## FILE EXCHANGE PROTOCOLS

## <u>AND</u>

## ZERO CONFIGURATION NETWORKING

James Werner Graduate Student in Information Technology Clarkson University Professor Matthews December 13<sup>th</sup>, 2006

# TABLE OF CONTENTS

1. Introduction	
1.1 Client-Server	
1.2 Peer-to-Peer	
1.3 Hybrid	
2. AppleShare File Protocol (AFP)	
2.1 AFP Mac OS X Releases	
2.2 Brief Background on AppleTalk	4
2.2.1 Sockets	4
2.2.2 Nodes	5
2.2.3 Networks	5
2.2.3.1 Non-Extended Networks	5
2.2.3.2 Extended Networks	6
2.2.4 Zones	6
3. File Transfer Protocol (FTP)	7
3.1 Data Transfer	7
4. Server Message Block (SMB)	
4.1 Client-Server Relationship	8
5. Network File System (NFS)	8
6. Andrew File System	9
7. WebDAV	9
7.1 WebDAV Background	9
8. Zero Configuration Networking	9
8.1 Using IP Addresses With and Without DHCP	
9. Conclusion	
10. Ethereal Traces	
10.1.1 AFP Over TCP Trace	12
10.1.2 AFP Over Bonjour (ZeroConf) Trace	

10.1.3 FTP Trace	14
10.1.4 SMB Trace	15
11. Comparison Table	16
Works Cited	17

### 1. Introduction

The purpose of this paper is to describe in detail some of the major file exchange protocols that are currently used on computing systems. The protocols that will be covered are AFP (Apple-Share File Protocol), FTP (File Transfer Protocol), SMB (Server Message Block), NFS (Network File System), WebDAV, and Zero Configuration Networking.

### 1.1 Client-Server

In a client-server networking architecture, personal computing applications connect to a server. Some different server types include database servers, file servers, and mail servers. A good example of this relationship would be viewing a website on a computer, with one's computer being the client and the machine where the website is hosting being the server.

### **1.2 Peer-to-Peer**

Peer-to-peer networks consist of mostly clients interacting with each other rather than using a shared server. In recent years, peer-to-peer networks have become increasingly popular and controversial for the use of file sharing. Prominent files sharing applications include *LimeWire, Morpheus*, and *Napster*. These networks work by having the client act as both a client and a server (commonly referred to as Ad-Hoc).

### 1.3 Hybrid

Hybrid networking consists of both client-server and peer-to-peer.

## 2. AppleShare File Protocol (AFP)

The AppleShare File Protocol is a product of Apple Computer, which was introduced with Mac OS 7.6 in circa 1996. AFP, as it is commonly referred to, is a presentation layer (used for the delivery and formatting of the information) protocol that is used for the exchange of files on Mac OS X and a classic Mac system. The first version of this was AFP 1.x, in System 7, which was later succeed by AFP 2.0 and then AFP 3.0 version in Mac OS X. AFP 3.0 relies entirely on TCP/ IP for communication, only using the companies' AppleTalk software as a service discovery protocol. <sup>1</sup> AFP had used an AppleTalk server until the current version of Mac OS X 10.4, which only utilizes AppleTalk for its communication medium (See Figure 2).

2.1 AFF	Mac	OS X	Releases
---------	-----	------	----------

AFP Release	Mac OS Version	Comments
3.0	10.0.3 Server	• Notable for its use of UNIX-style POSIX permis- sions model. The largest file size that could be transferred was two terabytes.

<sup>1</sup> AppleShare File Protocol

AFP Release	Mac OS Version	Comments
3.1	10.2 Server	• Changes for Kerberos support, Network File Sys- tem (NFS), and Secure AFP connections through a secure shell. The new capacity for file exchange was increased to 8 terabytes.
3.2	10.4 Server	• Added more support for Access Control Lists and the ability to view metadata files in Apple's Spot- light searching technology. All of these changes occurred in Mac OS X Server 10.4. AFP currently supports access control list permissions.

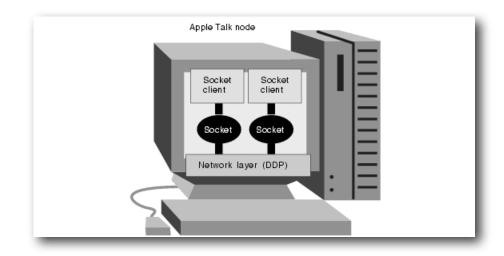
#### **2.2 Brief Background on AppleTalk**

AppleTalk began in 1984 with the introduction of the Macintosh computer. Over the past few years, Apple has been using TCP/IP functionality instead of AppleTalk. AppleTalk can be used as a hybrid (client-server and peer-peer). There are four components to the AppleTalk network: *sockets, nodes, networks,* and *zones.* 

#### 2.2.1 Sockets

A *socket* is an addressable location in an AppleTalk node. It is where the AppleTalk software directly interacts with the network layer. The software, which is limited to the upper layer, contains processes known as socket clients.

Socket clients have one or more sockets, which send and receive datagrams. Certain sockets can be assigned statically or dynamically. Statically assigned sockets are reserved for use by certain protocols or other processes. Dynamically assigned sockets are assigned by the Datagram Delivery Protocol (DDP) to socket clients upon request.<sup>2</sup>



<sup>2</sup> AppleShare File Protocol

Figure 1. An AppleTalk Socket.

#### 2.2.2 Nodes

*Nodes* are devices that are connected on the AppleTalk network. These devices can be any device, ranging from another Macintosh, to a Windows machine, or even a printer. Printers cannot upload files onto the network, but can only print. Each specific node on the AppleTalk network is restricted to only one network and one zone.

#### 2.2.3 Networks

AppleTalk *networks* are a series of nodes attached to a cable. This cable can be connected to a bridge or a router to further extend the connections of the network. There are two different kinds of networks involved: non-extended networks and extended networks.

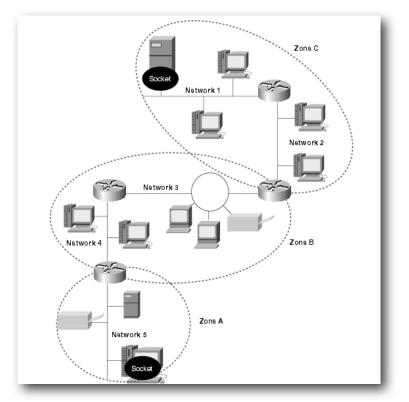


Figure 2. AppleTalk Internetwork Components.

### 2.2.3.1 Non-Extended Networks

*Non-extended networks* are physical segments that have only a single network segment, which is in the number range of 1 to 1024 (See Figure 3). Each specific node number must only have one zone configured and can transfer information over Ethernet or wireless. Currently, non-extended networks are not as widely used as extended networks.

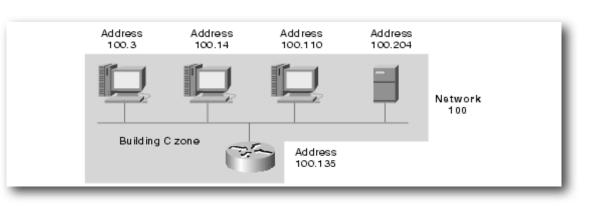


Figure 3. Non-Extended Network Diagram.

#### 2.2.3.2 Extended Networks

An *extended network* is a physical network segment that is given multiple network numbers (also known as a cable range) (See Figure 4). Extended networks can contain a single network number or multiple sequential network numbers. These extended networks can transfer information over Ethernet or wireless.

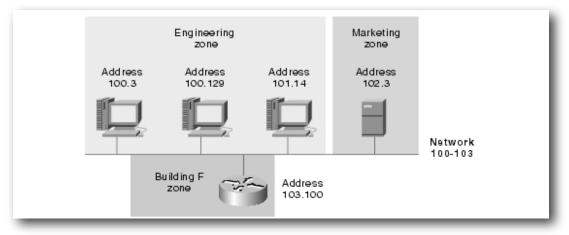


Figure 4. Extended Network Diagram.

#### 2.2.4 Zones

AppleTalk *zones* are groups of nodes or networks that are defined when the network administrator administers the network. In example, Clarkson University's buildings have their own subnetwork within Clarkson's network (machine1.snell.clarkson.edu, machine2.sc.clarkson.edu). Snell would be its own zone as well as the Science Center. As far as an autonomous system is concerned, AppleTalk zones are controlled by a single entity, with various subnetworks underneath it. Autonomous systems and AppleTalk zones essentially operate in the same fashion.

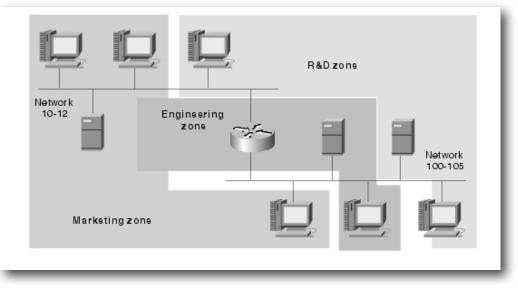


Figure 5. An AppleTalk Zone.

## 3. File Transfer Protocol (FTP)

File Transfer Protocol, or FTP, is a protocol used for the sharing of files across a network. Other functions of FTP include indirectly controlling a remote machine, and transfer data regarding reliability. The first version of FTP was first introduced in 1971, and was used as the first proposed file transfer mechanism.<sup>3</sup> In 1982, The File Transfer protocol definition was changed to: "a protocol for file transfer between HOSTs on the ARPANET, and its main function was to transfer files efficiently.<sup>4</sup>"

### 3.1 Data Transfer

Using FTP, files can only be exchanged through a data connection (See Figure 4). The control connection is for the transfer of commands, which describe the functions to be performed, and the replies to these commands.<sup>5</sup> There are other commands that deal with the transfer of data between host machines.

Included in FTP is the Protocol Interpreter and the Data Transfer Protocol (DTP). The user protocol interpreter has control over sending commands and translating the data it receives. The server protocol interpreter reads the commands, sends responses, and directs the DTP for initialization of the data connection for the transfer of data.

- 4 Postel, 2
- 5 Postel, 2

<sup>3</sup> Postel, 2

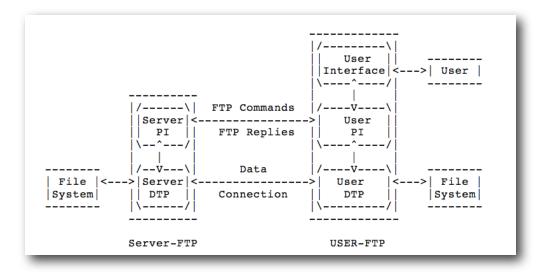


Figure 6. The FTP model.

## 4. Server Message Block (SMB)

The Server Message Block, or SMB, protocol was invented at IBM by Dr. Barry Feigenbaum. The idea for SMB originated when Dr. Feigenbaum wanted to take the DOS "Interrupt 33" local file access into a networked file system.<sup>6</sup> The version that is used now has been refitted by Microsoft. In 1998 Microsoft launched its own file protocol based off of SMB, called Common Internet File System (CIFS), intended for use with the Windows workgroup. In 1998, Sun Microsystems also created its own version of SMB called WebNFS.

The SMB protocol has been very significant to Windows. In the case of computer users who do not use Windows, an open source implementation called Samba was created for this specific purpose.

#### 4.1 Client-Server Relationship

SMB operates from a client-server approach. This approach is done when a client connects to the server, and the server replies to the message. SMB servers can provide access to let the servers file become visible on the network. Hard disks can be hidden (through changing permissions) on the network, and peripherals (printers, etc.) can be connected remotely. It is common for SMB to slow down network traffic, as the server broadcasts itself over an entire subnetwork or network.

## 5. Network File System (NFS)

The Network File System (NFS), protocol was created by Sun Microsystems in 1984. NFS allows a computer to connect and obtain files or data on a remote machine, but viewing the remote disk as local disk. The file system used for NFS contains a hierarchical with regular files that are encoded in UTF-8 for multilingual uses.<sup>7</sup>

<sup>6</sup> Server Message Block

<sup>7</sup> RFC 3530

The NFS mount function permits NFS clients to attach remote directory trees to a mount point in the local file system.<sup>8</sup> In order to mount an NFS file system, permissions are needed for the primary file system machine. Once the user has been authenticated, remote access is granted to the file system.

## 6. Andrew File System

The Andrew file system, or AFS, is a distributed networked file system with regards to scalability and security. AFS was first developed at Carnegie Mellon University. The files on the system are cached onto a local machine and when the files are located on the network, the local machine can read the files more quickly than through other networked file systems. AFS uses Kerberos to authenticate users, as well as implementing access control lists for the directories and users.

## 7. WebDAV

Web-based Distributed Authoring and Versioning called WebDAV, is a group of HTTP extensions for editing and managing files on remote web servers collaboratively. WebDAV is primarily a client-server relationship.

### 7.1 WebDAV Background

WebDAV first took shape when Jim Whitehead asked the W3C to hold special meetings about distributed authoring on the Web, and the possibility of proposing solutions to that problem. A web browser called *WorldWideWeb*, which was written by Tim Berners-Lee, had the option of editing and viewing web pages. As the internet progressed, Whitehead wanted to have editing and viewing pages included in newer software, since as it had faded out. After the W3C meeting, several people agreed that an IETF working group should be created, in order to standardize a new protocol. Soon after, the WebDAV group chose to focus in distributed authoring.

## 8. Zero Configuration Networking

The Zero Configuration Networking Group was chartered in 1999 to promote the work of Dynamic Configuration for IPv4 Link-Local Addresses that the group had created. This protocol was shipped with the Macintosh Operating System (9 and X), the Windows Operating System (98 through the present), and in nearly all network printers from most major suppliers.

Zero Configuration Networking (ZeroConf) allows a user to connect to devices, either wired or wireless, through a small network without a DHCP or DNS server (See Figure 7).

<sup>8</sup> http://www.ussg.iu.edu/usail

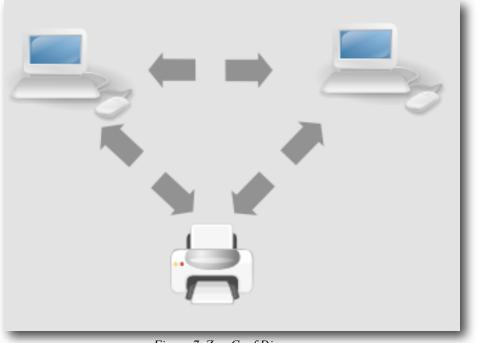


Figure 7. ZeroConf Diagram.

A computer can exchange files with another machine, sharing files via peer-to-peer, and can connect to a printer without the need of a LAN. With the exception of Zero Configuration Networking, problems can arise when an LAN shuts down and information cannot be transmitted.

#### 8.1 Using IP Addresses With and Without DHCP

An IP address is needed to connect to a device on a network. The two most ideal ways to retrieving an IP address are manual input and DHCP.

#### Manual

Windows and Macintosh lets one input the IP address manually through a graphical user interface (GUI) application.

#### DHCP

Once an IP address is located through the DHCP client, the computer can be configured to connect to other devices. In example, this is used for a client connecting to a DHCP server for an amount of time. In the case of Zero Configuration Networking, the DHCP client is not used in favor of the link-local address.

#### Selection of an IP Address

In 1998, Mac and Windows were outfitted with the ability to obtain a link-local address. If the DHCP server happens to fault, a link-local address is assigned with the prefix address of

169.254.x.x using Multicast DNS (mDNS). The host machine must complete a scan of the network using an Address Resolution Protocol (ARP) probe to resolve any conflicts, which then claims a link-local address so that the user can be assigned an address and use the network for printing or sharing a document on the local network. The address that the computer is assigned cannot be placed into service until the same, if any, address on the network is free for use.

#### IPv6

IPv6 is the next generation of internet protocol that will be implemented over the course of the next 5-10 years (IPv4 addresses would be exhausted within the next several years). Zero Configuration Networking currently only supports IPv4.

### 9. Conclusion

All of these major networking protocols transfer data from one point to another, whether it be peer-to-peer, client-to-server, or both. All major operating systems have connections to network protocol installed. AFP is currently the most useful file sharing protocol, as the ease-of-use and its hybrid connections cannot be matched by any other protocol. The AFP connection is built into the Mac OS X operating system, can connect very quickly, and the remote disk can be mounted as if it was a local drive. The transfer of data and the exchange of ideas will keep networks advancing for many years to come.

### **10. Ethereal Traces**

The AFP over TCP, AFP without TCP, SMB, and FTP traces are in example on pages 12 through 15. The complete details of these traces can be found on the project website at <a href="http://www.clarkson.edu/projects/itl/mpX52/fa2006/wernerjs/traces.htm">http://www.clarkson.edu/projects/itl/mpX52/fa2006/wernerjs/traces.htm</a>.

No	Time	Source	Destination	Protocol Info	Info
145	145 5.900720	128.153.144.154	63.111.69.12	НТТР	GET /weather/local/13699 HTTP/1.1
146	146 5.977156	63.111.69.12	128.153.144.154	НТТР	HTTP/1.1 200 OK[Unreassembled Packet]
147	147 5.987486	63.111.69.12	128.153.144.154	НТТР	Continuation or non-HTTP traffic
148	148 5.987574	128.153.144.154	63.111.69.12	TCP	52626 > http [ACK] Seq=301 Ack=2921 Win=64240 Len=0
149	49 5.995911	128.153.144.154	128.153.22.188	AFP	FPGetFileDirParms request: Vol=1 Did=2 Name=Audio.mov
150	150 5.997388	128.153.22.188	128.153.144.154	TCP	afpovertcp > 52566 [ACK] Seq=6123 Ack=1083 Win=32746 Len=0 TSV=1545539865 TSER=426559326
151	151 5.998506	128.153.22.188	128.153.144.154	AFP	FPGetFileDirParms reply
152	152 5.998557	128.153.144.154	128.153.22.188	TCP	52566 > afpovertcp [ACK] Seq=1083 Ack=6252 Win=49199 Len=0 TSV=426559326 TSER=1545539865
153	153 6.041575	LinksysG_78:92:b3	Spanning-tree-(for-bric STP	STP	Conf. Root = 8189/00:d0:00:98:a4:8f Cost = 123 Port = 0x8002
154	154 6.054265	63.111.69.12	128.153.144.154	НТТР	Continuation or non-HTTP traffic
155	155 6.064502	63.111.69.12	128.153.144.154	НТТР	Continuation or non-HTTP traffic
156	156 6.064676	128.153.144.154	63.111.69.12	TCP	52626 > http [ACK] Seq=301 Ack=5841 Win=65535 Len=0
151	57 6 N748N3	61 111 KO 17	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	нттр	
P Frame	e 149 (110 b	Frame 149 (110 bytes on wire, 110 bytes captured)	captured)		
<b>D</b> Ethen	net II, Src: 0	Ethernet II, Src: 00:17:f2:4b:6e:ce (00:17:f2:	7:f2:4b:6e:ce), Dst: Cisco_35:4f:dc (00:d0:d3:35:4f:dc)	35:4f:d	1c (00:d0:d3:35:4f:dc)
▷ Interr	net Protocol,	Internet Protocol, Src: 128.153.144.154 (128.1	28.153.144.154), Dst: 128.	153.2	.53.144.154), Dst: 158.153.22.188 (128.153.22.188)
D Transi	mission Contr	Transmission Control Protocol, Src Port: 52566	566 (52566), Dst Port: afpo	vertcp	(52566), Dst Port: afpovertcp (548), Seq: 1039, Ack: 6123, Len: 44
Data :	Data Stream Interface	ace			
¬ Apple	Apple Filing Protocol	lo:			
Col	mmand: FPG	Command: FPGetFileDirParms (34)			
Pad	F				
Vol	Volume id: 1				
DIC	DID: 2				
File	File bitmap: 0xe93f	93f			
Dir	Directory bitmap: 0xa33f	): 0xa33f			
Patl	Path: Audio.mov				

File Exchange Protocols a 10.1.2 AFP Ove					-								-	
	Info	FPGetFileDirParms reply: misc. err (-5014)	FPGetFileDirParms request: Vol=1 Did=974823 TTCP Retransmission1 EDGetFileDirParms request: VAI=1 Did=074823	arms reply: misc. err (	FPResolveID request	[TCP Retransmission] FPResolveID request	U reply	FPGetFileDirParms request: Vol=1 Did=2 Name=Audio.mov LTCP Retransmission1 FDGetFileDirDarms request: VAI=1 Did=2 Name=Audio mov	FDGetFileDirParms really	FPDelete request: Vol=1 Did=2 Name=Audio.mov	[TCP Retransmission] FPDelete request: Vol=1 Did=2 Name=Audio.mov	FDDalata raniv	187 46 R41757       107 168 1106       107 168 1106       107 168 1106       100 bytes captured)         Frame 173 (110 bytes on wire, 110 bytes captured)       101 17:12:40:66::ce ()0117:12:40:66::ce (), D5:13:108 (192.168.1.108 (192.168.1.106 (192	
	<sup>3</sup> rotocol	AFP	AFP	AFP	AFP	AFP	AFP	AFP	AFP	AFP	AFP	AFD	AppleCom 192.168.1.1( ort: afpovertu	
	Destination	192.168.1.106	192.168.1.108 192 168 1 108	192.168.1.106	192.168.1.108	192.168.1.108	901'T'89''''	192.168.1.108 192.168.1.108	192.168.1.106	192.168.1.108	192.168.1.108	192 168 1 106	192 168 1 106 es captured) :17:f2:4b:6e:ce), Dst: 192.168.1.106), Dst: 51762 (51762), Dst Pr	
	Source	192.168.1.108	192.168.1.106 192.168.1.106			192.168.1.106	801.1.801.261	192.168.1.106 192.168.1.106			192.168.1.106	182 46 841252 192 168 1 108	182         45         841357         103         168         106         459           Frame         173         (110         bytes on wire, 110         bytes captured)         Ethernet II, Src: 00:17:f2:4b:6e:ce (00:17:f2:4b:6e:ce), Dst: 192.168.11.         Transmission         control         proced         str. 102         168.11.         transmission         control         proced         str. 192.168.11.168         proced         str. 192.168.11.168         str. 192.168	
	No. + Time	150 40.444579	152 40.444738 154 40 447036	156 40.450839	158 40.450992	160 40.453475	102 40.4039/1	1/3 44.995100 174 44 996123	176 45.001325	179 46.063346	180 46.064325	182 46 841252	<ul> <li>Frame 173 46 841357 193 16</li> <li>Frame 173 (110 bytes on v Ethernet II, Src: 00:17:f2:4</li> <li>Internet Protocol, Src: 192.</li> <li>Transmission Control Protoco</li> <li>Data Stream Interface</li> <li>Apple Filing Protocol</li> <li>Command: FPGetFileDirP Pad</li> <li>Volume id: 1</li> <li>DID: 2</li> <li>File bitmap: 0xe93f</li> <li>Path: Audio.mov</li> </ul>	

# 10.1.2 AFP Over

File Exchange Protocols and Zero Configuration Networking

No	Time	Source	Destination	Protocol	Info
ICI	5.U05384	00'C'SCI'871	192.158.1.105	FIP -	Kesponse: 226 File receive UK.
154	154 5.066607	192.168.1.106	128.153.5.66	FТР	Request: SITE CHMOD 644 /afs/clarkson.edu/users/w/e/wernerjs/Audio.mov
157	157 5.232152	128.153.5.66	192.168.1.106	FTP	Response: 200 SITE CHMOD command ok.
159	159 5.262602	192.168.1.106	128.153.5.66	FTP	Request: SITE UTIME 20061210180010 /afs/clarkson.edu/users/w/e/wernerjs/Audio.mov
160	160 5.403667	128.153.5.66	192.168.1.106	FTP	Response: 500 Unknown SITE command.
162	162 5.404684	192.168.1.106	128.153.5.66	FTP	Request: QUIT
163	163 5.535003	128.153.5.66	192.168.1.106	FTP	Response: 221 Goodbye.
168	168 5.623846	192.168.1.106	128.153.5.66	FTP	Request: NOOP
170	170 5.757916	128.153.5.66	192.168.1.106	FTP	Response: 200 NOOP ok.
172	172 5.761064	192.168.1.106	128.153.5.66	FTP	Request: CWD /afs/clarkson.edu/users/w/e/wernerjs
173	173 5.911605	128.153.5.66	192.168.1.106	FTP	Response: 250 Directory successfully changed.
175	175 5.912453	192.168.1.106	128.153.5.66	FTP	Request: PASV
176	176 6.075530	128.153.5.66	192.168.1.106	FTP	Response: 227 Entering Passive Mode (128,153,5,66,130,35)
181	181 6.247323	192.168.1.106	128.153.5.66	Ц	Request: LIST - a
Frame	e 154 (129 b	Frame 154 (129 bytes on wire, 129 bytes captured)	captured)		
Etherr	net II, Src: 0	0:17:f2:4b:6e:ce (00:17	Ethernet II, Src: 00:17:f2:4b:6e:ce (00:17:f2:4b:6e:ce), Dst: Cisco-Li_9b:40:29 (00:0f:66:9b:40:29)	i_9b:4	0:29 (00:0f:66:9b:40:29)
D Intern	net Protocol,	Internet Protocol, Src: 192.168.1.106 (192.168.1.106),		3.5.66 (	Dst: 128.153.5.66 (128.153.5.66)
Transr	mission Contr	rol Protocol, Src Port: 520	772 (52072), Dst Port: ftp (	.21), St	D Transmission Control Protocol, Src Port: 52072 (52072), Dst Port: ftp (21), Seq: 228, Ack: 469, Len: 63
∀ File Tr	∀ File Transfer Protocol (FTP)	col (FTP)			
∆ SIT	TE CHMOD 64	14 /afs/clarkson.edu/users		\r\n	
ц	Request command: SITE	mand: SITE			
ď	Request arg:	Request arg: CHMOD 644 /afs/clarkson.edu/users/w	n.edu/users/w/e/wernerjs/Audio.mov	Audio.m	NOV

9.496435         192.168.1.106         5MB           9.502143         192.168.1.106         5MB           9.502143         192.168.1.106         5MB           9.502467         192.168.1.106         5MB           9.502467         192.168.1.106         5MB           9.502467         192.168.1.106         5MB           9.502467         192.168.1.106         192.168.1.106           9.508653         192.168.1.106         192.168.1.106           9.508653         192.168.1.106         192.168.1.106           9.508653         192.168.1.106         192.168.1.106           9.508653         192.168.1.106         5MB           9.508653         192.168.1.106         5MB           9.508653         192.168.1.106         5MB           140 (172 bytes on wire, 172 bytes captured)         5mB           aret Probocol, Src Port: 51751 (51751), Dat Port: netbios-san         5Session Service           DS Session Control Protocol)         5rc port: 51751 (51751), Dat Port: netbios-san           DS Session Service         00:17:12-4b.166.1.106         5MB           D ord Count (WCT): 15         6rot Count: 00         5Session Service           D ord Count (WCT): 15         6rot Count: 10         182           A Paramete	No	Time	Source	Destination	Protocol Info	
192.168.1.106 SMB 192.168.1.106 SMB 192.168.1.106 SMB 192.168.1.106 SMB ytes captured) ytes captured) (192.168.1.106), Dst: 192.168.1.108 (192.168.1.106), Dst: 192.168.1.108 (192.168.1.106), Dst: Port: netbios-ssn : 51751 (51751), Dst Port: netbios-ssn	140	9.496435	192.168.1.106	192.168.1.108		Trans2 Request, FIND FIRST2, Pattern: Vaudio.mov
192.168.1.108       SMB         192.168.1.106       SMB         ytes captured)       SMB         00:17:12:4b:6e:ce), Dst: AppleCom_20       20         (192.168.1.106), Dst: 192.168.1.108       SMB         :: 51751 (51751), Dst Port: netbios-ssn       3001)	142	9.502143	192.168.1.108	192.168.1.106		Trans2 Response, FIND_FIRST2, Error: STATUS_NO_SUCH_FILE
192.168.1.106         SMB           192.168.1.108         SMB           ytes captured)         SME           00:17:f2:4b:6e:ce), Dst: AppleCom_20         20           (192.168.1.106), Dst: 192.168.1.108         SME           :: 51751 (51751), Dst Port: netbios-ssn         2001)	144	9.502467	192.168.1.106	192.168.1.108	SMB .	Trans2 Request, FIND_FIRST2, Pattern: \Audio.mov
192.168.1.108         SMB           ytes captured)         00:17:12:4b:6e:ce), Dst: AppleCom_2G           012.1751 (51751), Dst Port: netbios-ssn         51751 (51751), Dst Port: netbios-ssn           :: 51751 (51751), Dst Port: netbios-ssn         0001)	146	9.508471	192.168.1.108	192.168.1.106		Trans2 Response, FIND_FIRST2, Error: STATUS_NO_SUCH_FILE
ytes captu 00:17:f2:: (192.16) :: 51751 ( :: 51751 ( :: 51751 ( :: 51051 )	148	9.508653	192.168.1.106	192.168.1.108		NT Create AndX Request, Path: \Audio.mov
00:17;f2: (192.16) :: 51751 ( 192.16) :: 51751 ( 192.16) :: 51751 ( 192.16) :: 51751 ( 192.16)	Frame	a 140 (172 b	ytes on wire, 172 bytes	s captured)		
(192.16) :: 51751 ( ::	<b>D</b> Ethern	net II, Src: 0(	0:17:f2:4b:6e:ce (00:1	17:f2:4b:6e:ce), Dst: AppleC	Com_20	0:81:e1 (00:30:65:20:81:e1)
:: 51751 (	D Intern	net Protocol, S	Src: 192.168.1.106 (19	92.168.1.106), Dst: 192.168	8.1.108	(192.168.1.108)
<ul> <li>&gt; NetBIOS Session Service</li> <li>&gt; SMB (Server Message Block Protocol)</li> <li>&gt; SMB (Server Message Block Protocol)</li> <li>&gt; Trans2 Request (0x32)</li> <li>word Count (WCT): 15</li> <li>Total Data Count: 34</li> <li>Total Data Count: 1664</li> <li>Max Setup Count: 0</li> <li>Max Setup Count: 10</li> <li>Max Setup Count: 10</li> <li>Max Setup Count: 0</li> <li>Flags: 0x0000</li> <li>Timeour: Return immediately (0)</li> <li>Reserved: 000</li> <li>Prameter Count: 34</li> <li>Prameter Count: 40</li> <li>Prameter Count: 1664</li> <li>Prameter Count: 10</li> <li>Prov. Prameter Count: 10</li> <li>Prov. Prameter</li></ul>	D Transn	mission Contri-	ol Protocol, Src Port: 51	1751 (51751), Dst Port: netb	oios-ssn	(139), Seq: 3152, Ack: 16785, Len: 106
<ul> <li>SMB fleader</li> <li>SMB Header</li> <li>Nord Count (WCT): 15</li> <li>Tinda Parameter Count: 34</li> <li>Total Parameter Count: 34</li> <li>Total Data Count: 0</li> <li>Max Parameter Count: 10</li> <li>Reserved: 00</li> <li>Parameter Count: 34</li> <li>Par</li></ul>	D NetBI(	OS Session S	service			
• F SNB Header           * Trans2 Request (0.32)           * Trans2 Request (0.32)           Total Parameter Count: 15           Total Data Count: 1644           Max Setup Count: 10           Max Setup Count: 0           Max Setup Count: 10           Max Setup Count: 10           Max Setup Count: 0           Max Setup Count: 0           Max Setup Count: 0           Reserved: 00           Parameter Count: 34           Parameter Count: 1           Reserved: 00           Sub Count: 10           Reserved: 00           Sub Count: 0           Sub Count: 10           Reserved: 00           Sub Count: 10	¬ SMB (	Server Mess	age Block Protocol)			
<ul> <li>&gt; Trans2 Request (0x.32) Word Count (WCT): 15 Word Count (WCT): 15</li> <li>Yotal Data Count: 10 Max Parameter Count: 10 Max Stata Count: 00 Flags: 0x0000 Parameter Offset: 68 Data Count: 34 Parameter Offset: 68 Data Count: 34 Data Count: 34</li></ul>	SMB	B Header				
Word Count (WCT): 15 Total Parameter Count: 34 Total Data Count: 10 Max Parameter Count: 1664 Max Data Count: 10 Reserved: 00 Primoeut: Return immediately (0) Reserved: 000 Parameter Oriset: 68 Data Cont: 0 Parameter Oriset: 68 Data Count: 0 Subcommand: FIND_FIRST2 (0x0001) Reserved: 00 Subcommand: FIND_FIRST2 (0x0001) Reserved: 00 PriND_FIRST2 Parameters	¬ Trar	ns2 Request (	(0×32)			
Total Parameter Count: 34 Total Parameter Count: 10 Max Parameter Count: 10 Max Setup Count: 10 Max Setup Count: 10 Max Setup Count: 10 Max Setup Count: 0 Reserved: 000 D Reserved: 000 Parameter Count: 34 Parameter Count: 34 Parameter Count: 34 Parameter Count: 1 Reserved: 00 Parameter Count: 1 Reserved: 00 Setup Count: 1 Reserved: 00 S	8	Vord Count (V	VCT): 15			
Total Data Count: 0 Max Parameter Count: 10 Max Stata Count: 16644 Max stata count: 16644 Max stata count: 16644 Max stata count: 0 Reserved: 00 Flags: 0x0000 Timeout: Return immediately (0) Reserved: 00 Parameter Offset: 68 Data Count: 1 Reserved: 0 Setup Count: 1 Reserved: 00 Subcommand: FIND_FTRST2 (0x0001) Byte Count (BCC): 37 Parameters	ŕ	otal Paramete	er Count: 34			
Max Parameter Count: 10 Max Data Count: 16644 Max Data Count: 16644 Max Data Count: 0 Reserved: 00 Flags: 0x0000 Timeout: Return immediately (0) Reserved: 0000 Timeout: Return immediately (0) Reserved: 0000 Parameter Count: 3 Parameter Count: 3 Parameter Count: 3 Parameter Count: 0 Data Offset: 68 Data Count: 0 Data Count: 0 Data Count: 0 Setup Count: 1 Reserved: 00 Setup Count: 1 Reserved: 00 Setup Count: 1 Parameter (BCC): 37 Padding: 00000 D FIND_FIRST2 (0x0001)	F	otal Data Cou	int: 0			
Max Data Count: 16644 Max Setup Count: 0 Reserved: 00 Flags: 0x0000 Timeout: Return immediately (0) Remout: Batum immediately (0) Reserved: 00 Data Offset: 68 Data Count: 1 Reserved: 00 Setup Count: 1 Reserved: 00 Sub Count: 37 Padding: 000000 FIND_FIRST2 Parameters	Σ	1ax Paramete	r Count: 10			
Max Setup Count: 0 Reserved: 00 Flags: 0x0000 Timeout: Return immediately (0) Reserved: 0000 Parameter Count: 34 Parameter Count: 34 Parameter Count: 34 Parameter Count: 34 Parameter Offset: 68 Data Offset: 68 Data Offset: 68 Data Count: 0 Setup Count: 1 Reserved: 00 Subcommand: FIND_FIRST2 (0x0001) Byte Count (BCC): 37 Padding: 000000 D FIND_FIRST2 Parameters	Σ	1ax Data Coul	nt: 16644			
Reserved: 00         Flags: 0x0000         Timeout: Return immediately (0)         Reserved: 0000         Parameter Count: 34         Parameter Count: 34         Parameter Count: 34         Parameter Count: 34         Parameter Offset: 68         Data Count: 0         Data Count: 0         Data Count: 1         Reserved: 00         Setup Count (BCC): 37         Byte Count (BCC): 37         Padding: 000000         P FIND_FIRST2 Parameters	≥.	1ax Setup Co	unt: 0			
<ul> <li>▷ Flags: 0x0000</li> <li>Timeout: Return immediately (0)</li> <li>Reserved: 0000</li> <li>Parameter Count: 34</li> <li>Parameter Cont: 34</li> <li>Parameter Offset: 68</li> <li>Data Count: 0</li> <li>Data Count: 0</li> <li>Data Count: 1</li> <li>Reserved: 00</li> <li>Subcommand: FIND_FIRST2 (0x0001)</li> <li>Byte Count (BCC): 37</li> <li>Padding: 000000</li> <li>D FIND_FIRST2 Parameters</li> </ul>	~	Reserved: 00				
		lags: 0x0000	-			
	F	imeout: Retu	rn immediately (0)			
	~	Reserved: 00	00			
	ď	arameter Col	unt: 34			
	ď	arameter Off	set: 68			
		) ata Count: 0				
		) ata Offset: 0				
	S	Setup Count:	1			
	2	teserved: 00				
	S	Subcommand	: FIND_FIRST2 (0x000	1)		
	ά.	yte Count (B(	CC): 37			
	đ	adding: 000(	000			
		IND_FIRST2	Parameters			

## **10.1.4 SMB Trace**

File Exchange Protocols and Zero Configuration Networking

## 11. Comparison Table

	Client-Server Peer-to-Peer Hybird	File Size Limit
AFP	Hybrid	16 TB*
FTP	Client-Server	50-200MB*
NFS	Client-Server	Varies
SMB	Client-Server	2 GB
AFS	Client-Server	2 GB
WebDAV	Client-Server	500 MB

\*AFP - 16 TB as of OS 10.2. OS 10.3 has not stated any kind of file limit.

\*FTP - 50-200 MB depending on the FTP client.

#### Works Cited

- 1. "Andrew File System." <u>Andrew File System</u>. 17 Oct 2006.Wikimedia Foundation, Inc.. 10 Nov 2006 < http://en.wikipedia.org/wiki/Andrew\_File\_System>.
- 2. "AppleShare Filing Protocol." <u>Apple Filing Protocol</u>. 17 Sep 2006. Wikimedia Foundation, Inc.. 19 Oct 2006 <a href="http://en.wikipedia.org/wiki/Apple\_Filing\_Protocol">http://en.wikipedia.org/wiki/Apple\_Filing\_Protocol</a>.
- 3. "Apple Talk." <u>Apple Talk</u>. 2006. Cisco Systems. 17 Oct 2006 <http://www.cisco.com/univercd/cc/td/doc/cisintwk/ito\_doc/applet.htm>.
- 4. Cheshire, Stuart. Zero Configuration Networking. 1st. Sebastopol, CA: O'Reilly, 2005.
- 5. Hertel, Christopher. Implementing CIFS. 1st. Open Publication License, 2003.
- 6. "Network File System." <u>Network File System</u>. Indiana University. 17 Oct 2006 <<u>http://www.ussg.iu.edu/usail/network/nfs/overview.html></u>.
- 7. Potsel, J. "FILE TRANSFER PROTOCOL (FTP)." <u>RFC 959</u> Oct 1985 17 Oct 2006 <<u>http://www.ietf.org/rfc/rfc0959.txt</u>>.
- 8. "Server Message Block." <u>Server Message Block</u>. 11 Oct 2006.Wikimedia Foundation, Inc., 17 Oct 2006 <a href="http://en.wikipedia.org/wiki/Server\_Message\_Block">http://en.wikipedia.org/wiki/Server\_Message\_Block</a>>.
- Shepler, S. "Network File System (NFS) version 4 Protocol." <u>RFC 3530</u> Apr 2003 17 Oct 2006 <<u>http://www.ietf.org/rfc/rfc3530.txt</u>>.
- Whitehead, Jim. "WebDAV Resources." <u>WebDAV Resources</u>. 04 Oct 2006. Oregon State University. 17 Oct 2006 < http://www.webdav.org>.