1. INTRODUCTION

In this Information Age, many new technologies are changing the face of office communication. High-speed data networks, desktop publishing, video-teleconferencing, and multimedia are the architects of this new age. Desktop personal computers (PCs) and workstations are becoming the norm. Communicating with electronic mail (e-mail) via local area networks (LAN) and abroad are commonplace. Within the Networks Division, we see evidence of the information age in the form of the Management Information Systems (MIS). Some of these information systems are the Automated Ground Network System (AGNS), Engineering Change Automation System (ECAS), Interactive Multi-Media/Computer Based Training (IMM/CBT), Facilities Automation System (FAS), and Technical Information Program (TIP). These projects provide services which are vital to how we function from day-to-day.

1.1 BACKGROUND

Currently, the elements of the Networks Division MIS are not compatible and do not provide a seamless integration to a common multimedia workstation. The ND-MIS exist on different platforms: UNIX workstations, PCs and Macs. Users of contrasting platforms can not access the services of the ND-MIS seamlessly from a platform of their choice. Public domain (PD) and commercial-off-the-shelf (COTS) software may allow for their integration.

The following report explains the experimental and investigative attempts to provide seamless integration of the ND-MIS to a computer platform of the users choice (UNIX workstation, PC, or Mac). There are hundreds of software products that help provide integration between the differing platforms. This report describes the results of the products provided by Farrallon, IBM, Insignia, Intercon, Microsoft, Micro-X, NCD, SunSelect, and White Pine.

The following table compares the diversity of the Networks Division Management Information Systems. Each system supports its own hardware platform and operating system. The table also lists the type of information presented to the user, e.g. text, graphics, audio, and video. All the ND-MIS share one common attribute, they all reside on an ethernet TCP/IP network.

Table 1. Networks Division MIS Comparison Chart

	Hardware	Operating system	Data Type	Network
AGNS	SUN	UNIX	Text, Graphics	ethernet, TCP/IP
ECAS	PC	DOS (future - Windows)	Text, Graphics	ethernet, TCP/IP
IMM/CBT	Macintosh	System 7	Text, Graphics, audio, video	ethernet, TCP/IP
FAS	PC	SCO UNIX	Text, Graphics, audio, video	ethernet, TCP/IP

1.2 HARDWARE AND SOFTWARE REQUIREMENTS

The sample of the ND-MIS shown in Table 1 was chosen because it represents the diversity of the different systems within the Network Division. In the investigation and integration, the actual ND-MIS were not used. Three computer platforms were set up: a Sun SPARC 2 workstation, running UNIX SunOS 4.4.1; a Gateway2000 486 DX2-66Mhz PC, 16 MB RAM, running DOS 6.2 and MS-Windows for Workgroups (WFW) 3.11; and a Macintosh Quadra 650, 24 MB RAM, running System 7.1. The three platforms were configured with TCP/IP networking and integration software from the aforementioned vendors. Since TCP/IP is our current networking environment and the forerunner of the Information SuperHighway, it makes sense to model the ND-MIS in this manner.

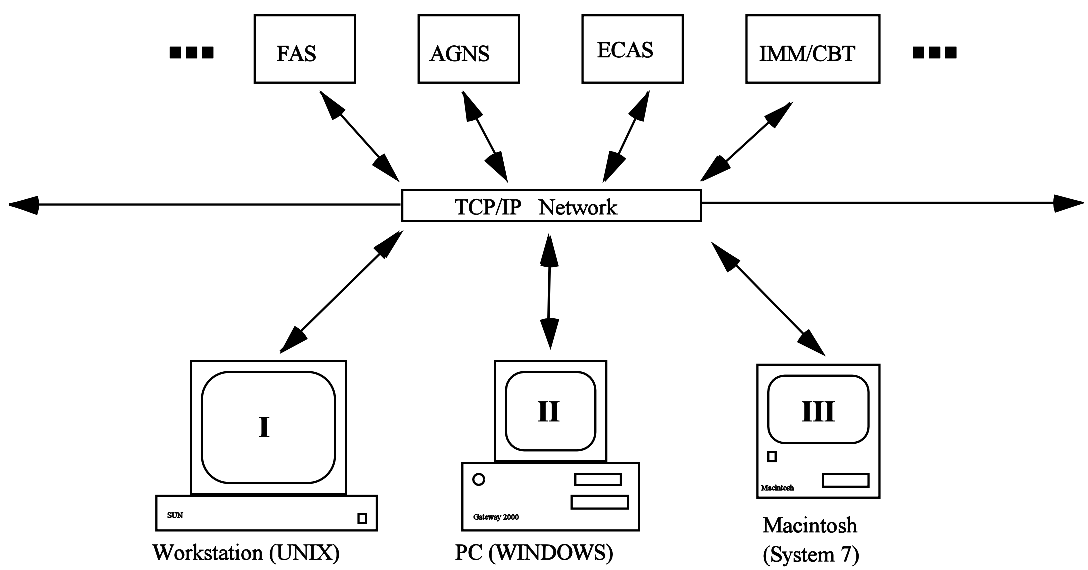

Figure 1. Current computing environment

1.3 PROBLEM STATEMENT

The primary objective of this project was to provide a cost saving, convenient, and expandable solution to cross platform integration. As stated in the opening remarks, users do not have seamless access to Networks Division MIS from a computer platform of there choice. For example, ECAS resides on a the PC-DOS platform. Therefore, if a user wishes to access the services of ECAS, they must use a PC; not a Macintosh or a UNIX workstation. Considering that most users take advantage of using one or more of the Networks Division MIS daily, it is likely that the user needs to have more than one computer platform at their desktop. This type of day-to-day operation can become quite costly. Depending on how often a user needs to access other services, it may not be convenient to borrow another co-workers computer or relocate to a communal terminal to complete their current task. This brings us to main goal of this project: provide a means for users to access any Networks Division MIS from any computer platform of there choice.

The secondary objective was to demonstrate the effectiveness of "open computing" in a distributed computing environment. The concept of "open computing" refers to non-proprietary control of a users computing environment or not being tied to one vendor for all your computing resources. A distributed computing environment is essentially how we operate within the Networks Division today. For example, ECAS servers located at ATSC (Columbia & Seabrook) and GSFC provide engineering documents and drawings to users within the GSFC community and abroad (WSC, MILA/PDL & BDA). See Figure 2. Note that, we are still limited by the cross platform bottleneck between PCs, Macs, UNIX workstations.

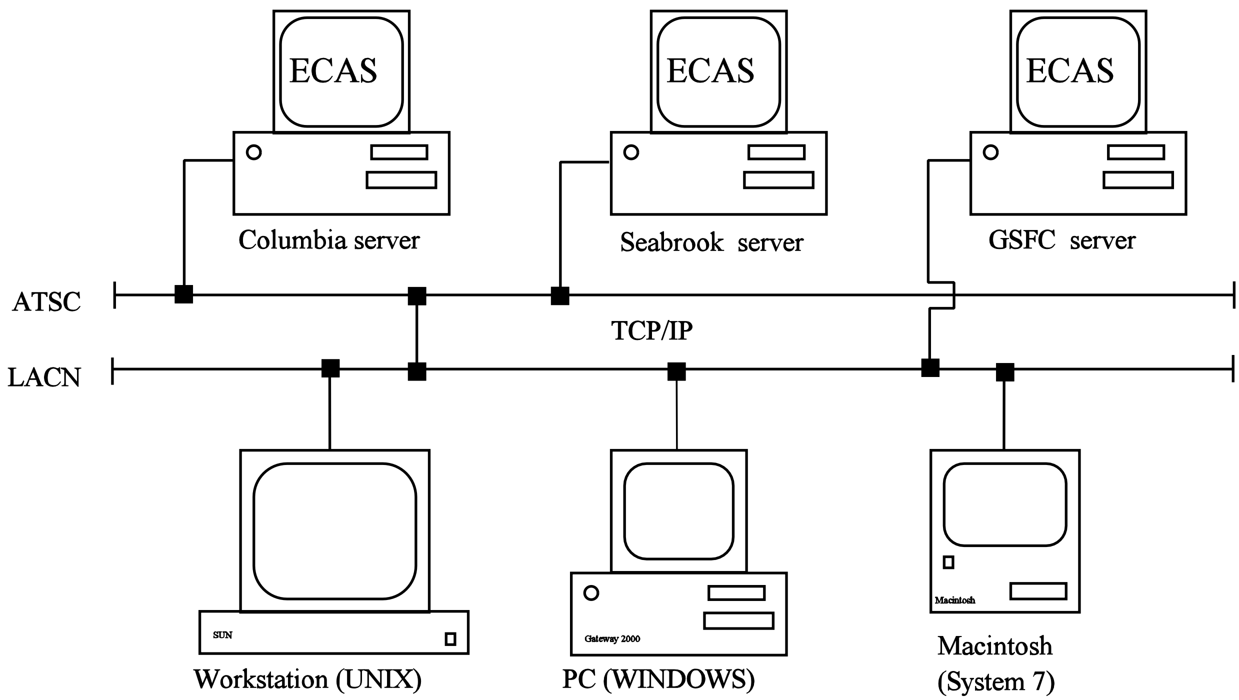

Figure 2. Proposed ECAS computing environment.

Despite the limitations due to differing platforms, the networking technology of the ethernet and TCP/IP communication standards affords us the ability to operate flexibly. With the appearance of such new network technologies as ATM and SONET, we will see the ease and convenience of distributed computing increase.

2. SOFTWARE DESCRIPTION

2.1 PC-TO-UNIX WORKSTATION INTEGRATION

In this section, I will discuss and define the PC-to-UNIX workstation integration in terms of the software used to implement the process. An interpretation of the findings is discussed in Section 4. There are several software packages that were used to implement this integration: SunSelect's PC-NFS, Insignia's SoftWindows, NCD's PC-Xware, and Microsoft's Windows NT. It should be noted that some software packages (e.g. NCD's PC-Xware) were used throughout the testing of the PC-to-UNIX workstation integration. Below, I define what each software package provides.

SunSelect PC-NFS

PC-NFS software enables PCs running DOS, with or without Microsoft Windows, to use resources on the network, such as files and printers. These resources may reside on workstations, minicomputers, and mainframes that have different hardware architectures and operating systems, including servers running SunOS or other versions of UNIX. These resources could be any of the Networks Division MIS projects.

PC-NFS uses the flexibility of the TCP/IP suite of protocols to provide the resources to the PC user. The Network File System (NFS) is a part of the suite of protocols that allow network communication between the PC and the UNIX workstation. The details of how NFS works and its relation to the OSI (Open Systems Interconnect) model will be discussed later.

Insignia SoftWindows

SoftWindows is an X Windows application which runs on UNIX workstations or servers. It emulates a high specification IBM PC/AT or compatible. SoftWindows enables UNIX workstations to run most Windows and MS-DOS applications alongside other UNIX applications. The strengths and weaknesses of using such a product for PC emulation will be discussed later.

NCD PC-Xware

PC-Xware is an X Windows server (X11R5) for PCs. It enables PCs to operate within the same environment as UNIX workstation via a network connection. The network connection can be either serial using SLIP (Serial Line Interface Protocol), PPP (Point-to-Point Protocol) or a direct ethernet connection to the local area network (LAN) using TCP/IP. PC-Xware provides the PC user with access to UNIX and X Windows applications residing on UNIX servers.

Microsoft Windows NT Advanced Server

The Windows NT Advanced Server operating system is a platform-independent, scalable server operating system. It is designed to be both a personal computer operating system (e.g. MS-DOS) and a network operating system. Microsoft Windows NT Advanced Server is a complete redesign from Windows 3.1. It is scalable to symmetric multiprocessing systems, where extra processors can be added for increased performance. However, due to hardware compatibility issues, the Windows NT Advanced Server integration could not be explored deeply.

2.2 MACINTOSH-TO-UNIX WORKSTATION INTEGRATION

In this section, I will discuss and define the Mac-to-UNIX workstation integration in terms of the software used to implement the process. An interpretation of the findings is discussed in Section 4. There are several software packages that were used to implement this integration: White Pine's Exodus and Intercon's NFS-Share and Planet X. Below, I define what each software package provides.

White Pine Exodus

Exodus is an X Windows server for Macintosh computers. It enables Macs to operate within the same environment as UNIX workstation via a network connection (LAN). Using Exodus, the Macintosh user can access UNIX programs and X Windows applications residing on UNIX servers. Using Exodus does not remove the user from the normal Macintosh environment, it simply adds X server capabilities.

Intercon NFS-Share

NFS-Share software allows Macintosh users to access resources on the network, such as files and printers. NFS-Share uses the Network File System to provide transparent access to file storage servers such as IBM mainframes, minicomputers, and workstations. Since NFS is part of the TCP/IP family, compatibility across the LAN is ensured.

Intercon Planet X

Planet X is an X Windows client for the Macintosh. It allows a user to operate a Macintosh remotely from any X Windows system such as a Sun, HP, or IBM UNIX workstation. The remote Macintosh's screen appears within an X Window on the workstation. Full function of the Macintosh is available via the UNIX workstation.

Macintosh Application Environment (MAE)

MAE is an X Windows application for UNIX workstations. It provides a virtual Macintosh interface for UNIX workstations. Essentially, it is a Macintosh emulator that allows UNIX workstation users to run unmodified 680x-based applications. In order to install this software, it requires a Sun SPARCstation running Solaris 2.3. Such a configuration was not available at the time of prototyping and could not be carried out.

2.3 PC-TO-MACINTOSH INTEGRATION

In this section, I will discuss and define the PC-to-Macintosh integration in terms of the software used to implement the process. An interpretation of the findings is discussed in Section 4. Only Farrallon's Timbukutu Pro provided limited PC-to-Macintosh integration using the AppleTalk protocol. Support for TCP/IP was not provided. Due to the limitations of the commercial software available, this stage of the integration was discontinued.

3. PROTOTYPE DEVELOPMENT

The prototype for the cross platform integration consists of a 486 PC, Macintosh Quadra, and a Sun Sparc2 workstation. These three architecturally unique computer platforms currently coexist on the LACN at GSFC. However, they interact in a very limited fashion.

The three computer platforms were configured with basic network operating system (NOS) software. The NOS was provided by the services within the TCP/IP protocol family. The Macintosh was configured with MacTCP version 2.02. The PC was configured with several network software packages: Microsoft-TCP and PC-Xware. The Sparc2 is a standalone UNIX workstation which has the NOS embedded in the operating system. All network services provided by TCP/IP are an inherent part of the Sun's daily operation.

With the basic network software installation complete, the PC and the Macintosh are able to communicate with *e-mail* (i.e. Eudora, Pine, Zmail, etc.), initiate *telnet* and *ftp* sessions, and *http* services (i.e. Lynx, Mosaic, Netscape, etc.). Users have the services mentioned above available to them today. However, the interaction between the three platforms remains minimal. The focus of our daily network activities is directed toward a UNIX server (like the Sparc2 mention previously) for network services.

For each platform prototype, the natural computing environment was preserved. For example, the cross platform software added to the PC still resembled the MS-Windows graphical user interface (GUI). The cross platform software added to the Macintosh preserved the "look and feel" of the System 7 GUI. Likewise, the cross platform software added to the Sparc2 preserved the GUI of X Windows and Motif applications.

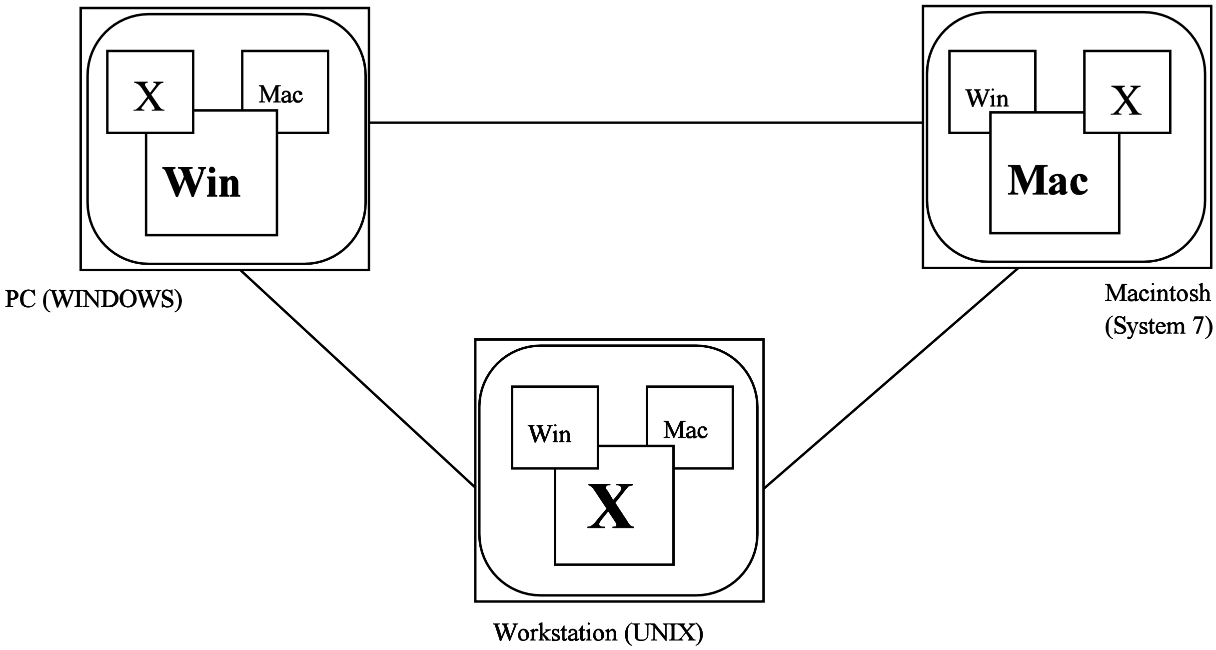

Figure 3. Description of the GUIs for the three platforms (PC, Mac, & UNIX workstation).

Once all three platforms were configured and operating at the same network level, the NFS and X Windows software packages were installed. The NFS protocol will allow the three platforms (DOS, Mac & UNIX) to share and transfer files transparently. The X Windows protocol will allow the three platforms to communicate between applications across the network in a client/server architecture. PC users will access *pc-native* and UNIX-based programs stored on the UNIX server. The Macintosh users will access *Mac-native* and UNIX-based programs stored on the UNIX server. The true test will be accessing *pc-native* programs stored on the Macintosh and accessing *Mac-native* programs stored on the PC. These findings will be discussed in the next section.

The PC-NFS software provides vital resources to accomplish the goal of cross platform integration for the PC. The NFS-Share software for the Macintosh provided the best resources for the goal of cross platform integration. Both PC-NFS and NFS-Share are NFS client software. The server side software was installed on the UNIX server. The server software, called a daemon, runs continuously on the UNIX server listening for requests from the client software installed on the PC and Macintosh.

The NFS protocol is the key to cross platform integration. Without it, the transparent file access for the user would not be possible. NFS is embedded into the UNIX kernel. Programs do not know whether they are being accessed locally or remotely. This is achieved by executing the corresponding system calls at a level external to the program in the UNIX operating system. Figure 4 describes the structure.

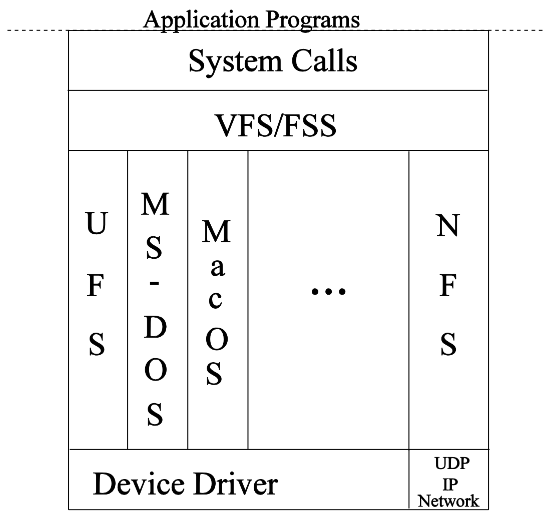

Figure 4. NFS and other file systems embedded in the UNIX kernel.

The developers of NFS integrated the client software in an architecturally clean manner by providing an internal system interface called the Virtual File System (VFS). VFS makes it possible to operate several different file systems at the same time. The access to the file system is managed at the layer before file system dependent layer (UFS, MS-DOS, MacOS) is called. By embedding NFS in the operating system, faster throughput is achieved when accessing file systems. This is what makes the NFS protocol so useful in solving the cross platform dilemma. The UNIX kernel is designed such that it can incorporate the UNIX File System (UFS), MS-DOS File System, MacOS File System, and others if needed.

In relation to the OSI reference model and the TCP/IP protocol family, NFS operates at the Transport Layer (See Figure 5). NFS uses UDP (User Datagram Protocol) to communicate to remote clients.

NFS requests are actually made through remote procedure calls or RPC. The RPC requests are then sent to UDP port 2049 for transport across the network. All of the these protocol requests and data transports are transparent to the user. All the user has to know is that he/she issued a command for a directory listing of **c:\windows**, for example. The ease of the user's normal and familiar computing is preserved. Clearly, the use of the UNIX operating system and NFS as the core of the cross platform solution looks promising.

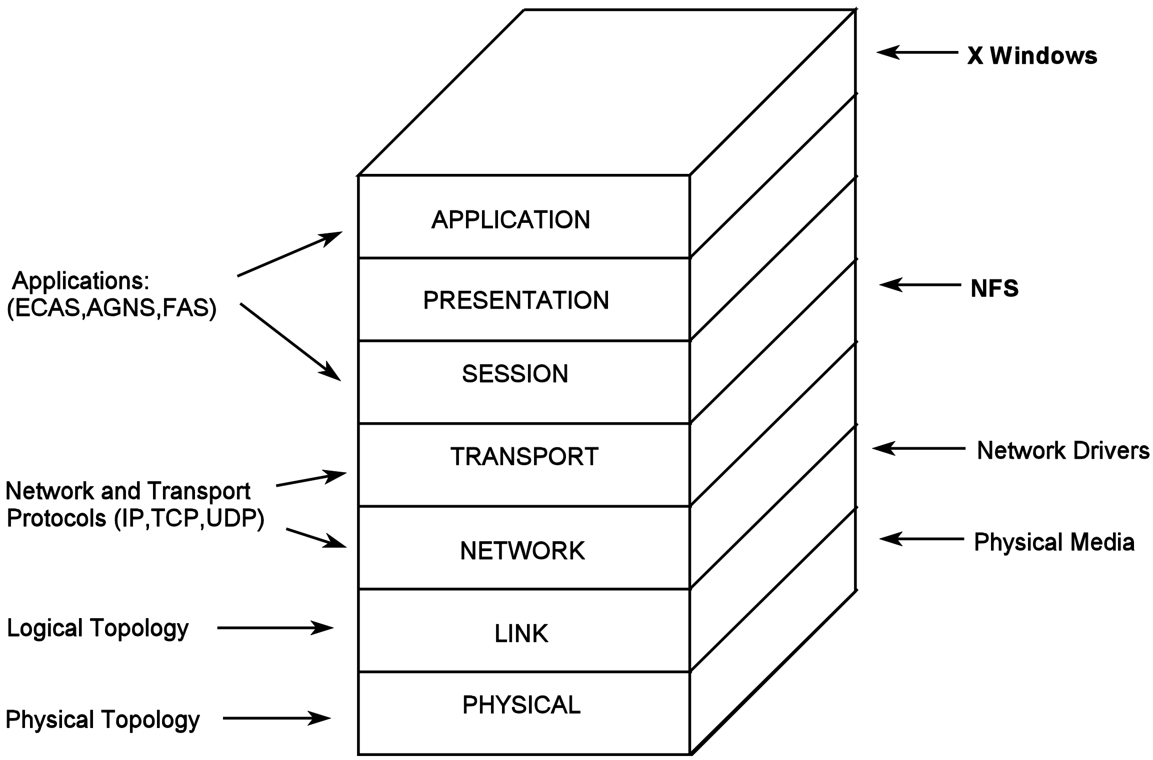

Figure 5. OSI reference model.

The PC-Xware and Exodus software packages provide a very useful approach to accessing UNIX-based applications for the PC and Mac, respectively. The two software programs are X Windows servers for the PC and Mac, respectively. Since the X protocol is independent on both hardware and software, it is not limited by being a PC-only or Mac-only application. However, X is not an application, but a method of communicating between two applications. Any operating system or hardware can run X. The X Windows server allows the PC and Mac users to take full advantage of the graphical environment produced by UNIX-based programs and applications.

4. CONCLUSION

4.1 SUMMARY OF FINDINGS

PC-to-UNIX Workstation Integration

The PC was successfully integrated to the UNIX workstation via NFS and X Windows. PC-native programs were stored on the UNIX server and accessed using NFS transparently from the PC. The PC recognizes the remote file system as just another drive. For example, if the remote file system was the **/usr** partition on the UNIX server, MS-DOS and WFW 3.11 would recognize it as drive "E:". As long as drive "E:" was setup in the users path, programs will run as if they were resident on the PC locally.

The X Windows protocol allowed the PC user to access, run, and display remotely UNIX-based programs and applications. The X Windows server software was installed on the PC. Through the underlying TCP/IP network, the PC user was able to access UNIX-based programs and applications. The UNIX-based programs and applications appear in the MS-Windows style window interface or the Motif style window interface. Only minor delays were experienced due to network traffic when running UNIX-based programs. The amount of RAM (random access memory) on the local machine influences the performance of the X Windows server. The user is limited to the number of windows that he/she can open before local memory is exhausted.

The UNIX workstation to PC integration was marginally successful using an MS-Windows emulator for UNIX. The MS-Windows emulator runs as a Motif application on the workstation. This allows PC programs and applications to be accessed through the emulator's interface. The UNIX workstation user will find the performance of the MS-Windows emulator is marginal at best. However, the user can run PC programs successfully, but at 80286 performance levels. The emulator also only supports up to Super VGA (800x600 resolution). See the recommendations section.

Mac-to-UNIX Workstation Integration

The Macintosh was successfully integrated to the UNIX workstation via NFS and X Windows. Mac-native programs were stored on the UNIX server and accessed transparently. To initiate a NFS connection, the Mac user must select it through the *Chooser*. Under the *Chooser*, a new icon appears called **NFS**. By selecting the NFS icon, the user can then choose which remote file system they wish to access. However, privileges and restrictions on the remote machine do apply. Once the remote machine is chosen, it appears on the desktop as an icon. Now all *Mac-native* programs stored on the remote machine can be accessed by the Mac user. The performance of the programs stored remotely was diminished in comparison to local file access. Expect to wait up to 5 seconds once a remote application is chosen or file system directory is listed (not to mention the time for normal network traffic delays). This can be attributed to the way the NFS client was designed for the MacOS. For experimentation and demonstration purposes, the performance was adequate.

The X Windows protocol affords the Mac user the same features as the PC user: access to UNIX-based programs and applications. The X Windows server software was installed on the Mac. Through the underlying TCP/IP network, the Mac user was able to access UNIX-based programs and applications. The UNIX-based programs appear in the MacOS style window interface or the Motif style window interface. The performance and operation of the UNIX-based applications using X Windows was excellent. The limiting factor was dependent upon the amount RAM available on the local machine and the network traffic load.

The UNIX workstation was successfully integrated to the Macintosh via X Windows client/server software. Through the use of the X Windows server installed on the Mac and the X Windows client software on the UNIX workstation, the UNIX user remotely accessed another Macintosh on the TCP/IP network with the proper privileges. The act of remotely controlling another Mac is quite different from the NFS scenario described previously. Although the UNIX user can remotely access another Mac, the user cannot run *Mac-native* programs seamlessly. See the recommendations section for further comments.

PC-to-Macintosh Integration

The investigation into the final branch of the cross platform was limited by the lack of available software. At the writing of this paper, there were no vendors that provided an NFS server solution for the PC-to-Macintosh integration. Since this project was initiated, planned, and described as using COTS software products, time and resources were not available to write software in-house. Only one vendor (Beame & Whiteside) provided an NFS server solution. However, the solution only provided access to other PCs.

In order to complete this branch of the cross platform integration, a fully-functional NFS server for the PC and Macintosh would have to exist. This NFS server software should incorporate the architecture of the UNIX kernel described in section 3. By doing this, the MS-DOS kernel could incorporate other file systems such as the MacOS file system. Alternatively, the MacOS could incorporate other file systems such as MS-DOS. However, there still exists the differences between data file formats of PCs and Macs. Without the aid of complicated, expensive, and non-expandable third-party software, communicating seamlessly between PCs and Macs remains difficult. See the recommendations section for further comments.

4.2 RECOMMENDATIONS

The primary objective of this project was to provide a cost saving, convenient, and expandable solution to cross platform integration. The secondary objective was to demonstrate the effectiveness of "open computing" in a distributed computing environment. Through the use of COTS products, PD software, and well established communications protocols, both objectives were satisfied fully in some areas and partially in other areas.

From Section 4.1, we see that the PC-to-UNIX integration was successful only in a semi-bi-directional sense. NFS and X Windows provide excellent resources that are simple to use, inexpensive and expandable. The UNIX-to-PC integration capability exists, but the performance of the PC emulators are not at productive speeds for daily use.

When examining the Mac-to-UNIX integration, we find successful integration, but also in a semi-bi-directional sense. Once again, NFS and X Windows providing vital resources to accomplish the objectives set forth. The UNIX-to-Mac integration compares directly to the UNIX-to-PC integration. The use of the X Windows client software to remotely control another Macintosh is convenient, but not essential for the needs of our everyday network activities.

Finally, we arrive at the PC-to-Mac integration. The fact that software vendors have not been able to provide a convenient and expandable cross platform solution is not surprising. The PC and the Macintosh have totally different computer architectures. The PC's microprocessor uses a different instruction set than the Macintosh's microprocessor. This accounts for most of the incompatibility. Incompatibilities that can not be solved with NFS or X Windows.

There is no doubt that open and distributed computing are effective within the Networks Division as well as NASA on a whole. Distributed computing allows us to remotely control tracking stations across the Internet, provide real-time satellite tracking to the Network Control Center, and more. Such leaps in technology is due in part to the Networks Division's and NASA's commitment to adopt accepted standards and technologies such as TCP/IP. Such practices should continue if we are to lead the world in aerospace technology.

In order to bridge the cross platform gap, I recommend that the Networks Division make a commitment to produce platform independent products and services. Products that are platform independent are easier to market to our customers. Networks Division projects such as TIP has already taken the lead with such concepts. With the use of the NCSA's Mosaic WWW browser, ND users can access TIP's ever growing document library from a UNIX workstation, PC, or Mac.

New projects that produce software products or disseminate information should give consideration to operating on multiple platforms. This concept may impose expensive initial developing, but the long term benefits are immeasurable. The freedom of independent computer platforms allows us to choose the most cost effective and technologically sound platform of the time. As technology changes, NASA's needs and goals will change. Cross platform independence will allow the Networks Division and NASA to adjust to a cost effective, convenient, and expandable solution to meet our rapidly changing environment.

CROSS PLATFORM INTEGRATION OF MULTIPLE COMPUTER SERVICES

Professional Intern Program Level II Written Report

Written by: E. Joseph Stevens

February 7, 1996

Networks Division
Code 531.2
NASA Goddard Space Flight Center
Greenbelt, MD

TABLE OF CONTENTS

LIST OF TABLES & FIGURES

LIST OF ABBREVIATIONS & ACRONYMS

AGNS	Automated Ground Network System
ATM	Asynchronous Transfer Mode
ATSC	Allied Signal Technical Services Corporation
BDA	Bermuda
ECAS	Engineering Change Automation System
FSS	File System Switch
FTP	File Transfer Protocol
GUI	Graphical User Interface
HTTP	Hyper Text Transfer Protocol
IMM/CBT	Interactive Multi-Media/Computer Based Training
IP	Internet Protocol
LAN	Local Area Network
MAE	Macintosh Application Environment
MILA/PDL	Merritt Island Launch Area/Ponce De Leon
MIS	Management Information Systems
MS-DOS	Microsoft Disk Operating System
MacOS	Macintosh Operating System
NCSA	National Center for Supercomputing Applications
ND	Networks Division
NFS	Network File System
NOS	Network Operating System
OSI	Open Systems Interconnect
PC	Personal Computer
PD	Public Domain
PPP	Point-to-Point Protocol
RAM	Random Access Memory
RPC	Remote Procedure Call
SCO	Santa Cruz Operation Inc.
SLIP	Serial Line Interface Protocol
SONET	Synchronous Optical NETwork
SPARC	Scalable Processor ARChitecture
SunOS	Sun Operating System
TCP/IP	Transmission Control Protocol/Internet Protocol
TIP	Technical Information Program
UDP	User Datagram Protocol
UFS	UNIX File System
VFS	Virtual File System
VGA	Video Graphics Array
WSC	White Sands Complex
WWW	World Wide Web