

12-2016

## **An open source approach to serve a large number of computer users using block-level streaming**

Max D. Torres

*The University of Texas Rio Grande Valley*

Follow this and additional works at: <https://scholarworks.utrgv.edu/etd>



Part of the [Computer Sciences Commons](#)

---

### **Recommended Citation**

Torres, Max D., "An open source approach to serve a large number of computer users using block-level streaming" (2016). *Theses and Dissertations*. 169.

<https://scholarworks.utrgv.edu/etd/169>

This Thesis is brought to you for free and open access by ScholarWorks @ UTRGV. It has been accepted for inclusion in Theses and Dissertations by an authorized administrator of ScholarWorks @ UTRGV. For more information, please contact [justin.white@utrgv.edu](mailto:justin.white@utrgv.edu), [william.flores01@utrgv.edu](mailto:william.flores01@utrgv.edu).

AN OPEN SOURCE APPROACH TO SERVE A LARGE  
NUMBER OF COMPUTER USERS USING  
BLOCK-LEVEL STREAMING

A Thesis

by

MAX D. TORRES

Submitted to the Graduate College of  
The University of Texas Rio Grande Valley  
In partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

December 2016

Major Subject: Computer Science



AN OPEN SOURCE APPROACH TO SERVE A LARGE  
NUMBER OF COMPUTER USERS USING  
BLOCK-LEVEL STREAMING

A Thesis  
by  
MAX D. TORRES

COMMITTEE MEMBERS

Dr. Fitratullah Khan  
Chair of Committee

Dr. Mahmoud Quweider  
Committee Member

Dr. Hansheng Lei  
Committee Member

December 2016



Copyright 2016 Max D. Torres

All Rights Reserved



## ABSTRACT

Torres, Max D., An Open Source Approach to Serve a Large Number of Computer Users Using Block-Level Streaming. Master of Science (MS), December, 2016, 33 pp., 6 tables, 18 figures, 12 references, 12 titles.

There are several options for providing a large number of computers to users for their daily tasks. A typical setup may consist of a large number of computers where each relies on an HDD consisting of the required software, sufficient RAM, a capable CPU that meets the software requirements, and a stable network connection. This thesis proposes the use of the open-source AoE protocol to stream an OS to a user computer from a central server. Since the streaming is done from a well-protected central storage, the AoE protocol is less prone to failures compared to the traditional approach based on having a local HDD. Furthermore, AoE being open source, it avoids the need to purchase per seat licenses for streaming. The expected outcome of the proposed design is to create a central system consisting of servers and storage components to serve a large number of end-user diskless clients efficiently.





## TABLE OF CONTENTS

	Page
ABSTRACT .....	iii
TABLE OF CONTENTS.....	iv
LIST OF TABLES.....	vi
LIST OF FIGURES.....	vii
CHAPTER I. INTRODUCTION.....	1
CHAPTER II. SURVEY STUDY.....	6
The Discovery Method.....	6
The Initialization Phase.....	6
The Connection to the Virtualized Storage Target.....	7
The Streaming Protocol.....	8
AoE Header.....	9
ATA Header.....	10
Config/Query Header.....	10
Compatibility.....	11
Performance and protocol security.....	11
Best Practices for AoE Environment.....	12
AoE protocol comparison with other methods.....	13
Network Utilization and Delay.....	14
Processing Delay.....	14

Propagation Delay.....	15
Queuing Delay.....	15
Congestion Delay.....	15
CHAPTER III. PROPOSAL.....	17
CHAPTER IV. METHODOLOGY.....	18
Virtual Server Access through Terminal Machines.....	18
Virtual Server Access through Fat Clients.....	18
CHAPTER V. EXPERIMENT.....	19
Proposed Design.....	19
Current Results.....	24
CHAPTER VI. CONCLUSION.....	30
REFERENCES.....	32
BIOGRAPHICAL SKETCH.....	33

## LIST OF TABLES

	Page
Table 1: Diskless/TFTP Servers.....	24
Table 2: VHD Files.....	24
Table 3: Diskless Clients.....	25
Table 4: DHCP Servers.....	25
Table 5: Single Diskless Client Boot Statistics.....	26
Table 6: Maximum Recommended Values for a Diskless Environment.....	31



## LIST OF FIGURES

	Page
Figure 1: Boot Process of a Diskless Client.....	4
Figure 2: The ATA Over Ethernet Protocol Header.....	9
Figure 3: The ATA Header.....	10
Figure 4: The ATA Config/Query Header.....	10
Figure 5: Throughput as a function of load.....	14
Figure 6: Degradation with respect to Network Utilization.....	16
Figure 7: Diskless Process Overview.....	20
Figure 8: DHCP Request.....	21
Figure 9: TFTP Request.....	21
Figure 10: DHCP Request iPXE Stage.....	22
Figure 11: AoE Target Booting.....	22
Figure 12: DHCP Request OS Stage.....	23
Figure 13: User Login.....	23
Figure 14: RAM Usage While Booting (GB).....	27
Figure 15: Bytes Received At Completion Of The Boot Process (GB).....	27
Figure 16: Various Boot Times.....	28
Figure 17: Diskless Server CPU Single Core Usage.....	29
Figure 18: Load Test on a Single Vblade.....	29



## CHAPTER I

### INTRODUCTION

Current technologies allow us to have several options when deciding how to implement a large number of computers to serve computer users. The most common method is to provide a user with a computer consisting of an HDD (Hard Disk Drive) with the required software installed, sufficient RAM (Random Access Memory) and a capable CPU (Central Processing Unit) to meet the software requirements and a stable network connection.

Throughout the years, an IT manager would normally see an increased number of users, higher bandwidth consumption, oversubscribed networks, overtasked staff, increased number of applications, and growing storage needs. Typically, this is not reciprocated with resources by the organization. Therefore, an IT manager does not end up with the needed resources to handle the growth in usage of computing resources. The IT manager will benefit from looking for ways to handle the services in a more efficient and scalable ways to minimize the service overhead thus maintaining the computing environment with fewer additional resources to handle the growth.

One popular choice is to use a virtualized environment to provide an end-user with a virtual desktop. This can be achieved in different ways. One method is with a thin client architecture where each user desktop is hosted as a virtual machine in a server over the network. In this case, the user desktop is a thin client and functions only as an input/output terminal. The actual user OS configuration and files are hosted remotely on the server inside the respective virtual machine.



With a virtualized environment, one can reduce maintenance costs and the time the staff is required to maintain and operate systems. However, while this kind of setup may reduce costs in the long run, there is still a substantial cost that is tied to client licenses or seats when using software. Vendors providing the software and hardware, required to implement a virtualized environment, normally charge by seat licenses. In a very large setup, where one has many users such as students, faculty and staff, the costs associated with licensing may present a setback. This thesis focuses on an alternative open source solution that provides users with diskless-clients having with the same capabilities as that of a common PC (Personal Computer). The difference is that the OS (Operating System) will be streamed through the network from a central server instead of being installed on the local HDD which, under the correct configuration, gives better performance. This gives to all users a more efficient desktop experience by providing the exact same OS configuration, software, and network every time a user logs in.

In order to achieve the aforementioned system, the following technologies are used in the system:

- *Pre Execution Environment (PXE)*: an environment that allows computers to boot to a specific configuration through the network interface.
- *iPXE*: an open source network boot solution that provides an enhanced implementation of the PXE environment. iPXE was chosen over other options such as PXELINUX due to the fact that it supports the AoE protocol as a boot option.
- *AoE Protocol*: an open source network protocol which uses the ATA interface to communicate to devices over an Ethernet network.
- *Trivial File Transfer Protocol (TFTP)*: a protocol used to transfer files between two or more devices through the User Datagram Protocol (UDP).

- *Dynamic Host Configuration Protocol (DHCP)*: a protocol to provide computers and other devices with an IP address. This is usually done by a DHCP server.
- *Vmtouch*: a tool used on a UNIX based server to load the VHD (Virtual Hard Drive) file contents into RAM to improve performance.
- *Microsoft Windows Operating System*: a VHD file containing the Windows OS configured specifically for a diskless environment. The environment proposed in this document has been tested in 64-bit Windows 7.

The implementation of the aforementioned diskless virtualized environment can be done by IT departments in numerous scenarios where there is the need to serve a large number of users with the same OS configuration and hardware capabilities. Below are some examples:

- **Libraries**: Users need the same hardware and software capabilities on all desktops. Upon rebooting, the PCs get a clean state as the OS configuration is loaded again from the server. With this approach the IT management makes sure the new user has no access to any of the information of the previous user, respecting the user's privacy.
- **Computer Labs & Classrooms**: The IT management would only need to make sure the software needed by students is available as part of the OS image being loaded and that the PCs meet the software requirements.
- **For certain Staff/Faculty**: The diskless virtualized environment can be used in some cases where a large number of staff and faculty need the same OS configuration. Having the OS available as Read Only, this kind of environment forces staff and faculty to store their data on secured departmental file shares reducing the risk of data loss.

The protocol used to stream the OS from the central server is *ATA over Ethernet (AoE)* protocol. *Pre Execution Environment (PXE)* protocol is used to start the boot-from-network

process on a user’s desktop. This is achieved through the customization of an open source PXE implementation using iPXE. The following diagram gives a general idea of the process:

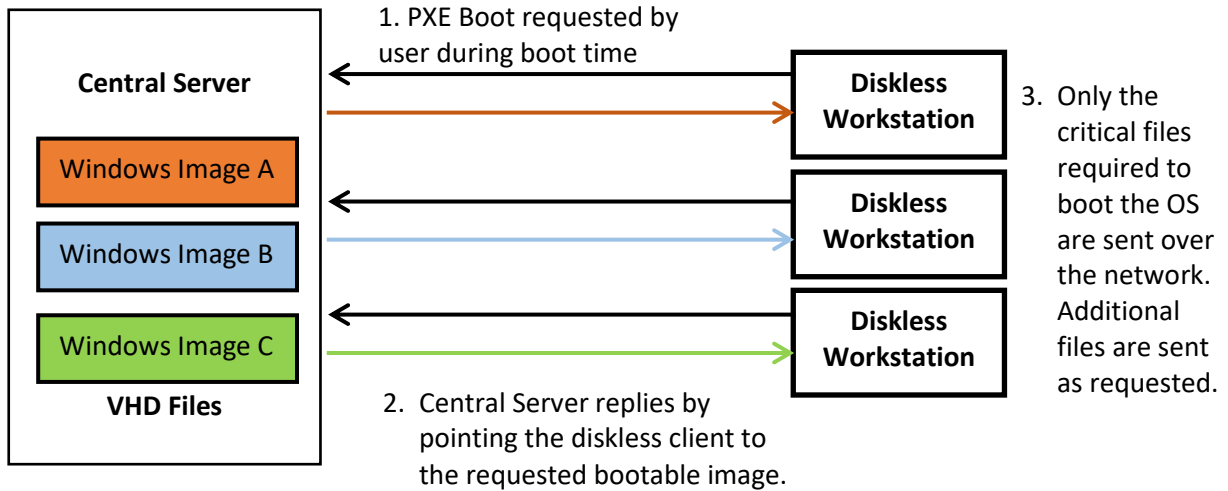


Figure 1: Boot Process of a Diskless Client

As shown in Figure 1, once the PXE settings have been configured on the central server, a user can boot to any preconfigured Virtual Hard Drive (VHD) file on the central server using a diskless workstation. When a VHD file is selected, only the OS files required to boot the diskless workstation are streamed using the AoE protocol. Once the machine has booted, additional files are sent dynamically as requested by the user’s computing environment.

This virtualized diskless-client environment can be implemented for a computer laboratory, workplace network, etc. when one needs to provide the same software configuration to a large number of users. The proposed configuration to implement the virtualized environment is described in detail in the experiment section of this document.

This thesis is organized as follows: Chapter 2 contains the Survey Study where the AoE protocol is described in detail as well as the compatibility, performance and security of this protocol is addressed. Chapter 3 contains the proposed setup to implement an AoE-based computer laboratory environment. Chapter 4 covers the methodology and review of current

related work in the area. Chapter 5 describes the experiment design and current results. Finally, Chapter 6 includes the conclusion and additional remarks on the proposed system.

## CHAPTER II

### SURVEY STUDY

#### **The Discovery Method**

The Dynamic Host Configuration Protocol (DHCP) is a protocol used to assign IP addresses to different devices such as servers and PCs on a network. Additional configuration information can be provided by a DHCP server if requested by a client. An IP address request to a DHCP server is the first step taken by the proposed virtualized environment when a diskless computer is booted. As in the case of any other desktop contacting a DHCP server, a diskless workstation sends a message to the DHCP server requesting an IP in order to boot. In response, the diskless workstation, also called diskless client, gets a dynamically assigned IP and a boot file information. This is then used by the diskless client to identify itself on the network and be able to communicate with a file transfer server to start the streaming process. To achieve this, a TFTP (Trivial File Transfer Protocol) server is used. The information required by the client to connect to the TFTP server is provided by the DHCP server using the *filename* and *next-server* DHCP options respectively. Once the diskless client makes contact with the TFTP server, the DHCP server is no longer involved in the process until OS boot time.

#### **The Initialization Phase**

The file that was provided by the DHCP in the previous request is known as the PXE boot file, more specifically, the iPXE boot file. This is the file that includes additional commands for the

diskless workstation. The following is a sample boot file given to a diskless workstation:

```
ln1    #!ipxe
ln2    dhcp net0
ln3    set keep-san 1
ln4    sanboot aoe:e1.0
```

The most important command in this file is *sanboot aoe:e1.0*. This command instructs the diskless client that a bootable AoE target is available at location: shelf 1, slot 0. The diskless client does not need to know if the AoE target actually resides in a physical NAS (Network-Attached Storage) device. While this is an option and such devices are sold by the company behind the AoE protocol, the setup proposed in this thesis utilizes an open source software based on AoE targets. This implementation called *Vblade* is a virtual Ethernet based blade that makes an AoE target available to clients on the network. This software based alternative as well as the drivers are distributed by Coraid Inc.

### **The Connection to the Virtualized Storage Target**

Vblade is a software based AoE target that requires an AoE initiator to export a local block storage target on the network, such as a VHD file. If a VHD file resides on the same server that has a vblade installation properly configured, the vblade service can be started to make the VHD file available to any clients on the same network pointing to the correct AoE target. An AoE target is defined by two parameters: [shelf].[slot].

Both parameters are assigned a number, such as 1.2, 3.10, 12.7, etc. This software based AoE target gives the option to the administrator of the system to decide if the diskless clients have access to the Windows based VHD file in Read-Write or Read-Only mode. RW mode can be used in diskless virtualized environments that are hosting multiple VHD files, each one used by one user at a time to avoid data corruption. An environment like this can be used if the

resources are available, however, this thesis focuses on a different approach. The focus of our approach is to use RW mode only while making modifications to the OS (e.g. software updates, driver installation, etc.) on a VHD file, e.g. when upgrading a software, or adding a software. Once the changes are completed and tested, the VHD file is made available to users in RO mode so that every user gets the same image at login time.

### **The Streaming Protocol**

The AoE protocol is mainly used in a SAN where it is used to access contents of a virtualized hard disk over the network. The protocol was created with the plan of being a simple protocol; its biggest trait relies on its simplicity (Coile 2009). This protocol was created by Coraid, Inc. as an inexpensive alternative to other SAN based solutions such as iSCSI. The primary use of this protocol is to offer to the user a SAN based storage system that relies on its simple authentication through MAC address verification. While this may seem like a security flaw, it is important to note that this is not a routable protocol. Its security relies on the infrastructure surrounding its current setup (Coile 2009). The proposed experiment implements VLANs to segregate reachable servers to achieve a higher level of network security.

There has not been much research on how to implement this protocol for purposes other than just a SAN based storage. The main function of this protocol, as the company behind the protocol states it, is to provide an inexpensive SAN based storage solution (Coraid 2009). This is the primary function of this protocol and there has not been much interest in using this protocol for other purposes such as OS streaming or host applications. The AoE protocol is at the heart of the work presented in this thesis.

The AoE protocol provides open standards that allows the user direct network access to disk

drives by client hosts such as web servers, mail servers, etc. While the recommended method to allow the aforementioned access is through commodity Ethernet switches provided by the company behind the protocols, reliable and fast access is also achievable at the software level. To ease the understanding of AoE protocol this subsection will break down the topics as follows: AoE Header, ATA Header, Compatibility, and Performance and protocol security.

### AoE Header

While the two classes of messages, ATA and Config/Query, each have their own fields, they both share a common format for the first 24 bytes of a message (Figure 2).

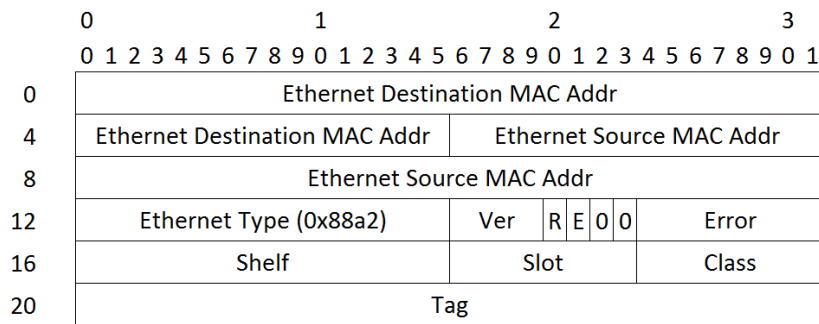


Figure 2: The ATA Over Ethernet Protocol Header

This common header provides enough information to send messages between client hosts and AoE servers. The common header has four functions. First, it provides a way to correlate responses with the requests. Second, it provides a way to discover the Ethernet address of an AoE server at some physical location in a rack storage blades. Third, the common header identifies requests from the responses. Lastly, the header contains error information. The two classes of messages, ATA and Config/Query are described below.



## ATA Header

The Advanced Technology Attachment (ATA) messages contain requests and responses to perform ATA transactions (Figure 3). In an ATA transaction, there are three possibilities: no data will be transferred, data will be written into the disk, or data will be read from the disk.

	0								1								2								3																	
	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
24	0 E O D O O A W								Err/Feature								Sector Count								Cmd/Status																	
28	lba0								lba1								lba2								lba3																	
32	lba4								lba5								0000000000000000																									
36	Data																																									

Figure 3: The ATA Header

An ATA message contains values to be put into parameter registers, and the flags field to control how these values are to be used. The functionality of ATA messages can be summarized as the method the AoE servers utilize to deliver the information to client hosts.

## Config/Query Header

This type of messages is used when advanced naming options are required in the current AoE setup. This is achieved by using a Config/Query message to set a configuration string of data of no more than 1024 bytes in an AoE server. This is later used by clients for matching and retrieval of this configuration data for server identification purposes (Figure 4).

	0								1								2								3																	
	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
24	Buffer Count																Firmware Version																									
28	00000000								AoE				CCmd				Config String Length																									
32	Config String ...																																									

Figure 4: The Config/Query Header

## **Compatibility**

Various implementations for different operating systems have been created to support AoE functionality. Coraid Inc., the company behind the AoE protocol, continues its support for the AoE protocol by releasing drivers periodically for both Unix and Windows based systems. Multiple third-party vendors have built upon these drivers to enhance the functionality of the protocol and compatibility for storage-related applications. One example is the StarWind Software company, a company that specializes in storage virtualization, that with their AoE Initiator solution allows Windows system clients use a graphical user interface to connect to remote storage devices over an Ethernet network using the AoE protocol. This software allows Windows systems to see the remote storage devices as if it was installed on the local machine.

## **Performance and protocol security**

A series of performance tests for ATA over Ethernet compared to Network Block Device protocols have been done in the past where different RAID setups have been used for such tests (Aatrokoski 2007). One of the important conclusions from these tests is the favorable performance impact of using jumbo frames for the AoE protocol (Aatrokoski 2007). Most of these tests on the AoE protocol have been done using software based targets. It is important to note that the usage of Jumbo frames is beneficial for both AoE and iSCSI protocols (Murray 2012, He 2009).

There have been few security assessments on the AoE protocol and many reviews claim that AoE is insecure. A white paper by *Security-Assessment.com* details the specifications of the AoE protocol, and claims that its simplicity is its weakness (Purvis 2006). The report includes the approaches an attacker can take against the AoE protocol and a general description on how to

perform such attacks. However, it also shows how these risks can be mitigated by segregating access to different hosts using VLANs, as it is proposed in this thesis.

While it has been stated by the AoE protocol own definition that this is not a routable protocol, there has been some research on it showing otherwise (Landowski 2011). However, the network setup proposed in this thesis restricts the protocol be available on only certain VLANs which makes it not routable.

### **Best practices for AoE Environment**

Since virtualization is a hot topic among IT managers, some best practices have surfaced and have been standardized when implementing and maintaining such systems. These best practices will be presented from the point of view of AoE environment which is initiated by *Pre eXecution Environment (PXE)* boot on a diskless client (Intel 1999).

A diskless client system cache plays an important role while streaming an operating system. While some methods that implement PXE boot have to copy the whole OS to the RAM to start using the system, the operating system streaming that can be achieved through the AoE protocol shows that only a small portion of the operating system is required in order to boot and start using an operating system on a diskless client. Tests on a default installation of a Windows 7 system, which is the OS used by the proposed environment, shows that only 0.7% of the total operating system “is needed for the machine to start booting, and 4% of the sectors are required to log in (Clerc 2010).”

Other tests have shown how large MTUs (Maximum Transmission Units), more specifically Jumbo frames set at 9000 bytes, can improve the overall performance of Ethernet traffic. The benefits of jumbo frames “offer throughputs much larger than a simplistic overhead

analysis might suggest (Murray 2012).” Further testing, as part of this proposal, revealed that a larger MTU also affects the performance of a TFTP server and the overall performance of the proposed setup. This is done by changing the TFTP configuration file on the TFTP server. Specifically, when the MTU is set to 9000, the TFTP option Blocksize should be changed accordingly to avoid fragmentation. A value of 8192 is used for the setup.

As any other implementation of a new system, the usage of the AoE protocol has advantages and disadvantages. Disadvantages of this setup is that the network administrator must be aware of an impending increase on network usage and must provide an effective management of resources on the central server that will be providing the virtual disk images. The network administrator should be also aware of the fact that while a central server may manage resources efficiently for more than one application, such as its CPU workload, it also requires additional configuration to handle cache and memory in a more effective way. The current proposal uses the Linux open source software vmtouch to load the VHD file into RAM to improve performance.

### **AoE protocol comparison with other methods**

As it was mentioned earlier, the usage of the AoE protocol to stream an operating system to a diskless-client was selected over other methods and protocols. The alternatives range from simple and commonly used methods such as booting from a local HDD, to other technologies that requires special switches and adapters such as Fibre Channel. Advantages of removing the hard disk drive from an end-user system is a reduced system failure rate and reduced power usage. Additionally, since AoE is open source it avoids seat licenses for streaming and operating system thereby reducing the maintenance costs.

## Network Utilization and Delay

Even the most reliable and robust networks experience different types of delays. While some of these delays are inevitable and have a predictable value, a knowledgeable network administrator should be aware of how these delays are impacting the network and what are the factors causing the delays. The ideal situation is to utilize as much of the available bandwidth without introducing congestion, and at the same time avoiding overloading the links, and operating between a safe range of network utilization as shown below (Jasem 2010):

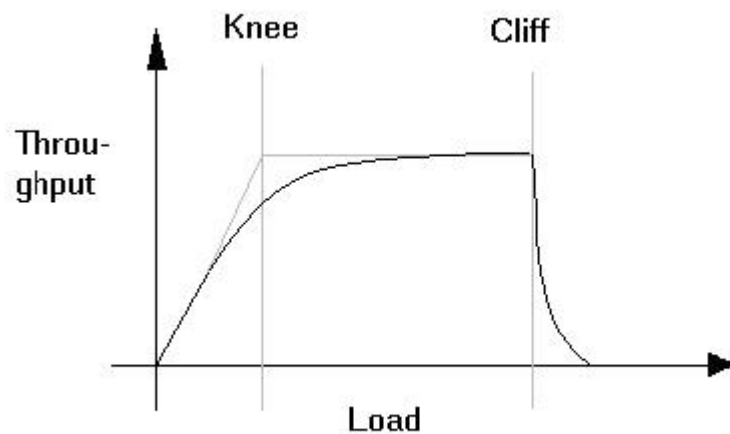


Figure 5: Throughput as a function of load (Jasem 2010)

Delays can be categorized into different type: Processing Delays, Propagation Delays, Queuing Delays, and Congestion Delays.

### Processing Delay

This type of delay is caused when network traffic is processed by inefficient hardware or software. The more traffic flows through the network, the higher this delay will be.

## **Propagation Delay**

This is the time it takes for the network traffic to traverse the length of the medium used. In other words, this is the time it takes for the signal to travel the medium physically.

## **Queuing Delay**

This is the time the network packets spend in routing or switching queues/buffers until these can be routed or switched to the destination. Flooded buffers is one of the causes for this type of delay. The average queuing delay can be determined by applying Little's law with the following formula:

$$Q_d = 1 / (\mu - \lambda)$$

Where

$\mu$ : Maximum packets per second that can be processed

$\lambda$ : Average packet arrival speed

## **Congestion Delay**

This delay is also called Transmission Delay, and it is the time it takes for the network packets to be transmitted through the medium due to the data transfer speed. Excessive traffic increases the congestion delay. The Congestion delay can be determined by the formula:

$$C_d = b / s$$

b: Total bits

s: Bits per second

An approximation of the total delay can be estimated by the formula:

$$\text{Delay: } D = D_0 / (1 - \mu)$$

$D_0$ : Delay at 0 utilization

$\mu$ : Network utilization (0 to 1)

By plotting this formula, the performance degradation with respect to network utilization can be seen in the graph below:

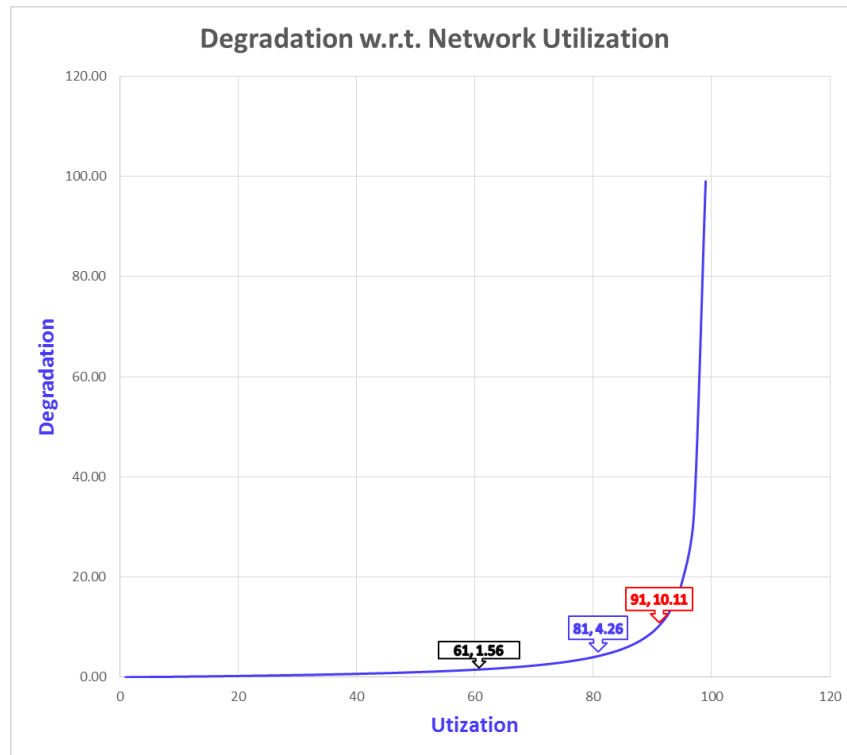


Figure 6: Degradation with respect to Network Utilization

To maintain a stable and reliable network, it is in the best interest of a network administrator to keep the sustained network utilization lower than 81%, where sustained utilization of 61% should be considered a warning to start thinking of improvements to push utilization below 50%.

## CHAPTER III

### PROPOSAL

It is proposed in this thesis that the AoE protocol in combination with PXE, and a SAN server be used to serve a Windows operating system to a diskless-client using VHD files to a large number of users.

The proposed method to experiment with this setup is to create different files storing a shared Read Only (RO) image. Examples of these files are Microsoft's VHD (Virtual Hard Disk) files, VMware's VMDK (Virtual Machine Disk), and Oracle's VDI (Virtual Disk Image) used on VirtualBox software. These files are typically used as a hard drive for a virtual machine and they may contain an exact replica of what one normally finds on a regular OS installation on a hard disk drive. The proposed file extension to use is VHD since it is the standard format for Windows-based virtualized images. Several VHD files with different OS configurations are tested to check which minimum hardware specifications are needed on the client as well as the server. Due to the high speed of the AoE protocol in combination with network reliability and speed, it is expected that streaming the OS through the network will not create a performance bottleneck on the diskless client.



## CHAPTER IV

### METHODOLOGY

#### **Virtual Server Access through Terminal Machines**

The thin client architecture with VDI (Virtual Desktop Infrastructure) is among the current options for a virtualized environment. This approach requires users to use “thin clients” as their terminal machines. A thin client is a PC with minimal hardware (sometimes diskless) that serves as a terminal to access a virtual machine hosted by a server on a remote location. A minimal amount of resources is used on the thin client since the heavy processing takes place on the server hosting the virtual machine. From the IT management perspective, this provides a way to centralize the maintenance on the server, reducing power consumption and costs, and improving security.

#### **Virtual Server Access through Fat Clients**

Another approach for implementing a virtualized environment is to use a Fat client, an HDD based PC, in combination with desktop virtualization. While this approach does not have the advantages seen in the thin client architecture approach such as reduced power consumption and costs, it does reduce issues related to hardware compatibility that may occur with a thin client. The presence of a hard drive on this approach adds an additional point of failure to the environment compared to the thin client.

## CHAPTER V

### EXPERIMENT

#### **Proposed Design**

The proposed setup will be conducted on different computer labs across campus which consist of different computer models with varying CPU speed, amount of RAM available, and network bandwidth. Different models, from basic systems to high end systems will be used to test the SAN-based image using the AoE protocol stressing the computer capabilities and checking under which conditions is the user able to manage the system with minimal delay or OS errors whatsoever. The following example is an overall picture of how one can setup the proposed method for a working environment:

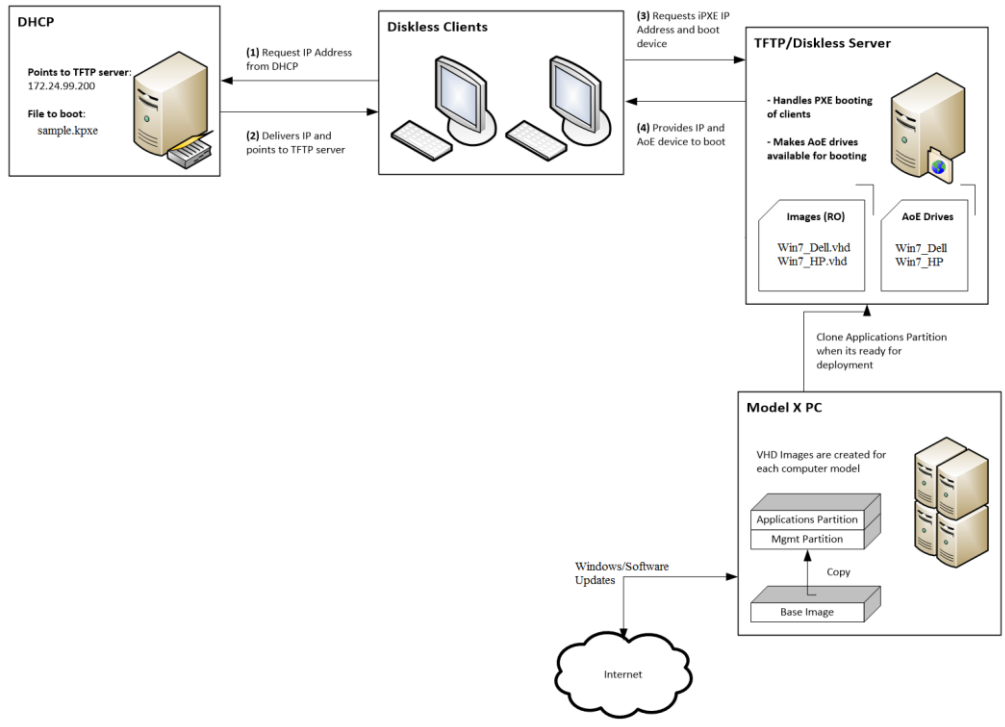
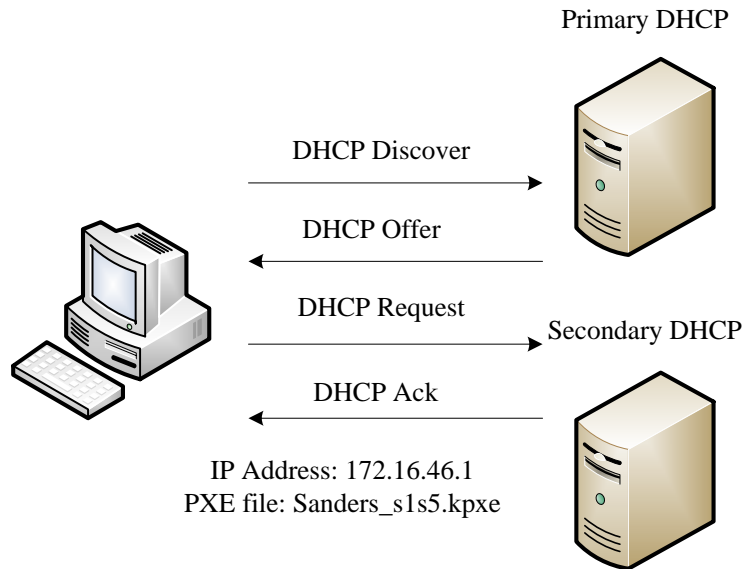


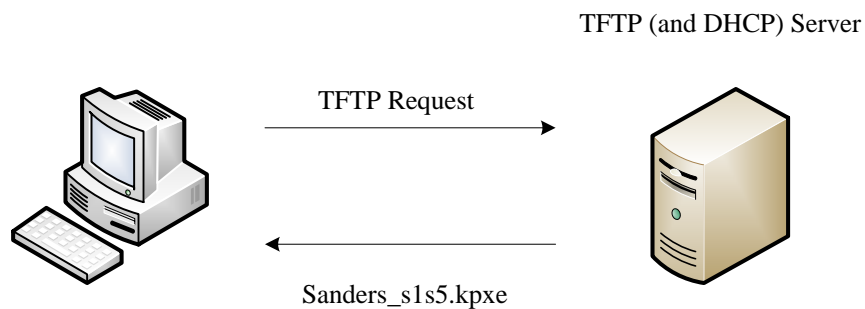
Figure 7: Diskless Process Overview

A step by step description of the proposed example is given below:



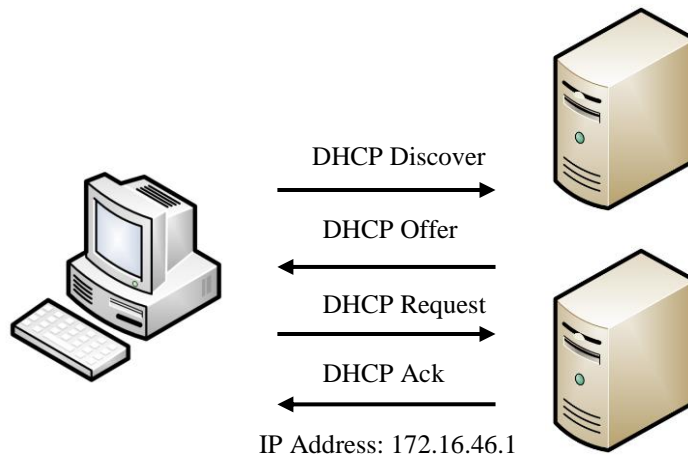
**At boot time, the client requests the name of the PXE file through a DHCP request. The information is obtained from whichever DHCP server responds first to the DHCP Discover**

Figure 8: DHCP Request



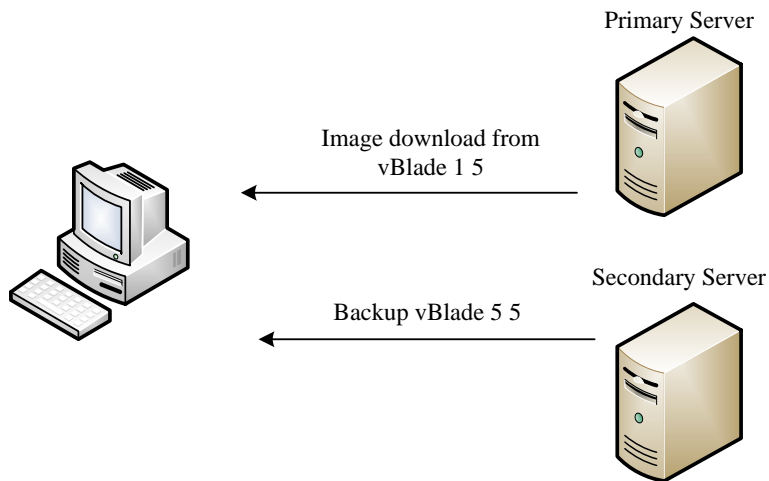
**The client requests and downloads the PXE file from the same server it got an IP address from. The PXE file is a short script telling the client from which vBlade to boot from**

Figure 9: TFTP Request



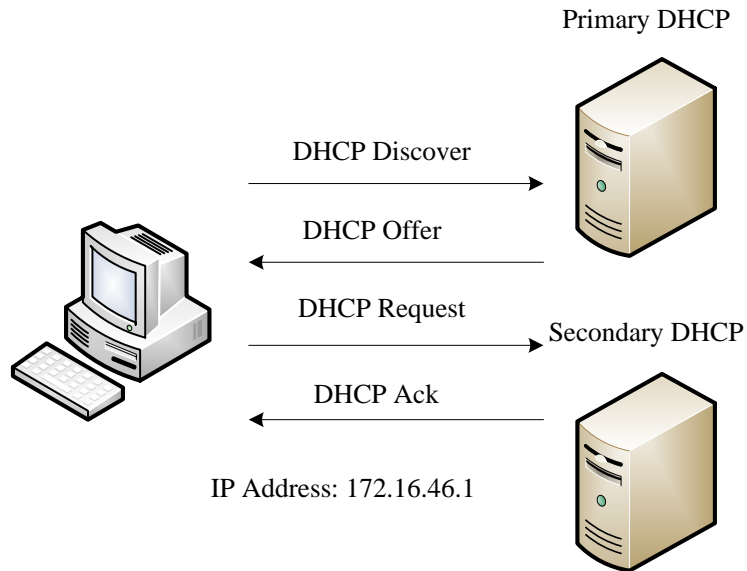
**The client loads the PXE image. The client requests an IP address. DHCP provides an IP address to the client.**

Figure 10: DHCP Request iPXE Stage



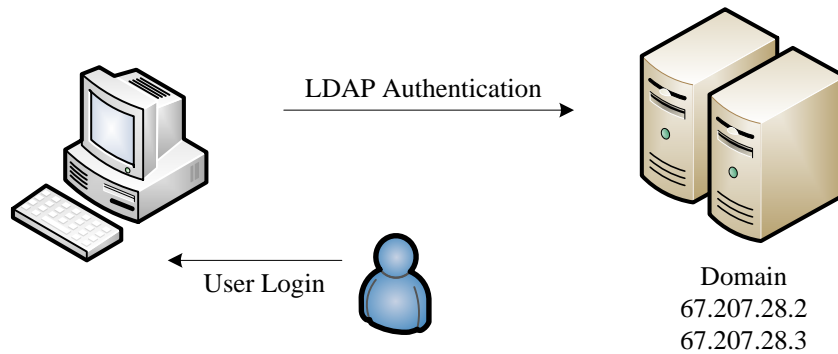
**Around 2.2 GB are downloaded and loaded into RAM on client for Windows to boot. If the vblade is not available, the other server contains a backup image on a vblade 4 shelves higher or lower (e.g. PRIM 1 5 and SECN 5 5; PRIM 2 4 and SECN 6 4).**

Figure 11: AoE Target Booting



Once the client has finished booting, Windows will request an IP address from DHCP

Figure 12: DHCP Request OS Stage



The client waits for an IP and for a secure LDAP connection with domain. User enters credentials and authenticates with the domain using LDAP.

Figure 13: User Login

## Current Results

The proposed setup is expected to yield a reduced boot time from a diskless client as well as an increased performance while opening applications on a diskless client compared to a local HDD. Results have been recorded using the following resources:

Four Diskless/TFTP servers were used to handle the load of the diskless clients.

Diskless Server	CPU	CPU Cores	RAM (GB)	HDD Size (GB)	Network Interfaces
1	8 x Intel Xeon E5-4640 2.4 GHz	8	630	300 + 1800	MYRICOM Myri-10G Broadcom NetXtreme Gigabit Ethernet Broadcom NetXtreme 1/10 Gigabit Ethernet
2	8 x Intel Xeon E5-4640 2.4 GHz	8	630	300 + 1800	MYRICOM Myri-10G Broadcom NetXtreme Gigabit Ethernet Broadcom NetXtreme 1/10 Gigabit Ethernet
3	8 x Intel Xeon E5-4640 2.4 GHz	8	630	300 + 1800	Intel I350 Gigabit Network Connection Broadcom NetXtreme II 10 Gigabit Ethernet
4	8 x Intel Xeon E5-4640 2.4 GHz	8	630	300 + 1800	Intel I350 Gigabit Network Connection Broadcom NetXtreme II 10 Gigabit Ethernet

Table 1: Diskless/TFTP Servers

VHD files with sizes ranging from 30 GB to 180 GB were used in the setup. Different sizes were used with the objective of testing the impact on performance and differences in boot times.

VHD Files	Disk Sizes (GB)	Disk Usage (GB)	OS	Compatibility
HP	30, 60, 90, 120, 150, 180	30, 60, 90, 120, 150, 175	Windows 7 (64 bit)	Hewlett Packard PCs
Dell	30, 60, 90, 120, 150, 180	30, 60, 90, 120, 150, 172	Windows 7 (64 bit)	Dell PCs

Table 2: VHD Files

The VHD files in this setup are compatible with different PC models. PC models used in the setup are shown in the table below.

#	Brand	Models	CPU	RAM (GB)	Network Interface
1	Hewlett Packard	HP Z220 HP Z230	Intel Core i7 3.2 GHz	24	Broadcom NetXtreme Gigabit Ethernet
2	Dell	Optiplex 745 Optiplex 755 Optiplex 760	Intel Core 2 Duo	4, 8	Intel 10/100/1000 Ethernet

Table 3: Diskless Clients

Two DHCP Servers were used in this setup. Diskless clients send a request to the DHCP server during two stages: the PXE boot stage and the OS boot stage.

DHCP Server	CPU	CPU Cores	RAM (GB)	HDD (GB)
1	2 x Intel Xeon X5670 2.93 GHz	2	4	42
2	2 x Intel Xeon X5670 2.67 GHz	2	4	42

Table 4: DHCP Servers

The proposed setup has been through rigorous testing and was implemented in a large diskless virtualized environment across campus consisting of 494 computers. The following data has been gathered from the implemented system:



Test #	VHD File Size (GB)	VHD Used Space (GB)	RAM Usage at boot (GB)	Bytes Received until Boot (GB)	Time to			
					DHCP Request Stage (s)	Start of OS critical files download (s)	OS Loading Logo	OS Login Screen (s)
1	30	30	1.74	0.89	13	39	68	96
2	30	30	1.75	0.848	13	40	64	124
3	60	60	1.91	0.96	14	40	67	106
4	60	60	1.92	0.97	13	39	71	105
5	90	90	1.5	1.9	13	40	75	161
6	90	90	1.5	1.9	13	40	67	163
7	90	90	1.5	1.9	13	38	72	110
8	120	120	1.68	1.16	13	40	69	141
9	120	120	1.78	1.3	13	40	68	105
10	150	116	1.83	1.6	21	56	160	168
11	150	116	1.83	1.6	21	58	127	137
12	150	116	1.83	1.2	21	55	129	143
13	180	130	1.6	1.5	20	59	122	123
14	180	137	2.26	2	13	39	80	135
15	180	137.2	1.78	1.8	21	57	322	342
16	180	150	1.76	1.14	13	42	89	167
17	180	150	1.75	1.3	13	39	75	181

Table 5: Single Diskless Client Boot Statistics

Please note that between test 9 and 10, the *VHD File Size* was increased to 150 GB and unused software was uninstalled which decreased the *VHD Used Space* from 120 GB to 116 GB. Note that this caused the *Start of OS critical files download* to take longer and delayed the *OS loading logo* screen from tests 10 to 14. Also, it is important to mention that between tests 13 and 14, additional software was installed that increased the *VHD Used Space* from 130 GB to 137 GB and then to 137.2 in test 15. However, this also increased the diskless client total boot time. To remediate this, settings such as data caching, and the creation of temporary folders were disabled for the new software installed. The startup of non-essential services was delayed to post-boot time. Test 16 reflects these changes. Additionally, for tests 16 and 17, the same changes were applied to the settings of all other software installed on the VHD file.

Results from Table 5 show that RAM usage at boot varies slightly regardless of VHD size and used space. This is true as long as the installation of new software does not add

additional services that are required to be started during boot time. It is important to note that the number of services, available at the startup, was tweaked at different steps to keep the RAM usage under 2GB. Refer to Figure 14 below.

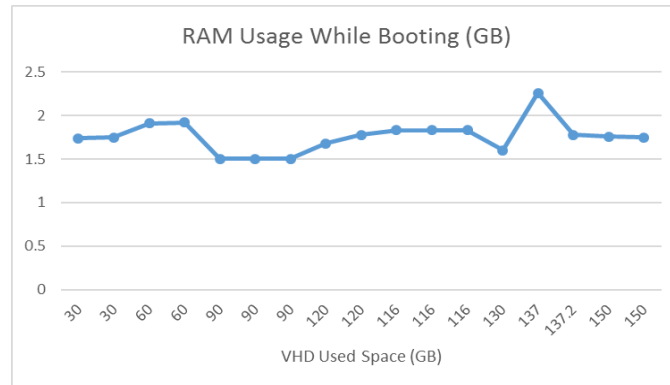


Figure 14: RAM Usage While Booting (GB)

Based on the results gathered in Table 5, Figure 15 shows the relationship between the number of GB received until boot and the VHD Used Space while the number of services, available at the startup, was tweaked at different steps to keep the RAM usage under 2GB.

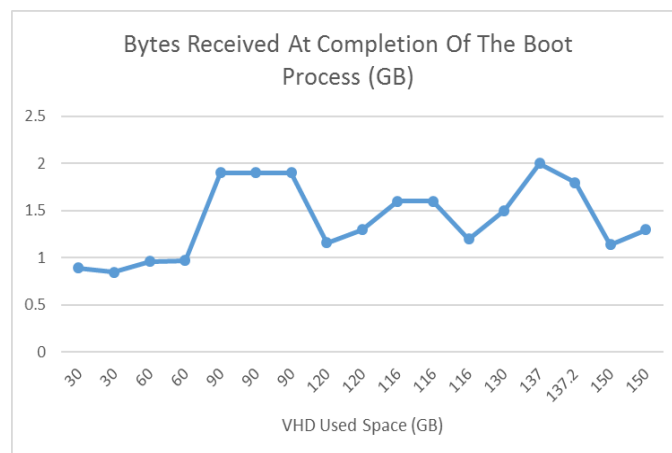


Figure 15: Bytes Received At Completion Of The Boot Process (GB)

With data collected in Table 5, the figure below shows how the number of seconds to get to the *DHCP Request Stage* and the Start of OS Critical Files Download remain unaffected with respected to the VHD File Size and Used Space. Variances in the number of seconds to get to

these stages are only affected by network performance. Note how the changes between test 9 and 10 affected the *OS Loading Logo* stage delaying its start. After applying the fixes between tests 15 and 16, we can see the large decrease in the total boot time as demonstrated in Figure 16.

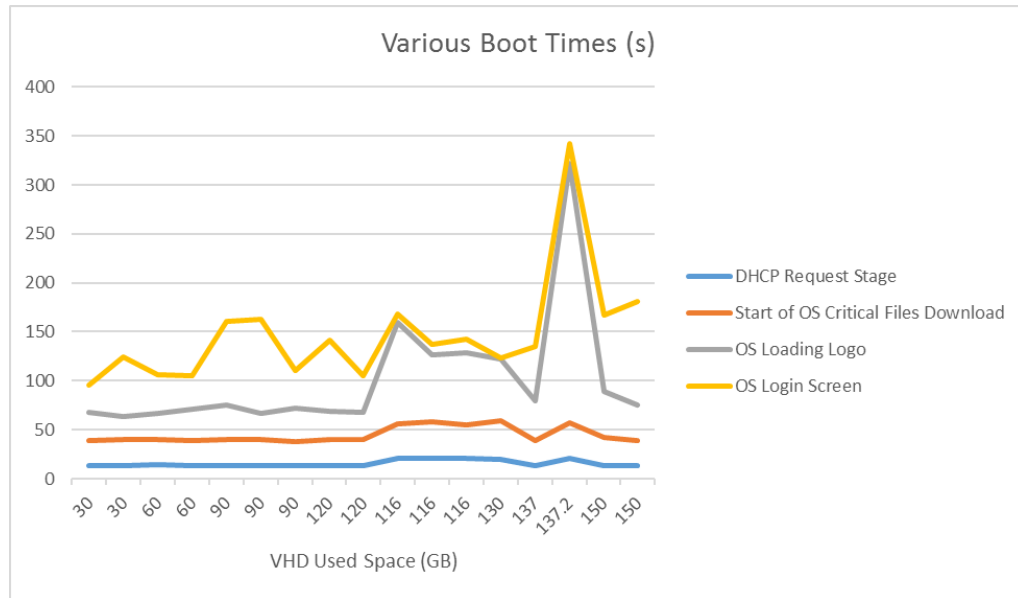


Figure 16: Various Boot Times

Note that Figures 14 to 16 compare different statistics with respect to VHD Used Space value instead of the VHD File Size. This is done since the VHD Used Space is determined by the quantity and configuration of the software on the diskless client which has an impact on performance. The VHD File Size is just the total size, or in other terms, the maximum container size. It is important to note that the number of services, available at the startup, was tweaked at different steps to keep the RAM usage under 2GB, and to avoid excessive boot times.

Figure 17 shows the CPU core percent used by one vblade instance on a server when serving one diskless client right after another during boot time. The instance peaks at around sixty six percent usage. Note that the hardware used for the Diskless/TFTP Servers have eight CPUs each with eight cores for a total of sixty-four cores for each of the four available servers.

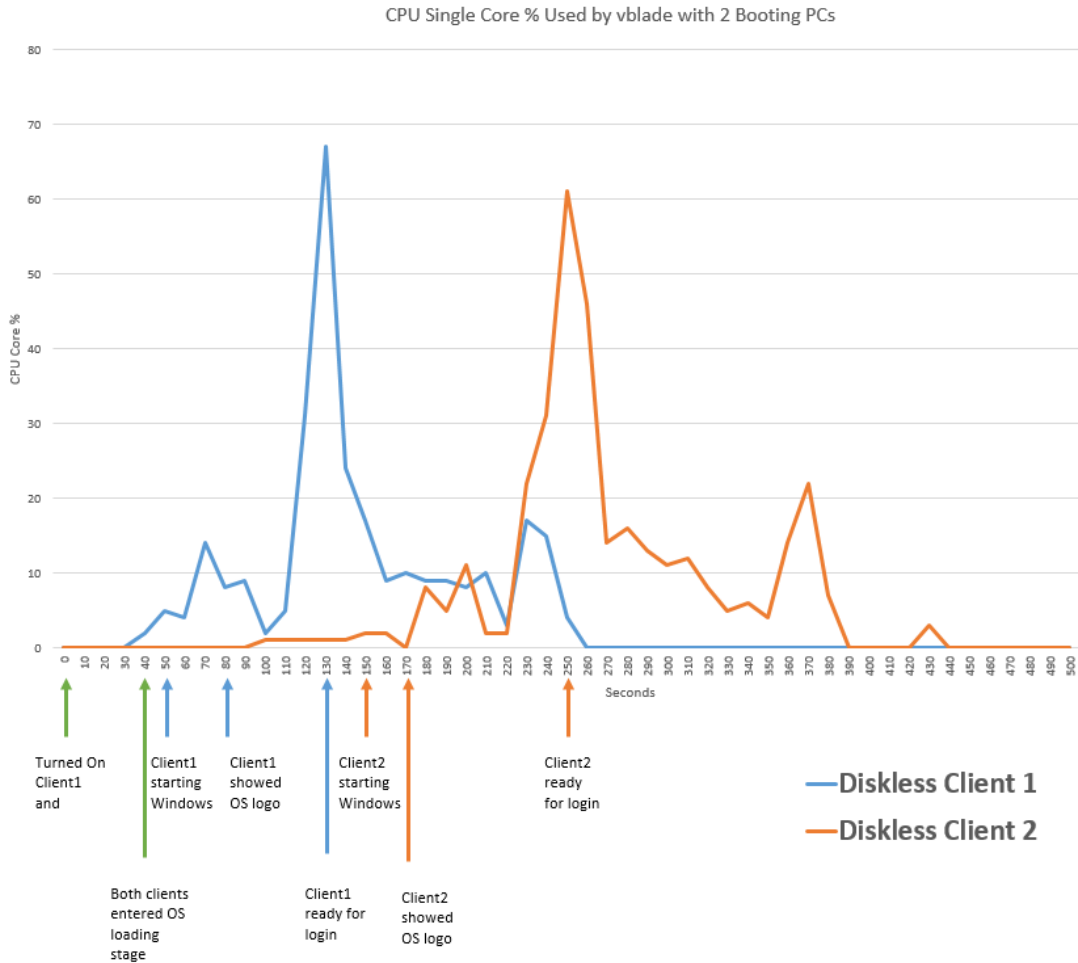


Figure 17: Diskless Server CPU Single Core Usage

Figure 18 shows the impact of booting diskless clients simultaneously on a single vblade:

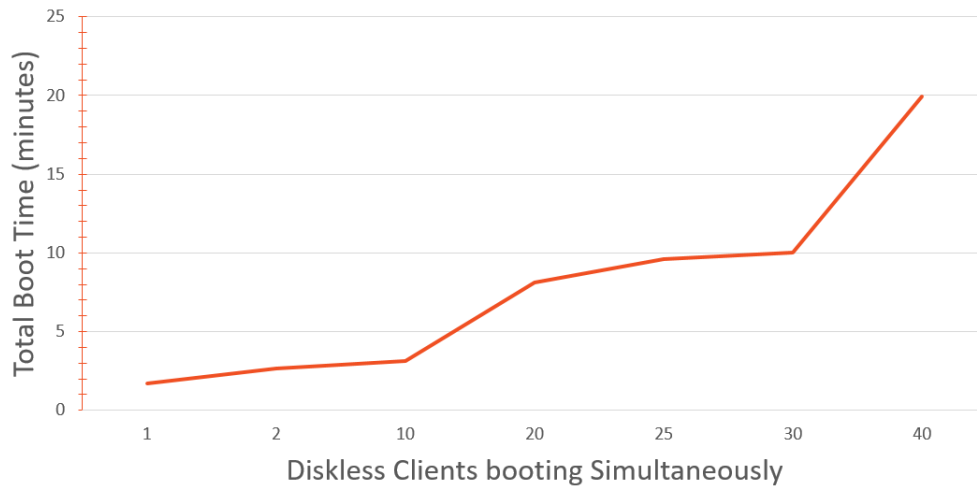


Figure 18: Load Test on a Single Vblade

## CHAPTER VI

### CONCLUSION

There are several options on how to provide a large number of computers to users for their daily tasks. It is our goal to find an alternative solution to the conventional methods keeping the following objectives in mind for improving the end-user computing experience: increase MTBF, increase overall computing speed, avoid per seat licensing costs, use a faster streaming protocol than iSCSI, reduce power consumption, provide a clean image every time a user logs in, reduce maintenance, and avoid protocols requiring special switches and adapters such as for Fibre Channel.

This thesis proposes the use of the open-source AoE protocol to stream an OS to a user computer from a central server. The expected outcome of this effort is to create a central system consisting of servers and storage components to serve a large number of end-user diskless clients efficiently and adhering to the goals mentioned above.

The proposed setup was implemented across campus involving student labs consisting of about 450 computers. With the findings from our current results using the resources described in Tables 1 through 4, we have been able to calculate the following estimated values recommended for implementing a diskless environment:

<b>Description</b>	<b>Maximum Value</b>
PCs per vBlade (per Server)	13
PCs per vBlade (2 Servers)	25
PCs per VLAN (per Server)	45
PCs per VLAN (2 Servers)	90
PCs per Interface (per Server)	90
PCs per Interface (2 Servers)	180
vBlades per VLAN (per Server)	4
VLANs per Interface (per Server)	2

Table 6: Maximum Recommended Values for a Diskless Environment

## REFERENCES

- A. Pitsillides, P. Ioannou, M. Lestas, L. Rossides, *Adaptive nonlinear congestion controller for a differentiated-services framework*, IEEE/ACM Transactions on Networking (TON), 2005
- M. Maia, M. Rocha, I. Cunha, J. Almeida, S. Campos, *Network bandwidth requirements for optimized streaming media transmission to interactive users*, WebMedia '06 Proceedings of the 12th Brazilian Symposium on Multimedia and the web, 2006
- B. Coile, S. Hopkins, *The ATA Over Ethernet Protocol*, The Brantley Coile Company, Inc., 2009.
- J. Aatrokoski, *ATA over Ethernet and Network Block Device performance tests*, Aalto University MRO PC-EVN Development and Tests, 2007.
- C. Purvis, M. Marquis-Boire, *Access over Ethernet: Insecurities in AoE*, security-assessment.com, 2006.
- M. Landowski, P. Curran, *AoE storage protocol over MPLS network*, MSST '11 Proceedings of the 2011 IEEE 27th Symposium on Mass Storage Systems and Technologies, 2011.
- Intel Corporation, *Preboot Execution Environment (PXE) Specification*, 1999
- D. Clerc, L. Garces-Erice, S. Rooney, *OS Streaming Deployment*, IBM Research, Zurich Laboratory, International Performance Computing and Communications Conference, 2010.
- D. Murray, T. Koziniec, K. Lee, M. Dixon, *Large MTUs and internet performance*, 13th IEEE International Conference on High Performance Switching and Routing (HPSR), 2012.
- C. He, W. Rao, *Modeling and Performance Evaluation of the AoE Protocol*, 2009 International Conference on Multimedia Information Networking and Security, 2009
- A. Hassidim, D. Raz, M. Segalov, A. Shaqed, *Network Utilization: the Flow View*, 2013 Proceedings IEEE INFOCOM, 2013
- H. Jasem, Z. Zukarnain, M. Othman, S. Subramaniam, *Evaluation Study for Delay and Link Utilization with the New-Additive Increase Multiplicative Decrease Congestion Avoidance and Control Algorithm*, Scientific Research and Essays Vol. 5, 2010

## BIOGRAPHICAL SKETCH

Max Daniel Torres, resides happily in Brownsville, Texas. His mailing address is 5975 Diamondback, Brownsville, TX 78526. Mr. Torres became a member of the Association for Computing Machinery (ACM) while earning his Bachelor of Science in Computer Science degree from The University of Texas at Brownsville. During this time, he gained valuable knowledge in software development and network devices while being employed as a part-time Computer Programmer for the Infrastructure, Telecommunication and Networks department of the university. He earned his B.S in Computer Science in 2009 from The University of Texas at Brownsville where he completed a campus wide production level project for his senior project *Multiple Media Messaging System Over VoIP*. This project involved a messaging system that uses different Java technology to deliver messages in a Voice over Internet Protocol (VoIP) environment using multiple media, such as, audio and text. Upon graduation, he was offered the position of Network Analyst where he gained further knowledge on the design and security of computer networks, web interfaces, and VoIP configuration. He started working towards his Masters of Science in Computer Science while working full time as a Systems Analyst for the IT Technical Services department and later the South Texas Diabetes and Obesity Institute, Medical School of The University of Texas Rio Grande Valley. He earned his Master of Science in Computer Science in 2016 from The University of Texas Rio Grande Valley.