic and capturing only packets to and from designated machines.

Activity on the file server's disk channel is traced using a *SCSI bus analyzer* [24] attached to the SCSI bus of the file server. The SCSI analyzer captures SCSI commands and bus states. The analyzer operates in a *promiscuous* mode on the SCSI bus, capturing all commands that pass over the bus. The SCSI bus analyzer works similarly to the network sniffer used to capture network traffic.

The following sections detail these three measurement devices and discusses their limitations as they relate to our analysis.

4.1.1 License Monitor

In the student computer labs, a local license monitor program [13] is used to monitor and control application use. The license monitor reports detail use times of various applications, allowing us to accurately record which applications are in use during our trace time periods.

The license monitoring software tracks all computers located in the student computer lab. However, there are several remote computers that use the student access file server that are not tracked by the monitor software. These include computer lab staff workstations and occasionally workstations used by faculty to prepare software for student use.

The reports generated from the license monitor are in ASCII text format making them easily portable into statistical analysis programs, or an electronic spreadsheet for analysis. For a detailed description of the report format, see Appendix A.

4.1.2 Network Sniffer

The network sniffer [31] is an application that runs on an IBM PC compatible machine with an installed Ethernet network interface. The application operates by placing the network adapter into promiscuous mode [12, 29] and receives all packets transmitted over the Ethernet LAN. By filtering incoming packets, using destination and source address fields, the sniffer can capture only packets to or from a specified Ethernet address. In our analysis, we capture only packets sent to and from our target file server. The sniffer stores packets on a local disk. To save time, the entire incoming packet is stored, eliminating any additional packet decoding beyond the address fields.

Because the sniffer captures entire packets, substantial disk storage is required to save the entire trace file. Also, the machine must be faster than network workstations to ensure that we have a very low packet loss. In the sniffer machine, we use a 2 gigabyte (GB) disk and a 66 MHz Pentium processor. During initial testing we found, with this configuration, we could reliably capture packets for two hour periods (about 1 GB of data) with an average packet loss of less than 1%.

We also considered using a hardware sniffer [20] to collect network traffic. There are two major advantages of a hardware sniffer over our software sniffer: low packet loss and timer resolution. The hardware sniffer is optimized to capture packets from the network and has almost no packet loss. In a simple test, the hardware sniffer captured 100% of network traffic, the software sniffer captured only 99.5%. The hardware sniffer also has a time resolution in the nano-second range whereas the software program's range is in micro-seconds. For our analysis, less than 1% packet loss and micro-second resolution is more than adequate. Additionally, the hardware sniffer and accompanying software we tested was unable to capture data directly to a local disk. This limited trace time to what data could fit into RAM. With 4 megabytes (MB) of RAM in our sniffer machine, we were limited to trace times of about 4 seconds. A recently released version of this product allows direct disk capturing, but it was not available at the time our tests were conducted.

Sniffer output is a binary file that includes a time stamp (at the start of a packet) and packet data received from the network interface. For analysis, the data is converted from this raw format into a smaller binary format of our own design. The size of each data packet is reduced by stripping off all but the protocol information needed for analysis. This greatly reduces storage requirements, making analysis of multiple traces easier to manage.

To analyze the data, we wrote additional software to summarize the smaller binary file and complete command statistics, request size distribution, intra-arrival time distribution, and response time distribution. Using this data, we analyze the traffic on the network related to the file server in question. The results and charts in Chapter 6 are derived directly from this data.

4.1.3 SCSI Bus Analyzer

To analyze the file server's disk traffic, a hardware monitor is used. NetWare provides hooks [23] in its operating system to monitor file activity, such as file open and close requests, but it does not have the facilities to trace disk access at the block level. A replacement disk driver [22], with tracing mechanisms could capture and store all the relevant data, could be used if available. A hardware SCSI bus analyzer [24] monitoring the SCSI bus in promiscuous mode, that does not register itself on the SCSI bus, is an ideal solution. It is a passive monitoring device that captures the commands and phases of the SCSI bus (similar in operation to the promiscuous Ethernet adapter used to monitor network activity). Activity on the bus is stored to a local disk for later processing.

The SCSI bus analyzer is an adapter card for an MS-DOS based PC. The card is connected as a SCSI device on the SCSI bus of the file server being monitored. The card does not register on the SCSI bus as an additional device; it is passive. Using additional software on the monitor machine, the analyzer card captures all the commands and IO requests on the SCSI bus. Each SCSI command and phase transition is stored in a binary file with time stamp and command information. The data transferred between SCSI devices is not captured, only command SCSI information, unlike the network sniffer which captured entire packets. To prevent data loss, a fast machine is required to record the SCSI data gathered by the promiscuous interface card. The card uses an internal RAM buffer of 3 MB to store incoming data. This internal buffer allows multiple commands to be buffered before saving them to disk, allowing high traffic situations to be accurately recorded with minimal data loss. A 33 MHz Intel 80486 based machine with a 2 GB disk drive was used as the SCSI bus monitoring machine, thereby providing high speed and ample storage space for collected trace files.

The SCSI analyzer is not without problems. Although it is a passive device on the server's SCSI bus, it must be interfaced to the file server's SCSI bus correctly (proper termination). It cannot be carelessly connected and disconnected from the SCSI bus, as with any SCSI device. On more than one occasion, disconnecting the card during SCSI activity caused the server to invalidate and dismount an important volume.

A software only solution (replacement disk driver or NetWare NLM) may encounter fewer interface problems. It would, however, require the use of server memory and may incur additional overhead while servicing data requests, perhaps distorting request response times. Perhaps most important is the additional memory that would be consumed by the monitoring software. This memory would not be available to the server's file cache, reducing the amount of memory available for caching workstation requests.

Like the network sniffer, the SCSI analyzer captures information to a binary data file. To use this file, we processed it with a custom program that extracts relevant statistics and data. Fortunately, the analyzer does not capture entire SCSI data packets, only command information for each transfer is stored. We can use the data generated from our custom program to graph, chart, and analyze the activity on the SCSI bus.

4.2 Measurements Taken

With both the network sniffer and SCSI bus analyzer we wanted to capture similar statistics. In particular, we should be able to compute read/write ratios, throughput, response time, and request size for both network and disk traffic.

We need to capture the request types for computation of read/write ratios and throughput. Time stamps of requests and corresponding replies are needed to compute response time statistics. The request sizes are needed to compute request size statistics. For disk requests, we need starting block request addresses to compute request locality and to find hot-spots of disk activity.

We will use the throughput and read/write ratios to correlate network and disk activity. The measurements we take, and the software we have written, allow us to make these comparisons.

4.3 Trace Period

In selecting a trace period, we attempted to choose times when the file server was most active. In the student computer labs, the license monitoring system tracks use. The reports generated can be used to identify heavy and average use periods. In the administrative areas we select trace periods beginning in the morning, just before offices open, and ending late afternoon. To help minimize one-time fluctuations in either environment several days are traced. This ensures we have enough data to accurately describe our environment.

In addition, we have to take into consideration the disk storage limitations of the network sniffer and SCSI bus analyzer.

The network sniffer is the most limited trace device, as it stores all captured packets including their data. A series of preliminary tests allowed us to determine we could consistently trace about two hours of traffic each day before filing our available disk storage. The data file is then processed, producing a smaller binary file with only relevant trace data. The condensed trace file can then be stored off-line (on tape or on another file server) until needed for analysis.

The SCSI analyzer produced much smaller data files and could trace almost indefinitely. To help balance the traces out and make them more manageable the SCSI bus analyzer was started each day at the same time as the sniffer. After the sniffer shut down, the SCSI bus analyzer continued to collect data throughout the day.

Chapter 5

Disk Workload Analysis

In this section we analyze the data collected by the SCSI bus analyzer. We start by describing the time periods traced and the amount of data collected. The read-to-write ratio, or mix of requests is then analyzed, followed by a presentation of request size statistics. We also analyze request throughput and response time distributions. We identify *hot spots* on the disk, by examining cylinder access data. The chapter concludes with an analysis of disk drive inter-request seek distances.

Throughout this section, we find that the workloads of the two file servers are considerably different, although they have some similarities. Some of these similarities are attributed to the fact that both servers run on the same model computer and use the same model of disk drives. We also observe some areas of disk drive access patterns that could be taken advantage of.

5.1 Trace Summary

Several disk traces were conducted in each environment. We traced the heaviest periods of network, and disk, activity. For the student access file server, heavy use periods were based on data collected

Date		Time Period	Overall Requests	System Disk	Application Disk
Student File Server					
4/11/94	Monday	9:00am - 6:00pm	358,845	290,825	68,020
4/12/94	Tuesday	9:00am - 6:00pm	301,529	232,122	69,407
4/13/94	Wednesday	9:00am - 6:00pm	435,053	344,620	90,433
4/14/94	Thursday	9:00am - 6:00pm	366,299	294,367	71,932
4/15/94	Friday	9:00am - 6:00pm	175,344	133,309	42,035
Overall			1,637,070	1,295,243	341,827
Administrative File Server					
4/26/94	Tuesday	9:30am - 5:00pm	79,593	57,412	22,181
4/27/94	Wednesday	8:00am - 5:00am	126,799	83,899	42,900
4/28/94	Thursday	8:00am - 5:00pm	88,622	60,295	28,327
4/29/94	Friday	8:00am - 5:00pm	74,764	42,356	32,408
5/02/94	Monday	8:00am - 5:00pm	98,021	57,764	40,257
Overall			467,799	301,726	166,073

Table 5.1: Disk Trace Summary

by a license monitor (Appendix A). For the administrative server, we traced disk IO during normal working hours (8:00am to 5:00pm). For comparison, we extracted data for common time periods from the traces. Table 5.1 details the time periods that were extracted for analysis from each disk trace. The table includes the number of requests observed during the time period, and the number of requests directed to each of the file servers' disk drives.

Table 5.1 shows that the student file server's disk subsystem was much busier than the administrative file server's. Comparing overall activity shows this difference. On average, the student file server's disk subsystem services 3.5 requests for every one request serviced by the administrative file server.

Both servers show more activity on the *system disk* than on their *application disk*. On the student file server, 80% of requests were for the system disk and 20% for the application disk. On the administrative file server, 65% of requests were for the system disk and 35% for the application disk. We can attribute much of the activity on the system disk to two causes: print queuing and operating system access. Recall that print queue files reside on the system disk, along with the file server's own operating system files.

On Fridays (4/15 and 4/29) we observed less activity than other weekdays. Fridays are typically *slow* days, where people arrive late and leave early.

A surprising occurrence is the definite peak in activity on Wednesdays, in both environments. The student file server processed over 435,000 requests and the administrative file server over 120,000. These two Wednesdays processed 50,00 to 100,000 more requests than other days traced.

5.2 Request Mix

In this section, we look at the read/write ratio, or the request mix, of the requests captured. The requests captured on the IO bus, via the SCSI analyzer, have already passed through the file server's cache; thus, the request pattern does not necessarily follow the patterns present on the network. This data is, however, very important for optimizing the disk drives for use with network file servers. It also give us insight to how the file server cache is working.

Date	Overall			System Disk	Application Disk
	Read%	Write%	R/W	R/W	R/W
Student File Server					
4/11/94	26.59	73.41	0.3622	0.2086	1.9844
4/12/94	28.65	71.35	0.4014	0.2109	1.8009
4/13/94	24.21	75.79	0.3195	0.2433	0.7115
4/14/94	28.10	71.90	0.3909	0.2442	1.5907
4/15/94	28.29	71.71	0.3944	0.1953	1.9560
4/18/94	24.17	75.83	0.3188	0.1889	1.5541
Administrative File Server					
4/26/94	40.12	59.88	0.6700	0.5354	1.1587
4/27/94	47.64	52.36	0.9100	0.6335	1.9421
4/28/94	58.64	41.36	1.4180	0.6967	3.1256
4/29/94	44.10	55.90	0.7890	0.5291	1.3256
5/02/94	48.77	51.23	0.9520	0.6189	1.8733
Overall	49.84	50.16	0.9935	0.6153	2.0607

Table 5.2: Disk Request Mix

All requests captured on the IO bus of both file servers are either read or write requests. In none of the traces were any other request types encountered. All traces started after the file server was powered on, and were terminated before it was shut down, reducing the possibility of encountering other requests, such as startup or shutdown requests. Table 5.2 summarizes the mix of request types for each trace. Figures 5.1 through 5.4 show, graphically, fluctuations in the read/write ratio throughout each day.

On the student server there were more write requests than read requests. Seventy-six percent of all requests were write data, 24.17% were read data, giving an overall read/write ratio of 1:3. The administrative file server was quite different, with 50.16% write and 49.84% read requests, resulting in a 1:1 read/write ratio. We believe these read/write ratios are more write intensive than typical workstation workloads. Others have found that a high write/read ratio is common when using a large read cache [19, 2]. With NetWare, there is a large cache, as described in Section 3.3.1.

To explain the difference in read/write ratios, we consider how each server is used. Students store data on floppy diskettes, and are only allowed temporary and intermediate storage on the file server. Administrative users store more data on the file server, both intermediate and long term files.

The large percentage of writes on the student file server reflects the large number of temporary files being created, cached, and written to disk. When the data is needed, it is still present in the cache, and is not read from disk. The creation and deletion of files does however, require updating of the file system's directory. These updates require write requests to the directory, on disk. Applications, when needed, are read from disk, and cached. Their frequent use keeps them current in the cache, reducing the need to read them from disk; thereby reducing the number of read misses and disk read requests.

The 1:1 read/write ratio observed on the administrative file server is partially explained by the file server's large cache. Furthermore, administrative users have the ability to store files long term on the file server. Thus, in addition to accessing applications that become resident in the cache, users

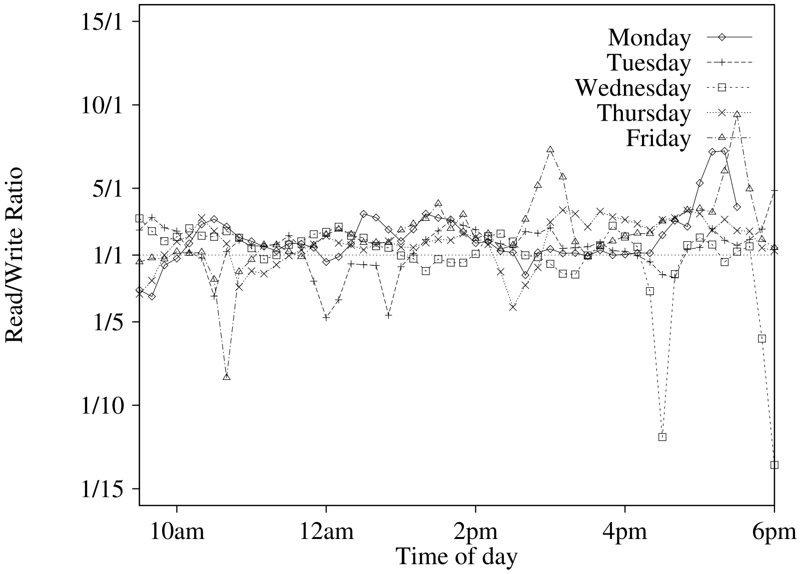

Figure 5.1: Read/Write Ratio (Student, Application Disk)

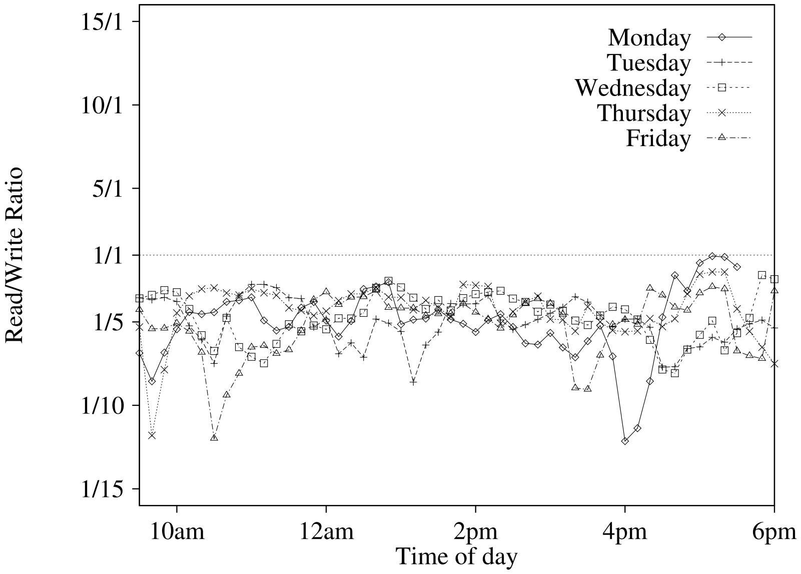

Figure 5.2: Read/Write Ratio (Student, System Disk)

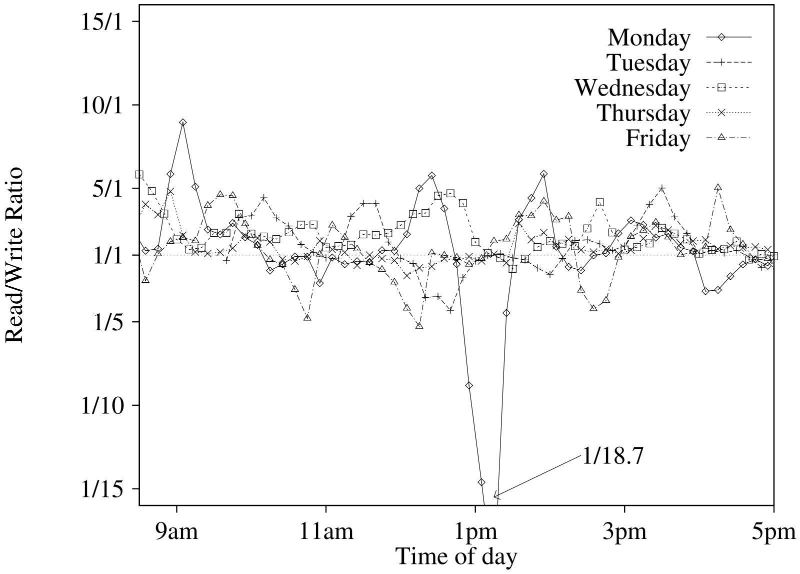

Figure 5.3: Read/Write Ratio (Administrative, Application Disk)

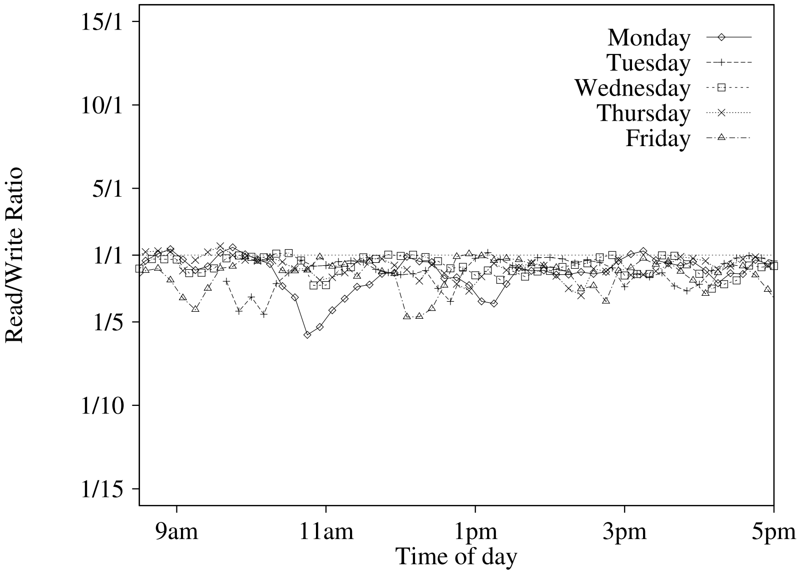

Figure 5.4: Read/Write Ratio (Administrative, System Disk)

access infrequently used, pre-existing files. These infrequently used files are purged from the cache, after a period of inactivity. When they are used again, perhaps hours or days later, they need to be read from disk, causing read requests on the disk subsystem.

On both servers, the system disk had a low read/write ratio, less than 2:3. This is expected, as print queue and most temporary files are stored on the system disk. Most of the activity on the application disk consists of loading applications and reading data. The ratio was closer to 1:1 on the administrative file server, as users have longer work sessions. Temporary files have a longer lifetime, and require less activity in this environment. Certainly, printing activity is much heavier in the student environment, which incurs additional writes to the system disk.

Both of the servers traced have high write/read ratios, which are different from what we expect for workstations or time-sharing systems. The large cache used by the file servers is partially responsible for this. The environment in which each server is used also affects this ratio; students continually create and delete temporary files and administration manipulates existing ones.

5.3 Request Size

In both environments, the file server's disk blocking size was set to 4 kilobytes (KB), equal to the size of the file server's *cache line*. The disk drives in both servers use a physical block size of 512 bytes. With these settings we expect read disk IO requests to be for multiples of 8 disk blocks (4 KB / 512 bytes per block = 8 blocks.) For write requests, we expect the file server to optimize disk communication and write data in 512 byte blocks or larger.

5.3.1 Read Requests

Examining read requests in the traces, summarized in Table 5.3, we observe that all reads are multiples of 8 disk blocks. It is curious that on the administrative file server *all* reads were for 8 blocks.

Date	Overall		System Disk		Application Disk	
	Mean	Std. Dev.	Mean	Std. Dev.	Mean	Std. Dev.
Student File Server						
4/11/94	18.17	5.64	21.06	6.03	14.97	2.68
4/12/94	19.97	4.71	23.04	3.56	17.21	3.83
4/13/94	22.97	3.48	23.28	3.11	22.42	3.99
4/14/94	22.89	3.76	22.74	3.97	23.11	3.42
4/15/94	19.68	4.75	22.89	3.80	17.16	3.80
Overall	20.97	4.89	22.60	4.29	18.98	4.83
Administrative File Server						
4/26/94	8.00	0.00	8.00	0.00	8.00	0.00
4/27/94	8.00	0.00	8.00	0.00	8.00	0.00
4/28/94	8.00	0.00	8.00	0.00	8.00	0.00
4/29/94	8.00	0.00	8.00	0.00	8.00	0.00
5/02/94	8.00	0.00	8.00	0.00	8.00	0.00
Overall	8.00	0.00	8.00	0.00	8.00	0.00

Table 5.3: Read Request Size (in 512 byte blocks)

There is absolutely no variation. We are still unsure as to what causes this peculiar operation of the administrative file server. We have found no definitive explanation why this happens. It is possible that a slightly different disk driver was being used on the administrative file server, causing it to only read 8 blocks of data. The administrative file server does have Macintosh network services loaded, but there is no indication that this would limit the size of disk accesses. The student file server has reads for 8, 16, and 24 blocks, but nothing larger.

The pattern of read request sizes varies on the student file server as Figures 5.5 and 5.6 show. Overall, more than 50% of read requests on the student server were for 24 blocks. The 50-percentile for number of blocks read on the student server was much larger than the 8 blocks on the administrative server, 16 blocks on the application disk and 24 blocks on the system disk. In addition, the request size mode on the student file servers' disks was also 16 and 24 blocks.

These figures suggest the student server's cache is either configured for a larger read ahead, or that the administrative server had an overly restrictive upper limit set on read ahead size. It is possible, but we feel it is unlikely, that the administrative server does not need larger request sizes for some other, unknown reason.

If there is a configuration parameter that limits request sizes on the administrative file server to 8 blocks, it is possible we are also limiting the size of requests on the student file server to 24 blocks. It would be interesting to see if the student file server would request 32, 40, 48, or larger sizes with this limit removed.

5.3.2 Write Requests

Examining write requests in the traces, summarized in Table 5.4, we see very small request sizes. Most write requests, on both file servers, are for a single block. In terms of performance, this makes sense. If only a single 512 byte block has changed the file server should not write an entire cache line (4 KB) of data to disk. In some cases, if the same block will be written again, it should not be written to disk at all.

The mean write request sizes were small compared to the mean read request size. On the student file server, the application disk had an overall mean request size of 3.70 blocks. The system disk's overall mean request size was 2.17 blocks. The administrative file server had even smaller mean request sizes, 2.17 blocks on the application disk and 1.60 blocks on the system disk.

Figures 5.7 through 5.10 show histograms of write request sizes, in blocks. The most common write request was for a single block, on all disks analyzed. In most cases, requests for a single block write account for more than 60% of all write requests.

What is curious is the large number of write requests for 8 blocks. The smallest number of 8 block writes was on the system disk of the administrative file server, Figure 5.10, which accounts for

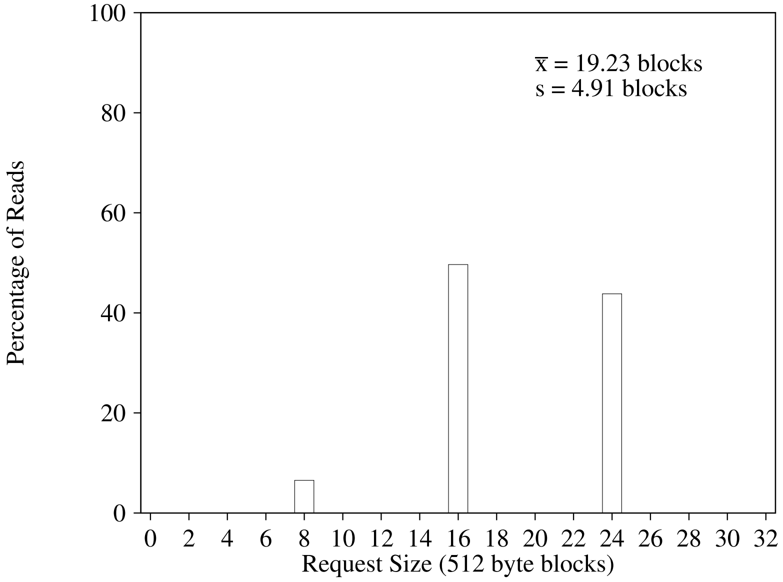

Figure 5.5: Read Request Size (Student, Application Disk)

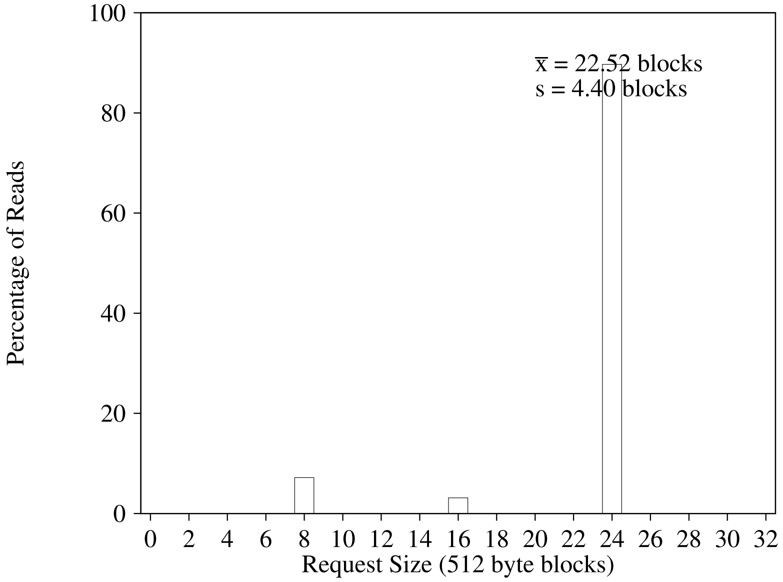

Figure 5.6: Read Request Size (Student, System Disk)

Date	Overall		System Disk		Application Disk	
	Mean	Std. Dev.	Mean	Std. Dev.	Mean	Std. Dev.
Student File Server						
4/11/94	3.04	3.06	2.99	3.08	3.59	2.77
4/12/94	3.00	3.01	2.92	3.04	3.55	2.78
4/13/94	3.11	3.10	2.98	3.08	3.81	3.16
4/14/94	2.81	2.91	2.68	2.89	3.80	2.97
4/15/94	2.61	2.78	2.50	2.76	3.51	2.80
Overall	2.96	3.01	2.86	3.00	3.70	2.96
Administrative File Server						
4/26/94	1.72	1.81	1.64	1.65	2.03	2.30
4/27/94	1.70	1.81	1.64	1.71	1.89	2.13
4/28/94	1.68	1.75	1.58	1.57	1.96	2.21
4/29/94	1.88	2.07	1.58	1.56	2.48	2.71
5/02/94	1.80	1.93	1.56	1.50	2.42	2.64
Overall	1.74	1.87	1.60	1.60	2.17	2.44

Table 5.4: Write Request Size (in 512 byte blocks)

4.4% of all write requests. The largest number of 8 block writes was on the student file server's application disk, Figure 5.7, accounting for 28.74%. The other disks on both servers fall between these two measures, as can be seen in Figures 5.8 and 5.9. This activity is a large factor, contributing to the variation of mean request sizes between the two file servers. The administrative file server had fewer writes for 8 blocks. On the student file server, however, 8 block requests account for at least 24% of all write requests (on both disks).

5.3.3 All Requests

Table 5.5 gives request size statistics, including both read and write requests. On the student file server, where writing traffic dominates, the mean request sizes were 6.51 and 12.54 blocks for the system and application disks, respectively. The administrative file server, where the read and write requests are about equal in number, the mean request sizes were 4.02 and 5.73 blocks.

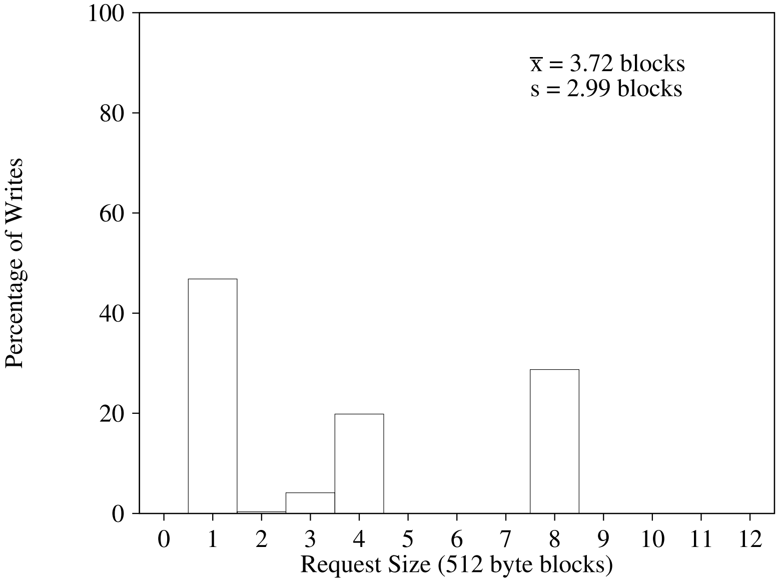

Figure 5.7: Write Request Size (Student, Application Disk)

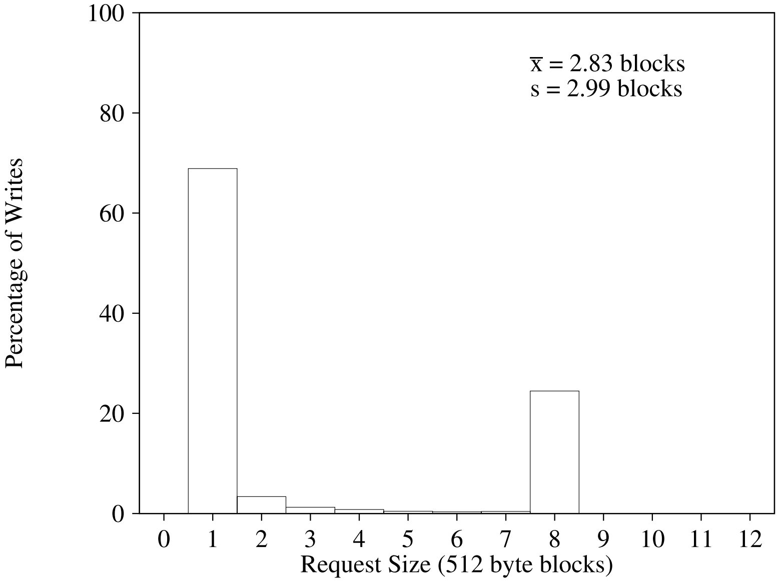

Figure 5.8: Write Request Size (Student, System Disk)

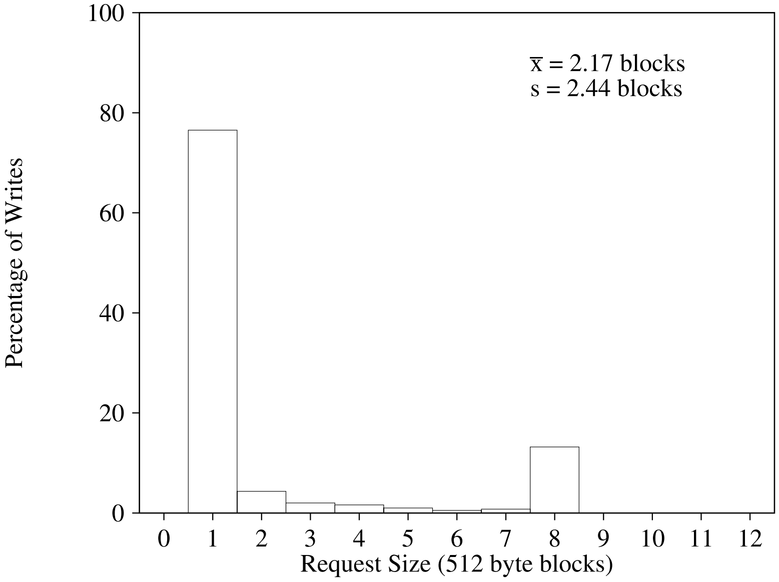

Figure 5.9: Write Request Size (Administrative, Application Disk)

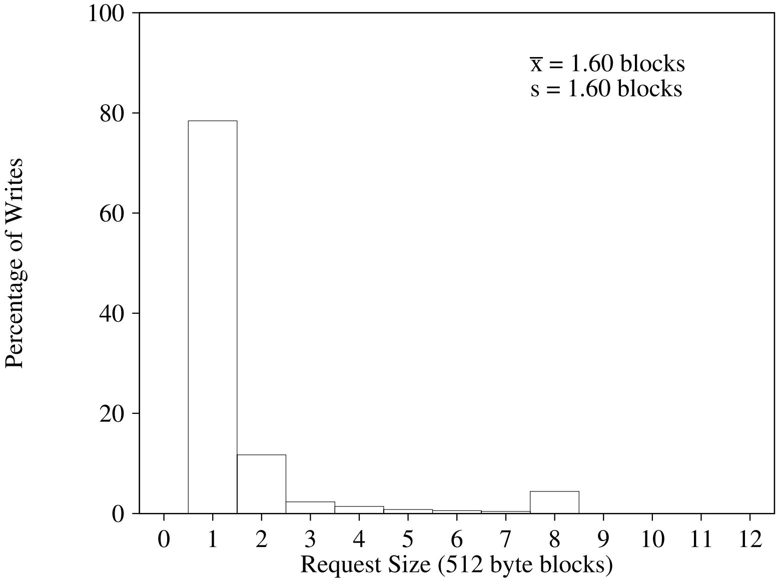

Figure 5.10: Write Request Size (Administrative, System Disk)

Date	Overall		System Disk		Application Disk	
	Mean	Std. Dev.	Mean	Std. Dev.	Mean	Std. Dev.
Student File Server						
4/11/94	7.06	7.75	6.10	7.80	11.16	6.02
4/12/94	7.64	8.36	6.30	8.14	12.12	7.46
4/13/94	7.93	9.09	6.94	8.61	11.68	9.85
4/14/94	8.54	9.61	6.88	8.74	15.35	10.01
4/15/94	7.44	8.43	5.83	8.10	12.54	7.35
Overall	7.77	8.75	6.51	8.34	12.54	8.61
Administrative File Server						
4/26/94	4.25	3.38	3.85	3.31	5.28	3.35
4/27/94	4.75	3.41	4.12	3.38	5.97	3.13
4/28/94	4.56	3.41	4.12	3.37	5.48	3.30
4/29/94	4.61	3.41	3.83	3.31	5.63	3.26
5/02/94	4.83	3.39	3.98	3.34	6.06	3.08
Overall	4.63	3.31	4.02	3.35	5.73	3.22

Table 5.5: Overall Disk Request Size (in 512 byte blocks)

The mode, most common request size, on the student file server is one block. Almost 50% of all requests were for only a single 512-byte block. On the administrative file server, the mode was 8 blocks, with a smaller, local peak of one block. Again, read requests are always 8 blocks here, and writes are mostly one block.

It is equally important to note the large deviation in request sizes on the student file server, 8.34 and 8.61 blocks, as compared to the 3.35 and 3.22 block deviation on the administrative file server. For each request, the administrative file server reads and writes a much smaller numbers of blocks.

5.4 Throughput and Arrival Time Measurements

In this section we examine the throughput of the disk subsystems on the two file servers. Because throughput is closely related to inter-arrival time, this analysis can be used in place of arrival

Date	Overall		System Disk		Application Disk	
	Requests	Kilobytes	Requests	Kilobytes	Requests	Kilobytes
Student File Server						
4/11/94	11.9	42.0	9.6	29.4	2.3	12.6
4/12/94	9.3	35.6	7.2	22.6	2.1	13.0
4/13/94	13.4	53.2	10.6	37.0	2.8	16.3
4/14/94	11.3	48.3	9.1	31.3	2.2	17.0
4/15/94	5.5	20.4	4.1	12.0	1.3	8.1
Mean	10.3	39.9	8.1	26.4	2.1	13.4
Administrative File Server						
4/26/94	3.9	9.3	2.6	5.3	1.3	4.0
4/27/94	2.7	6.2	1.9	3.8	0.9	2.9
4/28/94	2.3	5.3	1.3	2.5	1.0	2.8
4/29/94	3.0	7.3	1.8	3.5	1.2	3.8
5/02/94	2.7	5.8	1.9	3.7	0.8	2.1
Mean	2.9	6.8	1.9	3.8	1.0	3.1

Table 5.6: Daily and Mean Disk Throughput (Requests and Kilobytes per Second)

time characteristics. The throughput reflects the number of requests, per unit of time that the disk services. The inter-arrival time measurements reflect the time elapsed between the beginning of successive requests. By examining peak, or high, use periods we characterize our systems under heavy load. Heavy load periods are important to characterize, and optimize, because typically this is where response time of the disks suffers.

Table 5.6 shows the daily and throughput for both file servers traced. An initial observation is the larger number of requests and data transferred on the student file server. The student file server handled 10.3 disk requests per second, more than three times as many requests as the administrative file server which processed 2.9 disk requests per second. In KB per second, the student file server handled 39.9 KB per second, almost six times as much data as the 6.8 KB per second transferred on the administrative file server.

Examining the system disks reveals similar ratios. The student file server processed 8.1 requests per second, 4.3 times as many requests as the administrative file server which processed 1.9 requests per second. There were 6.9 times as many KB transferred each second on the student file server, 26.4 KB, as there were on the administrative file server, 3.8 KB.

The application disk has smaller throughput ratios. There were 2.1 requests per second on the student application disk per second, and 1 request per second on the administrative application disk. The student file server's application disk serviced 13.4 KB per second, whereas the administrative file server's application disk serviced 3.1 KB per second.

With the exception of Friday, 4/14, the number of requests and kilobytes transferred per second, did not vary greatly from one day to the next, within each environment. Although the deviation is not computed, Table 5.6 shows the small differences in the trace statistics.

Figures 5.11 through 5.14 show daily trace throughput graphically. Data points are plotted at ten minute intervals, along the X-axis, labelled *Time of day*. Each data point corresponds to the throughput, in IO requests per second, for the previous 30 minutes. For comparison, all the traces for a particular environment and disk device, are plotted on the same graph, with a different line and point style.

These figures allow us to identify several peaks in disk activity. Most of these peaks occur on the student file server. Only a few small peaks occur on the administrative file server and even then they are, relative to the student file server, very small. With the exception of the administrative file server's application disk, all the peaks occur in the afternoon, mostly between 2:00pm and 6:00pm.

5.5 Response Time Measurements

In this section we examine the response time distribution of the disk drives. The response time measurements reflect the time elapsed from the beginning of a request (arrival time) to its completion time. Response times are dependent on the disk drive type as well as the workload.

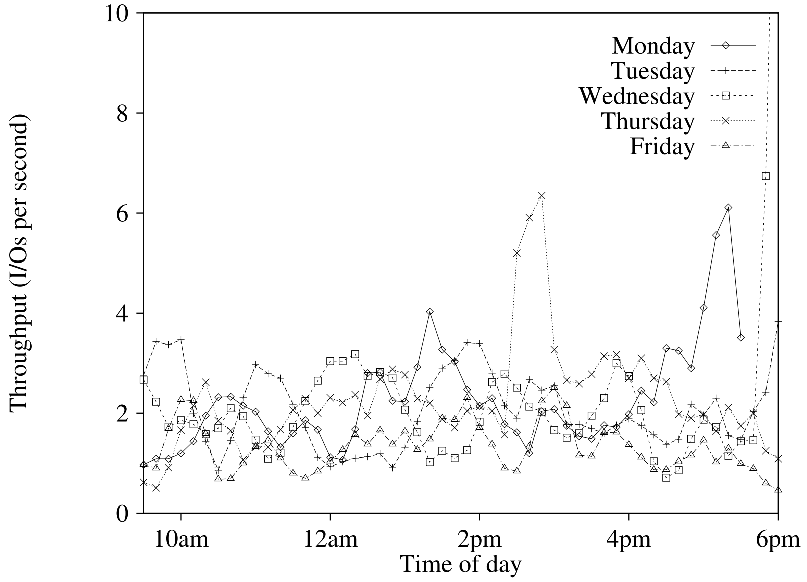

Figure 5.11: Daily Throughput (Student, Application Disk)

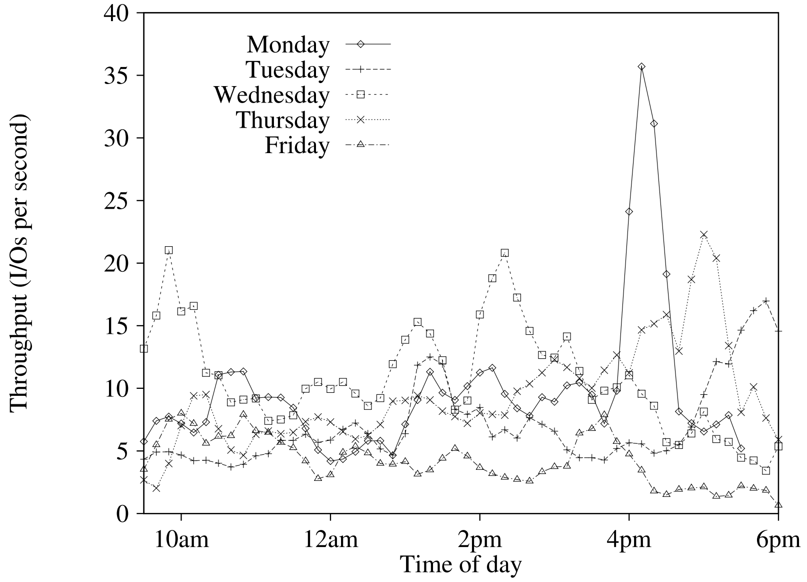

Figure 5.12: Daily Throughput (Student, System Disk)

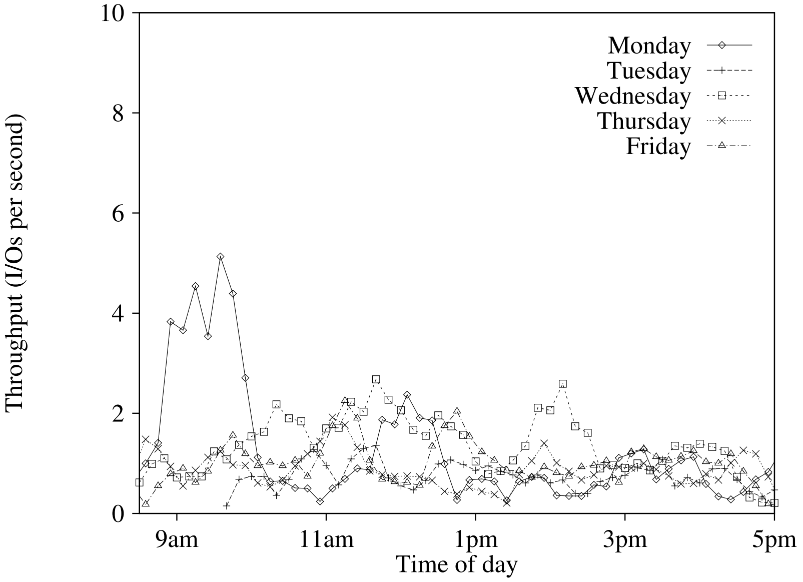

Figure 5.13: Daily Throughput (Administrative, Application Disk)

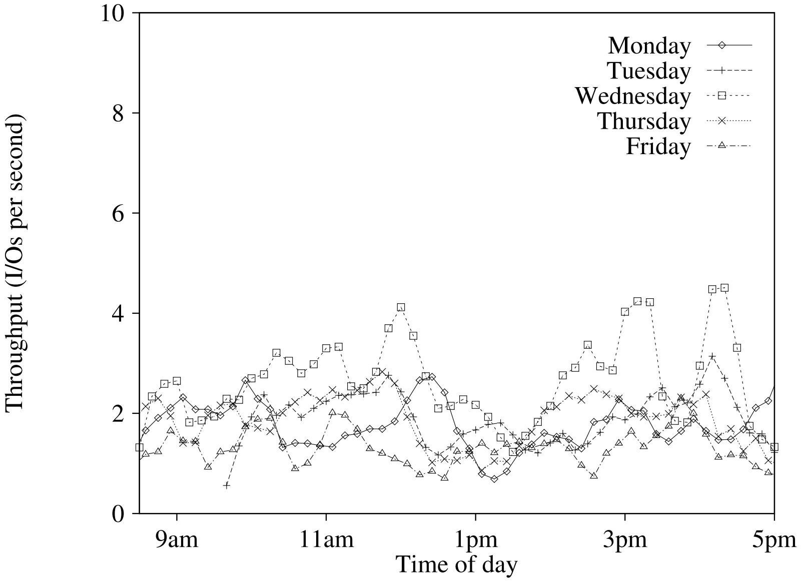

Figure 5.14: Daily Throughput (Administrative, System Disk)

Date	Overall		System Disk		Application Disk	
	Mean	Std. Dev.	Mean	Std. Dev.	Mean	Std. Dev.
Student File Server						
4/11/94	17.68	11.50	21.96	10.94	12.93	10.17
4/12/94	17.46	11.73	22.61	11.10	12.86	10.28
4/13/94	20.60	11.39	23.33	10.90	15.40	10.63
4/14/94	19.44	11.47	22.15	11.41	15.56	10.40
4/15/94	16.70	11.22	20.91	11.04	12.64	9.80
Overall	18.56	11.59	22.38	11.14	13.95	10.38
Administrative File Server						
4/26/94	14.38	9.83	15.22	9.83	13.01	9.67
4/27/94	13.91	10.54	15.60	10.05	11.99	9.70
4/28/94	14.60	9.92	15.81	9.90	12.86	9.67
4/29/94	13.08	9.77	15.77	9.78	10.92	9.21
5/02/94	13.59	10.17	15.11	10.19	12.33	9.97
Overall	13.91	9.99	15.48	9.98	12.17	9.71

Table 5.7: Read Response Time (ms)

5.5.1 Read Requests

Table 5.7 details the response time statistics for both file servers. The system disk and application disk response times are listed separately. The *overall* column lists the read request response time of the entire disk subsystem as a single entity. We use this column in a later chapter to compare disk subsystem characteristics to network activity.

The student file server's system disk had a mean response time, for read requests, of 22.38 ms. This is almost double the manufacturers average service time specification [14] of 12 ms (for all IOs). All of the other disks traced had mean response times that are closer to the manufacturers specifications. The student file server's application disk had a mean response time of 13.95 ms, 0.55 ms faster than the 14.5 ms specification [9]. The administrative file server's disk had mean response times of 15.48 ms and 12.17 ms for system and application disks respectively.

The standard deviations of all the means were within a 2 ms range, from 9.71 ms to 11.73 ms. The largest being the student file server's system disk and the smallest the administrative file server's application disk.

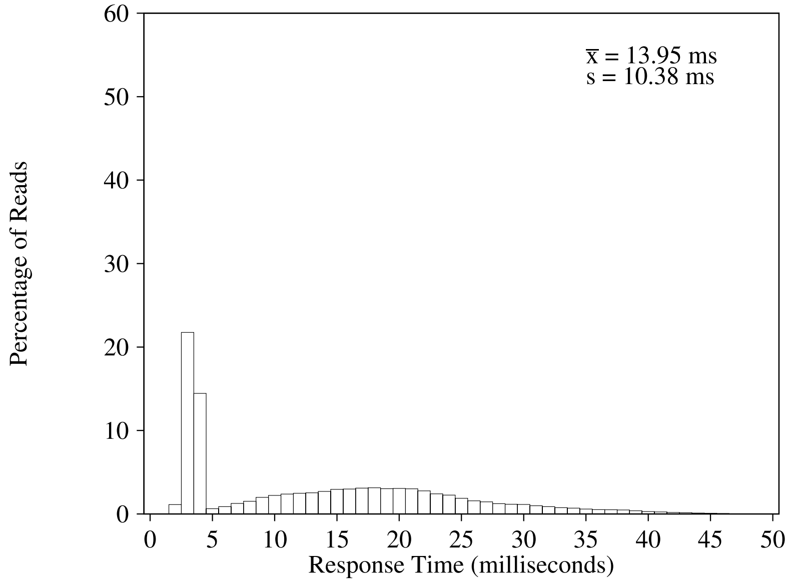

Figure 5.15: Read Response Time (Student, Application Disk)

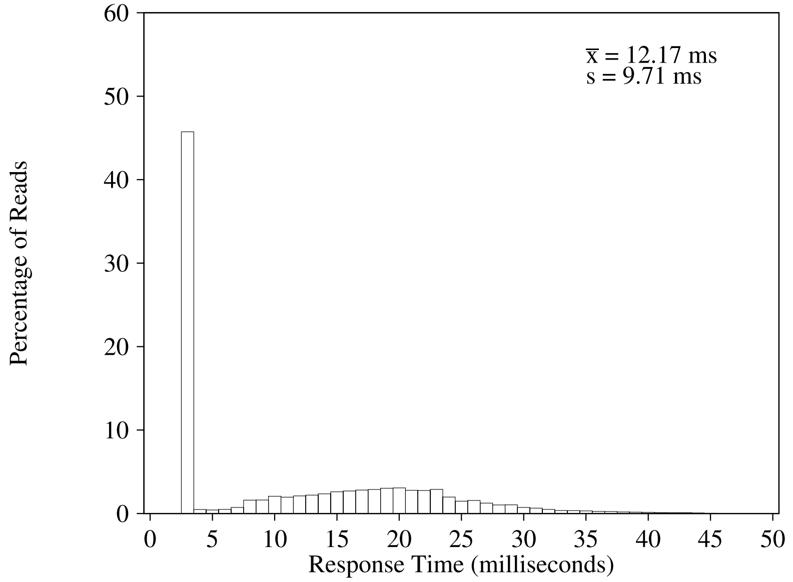

Figure 5.16: Read Response Time (Administrative, Application Disk)

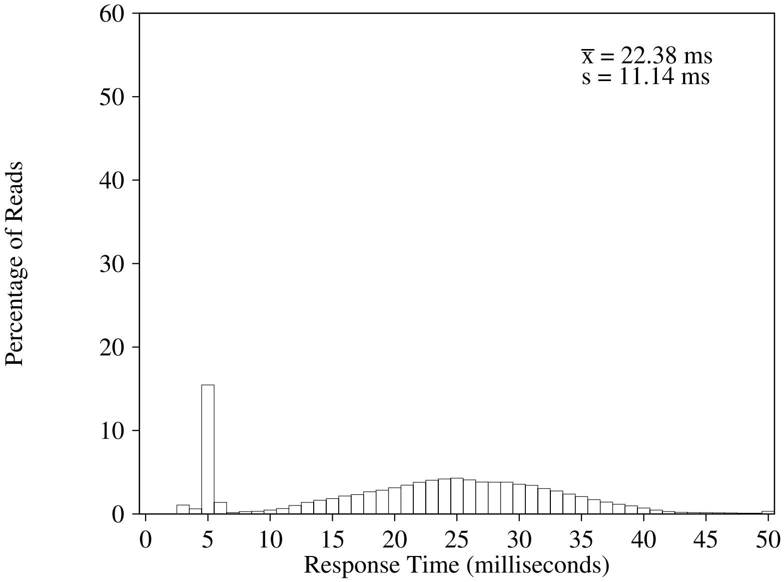

Figure 5.17: Read Response Time (Student, System Disk)

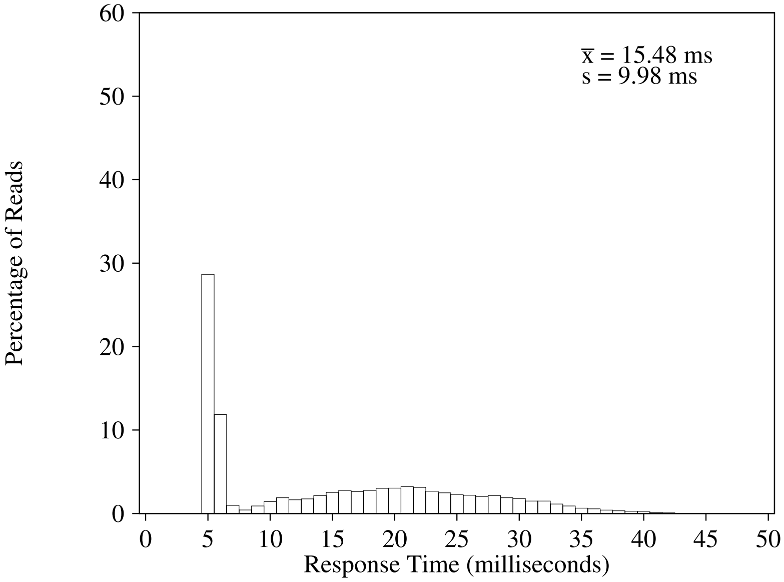

Figure 5.18: Read Response Time (Administrative, System Disk)

Figures 5.15 through 5.18 show histograms of read request response times. There were a large number of very fast, less than 6 ms, responses. These requests were most likely disk controller cache hits or requests that did not require any seek. Of all the disks measured, the administrative application disk had the highest hit/no-seek rate, at approximately 45%. Examining response times greater than 6 ms, the histograms resemble bell-shaped distributions, slightly skewed.

5.5.2 Write Requests

Table 5.8 shows write request response statistics, organized similarly to Table 5.7. The daily mean write response times are much closer to each other, and to the manufacturers specified average response time. The student file server had write request response times of 15.27 ms and 15.83 ms for system and application disks. The administrative file server had mean response times of 18.34 ms and 15.92 ms, respectively. The standard deviation of response time for the student file servers disks were 5.49 ms and 6.97 ms, for the system disk and application disk. The administrative file server had standard deviations of 6.89 ms and 7.08 ms, respectively.

Date	Overall		System Disk		Application Disk	
	Mean	Std. Dev.	Mean	Std. Dev.	Mean	Std. Dev.
Student File Server						
4/11/94	14.91	5.66	14.83	5.47	15.83	7.32
4/12/94	15.59	5.53	15.54	5.35	15.96	6.71
4/13/94	15.40	5.59	15.45	5.41	15.13	6.44
4/14/94	15.51	5.90	15.38	5.69	16.56	7.26
4/15/94	15.25	5.68	15.06	5.49	16.65	6.81
Overall	15.34	5.69	15.27	5.49	15.83	6.97
Administrative File Server						
4/26/94	17.87	6.86	18.34	6.78	16.13	6.86
4/27/94	17.63	7.07	18.11	6.80	15.92	7.71
4/28/94	17.73	7.17	18.24	7.19	16.15	6.89
4/29/94	17.42	7.01	18.36	6.86	15.56	6.91
5/02/94	17.92	6.93	18.68	6.80	15.96	6.89
Overall	17.72	7.02	18.34	6.89	15.92	7.08

Table 5.8: Write Response Time (ms)

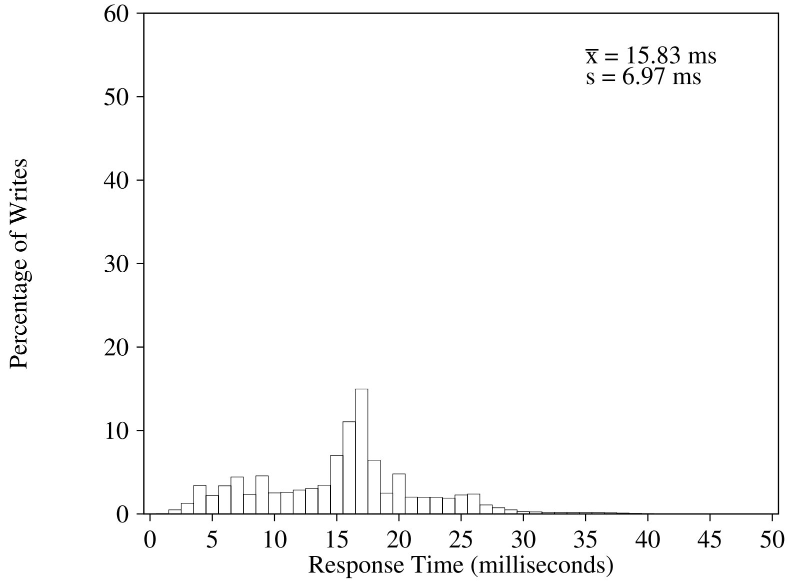

Figure 5.19: Write Response Time (Student, Application Disk)

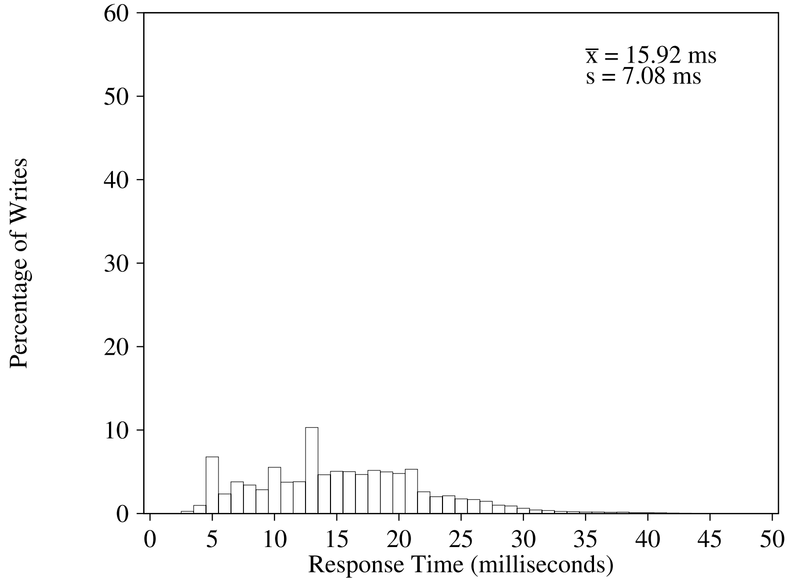

Figure 5.20: Write Response Time (Administrative, Application Disk)

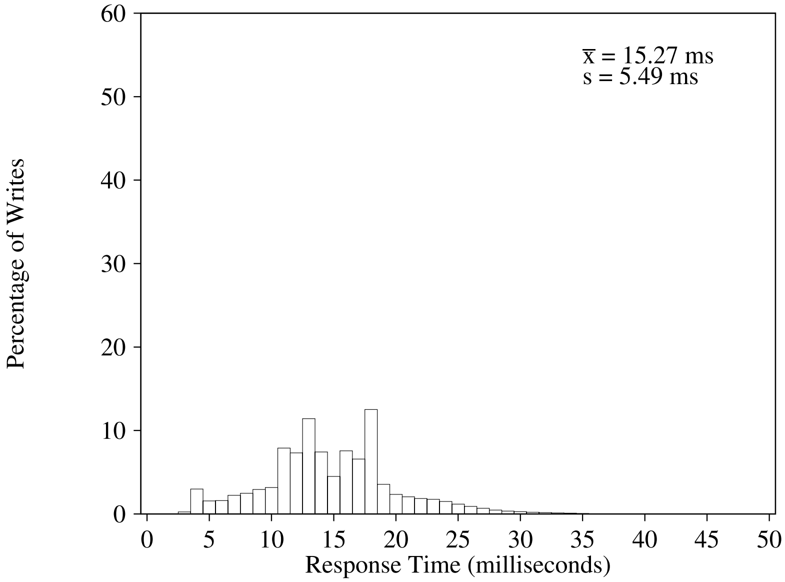

Figure 5.21: Write Response Time (Student, System Disk)

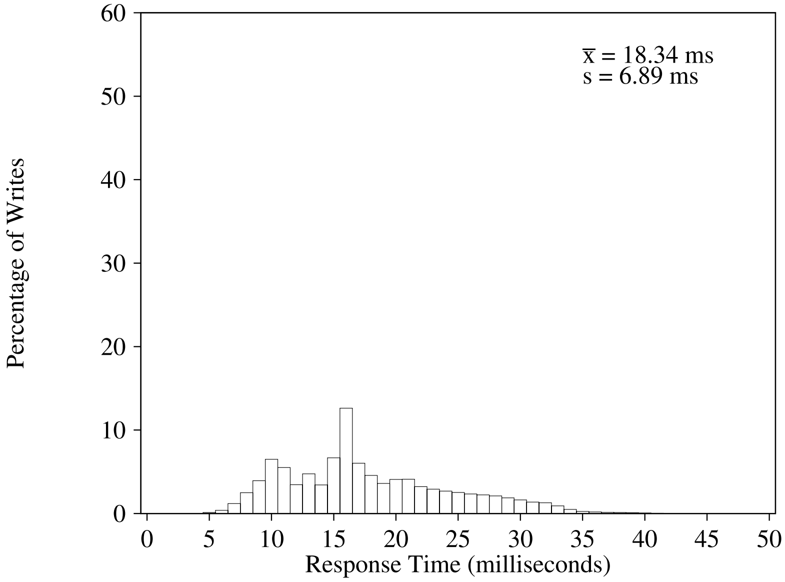

Figure 5.22: Write Response Time (Administrative, System Disk)

Figures 5.19 through 5.22 show write request response time histograms, similar to those shown for read request response time. Write response time histograms do not have a single, large spike below 6 ms as observed in the read request response time histograms. Furthermore, they suggest less pronounced bimodal distribution. This is most apparent on the student system disk, Figure 5.21. Other than this bimodal aspect, the histograms look somewhat normal, again, similar to the read response time histograms discussed earlier.

5.5.3 All Requests

As expected, overall response time measurements are almost the same on both servers (for similar disk drives). Both servers had the same configuration of disk drives, and service the same types of data (system data and application data). Differences in response time are primarily attributable to the different workloads the drives are subject to, different revisions of the disk controllers, and file server software revisions.

Date	Overall		System Disk		Application Disk	
	Mean	Std. Dev.	Mean	Std. Dev.	Mean	Std. Dev.
Student File Server						
4/11/94	15.65	7.76	16.06	7.25	13.90	9.41
4/12/94	16.10	7.78	16.72	7.18	14.02	9.23
4/13/94	16.65	7.75	16.98	7.52	15.41	8.47
4/14/94	16.63	8.10	16.79	7.77	15.96	9.28
4/15/94	15.62	7.55	16.02	7.06	14.20	8.97
Overall	16.20	7.85	16.59	7.43	14.74	9.15
Administrative File Server						
4/26/94	16.47	8.36	17.25	8.12	14.43	8.65
4/27/94	15.83	8.84	17.13	8.32	13.29	9.27
4/28/94	16.30	8.67	17.27	8.45	14.23	8.77
4/29/94	15.49	8.63	17.46	8.10	12.91	8.61
5/02/94	15.80	8.93	17.34	8.14	13.59	9.18
Overall	15.94	8.73	17.26	8.31	13.63	8.97

Table 5.9: Overall Response Time (ms)

Table 5.9, shows the student file server's system disk had about 1 ms faster overall mean response time, 16.59 ms, than the administrative file server's system disk, 17.26 ms. Neither system disk meets the average response time suggested by the manufacturer [14] of 12 ms.

The administrative file server's application disk had about 1 ms faster overall mean response time, 13.63 ms, than the student file server's, 14.74 ms. The administrative file server's application disk exceeded the manufacturers suggested average response time [9] of 14.5 ms. The student file server had only 0.24 ms slower than the specified average.

With both disks response times combined (the *overall* column in Table 5.9), the administrative file server's disk subsystem had slightly better response time with a mean of 15.9 ms. Comparatively, the student file server's disk subsystem had a mean response time of 16.2 ms.

Again, the overall response time distributions (Figures 5.23 through 5.26) are reminiscent of normal distributions. As with the write response histograms the distributions are slightly bimodal, reflecting use of the disk drive's internal cache to service requests.

Even though both servers use the same models of disk drives, their response time statistics are slightly different. Their mean and standard deviations are very similar. The differences that do exist are caused by the different workloads presented to the file servers. As our histograms show, the distributions are all similar, but with different modes.

5.6 Disk Hot Spots

In this section, we examine cylinder access patterns for each disk in our traces. Most importantly, we identify frequently accessed areas, called *hot spots*, on the disks analyzed. In fact, as we will see, some of these hot spots are accessed so frequently they accounted for 20 to 30 percent of all accesses to the disk.

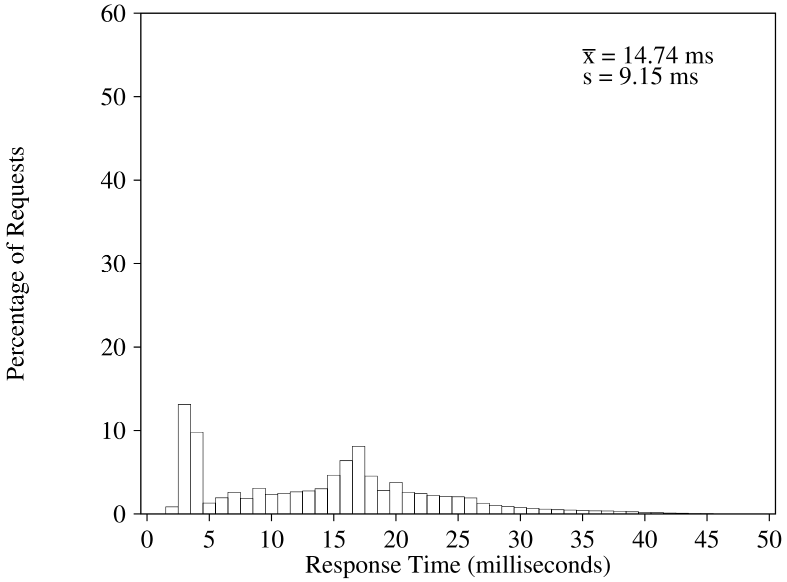

Figure 5.23: Overall Response Time (Student, Application Disk)

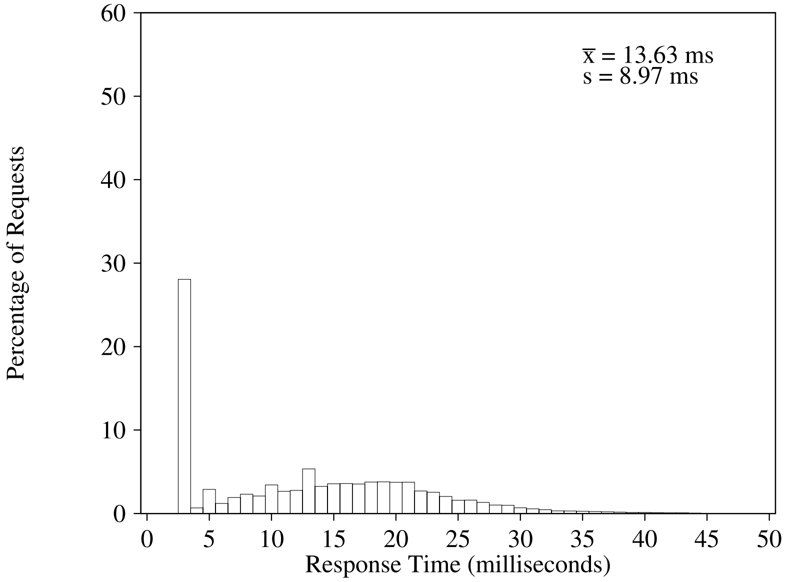

Figure 5.24: Overall Response Time (Administrative, Application Disk)

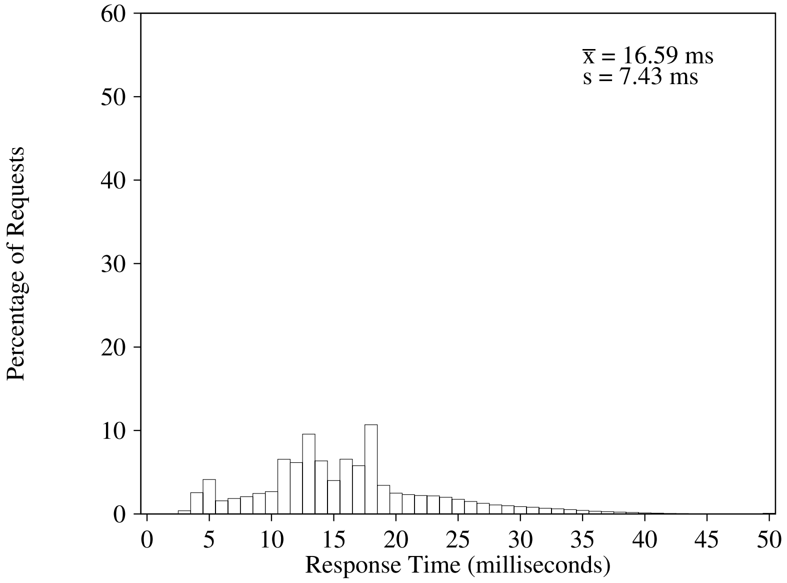

Figure 5.25: Overall Response Time (Student, System Disk)

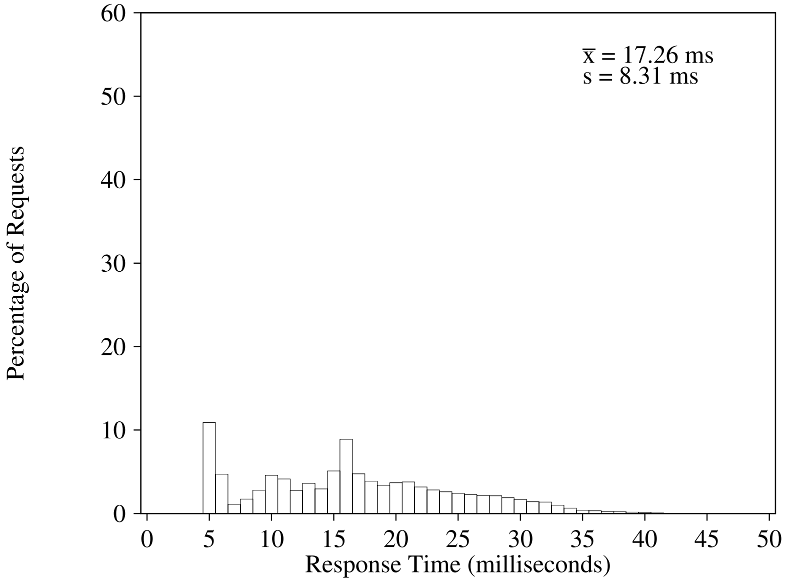

Figure 5.26: Overall Response Time (Administrative, System Disk)

5.6.1 Read Requests

On both the student and administrative file server, read requests span nearly the entire system and application disks. Table 5.10 details the ten most requested cylinders for reads. Only a few cylinders could be considered read hot spots, consisting of more than 2% of a disk accesses. The administrative file server's application disk had the most of these, with two cylinders reaching more than 5% of all reads for the disk.

Figures 5.27 and 5.28 show the read distribution, by cylinder, for the disks on the student file server. The predominant features being the peaks in read activity near cylinders 223, 1165, and 84 on the

System Disk			Application Disk		
Cylinder	% reads	cumulative %	Cylinder	% reads	cumulative %
Student File Server					
642	2.56	2.56	223	3.76	3.76
978	2.22	4.78	209	2.73	6.49
638	2.01	6.79	1165	2.33	8.82
630	1.83	8.62	84	2.32	11.14
977	1.61	10.23	1136	2.26	13.40
971	1.58	11.81	53	2.20	15.60
685	1.52	13.33	52	2.03	17.63
970	1.40	14.73	1135	2.00	19.63
536	1.31	16.04	204	1.74	21.37
981	1.24	17.28	1137	1.59	22.96
Administrative File Server					
725	2.76	2.76	834	5.80	5.80
781	2.25	5.01	686	5.17	10.97
721	1.78	6.79	687	4.61	15.58
595	1.77	8.56	836	3.93	19.51
813	1.52	10.08	835	3.89	23.40
817	1.32	11.40	837	3.64	27.04
728	1.29	12.69	832	2.68	29.72
796	1.27	13.96	35	2.07	31.79
815	1.15	15.11	84	2.05	33.84
816	1.13	16.24	838	2.00	35.84

Table 5.10: Top 10 Read Cylinders

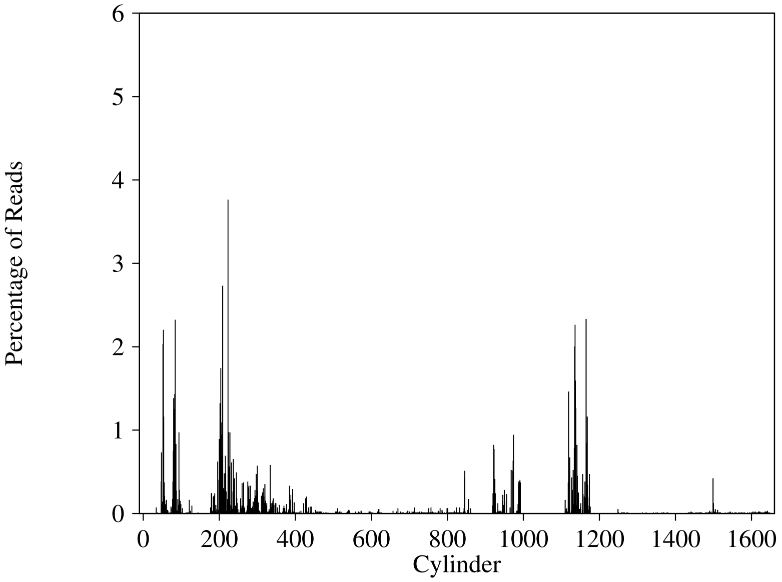

Figure 5.27: Read Access Distribution (Student, Application Disk)

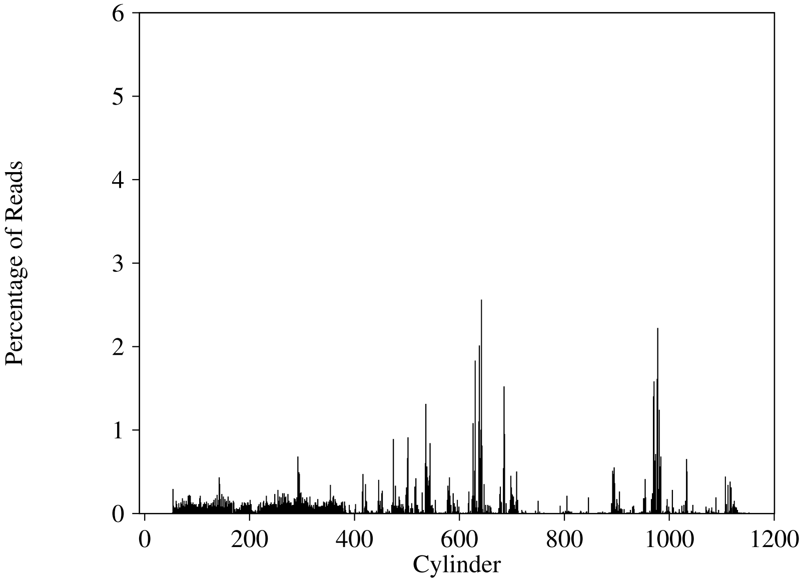

Figure 5.28: Read Access Distribution (Student, System Disk)

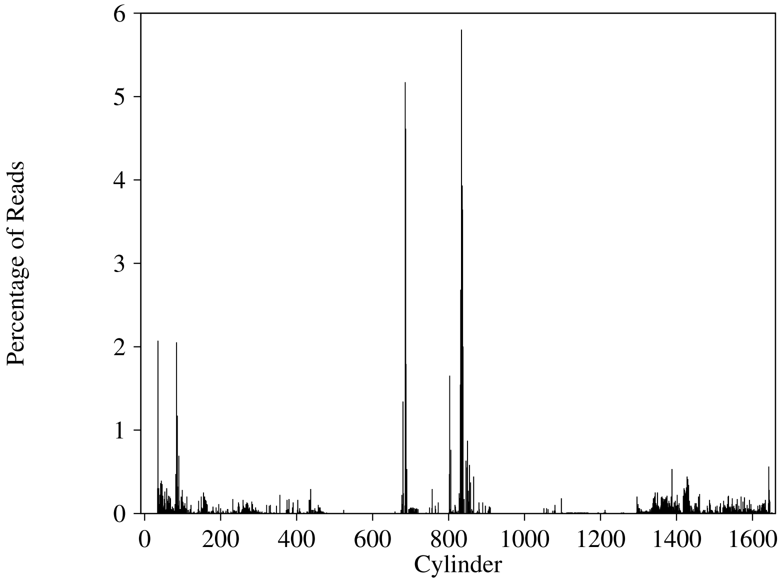

Figure 5.29: Read Access Distribution (Administrative, Application Disk)

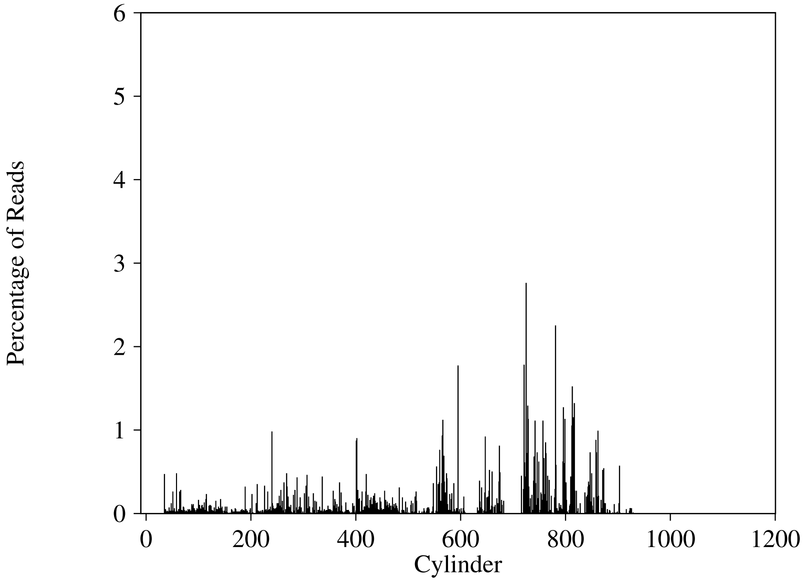

Figure 5.30: Read Access Distribution (Administrative, System Disk)

application disk, and cylinders 642 and 978 on the system disk. None of these peaks account for more than 4% of all read requests on the disk. Table 5.10 shows 17% of all read requests on the student file server's system disk were for the top ten cylinders. The ten most requested cylinders on the student file server's application disk account for 23% of all read requests.

The administrative file server's disks had a similar pattern, as can be seen in Figures 5.29 and 5.30. The peaks are in different areas of the disks, but again, no cylinder accounts for more than 6% of read requests. The most accessed cylinders on the administrative application disk were near cylinders 834, and 686. The system disk had hot spots near cylinders 725, 781, and 595 which are closer together than hot spots on other disks. Table 5.10 shows that the top ten accessed cylinders accounted for 16% of read requests on administrative file server's system disk, and 36% on the application disk.

Most non-hot cylinders receive near or less than one percent of all read requests. The ten most accessed cylinders combined account for a large percentage of the requested data, particularly on the application disks.

5.6.2 Write Requests

Write requests were not as well distributed as read requests. The write hot spots account for a very large percentage of all write requests. Table 5.11 shows the top ten requested cylinders, for write requests. This table shows a few, very frequently, accessed cylinders.

Figures 5.31 and 5.32 show write hot spots on the student file server's disks. There were four hot spots that, individually, account for 5% or more of all write requests on the application disk. The top ten hot spots combined, on the application disk, account for nearly 79% of all write activity. The system disk had only two hot cylinders, together accounting for 41% of all write activity.

The administrative file server, Figures 5.33 and 5.34, had similar, hot spots of write activity. On the

System Disk			Application Disk		
Cylinder	% reads	cumulative %	Cylinder	% reads	cumulative %
Student File Server					
53	21.65	21.65	1499	35.70	35.70
54	19.98	41.63	1258	9.82	45.52
170	0.99	42.62	1500	7.36	52.88
297	0.80	43.42	1248	6.21	59.09
294	0.79	44.21	1441	4.62	63.71
377	0.52	44.73	1395	4.43	68.14
184	0.51	45.24	1249	3.93	72.07
307	0.44	45.68	1394	3.46	75.53
147	0.43	46.11	843	1.74	77.27
248	0.43	46.54	438	1.59	78.86
Administrative File Server					
34	16.94	16.94	34	10.12	10.12
35	4.89	21.83	1097	6.40	16.52
33	4.54	26.37	1096	2.86	19.38
36	4.37	30.74	1458	2.76	22.14
457	4.35	35.09	1455	2.74	24.88
416	3.79	38.88	1644	2.08	26.96
419	3.17	42.05	44	1.75	28.71
56	2.71	44.76	57	1.49	30.20
60	2.65	47.41	58	1.42	31.62
460	2.57	49.98	355	1.35	32.97

Table 5.11: Top 10 Written Cylinders

system disk, cylinder 34 accounts for over 16% of all write requests. Strangely, on the application disk, the same cylinder number, 34, accounts for 10% of all write activity. The top ten accessed cylinders account for 50% of write activity on the system disk, and 35% of write activity on the application disk.

The student file server's system and application disk had hot spots on the low and high end of each disk respectively, as shown in Figures 5.31 and 5.32. The application disk's hottest write area was near the highest cylinder, the system disk's hottest area was near the lowest numbered cylinder. On the administrative file server, both hot areas were in the same location, at the lower end of the disk.

The administrative file server's hot write areas were not as clustered as the student file server's. The

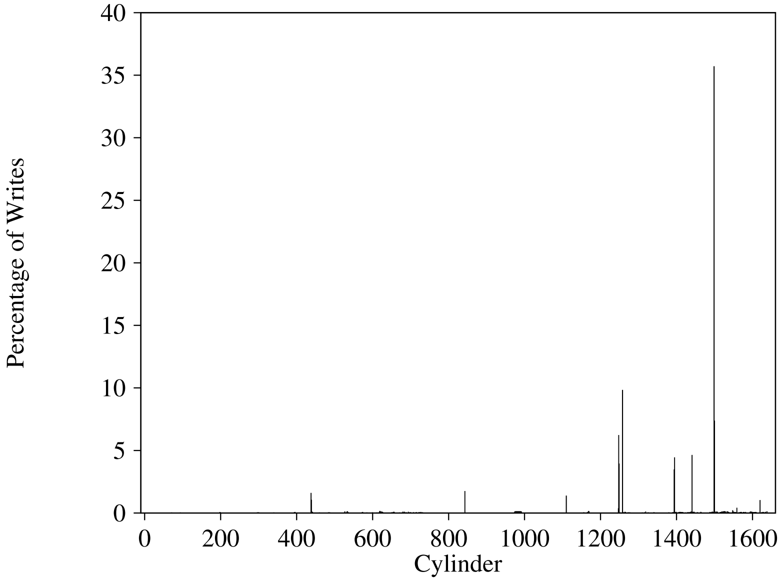

Figure 5.31: Write Access Distribution (Student, Application Disk)

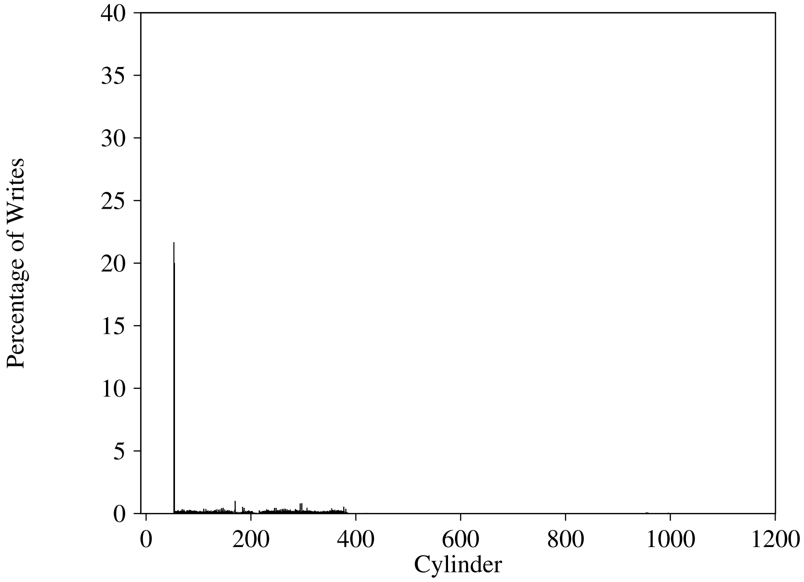

Figure 5.32: Write Access Distribution (Student, System Disk)

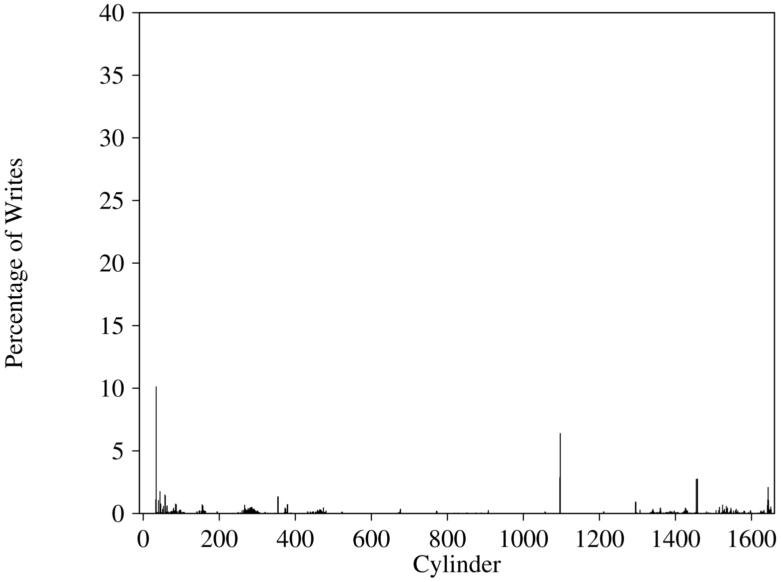

Figure 5.33: Write Access Distribution (Administrative, Application Disk)

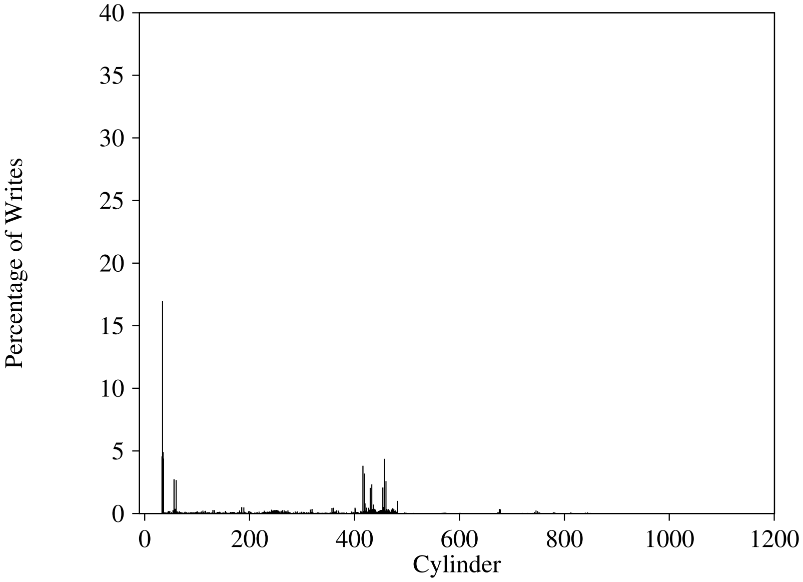

Figure 5.34: Write Access Distribution (Administrative, System Disk)

two hottest write areas on the administrative file server's application disk were about 1,000 cylinders apart. The two hottest write areas on the administrative file server's system disk were only about 400 cylinders apart. Comparatively, on the student file server, the application disk's two hottest areas were about 200 cylinders apart. The system disk on the student file server had only one large hot write area, at cylinders 33 to 36. The closer together the hot areas are, the less distance the drive will need to seek, when retrieving data. For best performance, we would like to have all hot spots close together, resulting in short seeks.

Unfortunately, we have been unable to determine which files reside on the hottest cylinders. It is possible that these cylinders contain the file directories. If these hot cylinders are file directories, the high number of write requests would be explained. In NetWare, directories are always cached, so they should be read only a few times by the operating system. Updates to directories need to be frequently flushed, written to disk, for integrity. Hence, the directory cylinders would have a large number of writes, and a low number of reads.

5.6.3 All Requests

Examining all requests, both read and write, we see patterns similar to write request accesses, because write requests dominate disk IO. Table 5.12 details the top ten requested cylinders. On the student file server, the top ten requested cylinders account for approximately 35% of requests, on each disk. On the administrative server, 26 to 30% of all requests were for the top ten cylinders, on each disk.

Figures 5.35 and 5.36 look somewhat like the write request results, Figures 5.31 and 5.32. Disk activity was, however, spread over the entire disk. On each disk there was only a single clear, dominating hot area, at cylinder 1499 on the application disk and cylinders 53 and 54 on the system disk. These hot areas were comprised mostly of write requests, as we saw earlier.

System Disk			Application Disk		
Cylinder	% reads	cumulative %	Cylinder	% reads	cumulative %
Student File Server					
53	17.65	17.65	1499	15.28	15.28
54	16.33	33.98	1258	4.14	19.42
170	0.83	34.81	1500	3.17	22.59
294	0.74	35.55	1248	2.62	25.21
297	0.70	36.25	223	2.18	27.39
642	0.47	36.72	1441	1.96	29.35
377	0.45	37.17	1395	1.87	31.22
184	0.43	37.60	1249	1.69	32.91
978	0.41	38.01	209	1.58	34.49
248	0.39	38.40	1394	1.46	35.95
Administrative File Server					
34	10.54	10.54	34	4.12	4.12
35	3.22	13.76	834	3.55	7.67
33	2.82	16.58	686	3.16	10.83
457	2.77	19.35	687	2.82	13.65
36	2.72	22.07	1097	2.60	16.25
416	2.39	24.46	836	2.40	18.65
419	2.07	26.53	835	2.38	21.03
56	1.69	28.22	837	2.23	23.26
60	1.66	29.88	832	1.65	24.91
460	1.65	31.53	35	1.29	26.20

Table 5.12: Top 10 Accessed Disk Cylinders

The administrative file server's overall access patterns resemble the write request access patterns, as Figures 5.37 and 5.38 show. The same write hot areas were present, somewhat diminished because of the 1:1 read/write ratio. Many hot areas were due to heavy read traffic, specifically near cylinders 700 and 800. As with the student file server's disks, requests were distributed over the entire disk, with small hot areas, each accounting for 10% or less of overall traffic.

One, simple performance enhancement, which would work for all of the disks analyzed here, would be to seek to the largest hot spot during idle periods. Looking at Figures 5.35 through 5.38, we can see, there are many requests *close* to these hot spots, which would also benefit from this idle seeking. We would benefit least with this algorithm, on the administrative file server's application disk.

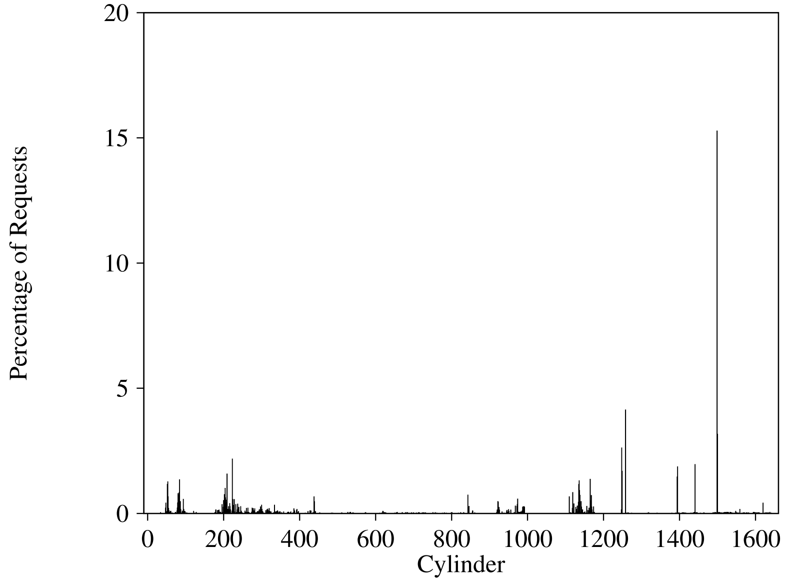

Figure 5.35: Overall Access Distribution (Student, Application Disk)

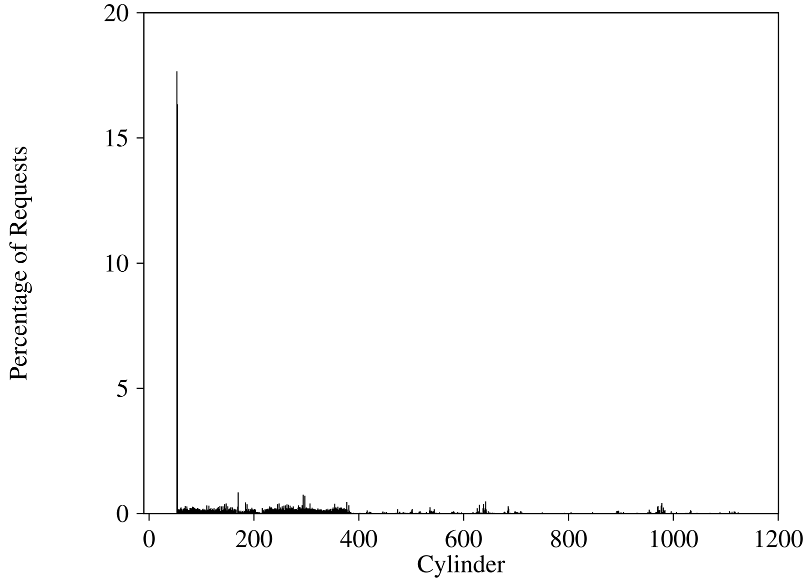

Figure 5.36: Overall Access Distribution (Student, System Disk)

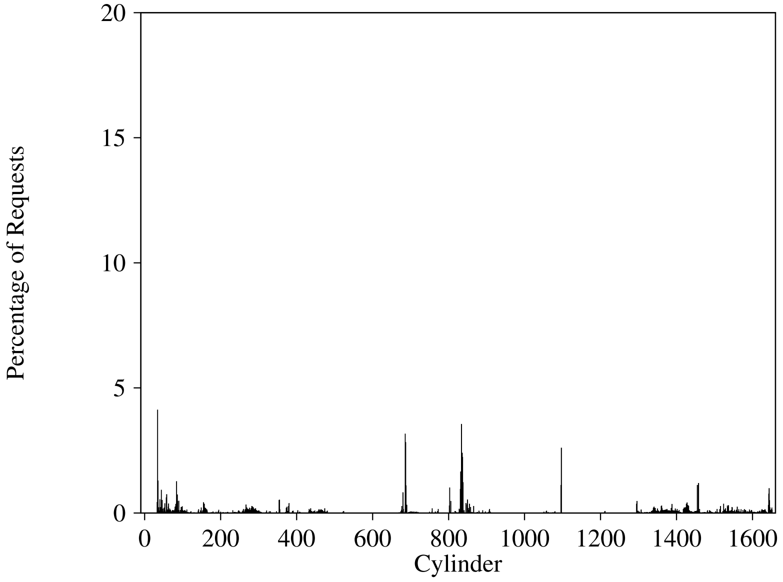

Figure 5.37: Overall Access Distribution (Administrative, Application Disk)

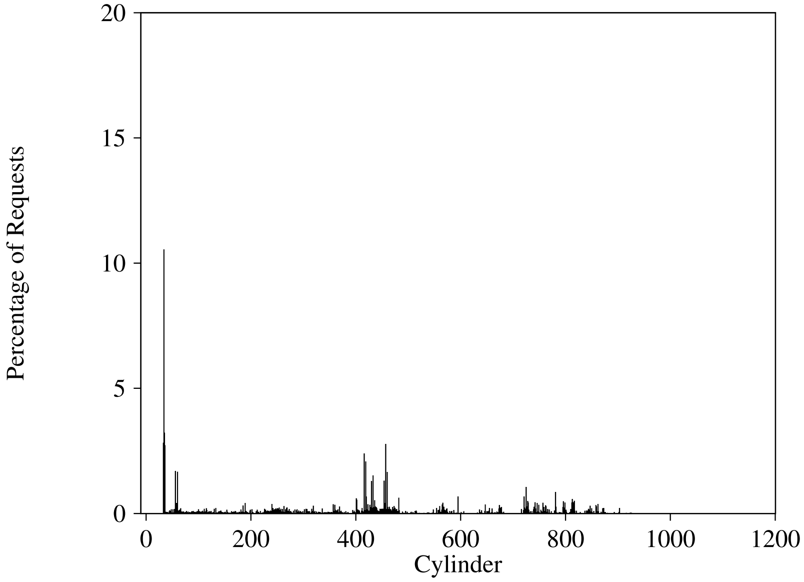

Figure 5.38: Overall Access Distribution (Administrative, System Disk)

An alternative enhancement would be to move the hot areas (the files or data they contain) to locations near each other. If the frequently accessed data were close to each other, the average time to seek to another hot spot would be less.

5.7 Disk Inter-request Seek Measurements

In this section we examine the distance a drive seeks between requests. This distance is computed as the distance from the last block read, in cylinders, to the first block of the next request. This measure is the distance the disk drive head must travel to retrieve a requested block. Small distances take less time since track (or head) switches are faster than cylinder switches and accessing the next sequential block is typically instantaneous. Because it does not make sense to divide our analysis up between read and write requests, we only analyze the seek distance between adjacent requests.

Both application disks have 15 surfaces, 1,658 tracks per surface, and 85 blocks per track. Both system disks have 14 surfaces, 1,199 tracks per surface, and 48 blocks per track. A seek of distance 0 reflects a request for the next sequential block, which may cause a track or cylinder switch, if it is the last block on the track.

We examine only cylinder seek distances, as they are more costly than block and track seeks. Cylinder seeks involve moving the read/write heads of the disk drive across the disk surface. Track seeks are merely electronic switches, between alternate read/write heads. Block seeks involve waiting for the appropriate block to circulate under the read/write heads at their current position.

Table 5.13 shows the cylinder seek distance mean for each trace, as well as the overall mean for each environment. On the student file server, the overall mean seek distance was 108.43 cylinders and 91.05 cylinders for application and system disks respectively. On the administrative file server, the seek distance means were 87.92 cylinders and 106.04 cylinders for application and system disk, respectively. These means indicate seeks were small relative to the size of the entire disk. The mean seek was less than 10% of the number of cylinders on the disk.

Date	System Disk		Application Disk	
	Mean	Std. Dev.	Mean	Std. Dev.
Student File Server				
4/11/94	82.48	186.61	100.67	261.97
4/12/94	89.92	185.69	106.37	263.06
4/13/94	97.44	197.22	105.83	253.03
4/14/94	96.23	191.99	127.57	260.45
4/15/94	83.77	180.89	97.22	234.92
Overall	91.05	190.08	108.43	256.56
Administrative File Server				
4/26/94	102.80	167.98	94.01	239.96
4/27/94	105.36	176.16	81.26	220.60
4/28/94	108.49	178.03	91.67	239.78
4/29/94	106.78	170.01	71.28	206.38
5/02/94	106.53	173.40	101.80	274.75
Overall	106.04	173.81	87.92	238.25

Table 5.13: Disk Request Seek Distance (512 cylinders)

Figures 5.39 through 5.42 show seek distance histograms. The most frequent seek distance in all figures is 0 cylinders (no seek). On the student file server's disks, more than 50% of all requests seek less than a single cylinder. On the administrative file server's disks, the percentage is less, but 0 cylinder seeks are, by far, the most common on every disk analyzed.

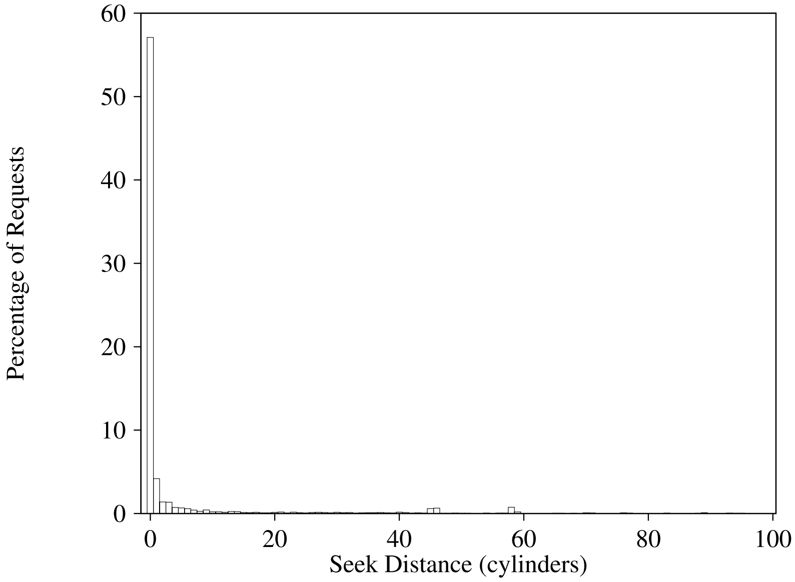

Figure 5.39: Seek Distance Distribution (Student, Application Disk)

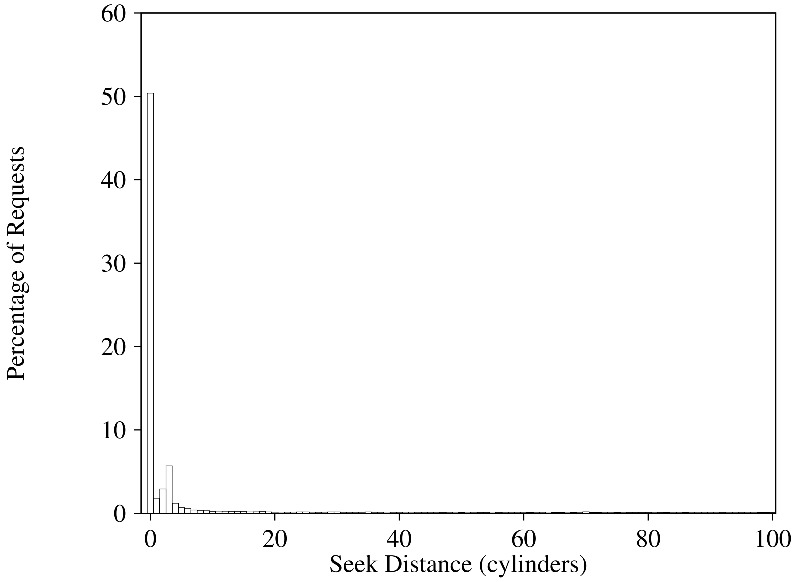

Figure 5.40: Seek Distance Distribution (Student, System Disk)

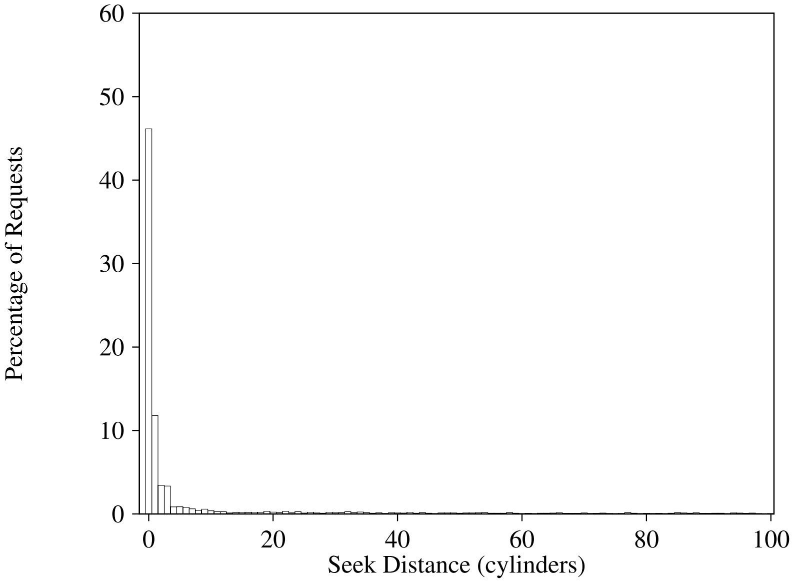

Figure 5.41: Seek Distance Distribution (Administrative, Application Disk)

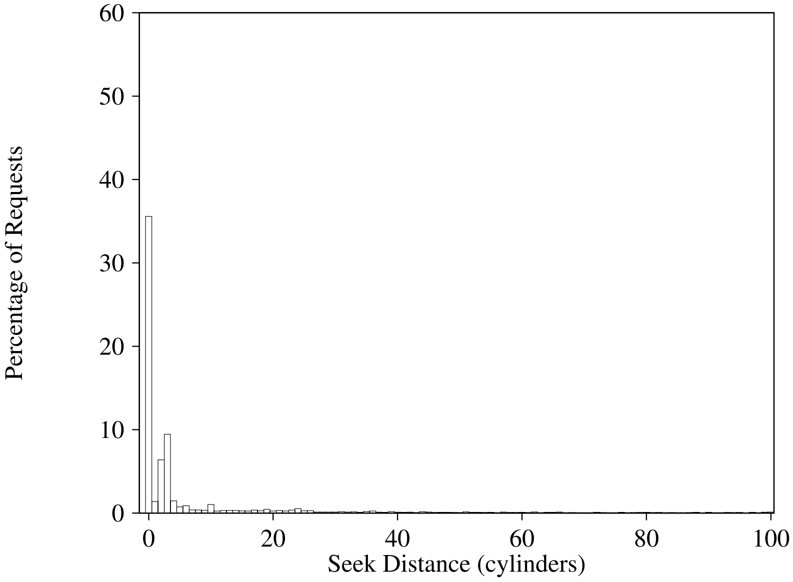

Figure 5.42: Seek Distance Distribution (Administrative, System Disk)

Chapter 6

Network Workload Analysis

In this chapter we analyze measurements gathered by the network sniffer. We begin by considering the number of file server requests, and determining the percentage of NetWare Core Protocol (NCP) file operations. The NCP request mix and size distributions are then derived. Finally, throughput and response time of network trace data is described and analyzed.

6.1 Trace Summary

As with our disk analysis, several traces of network activity were conducted in each environment. Each trace monitored millions of network packets over time intervals of variable length. The sniffer software filtered network traffic, and recorded only communications to or from the file server being monitored. To conserve space, after conducting each trace, a program was used to process the trace data and retain only the header of each network packet. The *reduced* data is what we analyze here. Furthermore, for our analysis here, we have extracted common time periods from each trace.

When network traffic was heavy, the network sniffer was unable to capture all transmissions but

			NCP	File	% File
Date		Time Period	Operations	Operations	Operations
Student Access File Server (USMP1)					
4/11/94	Monday	9 am -11 am	638331	602274	94.3%
4/12/94	Tuesday	9 am -11 am	bad data	—	—
4/13/94	Wednesday	9 am -11 am	1959233	1882269	96.1%
4/14/94	Thursday	9 am -11 am	bad data	—	—
4/15/94	Friday	9 am -11 am	1223197	1148940	93.9%
Mean			3820761	3633483	95.1%
Administrative File Server (PAYSON)					
4/26/94	Tuesday	8 am -12 noon	bad data	—	—
4/27/94	Wednesday	8 am -12 noon	923957	813667	88.0%
4/28/94	Thursday	8 am -12 noon	702327	604689	86.1%
4/29/94	Friday	8 am -12 noon	619159	508264	82.1%
5/02/94	Monday	8 am -12 noon	925182	825197	89.2%
Mean			3170625	2751817	86.8%

Table 6.1: Network Trace Summary

only failed to record less than one-tenth of one percent of the requests. In addition, three of the trace files (4/12/94, 4/14/94, and 4/26/94) were corrupt and could not be used in our analysis.

Table 6.1 summarizes the collected trace data. It details the trace days, times, the number of NCP requests sent to the file server, and the percentage of NCP file and directory operations that we analyze in this chapter. These include file open, close, read and write requests. Recall that each NCP request has an associated response from the file server. The NCP figures listed in Table 6.1 reflect only the number of NCP requests and not their associated responses.

In both environments the percentage of file operations was large, 95% for the student environment and 87% for the administrative. Non-file requests include such operations as file and directory maintenance, directory search, and time/date functions. As stated, our analysis concentrates on the read and write file operations, which compose most of our trace data.

We note here that the administrative environment had a slightly smaller percentage of file operations than the student environment. We attribute this to the fact that student workstations are diskless and

Date	Read%	Write%	Open%	Close%	Other%	Read/Write
Student Access File Server (USMP1)						
4/11/94	80.4%	8.5%	3.3%	2.8%	5.0%	9.5
4/13/94	78.7%	12.2%	2.6%	2.4%	4.1%	6.5
4/15/94	76.7%	10.8%	3.7%	3.4%	5.4%	7.1
Overall	78.4%	11.2%	3.1%	2.8%	4.6%	7.0
Administrative File Server (PAYSON)						
4/27/94	68.8%	4.8%	5.9%	4.4%	16.1%	14.3
4/28/94	70.6%	4.9%	6.8%	4.8%	13.0%	14.4
4/29/94	65.4%	5.1%	5.3%	4.8%	19.3%	12.8
5/02/94	70.5%	3.4%	3.7%	3.4%	19.1%	20.7
Overall	69.1%	4.5%	5.3%	4.2%	16.9%	15.6

Table 6.2: Network File Operation Statistics

use the file server for nearly all of their file system operations. In the administrative environment all workstations have local file systems. Therefore, administrative file server accesses are less frequent and comprise a smaller percentage of overall server operations.

6.2 Request Mix

In this section we consider the mix of requests sent from workstations to the network file servers. Specifically, we quantify the number and types of operations requested of the file servers. We are primarily concerned with file operations, particularly read and write requests. We also briefly consider file open and close operations.

In each trace we identified the read, write, open, close, and other requests. Other file requests include get/set file attributes, directory maintenance, printer queue, and file creation operations. Table 6.2 summarizes the percentage of these requests for each trace. We observe a high read/write ratio in all of the network trace data.

On the student file server, 78.4% of all file operations were read data. Write operations accounted for only 11.2% of file operations. Open operations accounted for 3.1% of operations whereas close operations were 2.8%. We attribute the small difference between open and close operations to the files opened during the trace period and closed after tracing had stopped. All other file and directory operations account for 4.6% of file operations.

On the administrative file server, 69.1% of file operations were read data; 4.5% of operations were write data. Again, we observed slightly more open requests than close requests, 5.3% and 4.2% respectively. Other file operations totalled 16.9% of all requests.

The student file server had a larger percentage of file data operations (read and write requests) than the administrative file server. The administrative file server, as was discussed in Chapter 2, services several printer queues, which may partially account for the high number of other operations.

Figures 6.1 and 6.2 show graphically the read/write ratio of requests. These are moving average graphs, with data points plotted at 10 minute intervals, representing the read/write ratio for the past 30 minutes. These graphs show a high read/write ratio throughout all of the analyzed time periods.

The student file server's read/write ratio was nearly always greater than 5:1, with a minimum ratio of approximately 5:1. The maximum ratio, 13:1, occurred on Friday, at the beginning of the trace period. In Figure 6.1 we observe that the read/write ratio was fairly uniform throughout the trace. The beginning and end of the traces are the only times there were large fluctuations in the ratio.

The administrative file server reached a minimum ratio of approximately 5:1, late Friday afternoon, and a maximum of 149:1, early Friday morning. In Figure 6.2 we observe particularly volatile read/write ratios at the beginning of each trace, early in the morning. After 10:00am the ratios became more uniform, ranging between 5:1 and 30:1. This can be attributed to employees starting up workstations and loading applications soon after they arrive. As the day progresses, they utilize the file server more for data storage and less for applications.

Overall the mix of requests is not surprising. The percentage of reads and writes is what we expected. A small difference in open and close percentages is apparent, and attributed to the extraction of trace data while the system is still active.

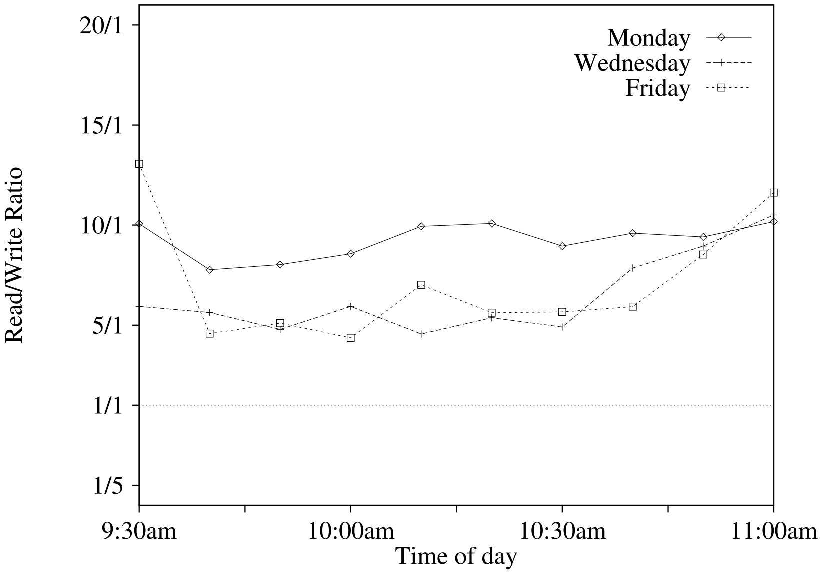

Figure 6.1: Network Read/Write Ratio (Student File Server)

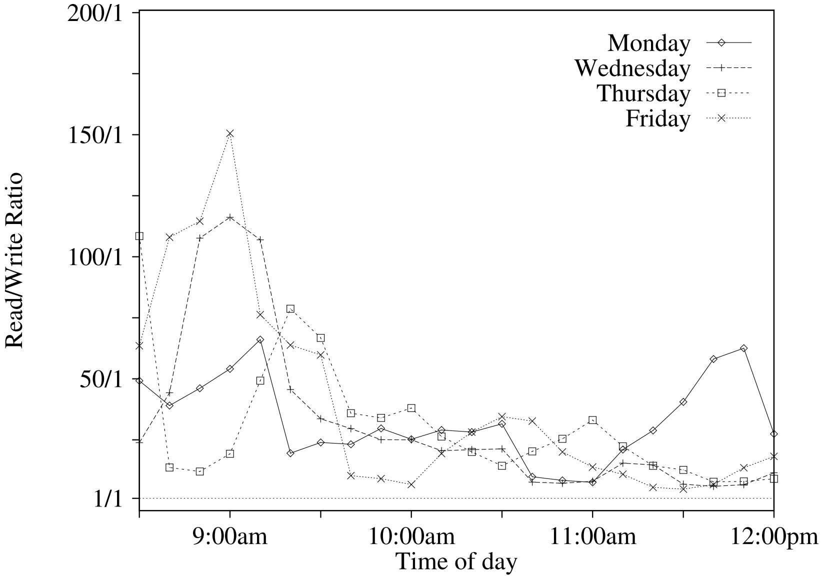

Figure 6.2: Network Read/Write Ratio (Administrative File Server)

Date	Read		Write		Overall	
	Mean	Std. Dev.	Mean	Std. Dev.	Mean	Std. Dev.
Student File Server						
4/11/94	541.8	423.2	398.4	405.6	442.9	433.9
4/13/94	420.0	397.0	402.1	383.8	364.8	394.7
4/15/94	465.6	389.6	370.2	360.3	373.0	391.8
Overall	454.1	400.2	389.5	381.1	380.2	400.7
Administrative File Server						
4/27/94	589.7	418.9	232.8	291.0	367.0	434.1
4/28/94	591.2	427.1	204.0	277.6	367.9	439.2
4/29/94	637.9	419.2	335.2	382.0	356.7	443.3
5/02/94	595.4	421.9	309.0	361.4	383.8	440.9
Overall	600.2	422.1	264.83	330.2	370.1	439.2

Table 6.3: NCP Request Size (in bytes)

6.3 Request Size

In this section we quantify the size of two types of requests, those that request data from the file server (read requests), and those that submit data to the file server (write requests). We also examine the size distribution of all NCP operations; including open, close, and other non-file operations. Table 6.3 data summarizes these results.

Because of network limitations, i.e. the maximum transmission unit of Ethernet, requests for large amounts of data are segmented into smaller requests. A client's read request for more than 1KB of data will be split up into 1KB or smaller requests. Similarly, a request with more than 512 bytes of data that must travel through the inter-network router, will be segmented into 512-byte or smaller requests. The effect of this segmentation is observed as large peaks in our distributions at these two limiting sizes. Using a scatter plot based on time, Bodnarchuk and Bunt [5] located *clusters* of requests and estimated the actual size of data requested by clients in a network.

6.3.1 Read Requests

Examining only read requests, columns 1 and 2 in Table 6.3, we observe the mean read request size was 454 bytes, with a standard deviation of 400 bytes for the student file server. The administrative file server had a mean read request size of 600 bytes, and a standard deviation of 422 bytes. As measures of central tendencies however, these mean values are somewhat misleading.

Figures 6.3 and 6.4 are histograms of request size. These figures reveal a different picture than the mean values give. The most common request size was 1024 bytes, the maximum allowed on our LAN. The second most common request size was 512 bytes, the maximum allowed through the inter-network router. In both environments the mode, the most commonly requested packet size, was 1024 bytes and the median was also 1024 bytes. These statistics combined with the histograms, reveal the true nature of our collected data.

The large number of requests for the maximum packet size implies that applications request data transfers larger than our network configuration allows. In fact, the server request size limitation is imposed by a very low network layer, the IPX network protocol.

6.3.2 Write Requests

As with read request size statistics, the write request statistics in Table 6.3 are somewhat misleading. The mean write request size on the student file server was 389 bytes, with a standard deviation of 381 bytes. The mean write request size on the administrative file server was 265 bytes with a standard deviation of 330 bytes.

Figures 6.5 and 6.6 show the write request size histograms for both file servers. In these figures we can see that the mode on the student file server, was clearly 1024 bytes. The mode on the administrative file server, 64 bytes, is not clear in the figure and is perhaps, more misleading than the mean.

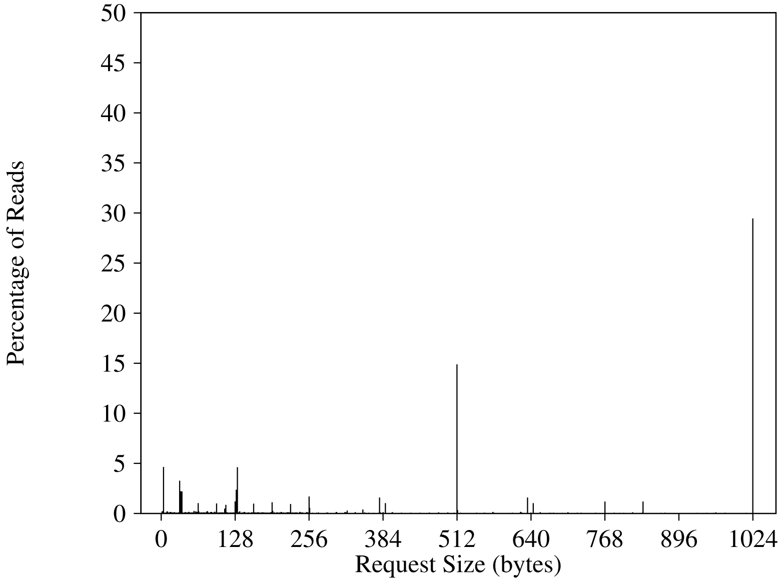

Figure 6.3: NCP Read Size Distribution (Student File Server)

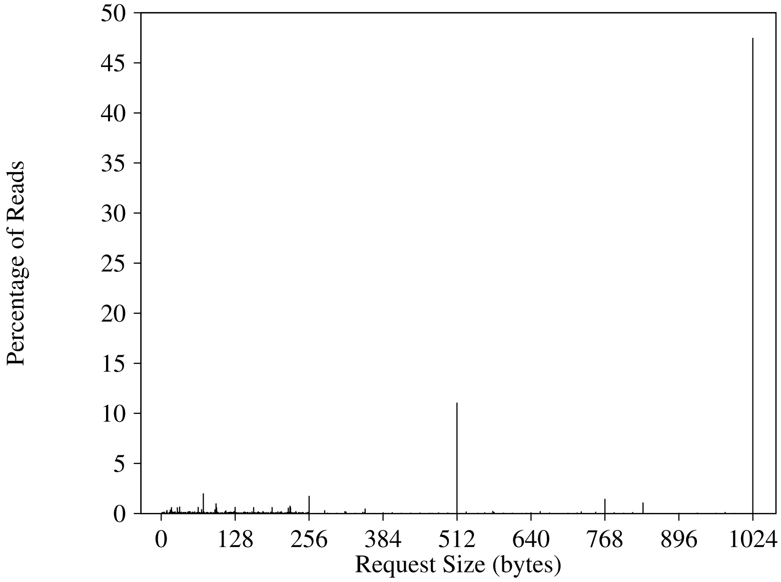

Figure 6.4: NCP Read Size Distribution (Administrative File Server)

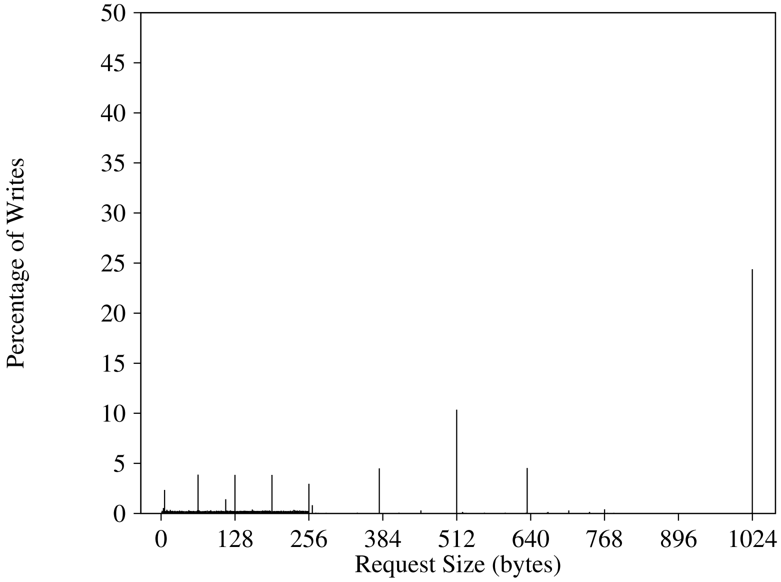

Figure 6.5: NCP Write Size Distribution (Student File Server)

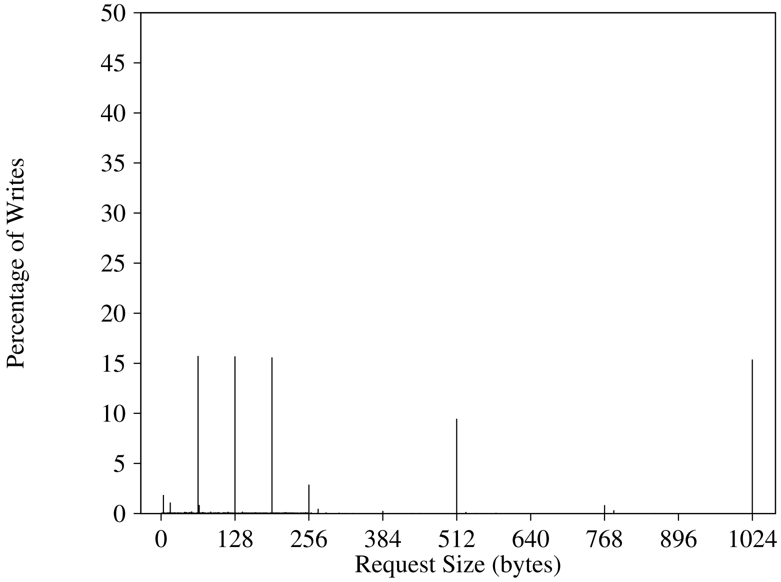

Figure 6.6: NCP Write Size Distribution (Administrative File Server)

More requests are clustered at the lower end of the histograms, for both servers. This is reflected in the lower median values of 192 bytes on the administrative file server and 256 bytes on the student file server. In fact, on the administrative file server, several peaks exist at the low end of the figure. These peaks all lie at multiples of 64 bytes.

6.3.3 Overall Requests

Table 6.3 details the overall request size statistics for each trace and gives overall statistics for all the traces. The student file server had an overall mean request size of 380 bytes, with a standard deviation of 400 bytes. The administrative file server had a mean request size of 370 bytes, and a standard deviation of 439 bytes. As with our read and write request size statistics, these statistics do not clearly represent the request size data.

Figures 6.7 and 6.8 show histograms of overall request size, on both file servers. These figures include read, write, directory, printer, and other NCP operations. Unlike the read and write size histograms, the overall size histograms are dominated by a large number of *zero-byte* requests.

The zero-byte requests are operations with no NCP data. They consist of status requests, acknowledgements and other operations that contain any needed information in the NCP header. The administrative file server encountered more zero-byte requests than the student file server, due to the larger number of non-file operations observed in Section 6.2.

Both the 512-byte and 1024-byte peaks observed on the read and write histograms propagate to our overall histogram, giving a similar appearance to the read size histograms.

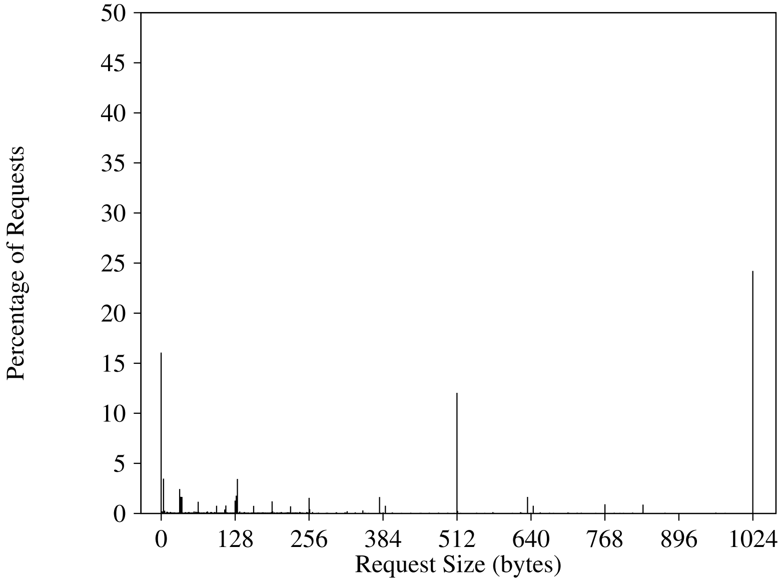

Figure 6.7: Overall NCP Size Distribution (Student File Server)

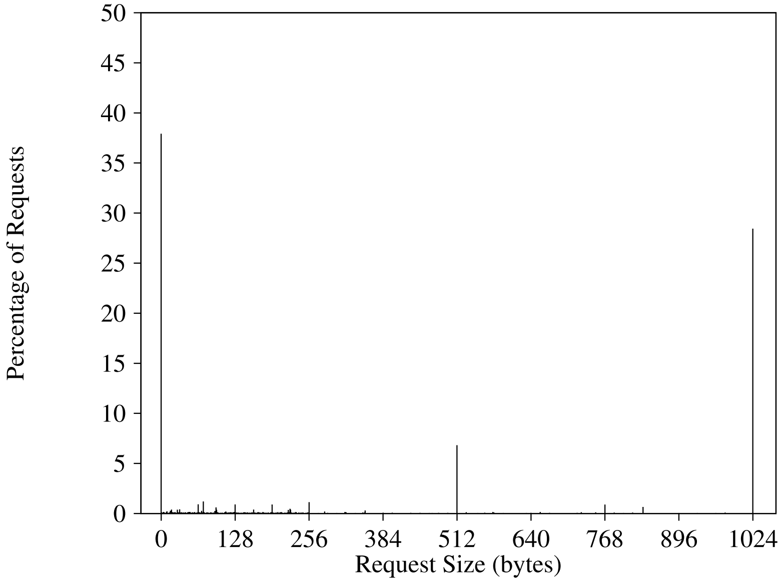

Figure 6.8: Overall NCP Size Distribution (Administrative File Server)

Date	Read		Write		Overall	
	IOs/sec	KB/sec	IOs/sec	KB/sec	IOs/sec	KB/sec
Student File Server						
4/11/94	74.04	35.57	9.71	2.78	83.75	38.35
4/13/94	222.83	84.43	38.88	12.52	261.70	96.95
4/15/94	136.59	55.64	23.20	6.24	159.79	61.88
Mean	144.45	58.55	23.93	7.18	168.41	65.73
Administrative File Server						
4/27/94	37.28	16.28	3.86	0.45	41.13	16.73
4/28/94	28.85	13.04	3.18	0.32	32.03	13.35
4/29/94	21.16	9.59	2.41	0.40	23.56	9.99
5/02/94	38.15	16.79	2.88	0.42	41.03	17.20
Mean	31.36	13.93	3.08	0.39	34.18	14.22

Table 6.4: Daily and Mean Network Throughput (IOs and KB per Second)

6.4 Throughput

In this section we examine the throughput of the two file servers, in requests per second and kilobytes (KB) per second. We examine read and write throughput separately, to gain more insight about the traffic traced. In addition, we examine the overall throughput to characterize the entire file servers' operation.

6.4.1 Read Requests

As observed in earlier sections, read requests constituted the majority of file server traffic. Trace data collected and derived statistics in columns 2 and 3 of Table 6.4 confirm this observation.

Table 6.4 shows that there were, including all traces on the student environment, an average of 144 read requests per second and 58.5 KB of data read per second. Here *data read* is the data portion of read requests, not including network headers. The administrative file server had a similar mix of request types, and is reflected in the 31 read requests per second and 14 KB data read per second.

Figures 6.9 and 6.10 show the moving average graphs for read throughput on the student and administration file servers, respectively. As before, these moving average graphs are plotted at 10 minute intervals and each data point represents the throughput for the past 30 minutes. Figure 6.9, the student file server, shows three very distinct throughput curves. The lowest throughput, Monday, ranged from 50 to 100 reads per second. The heaviest throughput occurred on Wednesday, where throughput ranged from 150 to 250 reads per second.

On the administrative file server, Figure 6.10, the average throughput was between 10 and 40 reads per second. Of all the trace data there was only one prominent peak, on Monday, where throughput reached 60 reads per second.

6.4.2 Write Requests

Write IO throughput in Table 6.4 is a much smaller portion of the overall throughput than are read IOs. The student file server had on average 24 write requests per second and 7.18 KB of data written per second. The administrative file server averaged only 3 write requests per second and less than 512 bytes per second, with 0.39 KB of data written per second.

Figures 6.11 and 6.12 graph the moving average write throughput on the student and administration file servers, respectively. The student file server's lowest throughput again occurred on Monday, where throughput remained below 10 write operations per second. The heaviest write throughput occurred on Wednesday, ranging from 20 to 50 writes per second. The administrative file server had a much lower throughput, less than 8 write operations per second in all traces.

These data demonstrate the small percentage of write requests as discussed earlier, and the small write request sizes. These results suggest that users of these systems were heavy consumers and a light producers.

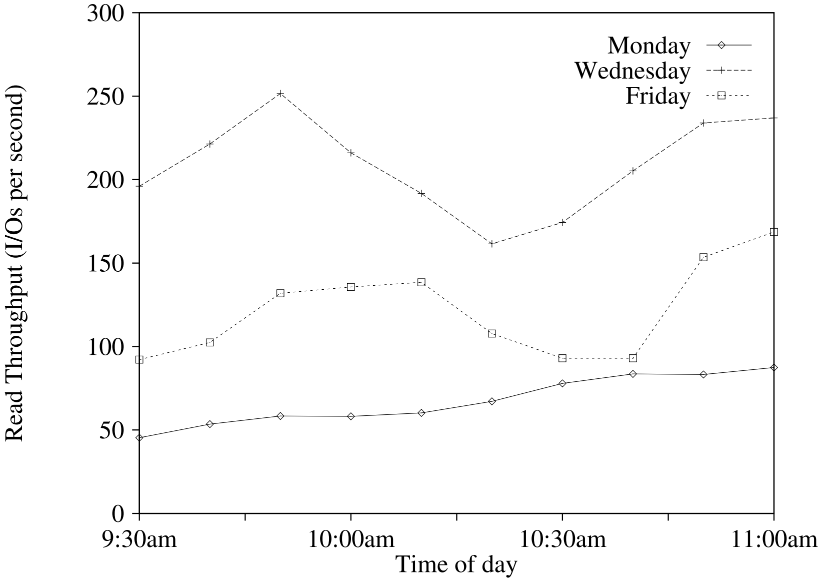

Figure 6.9: Network Read Throughput (Student File Server)

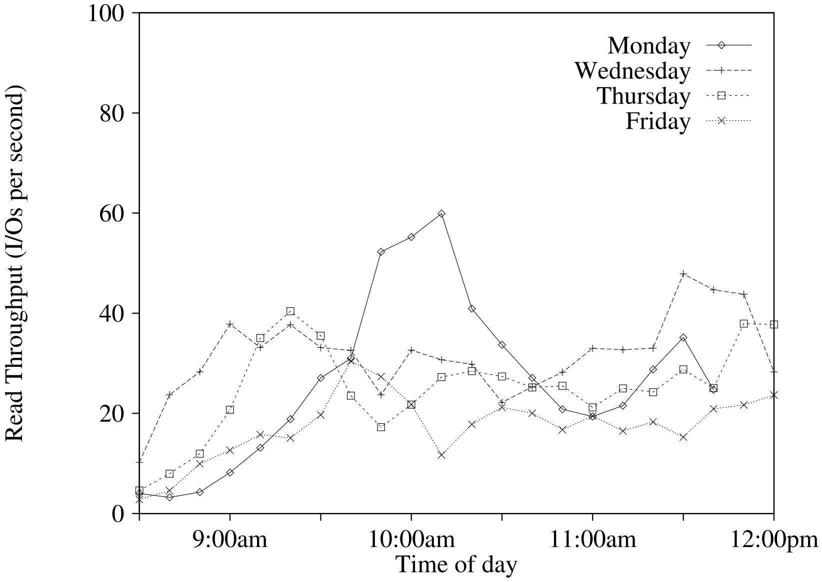

Figure 6.10: Network Read Throughput (Administrative File Server)

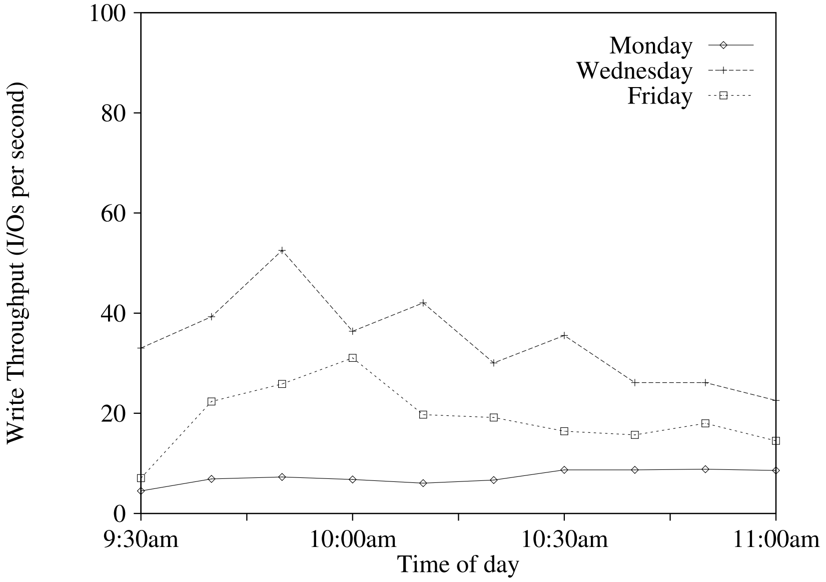

Figure 6.11: Network Write Throughput (Student File Server)

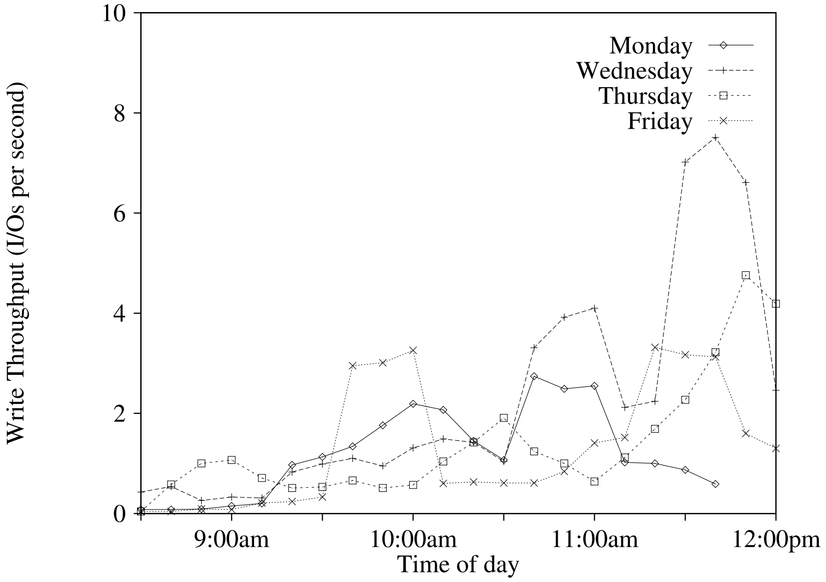

Figure 6.12: Network Write Throughput (Administrative File Server)

6.4.3 Overall Requests

Table 6.4 also shows the overall throughput for each trace, and the mean overall throughput for all traces in each environment. On the student file server, throughput was highest on Wednesday (4/13/94), at 267 requests and 97 KB per second. The mean throughput, of all student file server traces was 168 requests per second and 66 KB per second. The mean transfer size of just under 512 bytes agrees with our earlier observations of transfer size.

The administrative file server had its highest throughput on Monday (5/2/94), 41 requests and 17 KB transferred per second. The mean throughput, of all the administrative file server traces, was 34 requests per second and 14 KB per second. This gives a mean transfer size just under 512 bytes, agreeing with earlier observations.

Daily average throughput, for the student file server, is plotted in Figure 6.13. In this figure we can clearly see Wednesday's heavy traffic. The highest throughput was just after 9:45am, reaching 400 requests per second. The other traces ranged between 100 and 200 requests per second.

Figure 6.14 plots, for each trace, the administrative file server's throughput. Two peaks are apparent in the figure. The first peak was near 10:00am on Monday (5/2/94) at 99 requests per second. The second peak was near 11:30am on Wednesday (4/27/94) at 112 requests per second. Other than these two peaks in activity, throughput ranged between 20 and 60 requests per second. Throughput was lower at the beginning of each trace, which can be attributed to start up, and arrival of users at the beginning of the work day.

The student file server had much heavier use than the administrative file server. Its 30-minute throughput values ranged from 100 to 200 requests per second. This is much higher than the average administrative throughput, which peaked at just over 100 requests per second and averaged between 20 and 60 requests per second.

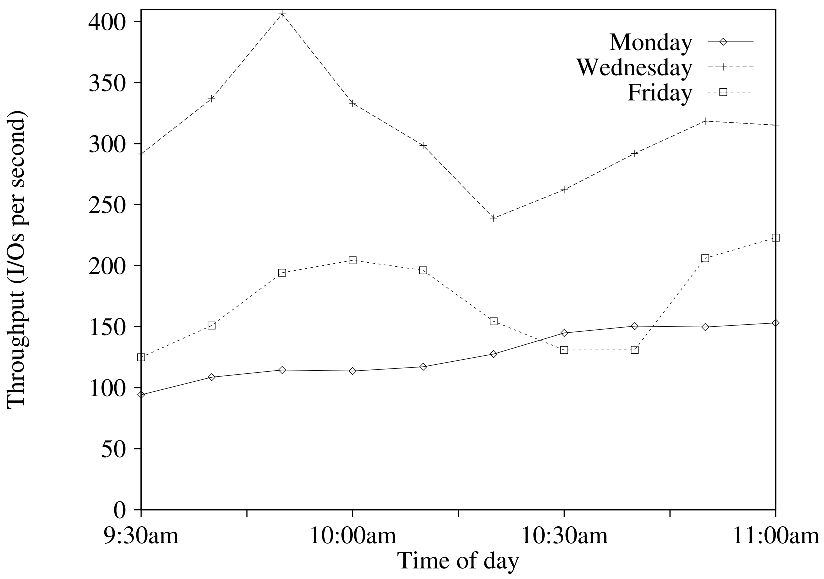

Figure 6.13: Network Throughput (Student File Server)

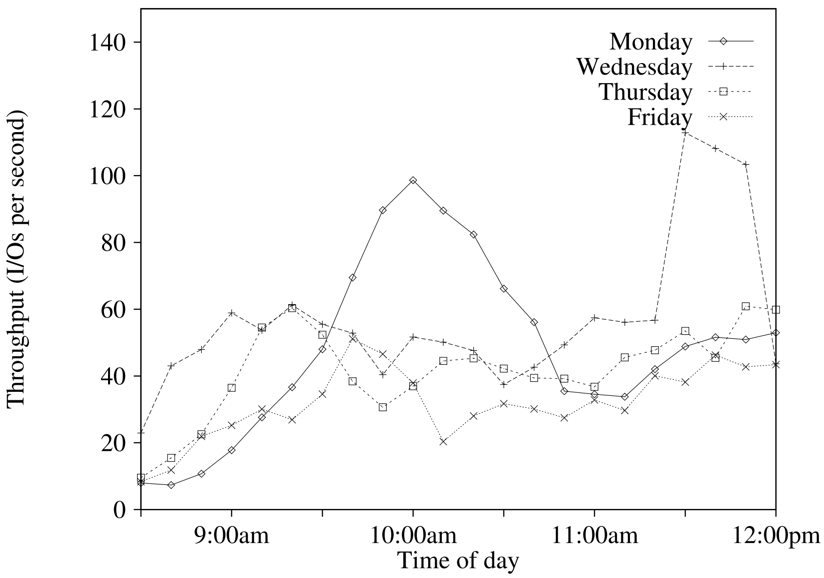

Figure 6.14: Network Throughput (Administrative File Server)

Date	Read		Write		Overall	
	Mean	Std. Dev.	Mean	Std. Dev.	Mean	Std. Dev.
Student File Server						
4/11/94	16.78	112.46	10.85	88.51	16.04	109.77
4/13/94	1.50	12.83	1.02	13.45	1.43	12.38
4/15/94	1.13	7.53	0.73	6.30	1.07	7.37
Overall	3.51	43.53	2.09	32.47	3.31	42.11
Administrative File Server						
4/27/94	2.44	12.29	1.94	11.03	2.40	12.17
4/28/94	2.13	7.88	1.59	6.74	2.07	7.78
4/29/94	2.07	6.38	1.59	6.67	2.02	6.41
5/02/94	2.21	8.54	1.69	8.39	2.17	8.53
Overall	2.23	9.36	1.72	8.65	2.19	9.30

Table 6.5: NCP Response Time (ms)

6.5 Response Time Measurements

In this section, we examine the mean response time of requests on each file server. Response times are measured as the difference in time from the beginning of the request packet, when transmission on the physical network begins, to the beginning of the reply packet. This method has an unfortunate shortcoming; it does not include the data transfer time of read requests. That is, the time for a read request's response packet, which contains data, to propagate across the network is not measured. This is because we use the time network transmission starts for our computation. Transmission end times are not recorded. Write request data transfer time, where data is transferred as part of the request, is measured.

6.5.1 Read Requests

Table 6.5 details the mean and standard deviation of read request response time in each trace, as defined earlier. Here we will examine read response times below the 90-percentile. Then, to gain insight about file server cache misses, we consider the response times above the 90-percentile.

On the student file server, the mean response times ranged from 1.13 ms to 16.78 ms. Monday's trace appears to contain some bad data, with several very long response times raising the mean response time to extremely high levels. The large standard deviation, 112.46 ms supports this observation. The overall mean read response time was 3.51 ms, with a standard deviation of 43.52 ms.

On the administrative file server, the mean response time was 2.23 ms, with a standard deviation of 9.36 ms. All the mean response times were in the range of 2.07 ms to 2.44 ms. The standard deviations were also more consistent than in the student environment, ranging from 6.38 ms to 12.29 ms.

Read request response times, for all student file server traces, less than 5 ms are shown as histograms in Figures 6.15 and 6.16. The histograms show the percentage of requests along the vertical axis, and response time along the horizontal axis. Each bar in the histogram represents a 0.1 ms interval. For example, the bar at 0.5 ms represents the percentage of response times from 0.5 ms to 0.59 ms.

A portion of the student file servers read response time histogram is shown in Figure 6.15. This figure shows response times from 0 to 5 ms, which account for 98.20% of all read requests. The largest peak in the figure is at 0.2 ms, accounting for 15.8% of read requests. The 0.3 ms interval accounts for an additional 15.37% of requests. These two intervals (0.2 and 0.3 ms), along with nearby intervals constitute nearly 50% of all request response times. The median response time was 0.6 ms and the 90-percentile was 1.5 ms.

A portion of the administrative file server's read response time is shown in the histogram Figure 6.16. This figure shows response times from 0 to 5 ms, accounting for 96.03% of read requests. It is clear in this figure that most requests had response times less than 2.0 ms. The largest peak in the histogram, at 1.7 ms, accounts for 15% of read requests. The median read request response time was 1.3 ms and the 90-percentile was 2.1 ms. Both of these statistics are noticeably larger than on the student file server, indicating the administrative file server's cache may be less effective than the student file server's.

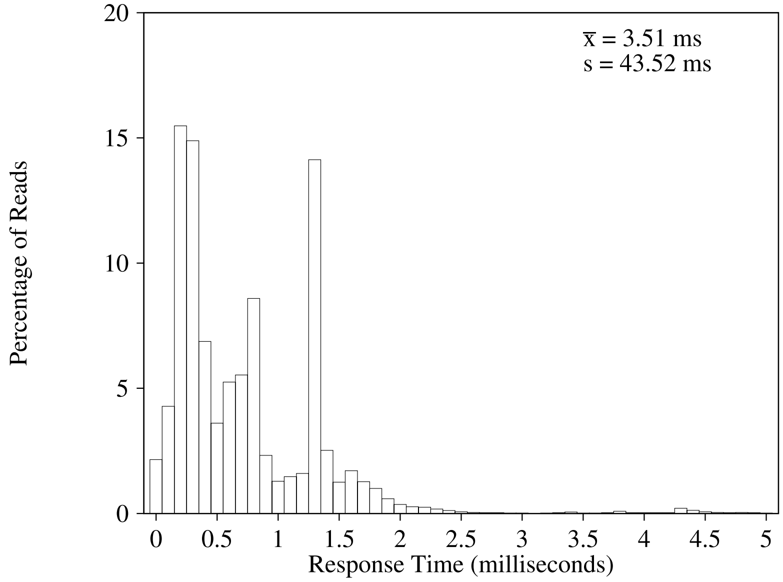

Figure 6.15: Network Read Request Response Time Histogram, <= 5 ms (Student File Server)

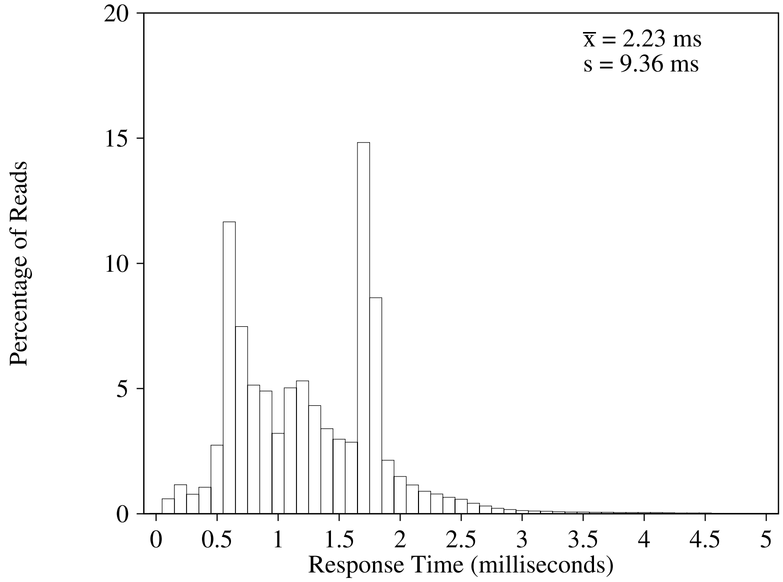

Figure 6.16: Network Read Request Response Time Histogram, <= 5 ms (Administrative File Server)

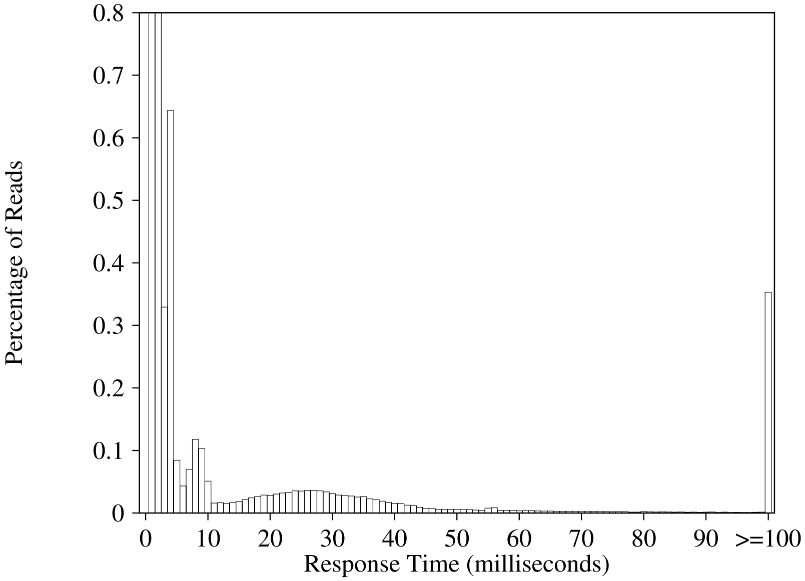

Figure 6.17: Network Read Request Response Time Histogram (Student File Server)

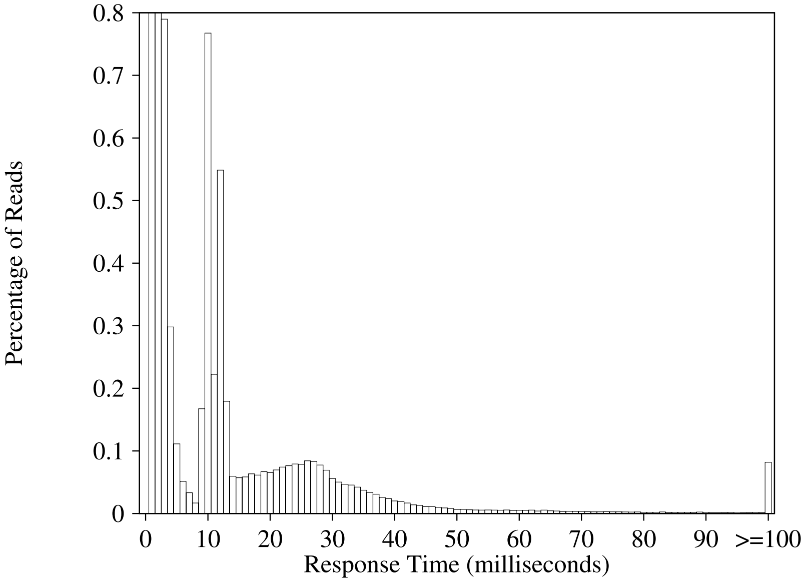

Figure 6.18: Network Read Request Response Time Histogram (Administrative File Server)

Figures 6.17 and 6.18 show the complete read response time histograms of both file servers. In these figures, we graph response times up to 100 ms and each bar represents 1 ms intervals. The 100 ms bar represents all requests with a response time greater than or equal to 100 ms. The Y-axis is scaled to emphasize the requests with response times larger than 5 ms; consequently, bars of time less than 5 ms exceed the height of the graph. These graphs show a clear bimodal distribution, with file server cache hits and misses as distinct peaks in activity.

From the discussion of disk activity in Chapter 5, we know the minimum disk subsystem response time was approximately 3 ms. Thus, network response times less than 3 or 4 ms must have been serviced from the server's file cache. We estimate that network response times between 4 and 10 ms would be server file cache misses and disk controller hits, or zero-cylinder seeks. In Chapter 5 we also observed that disk controller cache hits and zero-cylinder seek response times ranged from 3 to 6 ms. The 4 to 10 ms figure reflects the additional overhead that maintenance of the server's file cache imposes and disk queuing delay time. Network response times larger than 10 ms are server file cache and disk controller cache misses and require disk access.

The student file server histogram, Figure 6.17, supports these observations. A large percentage, more than 98%, of requests were serviced with a response time less than 4 ms (server cache hits). Of the remaining requests, a large percentage were serviced in less than 10 ms. These correspond to requests that required disk subsystem access, but probably were stored in the disk controller's cache (disk cache hits or zero-cylinder seeks). The remaining request response times appear to be distributed similarly to those of disk accesses observed in Chapter 5, only displaced by the time required by the file server to service the request.

The administrative file server histogram, Figure 6.18, also supports our observations. Again, a very large percentage of response times were less than 4 ms, 97%, and a large percentage of the remaining requests were serviced less than 10 ms. The remainder of the histogram follows a skewed, bell-shaped distribution, similar to the distributions of controller cache misses described in Chapter 5.

6.5.2 Write Requests

Write request statistics in Table 6.5 detail the mean and standard deviation of response times for each trace. The student file server, again, had relatively large mean response time on Monday (4/11/95), as it did for read requests.

Overall, the student file server had a mean write request response time of 2.09 ms, and a standard deviation of 32.47 ms. These overall statistics are inflated by the Monday statistics. Neither of the other two student file server traces analyzed had a mean response time of more then 1.02 ms, or a standard deviation of more than 13.45 ms. This supports our supposition that Monday's trace data had errors.

The administrative file server had an overall mean write request response time of 1.72 ms, and a standard deviation of 8.65 ms. These statistics are comparable with the daily statistics, of which the largest mean response time was 1.94 ms, with an 11.03 ms standard deviation.

Figures 6.19 and 6.20 show histograms of write request response times less than 5 ms. On the student file server, shown in Figure 6.19, more than 35% of all write requests had response time of 0.2 to 0.4 ms. The median response time was only 0.4 ms and the 90-percentile was 1.0 ms. Clearly most write requests were serviced by the file server's cache. Only 0.56% of write requests had response time larger than 5 ms, these are not shown. Recall that a *cache hit* in a *lazy-write* cache only schedules data to be written and returns, to the workstation, an acknowledgement. The data is written to disk at a later time. *Write-through* caching would write data to disk immediately and only when the write request completed would an acknowledgement be returned.

Figure 6.20 shows the write response time histogram for the administrative file server. In this histogram there are two distinct peaks in response time, at 0.6 ms and 1.2 ms, accounting for 35% of write requests. The median write request response was 1.2 ms, and the 90-percentile was 2.4 ms. On this file server, 1% of write requests had a response time larger than the 5 ms extent of this figure.

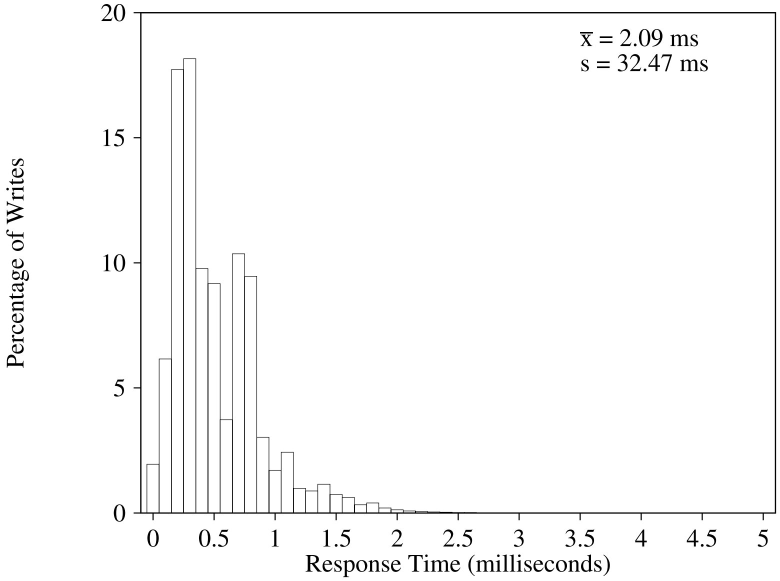

Figure 6.19: Network Write Request Response Time Histogram (Student File Server)

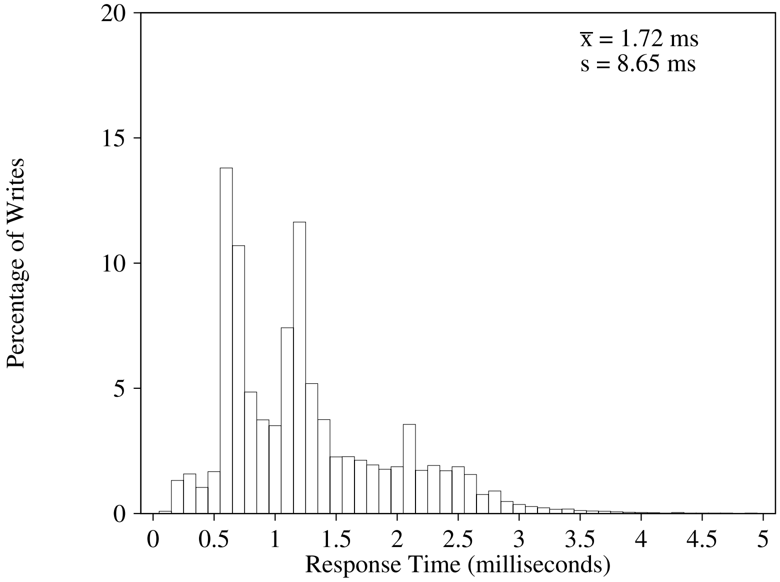

Figure 6.20: Network Write Request Response Time Histogram (Administrative File Server)

The administrative file server's write response time was more dispersed than the student file server's. Unlike read response time, a larger percentage of requests were serviced with a response time of less than 5 ms. This is due to the lazy-write file caching used by the file server. Similar to read response times, the administrative file server's mean and percentile statistics are larger, indicating slower response times than the student file server.

6.5.3 Overall Requests

Table 6.5 shows, for each trace, the mean and standard deviation of overall response time. The student file server had a mean overall response time of 3.31 ms, and a standard deviation of 42.11 ms. The administrative file server had a mean overall response time of 2.19 ms and a standard deviation of 9.30 ms.

Again, Monday (4/11/95), on the student file server was very different than the other two traces collected, with a mean response time of 16.04 ms. The largest mean of the two other traces was 1.43 ms. The administrative file server's largest mean and standard deviation was 2.40 ms and 12.17 ms respectively.

Figures 6.21 and 6.22 show histograms for overall response time, again only for response times up to 5 ms. In Figure 6.21, the student file server's overall response time histogram, there are a few large peaks, which correspond closely to the peaks in the read response time histograms. Recall that read requests dominate traffic, so this is not surprising. The median overall response time was 0.60 ms, and the 90-percentile was 1.40 ms. These percentile statistics are also similar to the read response time statistics.

The administrative file server, Figure 6.22, has two prominent peaks, at 0.6 ms and 1.7 ms, which again correspond to peaks on the read response time histogram for this server. The median overall response time was 1.2 ms, and the 90-percentile was 2.1 ms.

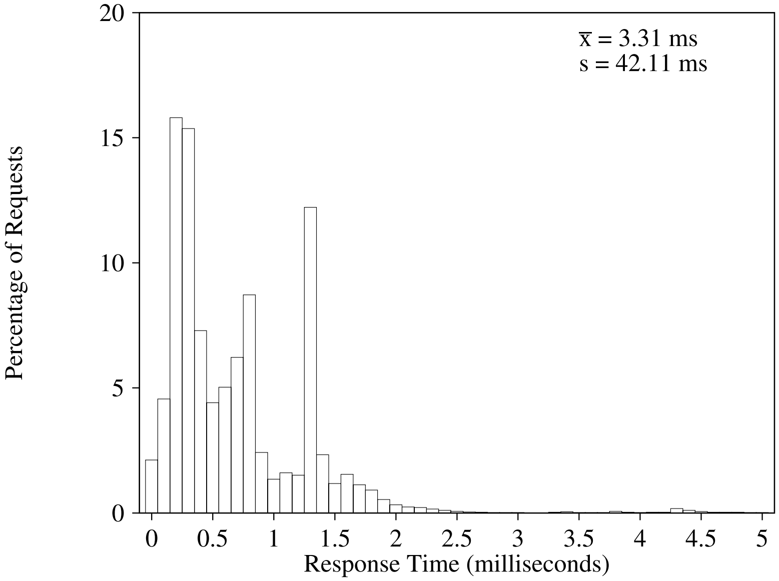

Figure 6.21: Network Request Response Time Histogram (Student File Server)

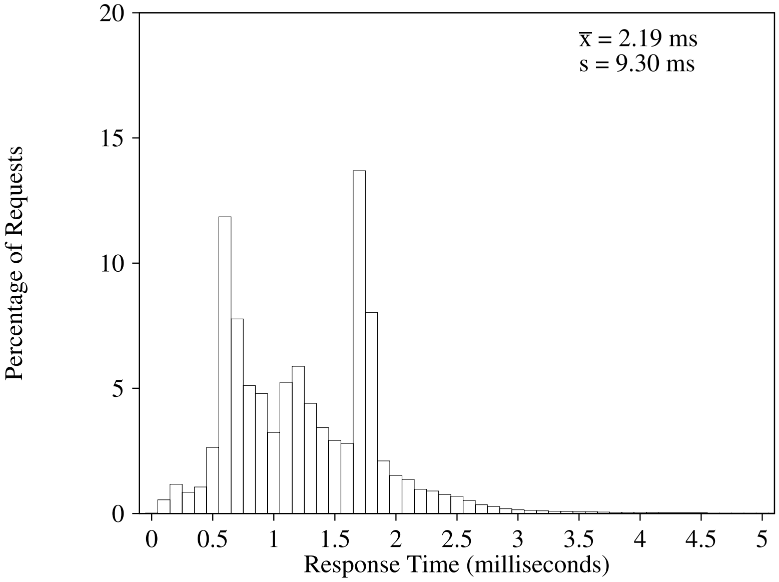

Figure 6.22: Network Request Response Time Histogram (Administrative File Server)

Not surprisingly, the overall response time histograms resemble the read response time histograms. Read requests dominate the traffic on both file servers, causing overall response time to be weighted towards the statistics for read requests.

The student file server, shown in Figure 6.19, had one large peak and several smaller ones. The large peak at 0.2 ms and 0.3 ms together accounted for 36% of all write requests. Other peaks each account for 10% or less of write requests. Fifty-percent of write request response times were less than 0.4 ms, and 90% were less than 1.0 ms. Both of these percentile measures are less than their corresponding read response time percentiles.

Chapter 7

Comparison of Network and Disk Workloads

In this chapter we compare the workload traces of both the disk subsystem, analyzed in Chapter 5, and network traffic, analyzed in Chapter 6. We make a time correspondence between network trace data and disk trace data. With this correspondence, we analyze the impact that network traffic has on disk traffic and draw conclusions about file server performance.

We start by defining how network requests are mapped to disk requests. Then we proceed with a comparison of the traces read/write ratio, and throughput.

7.1 Corresponding Network Requests and Disk IO

To compare network trace data and disk trace data, we first need to map network (NCP) requests to disk requests. In some cases this mapping is obvious; a NCP read request will result in a disk read request. Other NCP requests are not as easily translated. A more complex example is an open file

Operation	NCP	Opeartion
Description	Code	Classification
File Search Init.	$0x3e$	disk read
File Search Cont.	$0x3f$	disk read
Search for File	$0x40$	disk read
Close File	$0x42$	disk write
Create File	$0x43$	disk write
Erase File	$0x44$	disk read and write
Rename File	$0x45$	disk read and write
Set Attributes	$0x46$	disk write
Get File Size	$0x47$	disk read
Read File	$0x48$	disk read
Write File	$0x49$	disk write
Set Time and Date	$0x4b$	disk write
Open File	$0x4c$	disk read
Create New File	$0x4d$	disk write

Table 7.1: Translation of NCP to IO Operations

request. This NCP request requires an update to the file directory to set the last-accessed date of the file. Thus, an open file request causes a disk write request.

Table 7.1 classifies each NCP file operation as a disk read or disk write request. This classification indicates the type of disk operation the NCP operation translates to, if it were not using a file server (or cache). We use this mapping of requests throughout this chapter to allow us to compare the two data sets.

7.2 Read/Write Ratio

In this section, we compare measured disk read/write ratio and the read/write ratio that would be expected to result from NCP operations observed during the same trace period. For the purposes of comparison, we selected data from the disk traces for the same time period as the traced network traffic traces. Table 7.2 shows the read/write ratios from both trace environments for the same time periods. Both request ratios, (IOs) and kilobytes (KB) of data transferred, are shown.

Date	Disk R/W Ratio		Network (NCP) R/W Ratio	
	IOs	KB	IOs	KB
Student File Server				
4/11/94	0.33	2.37	7.63	7.63
4/13/94	0.31	2.43	5.73	5.73
4/15/94	0.24	2.02	6.89	5.89
Mean	0.29	2.27	6.75	6.42
Administrative File Server				
4/27/94	1.08	5.11	9.65	9.67
4/28/94	0.87	4.21	9.07	9.07
4/29/94	0.81	3.62	8.78	8.80
5/02/94	1.07	4.72	13.25	13.24
Mean	0.96	4.42	10.19	10.20

Table 7.2: Disk and Network Read/Write Ratios

The NCP read/write ratios in Table 7.2 are different than those presented in Chapter 5. This is because the ratios presented in Chapter 5 are a comparison of NCP reads to NCP writes, whereas the ratios presented here are NCP operations classified as reads and NCP operations classified as writes, from Table 7.1.

In Table 7.2, we observe that network read/write ratio is much larger than disk read/write ratio. In the student environment, the network read/write ratios are about twenty times greater than the disk read/write ratios. In the administrative environment, network read/write ratios are about ten times greater than disk read/write ratios.

Figures 7.1 through 7.7 show both the disk and network read/write ratio curves for the traced environments. The figures are plotted using the moving average method described earlier. Data points are plotted at ten minute intervals. Each data point represents the read/write ratio for the past thirty minutes.

Both disk and network trace read/write ratios for the student environment are plotted in Figures 7.1 through 7.3. We observe that client network read/write ratios, as well as disk read/write ratios remain somewhat uniform throughout the trace period. While client network read/write ratios range

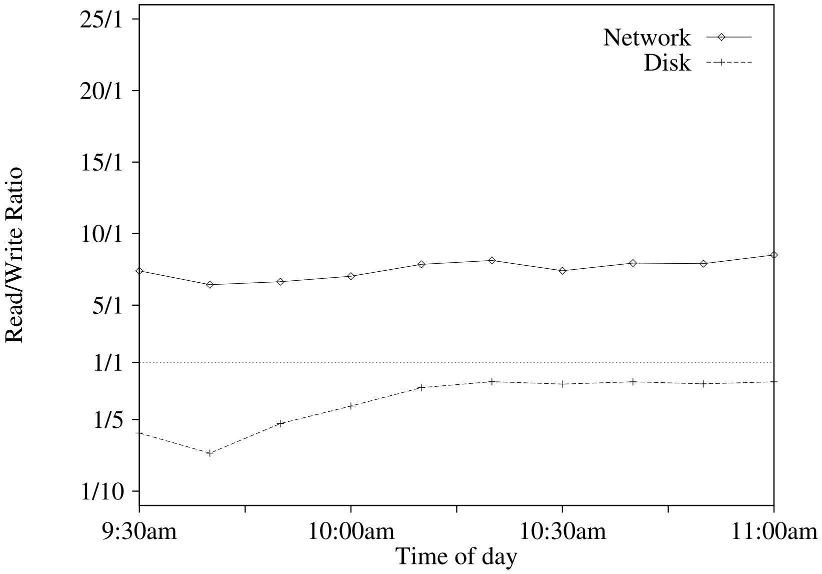

Figure 7.1: Student File Server Read/Write Ratio, Monday 4/11

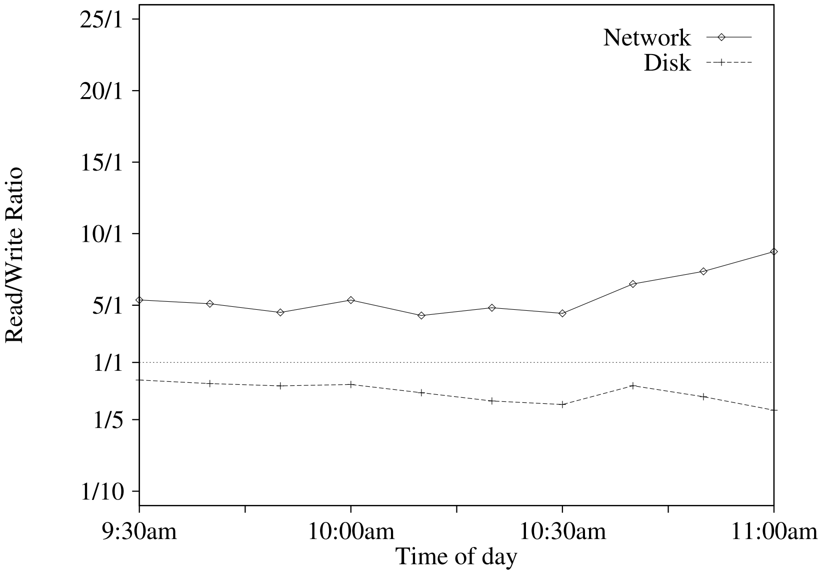

Figure 7.2: Student File Server Read/Write Ratio, Wednesday 4/13

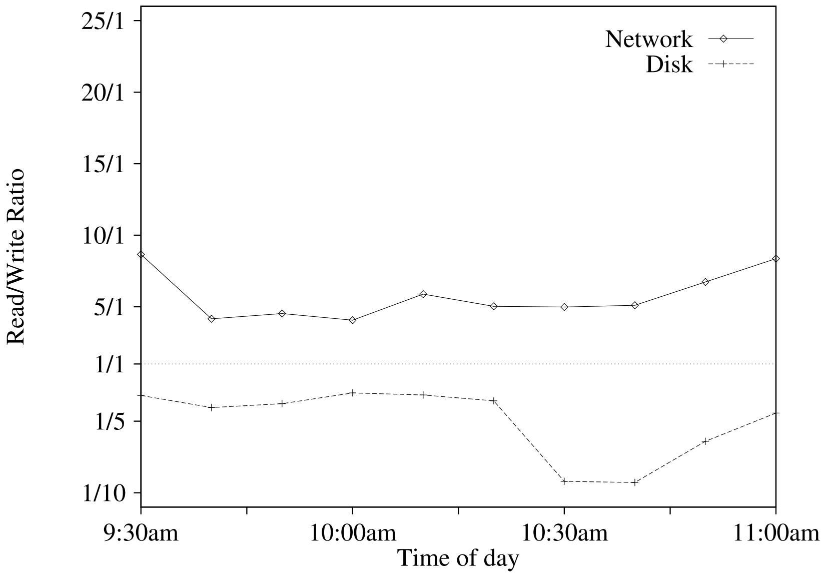

Figure 7.3: Student File Server Read/Write Ratio, Friday 4/15

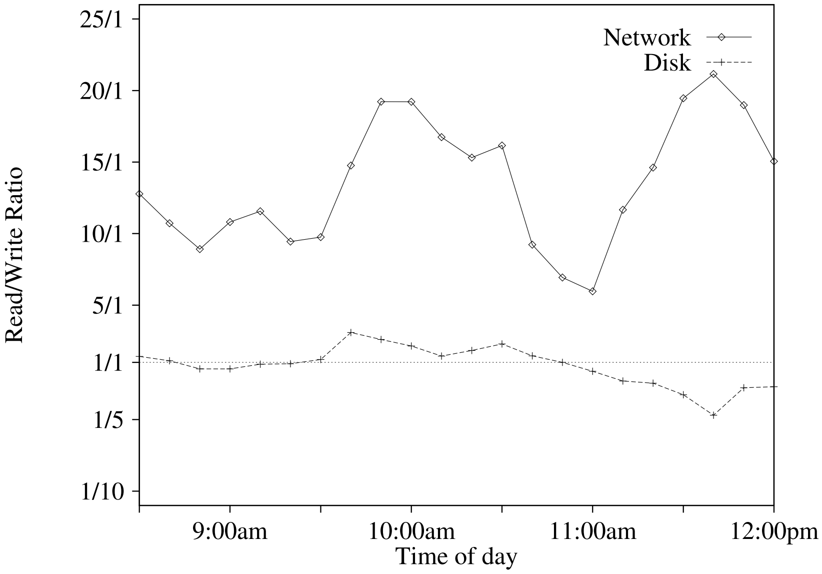

Figure 7.4: Administrative File Server Read/Write Ratio, Monday 5/2

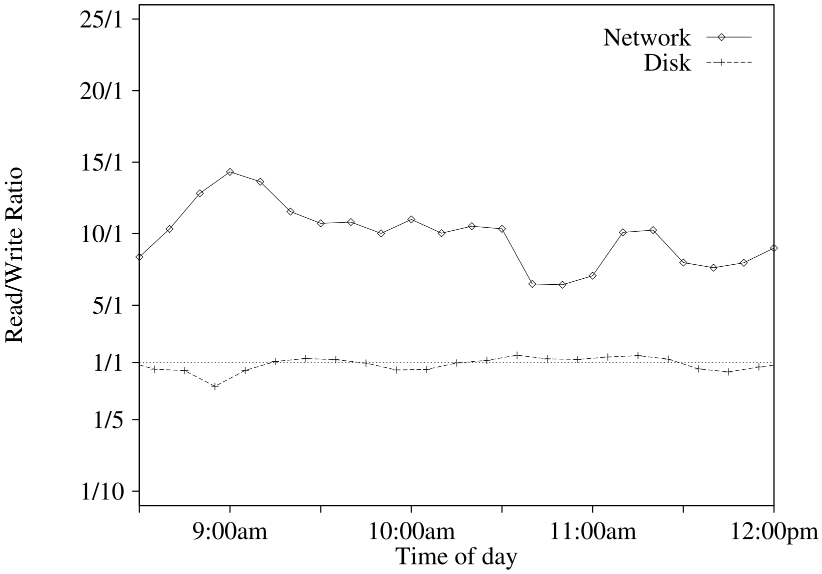

Figure 7.5: Administrative File Server Read/Write Ratio, Wednesday 4/27

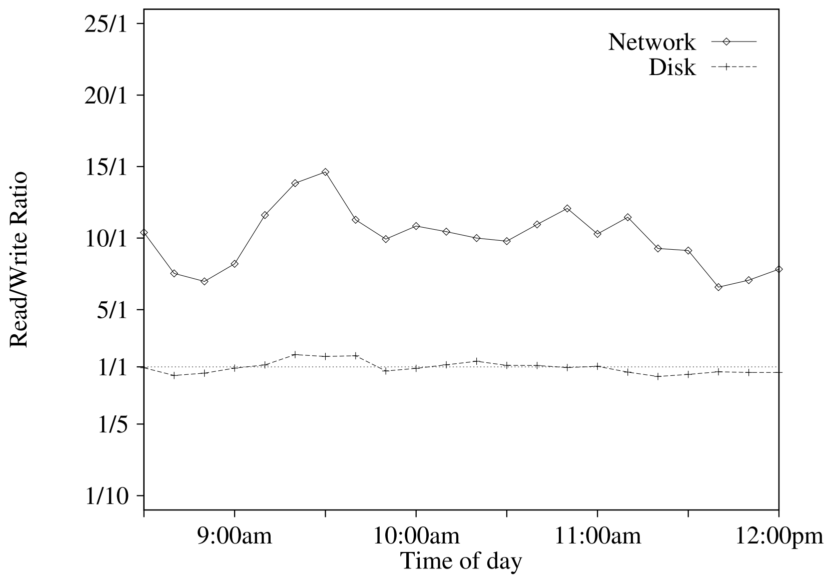

Figure 7.6: Administrative File Server Read/Write Ratio, Thursday 4/28

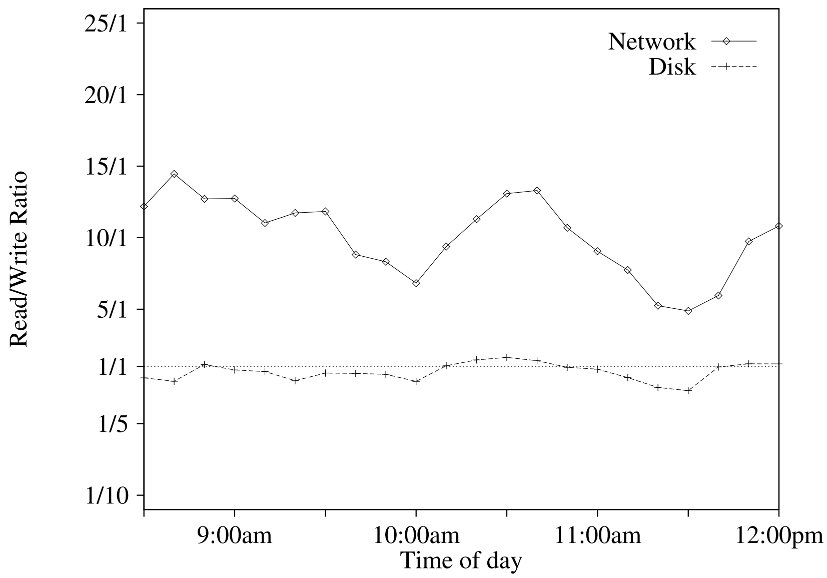

Figure 7.7: Administrative File Server Read/Write Ratio, Friday 4/29

from a low of 5:1, Figure 7.3, to a high of 10:1, disk read/write ratios have a low of 1:10 and a high of 1:2.

Data for the administrative environment are shown in Figures 7.4 through 7.7. Fluctuation in the network read/write ratios have no apparent effect on the disk read/write ratios. While the network client read/write ratios range from a low of 5:1 to a high of 20:1, Figure 7.4, disk read/write ratios remain relatively uniform at about 1:1, ranging from a high of 3:1 to a low of 1:3.

We conclude that most client file reads to the server are handled by the server cache without disk access and that most observed accesses in client read operations are to the same cached files, rather than to new files. Similarly, we observe, in Figures 7.1 through 7.7, that disk read/write ratio remains at about 1:1, regardless of fluctuations in network client read/write ratios.

We do not see a clear correlation between the read/write ratios of disk and NCP operations. Large changes in the mix of NCP requests have little effect on the read/write ratio of disk IOs. This would indicate the file cache has a high number of *hits*.

7.3 Throughput

In this section we compare disk throughput and network throughput. Again, we use trace data from the network sniffer and disk analyzer during the same time periods. Table 7.3 shows read, write, and overall throughput for both disk and NCP trace data. This table summarizes the data based on IO requests per second. We do not consider the amount of data (KB) transferred per second.

Table 7.3 shows clearly the large difference in network throughput and disk throughput. The network interface of the file server is managing an order of magnitude more requests than the disk subsystem. The difference is greatest in the student environment, where there is a mean overall disk throughput of 10.55 IOs per second and a mean overall network throughput of 168.42 IOs per second.

Date	Read		Write		Overall	
	disk IOs	NCP IOs	disk IOs	NCP IOs	disk IOs	NCP IOs
Student File Server						
4/11/94	2.45	74.04	7.50	9.71	9.95	83.75
4/13/94	3.64	222.83	10.67	38.88	14.31	261.71
4/15/94	1.44	159.79	5.95	23.20	7.39	159.79
Mean	2.51	152.22	8.04	23.93	10.55	168.42
Administrative File Server						
4/27/94	2.07	37.28	1.92	3.86	3.99	41.13
4/28/94	1.28	28.85	1.47	3.18	2.75	32.03
4/29/94	1.07	21.16	1.32	2.41	2.40	23.56
5/02/94	1.07	38.15	1.63	2.88	3.38	41.03
Mean	1.54	31.36	1.59	3.08	3.13	34.44

Table 7.3: Compared Daily and Mean Throughput (IOs per Second)

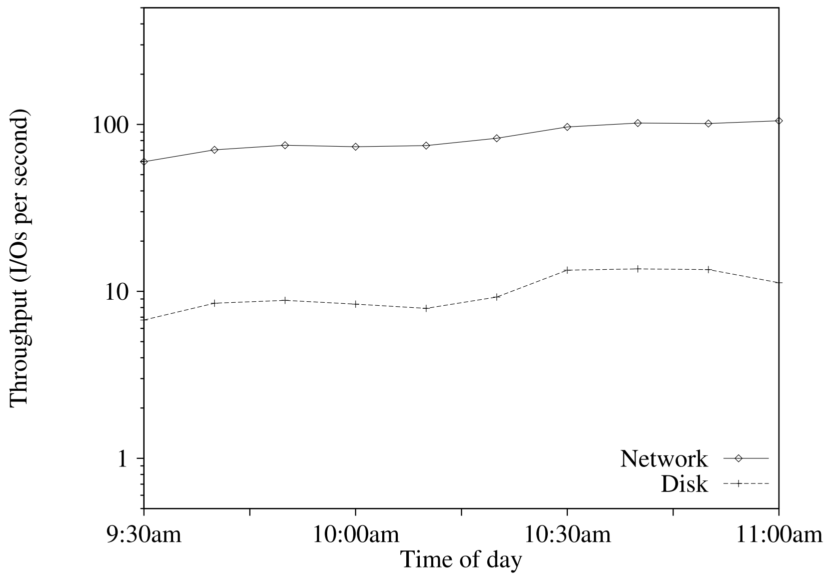

Figure 7.8: Student File Server Throughput, Monday 4/11

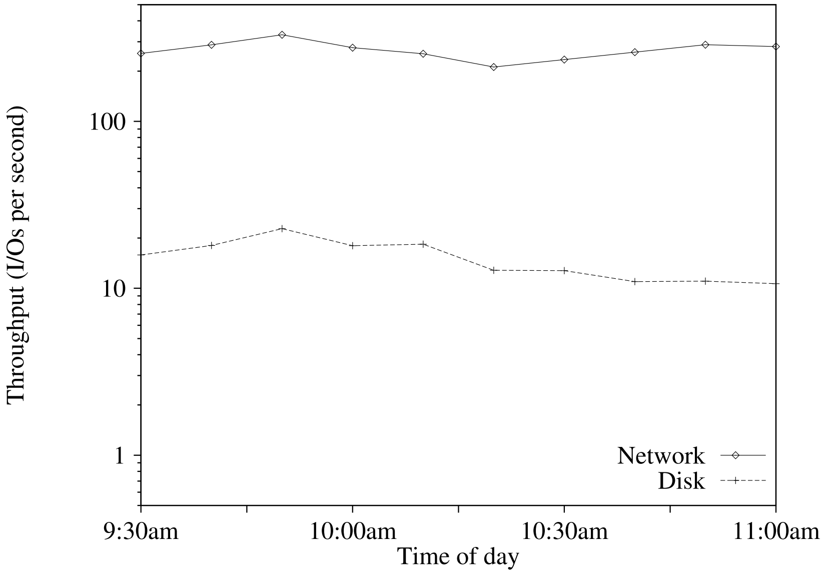

Figure 7.9: Student File Server Throughput, Wednesday 4/13

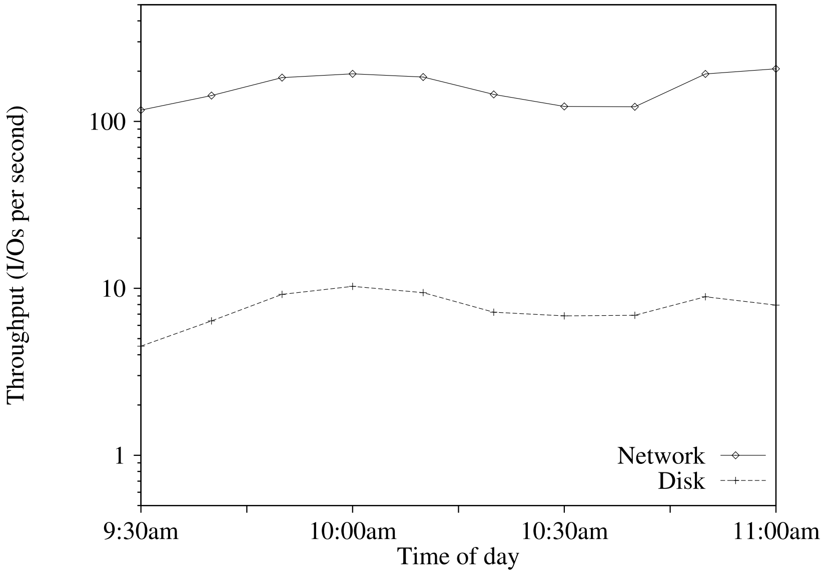

Figure 7.10: Student File Server Throughput, Friday 4/15

Date	Overall	Read	Write
Student File Server			
4/11/94	0.964	0.898	0.975
4/13/94	0.812	0.614	0.977
4/15/94	0.945	0.754	0.902
Administrative File Server			
4/27/94	0.552	0.342	0.924
4/28/94	0.957	0.898	0.762
4/29/94	0.776	0.682	0.710
5/02/94	0.883	0.798	0.924

Table 7.4: Disk and Network Throughput (IOs per Second) Correlation Coefficients

Figures 7.8 through 7.10 show the moving average disk and network throughput for the student file server. These figures use a logarithmic scale for the Y-axis, to show both disk and network activity in detail. In these figures, it appears that disk and network throughput, measured in IOs per second, is somewhat correlated. Peaks and valleys in network throughput are nearly mirrored in the disk throughput. From these figures however, it is hard to determine the degree of correlation. Both disk and network throughput curves appear more volatile when plotted on a normal scale. We also observe that network throughput is an order of magnitude larger than disk throughput.

Figures 7.11 through 7.14 show the moving average disk and network throughput for the administrative file server. In these figures, the correlation we saw between network and disk throughput is more apparent. These curves are much more volatile, and nearly mirror each others peaks and valleys. Again, we observe that network throughput is an order of magnitude larger than disk throughput.

In both the student and administrative environments throughput appears to be correlated. However, in some cases, particularly Figure 7.12 from 11:30 am to 12:00 noon, a peak in network throughput has no apparent impact on disk throughput. For this reason we computed the correlation coefficients between network and disk throughput. Table 7.4 shows the overall, read, and write request correlation of network and disk throughput.

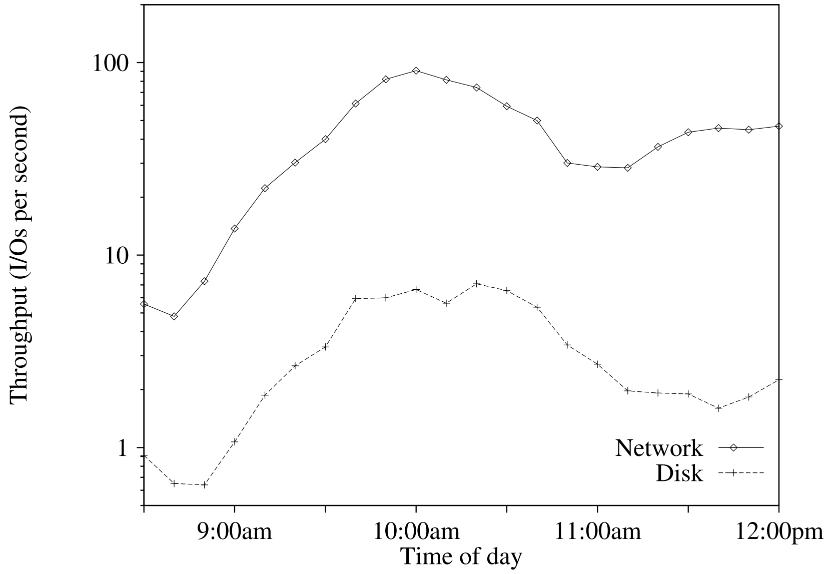

Figure 7.11: Administrative File Server Throughput, Monday 5/2

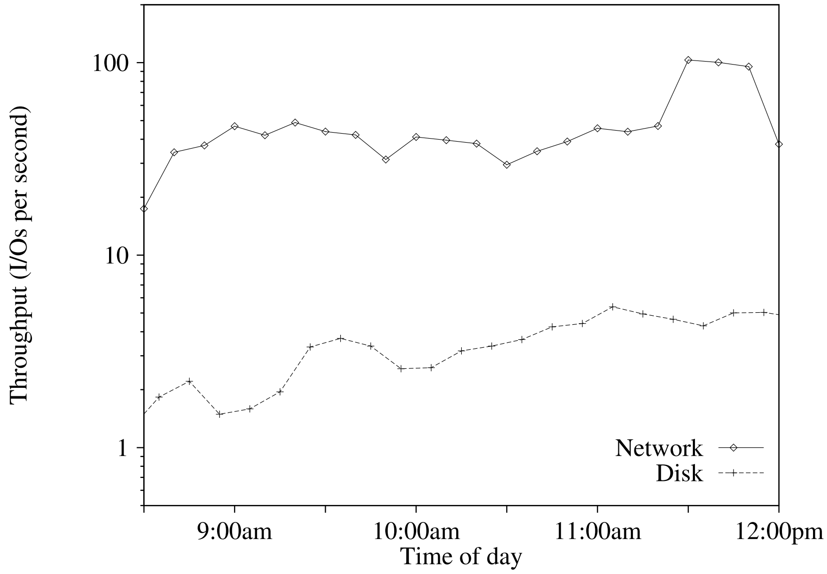

Figure 7.12: Administrative File Server Throughput, Wednesday 4/27

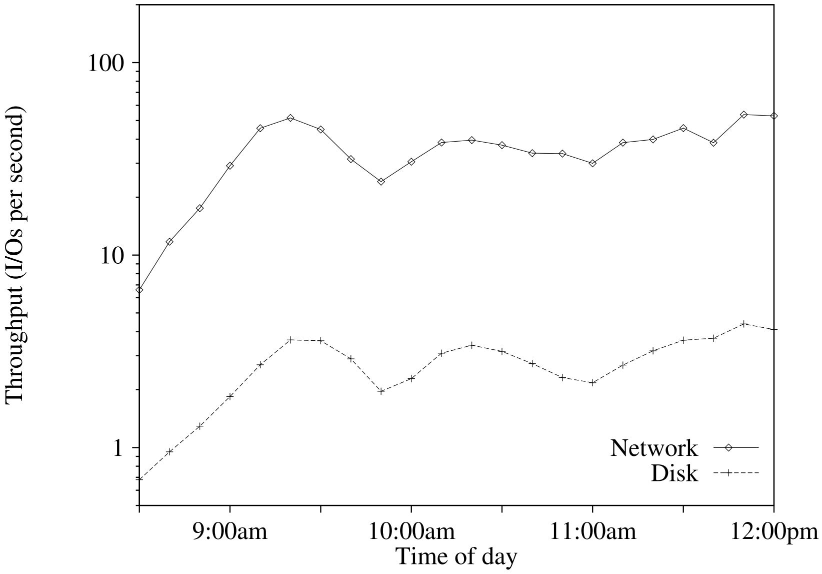

Figure 7.13: Administrative File Server Throughput, Thursday 4/28

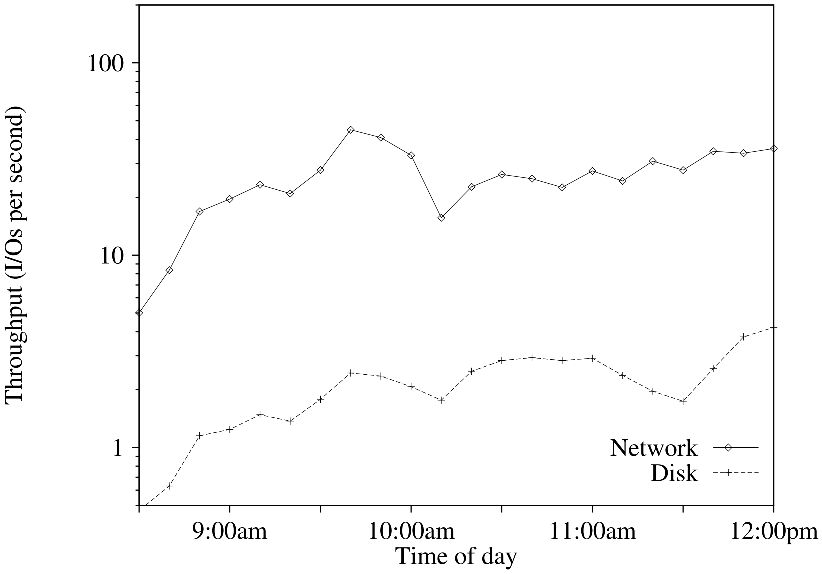

Figure 7.14: Administrative File Server Throughput, Friday 4/29

In both the student and administrative environments, overall network and disk throughput are highly correlated. The correlation coefficients ranged from a low of 0.552 on the administrative file server, to a high of 0.964 on the student file server. The low, of 0.552, is much lower than most other correlation values. The second lowest network to disk throughput correlation is 0.776. A correlation coefficient of 1 implies dependence.

This correlation is more apparent in the highly volatile throughput curves of the administrative environment, Figures 7.11 through 7.14. But, both the administrative and student environments have a high correlation between network and disk throughput.

Figures 7.15 through 7.21 are regression plots of network IOs as a predictor of disk IOs. In these figures, we graphically show the high correlation of network and disk IOs in most of our traces. The equation for each regression line and their corresponding R-squared values are give on each figure. In most of the examined trace data, there was a high correlation within the trace. However, if we combine all the data points from all traces in a particular environment onto a single curve, the correlation coefficient is low. Each trace has a very distinct regression line. These differences prevent us from making generalized conclusions about the correlation between network and disk IOs for. Such a generalization would require study of several more file servers.

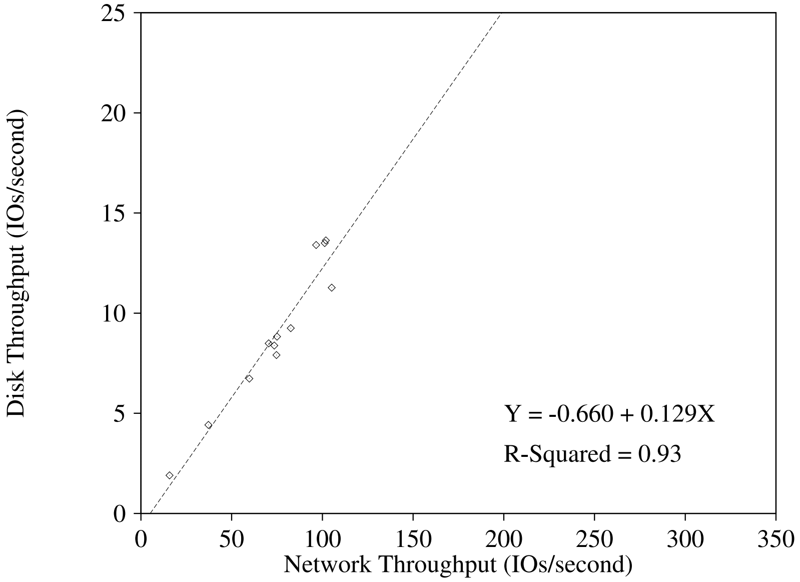

Figure 7.15: Regression Analysis Student File Server, Monday 4/11

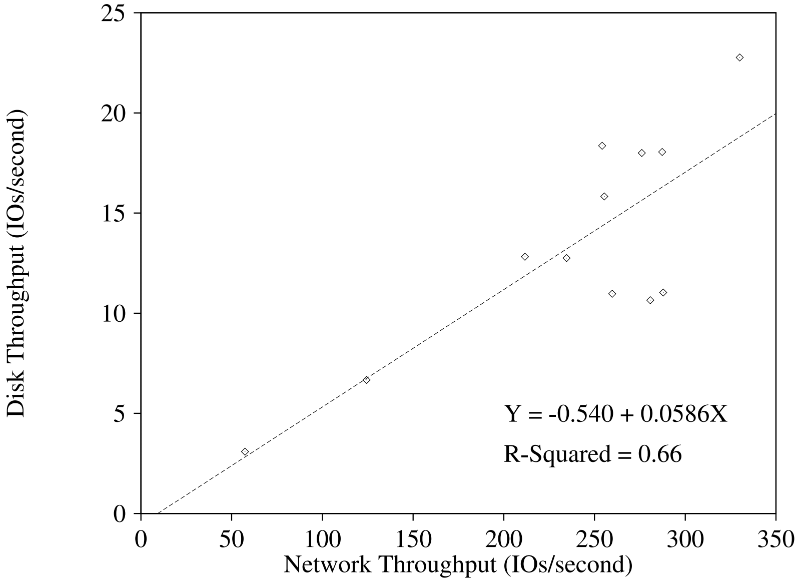

Figure 7.16: Regression Analysis Student File Server, Wednesday 4/13

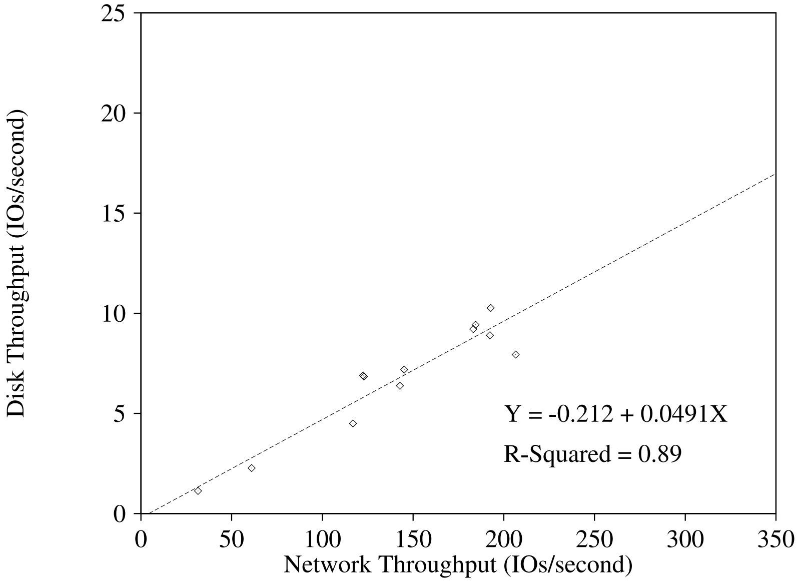

Figure 7.17: Regression Analysis Student File Server, Friday 4/15

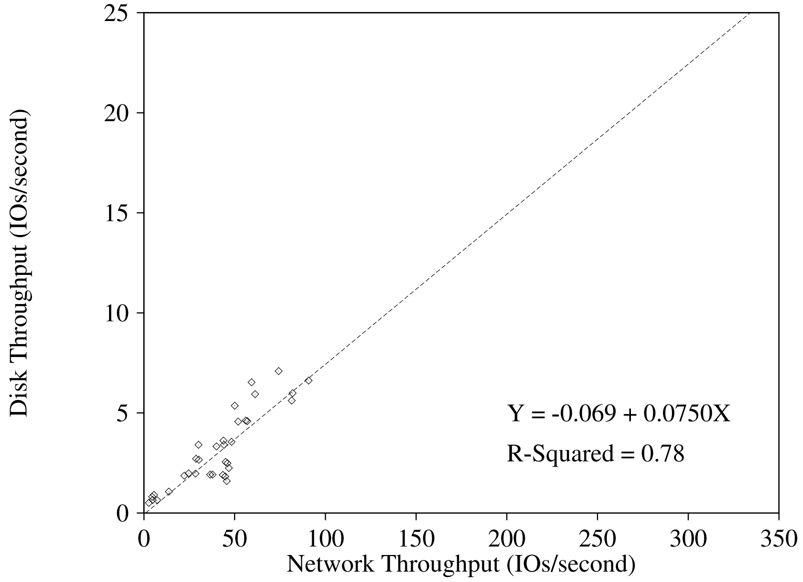

Figure 7.18: Regression Analysis Administrative File Server, Monday 5/2

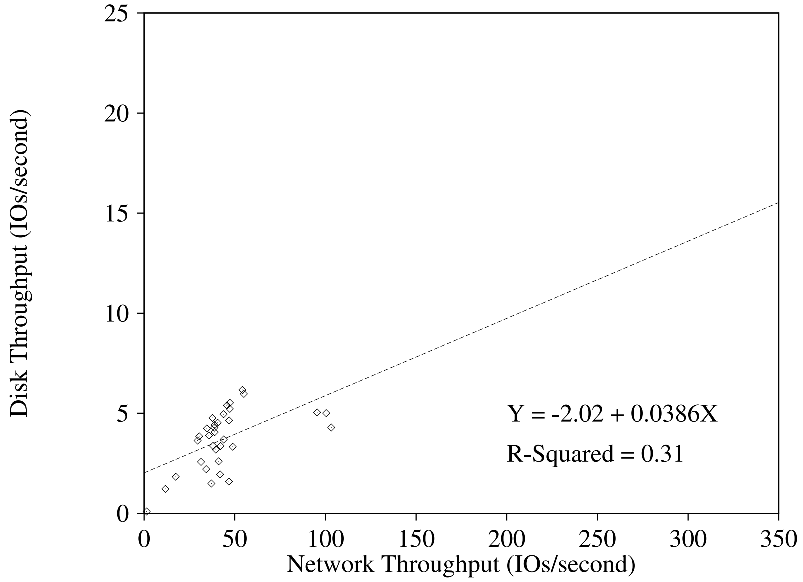

Figure 7.19: Regression Analysis Administrative File Server, Wednesday 4/27

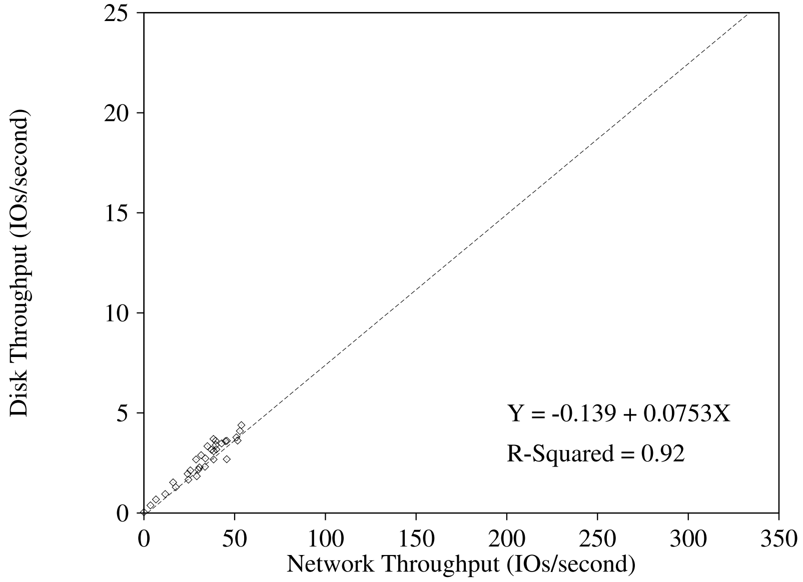

Figure 7.20: Regression Analysis Administrative File Server, Thursday 4/28

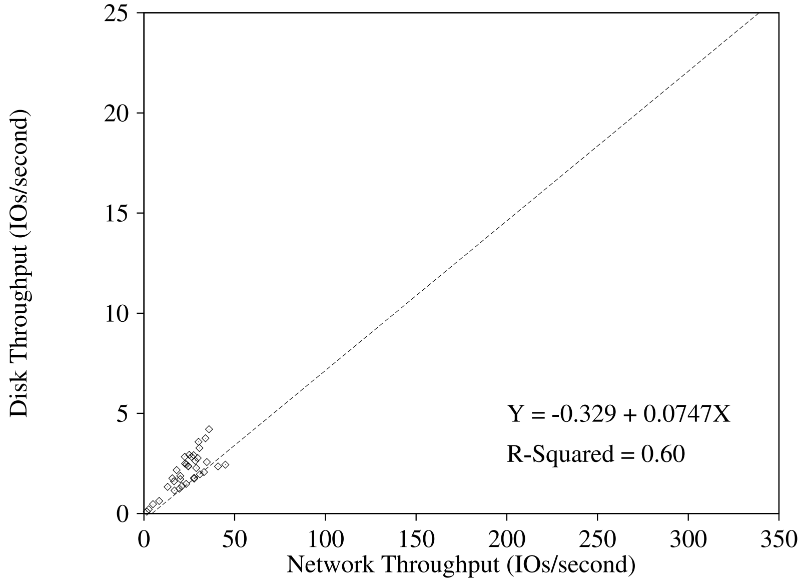

Figure 7.21: Regression Analysis Administrative File Server, Friday 4/29

Chapter 8

Summary and Conclusions

We have examined and characterized file server disk and network workload traces for two local area network (LAN) environments. We then combined the two workload analyses and compute their correlation. Both measured systems were in a university environment, but used for different applications.

8.1 File Server Disk Workloads

Our analysis of the disk workloads on each file server provides qualitative and quantitative information about file server disk traffic. These results can be used to parameterize synthetic workloads for server disk studies. We presented request size, throughput, read/write ratio, seek distance, response time, and cylinder access statistics and measure the read/write ratio and throughput fluctuation throughout the day.

A file server's file cache is designed to take advantage of locality of client data requests. We would expect server disk accesses to consecutive, or nearby, data blocks to be resolved within the file cache.

However, our analysis shows disk requests had a large degree of locality in both the seek distance and cylinder access distributions. Response time statistics, combined with this locality demonstrate that disk adapter and/or controller caching is an effective method of increasing performance in a file server environment.

The large number of single block write requests, and the large percentage of write requests overall, suggest that the server's *lazy-write* cache algorithm was ineffective. The large write *hot-spots* also support this conclusion. These observations, when taken together, suggest that further tuning of disk write request performance could increase overall server performance.

The diskless student environment had a higher throughput than the diskfull administrative environment. Also, the student file server's disk workload was more write intensive than was the administrative file server's. The response time and seek distance distributions were similar in both environments and both environments had intense disk write *hot-spots*.

8.2 File Server Network Workloads

Our analysis of network workloads on each file server also provides qualitative and quantitative information on network traffic. These results can be used to tune file server performance, or for synthetic workload models for network file server studies. We presented measurements of request size, throughput, read/write ratio, and response time statistics. Also, we measured the throughput and read/write ratio fluctuation throughout each day.

Client request response time was significantly improved by the large file server cache. In both environments, we found more than 90% of client request's were serviced in less than 5 ms. The file server's cache is effective in servicing multiple workstations. Requests ranged in size from zero bytes (status requests, etc.) to a maximum of 1024 bytes, the largest size allowed on our LAN.

As with the disk workloads, the diskless student environment had a much higher throughput than the administrative environment. We observed peaks in network activity in both environments, with most fluctuations in the administrative environment. The read/write ratios of the workloads were read intensive, as expected from client workstations. The student environment had an average read/write ratio of 7:1 and the administrative environment had an average of 15:1.

8.3 Combined Analysis

Finally, we compare our analyses of disk and network workloads and draw conclusions about how network traffic affects disk activity. We examine the read/write ratios and throughput of server disk and network client IOs during the same time periods and calculate correlation. This analysis is useful in understanding the file cache mechanisms of NetWare, for tuning file servers, and for future capacity planning.

We found that disk workload throughput was highly correlated to network workload throughput. We observed this even though network workload throughput was an order of magnitude larger than the disk throughput. The correlation coefficients for each trace period ranged from a low of 0.552 to a high of 0.964, with a mean of 0.841. Graphically, it did not appear that the network workload read/write request ratio was correlated to the disk workload's read/write ratio. We did not compute read/write ratio correlation statistics.

The file server cache was effective in reducing client read request response time and the number of read requests made on the disk channel. More than 90% of requests, in both environments, were serviced from the file server's file cache and did not incur disk requests. Those requests that were not serviced from the file cache were often serviced from the disk controller's cache. The *lazy-write* algorithm used for client write requests was effective in reducing client write request response time, but not at eliminating numerous disk write requests.

8.4 Future Work

Our traces were conducted on servers in a university environment, running the Novell NetWare operating system. There are many other types of network file server environments such as database, mixed diskless/diskfull systems, print servers, and a variety of network operating systems that were not represented in this study. Our environments also service particular applications, such as word-processing and spreadsheets. Different disk and network workloads may result from use of different client applications. Our observations may not generalize to other environments. Traces of servers in many other environments will be needed to make general conclusions.

Since our traces were captured and analyzed, both file servers have been replaced with newer equipment. Many workstations on both LANs have also been upgraded or replaced. In addition, numerous workstations have been added in both environments. It would be interesting to see if our observations are valid for the current environments. Measurements of the new file servers would also provide useful information about the capacity planning process.

Overall, we feel there is valuable information to be gained from tracing more and different environments. Specifically, further study could determine if our observations pertain to different environments, provide generalizations for file server behavior in particular environments or in all environments. Also, a study of the upgraded systems in the same student and administrative environments would allow us to characterize performance and workload changes that occur over time.

Appendix A

License Monitor, ASCII Report Format

The license monitor stores its data in a Paradox[15] table. Table A.1 contains a description of the fields contained in the table.

Field name	Field description
Station	name of the workstation this record is for
Application	application that is being operated on
Start date	date the application was launched
Start time	time the application was launched
End date	date the application terminated
End time	time the application terminated

Table A.1: License Monitor Database Fields

Appendix B

Network Sniffer, Binary Data File Format

The network sniffer[31] stores its data in a binary file. The format of the binary file is detailed in Tables B.1 through B.3. All packets have the fields listed in Table B.1 which are followed by either the request fields, shown in Table B.2, or the reply fields, shown in Table B.3.

The raw binary data file from the network sniffer is processed by a program we wrote. The output of that program is a *compressed* version of the entire network header. The format of the compressed binary data file contains the fields shown in Table B.4.

Field name	Size	Field description
Sniffer fields		
device	2 bytes	logical network interface
time stamp	4 bytes	time stamp in microseconds
status	2 bytes	status bits (internal use)
size	2 bytes	packet length in bytes
ptr buff	4 bytes	pointer to packet's data (internal use)
pre next	4 bytes	pointer to next packet (internal use)
Ethernet fields		
destination	6 bytes	packet destination address
source	6 bytes	packet source address
type	2 bytes	packet protocol identifier
IPX fields		
check sum	2 bytes	IPX check-sum value
length	2 bytes	IPX data length
control	1 byte	IPX control bits
type	1 byte	IPX packet type
destination network	4 bytes	IPX destination network
destination node	6 bytes	IPX destination node
destination socket	2 bytes	IPX destination socket
source network	4 bytes	IPX source network
source node	6 bytes	IPX source node
source socket	2 bytes	IPX source socket
NCP fields		
type	2 bytes	NCP request type
sequence	1 byte	NCP sequence number
connection low	1 byte	low-order byte of NCP connection number
task	1 byte	NCP task identifier
connection high	1 byte	high-order byte of NCP connection number

Table B.1: Network Sniffer Header Fields

Field name	Size	Field description
function	1 byte	NCP operation to be performed
subfunction length	2 bytes	length of subfunction data
subfunction	1 byte	subfunction identifier
request data	from IPX length	data if any

Table B.2: Network Sniffer NCP Request Header Fields

Field name	Size	Field description
completion code	1 byte	NCP request completion status (return code)
connection status	1 byte	NCP connection status
reply data	from IPX length	returned data if any

Table B.3: Network Sniffer NCP Reply Header Fields

Field name	Size	Field description
time stamp	4 bytes	time stamp in micro-seconds
gated	1 byte	0 if source node and source address are the same (the packet did not travel through a gateway), 1 otherwise
node	6 bytes	Ethernet address of workstation
psize	2 bytes	size of entire packet
connection	2 bytes	NCP connection number
task	1 byte	NCP task identifier
sequence	1 byte	NCP sequence number
operation	2 bytes	NCP function and subfunction
offset	2 bytes	offset of data within IPX packet
size	2 bytes	size of NCP data

Table B.4: Network Sniffer Compressed Data File Fields

Appendix C

SCSI Analyzer, Binary Data File Format

The SCSI bus analyzer[24] stores its data in a binary file. The binary file contains the fields shown in Tables C.1 and C.2 for each SCSI operation.

Field name	Size	Field description
flags	1 byte	status of SCSI bus
it	1 byte	initiator and target (bits)
lun	1 byte	logical unit number of target
qtype	1 byte	
qid	1 byte	
count	4 bytes	
operation	1 byte	SCSI operation code
lbn	3 bytes	starting logical block number
length	1 byte	number of logical blocks requested
reserved	7 bytes	structure padding
ltm stamp	4 bytes	time stamp at beginning of request
end stamp	4 bytes	time stamp at end of request

Table C.1: SCSI Bus Analyzer Trace Fields, (6) Commands

Field name	Size	Field description
flags	1 byte	status of SCSI bus
it	1 byte	initiator and target (bits)
lun	1 byte	logical unit number of target
qtype	1 byte	
qid	1 byte	
count	4 bytes	
operation	1 byte	SCSI operation code
lbn	4 bytes	starting logical block number
reserved	1 byte	reserved in SCSI II specification
length	2 bytes	number of logical blocks requested
control	1 byte	SCSI request control byte
reserved	3 bytes	structure padding
ltm stamp	4 bytes	time stamp at beginning of request
end stamp	4 bytes	time stamp at end of request

Table C.2: SCSI Bus Analyzer Trace Fields, (10) Commands

Bibliography

[1] American National Standard for Information Systems. *Small Computer System Interface (SCSI), X3T9.2 Revision 17B*, 1985.

[2] Mary G. Baker, John H. Hartman, Michael D. Kupfer, Ken W. Shirriff, and John K. Ousterhout. Measurements of a distributed file system. In *Proceedings of 13th ACM Symposium on Operating Systems Principles*, pages 198–212. Association for Computing Machinery SIGOPS, October 1991.

[3] P. Biswas and K. K. Ramakrishnan. File chanracterizations of vax/vms environments. In *10th International Conference on Distributed Compuing Systems*, pages 227–234, May 1990.

[4] P. Biswas, K.K. Ramakrishnan, Prabudda Biswas, and D. Towsley. Trace driven analysis of write cacheing policies for disks. In *1993 ACM Sigmetrics & Performance*, pages 13–23, June 1993.

[5] R. R. Bodnarchuk and R. B. Bunt. A synthetic workload model for a distributed system file server. *Proc. the ACM SIGMETRICS'91, Conf. on Measurement and Modeling of Computer Systems*, pages 50–59, 1991.

[6] Laura Chappell. *NetWare LAN Analysis*. SYBEX Inc., 2021 Challenger Drive, Alameda, CA 94501, 1990.

[7] Patrick H. Corrigan and Aisling Guy. *Buliding Local Area Networks With Novell's NetWare*. M&T Books, M&T Publishing, Redwood City, CA 94063, 1989.

[8] John K. Ousterhout et. al. A trace driven analysis of the unix 4.2 bsd file system. In *Tenth Symposium on Operating System Principles*, pages 15–24, December 1994.

[9] Fujitsu of America. *Fujitsi M2266A 1.2 Gb Disk Drive*, 1993.

[10] R. Gusella. A measurement study of diskless workstation traffic on an ethernet. *IEEE trans. on commun.*, COM-38, 9:1557–1568, 1990.

[11] Matt Hagen and Morgan Adair. Netware v3.x operating statistics exposed. *Novell NetWare Application Notes*, July 1991.

[12] Fred Halsall. *Data Communications, Computer Networks and Open Systems*. Addison-Wesley, Reading, Massachusetts, 1992.

[13] Stephen Houser. *Application License Server*. University of Southern Maine, 96 Falmouth Street, Portland, ME 04103, 1994.

[14] IBM. *IBM Type 0664 Disk Drive*, 1993.

[15] Borland International Inc. *Borland Paradox Database*. Borland International Inc., 1800 Green Hills Road, P.O. Box 660001, Scotts Valley, CA 95067, 1994.

[16] Ron Lee. Identifying test workloads in lan performance analysis. *Novell NetWare Application Notes*, May 1992.

[17] Ron Lee. Workload characterization, evaluating performance tests based on workload. *NetWare Connection*, pages 22–26, July/August and September/October 1993.

[18] Ron Lee. Workload characterization of netware clients and servers using a protocol analyzer. *Novell NetWare Application Notes*, July 1994.

[19] M. N. Nelson, B. Welch, and J. Ousterhout. Caching in the sprite network file system. In *Preprints for the Eleventh ACM Symposium on Operating Systems Principles*, pages 34–47, [11] 1987.

[20] Network General Corporation, 4200 Bohannon Drive, Menlo Park, CA 94025. *Network General Network Analyzer*, 1993.

[21] Novell Inc., South Provo, Utah, 84606. *NetWare System Interface Technical Overview*, 1990.

[22] Novell Inc., South Provo, Utah, 84606. *NetWare Device Driver Functional Specification*, 1992.

[23] Novell Inc., South Provo, Utah, 84606. *NetWare NLM Programming Guide*, 1992.

[24] Peer Protocols Inc., 4101 Westerly Place Suite 105, Newport Beach, CA 92660. *Peer Protocol SCSI Analyzer*, 1994.

[25] Willie Tejada Janice Perkins and Robert Jones. Netware v3.11 server tuning and optimization. *Novell NetWare Application Notes*, June 1992.

[26] J. Love R. Karedla and B. G. Wherry. Cacheing strategies to improve disk system performance. *IEEE Computer*, 27:38–46, March 1994.

[27] K.K. Ramakrishnan, Prabudda Biswas, and Ramakrishna Karedla. Analysis of file i/o traces in commercial computing environments. *Performance Evaluation Review*, June 1992.

[28] Alan J. Smith. Disk-cache miss ratio analysis and design considerations. *ACS Transactions on Computer Systems*, pages 161–203, August 1985.

[29] Pawel Szzerbina. Novell's netware core protocol. *Dr. Dobbs Journal*, 207:123–132, November 1993.

[30] Andrew S. Tanenbaum. *Computer Networks*. Prentice Hall, Inc., Englewood Cliffs, New Jersey, 07632, 1981.

[31] Univeristy of Delft. *Delft Network Analyzer Project*, 1993.