The following paper was originally published in the
Proceedings of the Large Installation System Administration of Windows NT Conference
Seattle, Washington, August 5–8, 1998

Providing Reliable NT Desktop Services by Avoiding NT Server

Thomas A Limoncelli, Robert Fulmer, Thomas Reingold, Alex Levine, and Ralph Loura
Lucent Technologies, Bell Labs

For more information about USENIX Association contact:

1. Phone: 510 528-8649
2. FAX: 510 548-5738
3. Email: office@usenix.org
4. WWW URL: http://www.usenix.org/

Providing Reliable NT Desktop Services by Avoiding NT Server

Thomas A. Limoncelli, Robert Fulmer, Thomas Reingold,
Alex Levine, Ralph Loura*
Lucent Technologies, Bell Labs
Murray Hill, NJ, 07974
ntdesk@research.bell-labs.com

Abstract

We have developed a reliable, stable NT Desktop environment for our customers. The services we provide include: Standard desktop applications (word processing, spreadsheet, etc.), access to UNIX compute servers, file storage and backups, e-mail, printing, calendar, netnews, web, and Internet access. We founded our architecture by selecting open, standard protocols rather than specific applications. This decoupled our client application selection process from our server platform selection process. We could then choose the server based on our needs for reliability, scalability, and manageability and let customers independently choose their clients based on their needs of platform (NT or UNIX), features, and preferences. We can now choose between competing server products rather than be locked into the (potentially difficult to manage) server required for a particular client application. This created a "no compromises" environment on the desktop as well as in our server room. Our customers are happy because the "tail" doesn't "wag the dog". Our ability to manage this infrastructure is superior because the dog doesn't wag the tail either. The resulting system gives us a strong base to build new services.

1 Introduction

Our department is responsible for providing desktop and back-end (server) computer and network services for approximately 600 customers in the "research" part of Bell Labs, a division of Lucent Technologies. We have a very stable and manageable system that can scale as needed. In this paper we hope to refute several myths: (1) It is impossible to provide reliable NT Desktop services. (2) It is impossible to integrate NT and Unix into one coherent environment. (3) Adding NT desktops means getting rid of all UNIX back-end servers. We prove these by demonstration.

2 Background

Bell Labs is biased towards open systems and publicly available standards. As the inventors of UNIX, we have a large and stable UNIX environment. When PCs first appeared they were often castigated to separate networks for security reasons and we required users to support themselves. Demand for official support grew at the same time as self administered machines were causing havoc. Finally we decided to give them official support to limit their damage. As NT grew in popularity in the industry, our customers needed to use it and we integrated it into our environment.

Our users (whom we call "customers" [Smallwood]) expect us to provide certain services and leave them alone to be self-sufficient for selecting their own tools, etc. We have a very technical customer base. For example, we provide file service but they select their own development environment. Since they preferred this with their UNIX environment, we adopted the same policy as we began official support of PCs. There are three categories of services we provide to our PC users.

The first category of services is related to deployment. We install new PC hardware, load the operating system, connect it to the network, and install and configure all applications. We also handle all account creation/deletion services and manage the

*Currently an independent consultant

NT Domains. These are outside of the scope of this paper, but are touched on as appropriate.

The second category is the main desktop applications that we provide. These include office automation, e-mail, calendar, web browser, web publishing tools, and access to UNIX compute servers (via X windows and telnet).

The third category is the "back end" centralized services we provide such as file storage (with access from NT or UNIX), backups and restores, printing, netnews (bulletin board system), web servers (intranet and external), and Internet access.

3 The Philosophy

Two philosophical rules guide our decisions as we develop our environment. The first rule is to select open systems and protocols whenever possible. The other is to keep things simple. We have learned these lessons the hard way, and now they serve us well when we follow them.

3.1 The Rule of Open Systems

Customers want an application that has the features and ease of use that they need. System Administrators (SAs) want an application whose server is easy to manage. Traditionally either the customers or SAs have more "power" and make the decision in private, surprising the other with the decision. If the SAs make the decision the customers consider them fascists. If the customers make the decision it will no doubt be a package that is difficult to administer which will make it difficult to give excellent service to the customers.

There is a better way that strikes a balance that lets everyone win. We select protocols based on open standards and permit each "side" to select their own software. This decouples our client application selection process from our server platform selection process.

Customers are free to choose the software that best fits their own needs, biases, and even platform. We have a corporate standard client (software) that receives official support but many of our customers are happy self-supporting their own rogue selections. We can not force people to use software they don't like, so we must "use the carrot, not the stick" and draw customers to our recommended software with incentives of reliability and support.

We SAs can independently choose the server based on our needs for reliability, scalability, and manageability. We can now choose between competing server products rather than being locked into the (potentially difficult to manage) server software and platform required for a particular client application. In many cases we can even choose our hardware and software independently if a vendor supports multiple hardware platforms.

For comparison, the opposite strategy would be to let the customers select the application without the informed consent of the staff that would be running the servers. For example, a local (New Jersey) pharmaceutical company selected a particular proprietary e-mail package for their PC users after a long evaluation. Their selection was based on user interface and features with no concern for ease of server management, reliability, or scalability. The system turned out to be very unreliable when scaled to a large number of users. In particular, data corruption problems were frequent and result in having to send the e-mail database to the vendor through the Internet for de-mangling. The system also stores all messages from all users in a single large file which must be kept writable by anyone, which is a security nightmare. Because the package is not based on open protocols the system support staff can not seek out a competing vendor which would offer a better, more secure, reliable, server. Because of the lack of competition the vendor considers server manageability low priority and ignores the requests for server-related fixes and improvements.

The answer is to strike a balance by decoupling the client and server selections. Open protocols permit us to do this because we can select clients from one vendor and servers from another. These two vendors' products talk to each other because the protocol between them is created in an open forum and is publicly documented. Anyone could make a compatible client or server. Consumers can choose between any number of clients or servers. End-users can select from an array of clients, possibly even switching between different ones for different tasks. SAs gain similar advantages. Vendors of server software are forced to compete with each other on a level playing field [Fair]. If the current server software begins to lag behind the competition, SAs can opt

to switch to an alternative vendor without forcing users to change clients. Vendors are more responsive to their customers when they know that their customers can leave them without significant pain.

Critics would say that the customers are the center of the universe and therefore their needs override any concerns of the IS staff. The IS staff should be able to learn any system that the customer selects. However, the reliability and scalability of the server is as much an issue to the customer as is a good user interface. Customers may not feel scalability is important, but they will understand that a mail system flooded by chain letters should not buckle and be down for a day as it is repaired. They might not be concerned by whether or not the IS staff will find it easy to manage the server. However, they will be frustrated if they have to wait a week for what seems to be a simple request, but is actually a major undertaking due to the way the server was designed. All of these "secondary" issues are important to the customer but usually only after a disaster has educated them the hard way. The SAs have a responsibility to present these concerns to the end-users in hope that they will be adopted as concerns of their own. To do this the SAs must partner with customers, use the customers language, not technical jargon and other techniques described in [Bashein].

3.2 The Rule of Simplicity

Supporting a mixed NT and UNIX environment is very complicated. Others have met this complexity by building larger, even more complicated systems to address the various issues. We feel that is the wrong direction. We looked to break the problem into smaller, more simple, chunks. Simple chunks can be implemented. Difficult chunks can be thought about, pondered and researched until we find simplifying principles and constructs that turn them into simple chunks. If something can not be simplified we would rather leave it unimplemented than create a monster that can not be tamed.[1]

While this sounds like we leave a lot unimplemented the opposite is true. Delaying the difficult chunks gives us more time to implement the first chunks "the right way". When those are complete, we have a better understanding of the system and those "more difficult chunks" become easier to simplify. Often we discover that those "difficult chunks" were hogwash that were not needed anyway. Either way, our strategy achieves more because we can remain focused on a smaller set of issues at a given time. It is with great hubris that someone thinks they can plan out an entire system without the benefit of the knowledge gained by first having solved smaller portions of the problem.

[1] A cynical version of this is described as "The New Jersey Approach" in [Gabriel].

4 The Services

We will now explain how we engineered each application using those two philosophies. Some fit easily into our philosophies. Others presented challenges.

4.1 E-Mail

The protocols we require for e-mail are the historic Internet standards that the world uses:

> Transport of mail must be via Simple Mail Transport Protocol [RFC821]
>
> Mailboxes must be accessed over the network via Internet Message Access Protocol - Version 4 (IMAP4) [RFC1730]
>
> Mailboxes must be stored in UNIX Mail format and be accessible from a UNIX platform

To meet these requirements, our supported client is Netscape Mail, which access mailboxes via IMAP4, on PC or UNIX. Some UNIX-oriented users use `elm`, `mh`, `mutt`, or even `/usr/ucb/mail`. Our servers are Sun Solaris 2.6 machines running Sendmail 8.8.8 [Allman] as an Mail Transport Agent (MTA) and procmail [Berg] as a Mail Delivery Agent (MDA). IMAP4 protocol is supplied by Sun's SIMS 2.1 IMAP4 Server product. We are evaluating the University of Washington IMAP4 server and may switch to it in the future. Because both IMAP4 servers store mailboxes in UNIX Mail format, we can change servers with little effort.

We selected Solaris over NT because UNIX scales better, is faster, can be made more secure, and it is easier to debug problems [Standish] [Kirch] [Petreley]. Sendmail is one of the few MTAs that we know of that is flexible enough to handle our

complicated configuration requirements. Our alias management system is complicated due to our large size (we import alias information from many sources to build our alias database). We have a fine, robust mail system via our UNIX servers. Why reinvent the wheel when we can give NT users access to our current wheel? Open protocols permit us to do just that.

4.2 File Services

Depending on our customer needs, different file service options are available. Some customers need access only from NT, others only from UNIX, others need to access their files from both. We feel that eventually all customers will need access from both; even UNIX-only users will want to share data with NT-only users.

NT clients access file servers using the Common Internet File System (CIFS) protocol [Leach] (which is Microsoft's new name for the Server Message Block (SMB) file protocol [SMB].) UNIX clients use the Network File System (NFS) [Stern] protocol. Some file servers support one or both of these protocols. File servers come in all sizes from small to extremely large. We feel at this time the file service marketplace can be summarized as in Figure 1.

Large	EMC Auspex		
Medium	NetApp	NetApp	NetApp
Small	Sun	SAMBA	NT Server
	NFS-Only	*Both*	*CIFS-Only*

Figure 1: The File Server Market

For small-scale NT file service, NT Server is appropriate. A UNIX Server is appropriate for small-scale NFS service. If the data must be accessed by both, a UNIX server running SAMBA [SAMBA] or Syntax TotalNET [TAS] is fine. We have multiple terabytes of data and it almost always needs to be accessed from both kinds of clients. Therefore those solutions have been nearly phased out in the past year.

For medium-scale file service with CIFS and NFS we choose Network Appliance Filer (referred to as the NetApp Filer) [Hitz1] dedicated file servers. A typical user has a directory on a NetApp Filer that is exported via NFS for access from UNIX and as a "share" available to NT systems. Some customers have requested to have the share only be a portion of their directory structure, usually a sub-directory called "PC". We get this request less often now. We are very happy with the NetApp Filers' ability to solve the problem of integrating NT and UNIX. On top of that we get snapshots, RAID and other features. Performance is exceptional for NFS as well as CIFS, almost dispelling the general perception that CIFS's design prevents fast implementations from existing.

Management of the NetApps is "free" since they access NIS for UNIX account data and NT Domain for NT account data. By keeping user names in sync, these systems require very little new administration tasks. The NetApp "does the right thing."

While the NetApp Filers are not inexpensive, we find their total cost of ownership is on par with other solutions. We occasionally price out an equivalent PC-based server for comparison and generally find the price per megabyte comparable after including RAID and other features. Such a system would not integrate NFS and CIFS as well, nor would performance be as good. Also, with our UNIX background, we find the Network Appliance File Servers easier to manage.

Our UNIX NIS configuration is automated and includes home spun components that update our NetApps (see [Limoncelli] for complete details). In fact, since the updates are automated as part of our NIS push system, additional NetApps require nearly zero additional work once it is included in our NIS database of NetApps. We encourage each large (200-400) group of customers to procure its own NetApp Filer which we then manage.

Because each small group of customers procures its own NetApp Filer, we have not had to evaluate solutions for large amounts of data. We have a large amount of data, but we have developed ways to manage it efficiently as many medium chunks of data.

4.3 Backups

Backups must be reliable. They must be on standard tapes in a format that can be read even without access to the backup software. To be cost effective a single server must be able to back up many machines and daily human activity must be minimal and simple.

We struggled with home grown solutions on our UNIX systems for years often employing full- or part-time staff just to change tapes. Later we adopted commercial software to gain quickly the new features we needed: Robot controlled tape library/jukebox, better tape handling, ability to back up non-UNIX systems, and enhanced index of tape contents.

We selected BudTool from IntelliGuard [BudTool] but other commercial products would serve our needs. BudTool's differentiators are (1) tapes are in UNIX "dump" format and therefore can be read without BudTool; (2) excellent support for NetApps; (3) use of the Network Data Management Protocol (NDMP) [Stager] open standard for for backup systems to access filesystems, tape drives, etc. over the network. This permits our backup servers to back up file servers to local tape drives or to the tape drives on other machines in the network.

BudTool is as complicated to manage as you make it. We maintain a relatively simple configuration and four huge tape jukeboxes and libraries. We have found their on-site support to be invaluable and worth the expense.

We do not back up our desktop systems. Customers know they are not to store data on them. We help customers conform to this policy by providing sufficient network bandwidth and fast file servers. That is, we make it more appealing to keep data on the server rather than being fascist about enforcing the policy. If a desktop disk dies we can replace it and reload the operating system and applications in about an hour using AutoLoad [Fulmer]. NDMP clients for NT are becoming available now so we are re-evaluating our desktop backup policy.

4.4 Printing

We find that printing is constantly fraught with problems due to printer jams and out of control print jobs that need to be cancelled. Doesn't anyone just print ASCII anymore? Network printing is fraught with badly written protocol stacks on the client and printer end.

In particular, most network printers prefer to have only one machine talking to them at a time (this defies our definition of a "network printer" but this is the reality we have to deal with). NT clients lose a lot of printing features unless they are talking directly to NT servers. Our bias is to spool to some central machine (or its hot standby) so that we have a single point of control when jobs need to be cancelled, etc.

Our solution is as follows: UNIX clients funnel jobs to a UNIX print server using the LPD protocol [RFC1179]. NT clients funnel jobs to an NT print server using the native NT print protocols. The NT print server funnels jobs to the UNIX print server using LPD. The UNIX print server is the only machine that directly talks to printers.

The UNIX print server is a Sun Solaris 2.6 host running LPRng [Powell] which is freely available software. LPRng completely replaces the printing system on Solaris, but is backwards compatible. LPRng's strong point is that it compensates for the broken LPR implementations frequently found in today's network devices (both clients that send jobs to it and the network printers that receive jobs). It converts non-PostScript data into PostScript. It detects and compensates for badly formatted PostScript data. It also does a fine job of accepting from and sending to properly functioning LPD implementations.

Funneling all jobs to a single UNIX server means a single spool to access when bad jobs need to be removed. In fact, LPRng can be configured to permit anyone to kill a job in the queue, something we do since we trust our customers. We have one machine spool the print jobs for each group of printers. This machine is a single point of failure but many of our UNIX compute servers can function as a stand-in for our print spooler when needed. All of our configuration files refer to the spooler by a DNS alias (CNAME) rather than the host's real name. If the spooler dies, a simple change to the DNS will direct all print jobs to a replacement machine.

4.5 Access to UNIX Hosts

Access to our UNIX servers is provided by Exceed [Exceed], an X Windows package for NT. This essentially turns the PC into a fully functioning X Terminal. In fact, we have reduced the number of X Terminals we purchase as a PC can cost about the same, yet can run PC applications locally. In particular, Netscape runs much faster on a PC compared to running it on a UNIX server and displaying it elsewhere via X Windows (whether the final display is on a PC or X Terminal.)

4.6 Internet/Intranet Services

Our standard desktops are loaded with Netscape's web browser. Customers choose their own tools to generate web pages; many prefer to use text editors such as `vi` or `emacs` from the UNIX servers. For customers that need collaborative document features, we use Netscape Enterprise Server [NSES] which is a Java-based application (and therefore runs on NT and all UNIX platforms) that lets users edit pages, lock/unlock them, use revision control, and control who may/may not edit a file. Because open protocols are used, our clients and servers can be of different platforms.

We choose UNIX for our web servers because management and scalability is critical. We securely mirror our web sites using "Stage" [Ches] which is only available on UNIX but is available to the public in source code form. Customers that wish to have external web pages maintain them on an internal server which is mirrored to the outside using "Stage". If our external web server were broken into, we could format the disks, reload the software, and "Stage" would copy the web pages back into place.

Our web proxy/cache is a Netscape Proxy Server [NSPS] running on a Sun Sparc Ultra running Solaris 2.6. While we use a transparent firewall [LMF] (i.e. one that does not require SOCKS or other proxies) users are encouraged to point their web browsers at our web proxy because it caches web pages.

Most of our customers prefer to host their web pages on our server. Customers with special requirements run their own web servers from their desktops. If traffic to their web site becomes considerable, they have the option of moving the data to a central server which has better network connectivity. However many of our customers are developing small, experimental CGI or Java applications and a personal web server suits their needs. In this case, they choose which server they prefer to install and run.

Our Discussion/Bulletin Board ("groupware") service is based on the Network News Transfer Protocol (NNTP) [RFC977], an open standard, so that NT as well as UNIX clients may access it. It is our pre-existing netnews server. Our netnews service is provided using InterNetNews (INN) [Salz] on a Sun Sparc Ultra with Sun On-line Disk Suite and enough disk space to store a month of news (except the "adult" stuff).

4.7 Calendar Management

There is a trend to do more team-based collaborative work in Bell Labs. That means more meetings than ever before. We now facilitate this process with a shared calendar server. Since there are no ratified standards for calendar server/calendar client interaction, we placed our bets on a vendor that we believe will adopt the IETF standards as they are born. We have an additional requirement that the client must be available for Sun Solaris, SGI IRIX and NT. This reduced our choices to Netscape Calendar Server and Client. We run the server on a Sun Solaris 2.6 server.

This is a recent addition to our network and so far we are happy with the fact that NT, Solaris and IRIX users can finally share calendars. Our Palm Pilot users are ecstatic with the interoperability.

4.8 Name Service and Account Administration

Our NT Domain service is provided by a pair of NT servers which are our Primary Domain Controller (PDC) and a Backup Domain Controller (BDC). These machines are also our Windows Internet Name Service (WINS) servers. We have "engineered them for reliability" in stupid, brute force ways because, unlike our UNIX servers, they can not easily be remotely rebooted or maintained. For example, installing new software often requires a reboot. Reliability is achieved by running no other services on them. We are unhappy that reliability has to be achieved this way. We are investigating other options: new software for UNIX servers

that turn them into PDCs; replacing NT Domain all-together with Light-weight Directory Access Protocol (LDAP) [RFC2251], an open standard for directory services; and other options.

Domain Name Service (DNS) [RFC1035] and Dynamic Host Configuration Protocol (DHCP) [RFC2131] services are provided by our existing UNIX servers. We require authenticated logins (single-use passwords via Hand Held Authenticators) to these machines since so much depends on DNS being reliable and authentic. For DHCP we use ISC's free DHCP reference implementation [DHCP] and are extremely happy with its flexible configuration file format. We actually generate the configuration file from our NIS data with a `perl` script. SAs don't actually have to know how to modify the DHCP database. They enter certain information and a perl script generates the rest. We also use the ISC "BIND" DNS software [BIND].

5 No "single log on" yet

As we mentioned earlier, we aim to simplify the complex and delay implementing what we haven't yet simplified. The challenge of a single network login is in that later category.

Currently every user has two accounts, a NT Domain login and a UNIX (NIS) login. While other NT and UNIX integration papers have focused on integrating logins we saw it as a holy grail that would waste our time if we pursued it.

We chose to use human discipline instead. That is, our SAs know to always use the same user name for a customer when creating their NT and UNIX logins. That is, my NT Domain user name is "tal" and my UNIX NIS login is also "tal". If I wish to change my password, I must do it twice, once for each system. (Or, I may choose to maintain separate passwords).

The ability of a NetApp to be programmed so that that "tal" in UNIX is "tal" in NT means the problem of dual platform file access is solved automatically. (NetApp permits us to specify exceptions to this rule if need be.) File permissions are handled like magic based on which protocol the request came from. That is, a request received by CIFS has NT file semantics and a request received by NFS has UNIX (POSIX) semantics [Hitz2].

In hindsight, if an integrated NT and UNIX environment simply means the same user name gets you to the same set of files then we achieved our goal without trying. Thus demonstrating the superiority of our "delay the complex" philosophy. We "missed the bullet" on that one.

If you feel that we are rationalizing the fact that we shirked our responsibility to achieve 100% perfect integration we have two responses:

First, we invite you to interview our customers who feel we have provided for the integration they needed. If their needs are met, the features we didn't complete aren't needed or we can surprise and delight them with such features when they do arrive.

Secondly, we could have spent the last two years developing a single login system and not had time to complete the other services we created. By the time we would have been done, LDAP (an open standard) has arrived. We would be left with a home-grown, incompatible single-login system that would break with every new release of NT and fewer of our other services would be complete. Instead we have a solid foundation to build on and are ready to embrace the coming third party products based on the newly developed standards. We are currently investigating the new single-login options available to us.

6 Protocol Summary

Figure 2 summarizes each application, the protocol selected, and the client and server used. We feel we accomplished our open protocol goal as well as possible given the challenges presented to us and made compromises only when essential. For example, while the CIFS protocol is relatively closed (compared to, for example, NFS) it would not have been cost effective to replace the file service client software on the NT systems. In this case, simplicity suggested that we let clients use the protocol that they use best rather than shoehorn them into a new protocol. The ease of use features of NT's native printing system forced us to use it on clients but only to bring print jobs to a central machine that would store the jobs then forward them using an open protocol. Some protocols were in-house (stage) while others are experimental (the calendar-related protocols). Overall, we were able to build the environment we wanted and do so by using open protocols.

Application	Protocol	Open/Closed	Preferred Client	Server
E-mail Reading	IMAP4	Open	NS Mail	Sun SIMS
E-mail Transport	SMTP	Open	n/a	Sendmail 8.8.8
File Service (UNIX)	NFS	Open	Native UNIX OS	NetApp ONTAP
File Service (NT)	CIFS	Closed	Native NT OS	NetApp ONTAP
File Backups	NDMP	Open	NetApp, UNIX, etc.	BudTool
Printing	LPD / PostScript	Open	LPRng*	LPRng
UNIX Access	X / Telnet	Open	Exceed	n/a
Web Access	HTTP	Open	NS Communicator	LMF Firewall
Web Servers	HTTP	Open	NS Communicator	NSES on UNIX
Web Mirroring	Stage	Free	`stage`	`staged`
Collaborative Publishing	HTTP/FTP/Java	Open	NS Communicator	NSES on UNIX
Bulletin Board	NNTP	Open	many/any	INN
Calendar Mgmt	CAP, CIP, etc.	Open	NS Calendar	NS Cal Server
Host Name Service	DNS, DHCP	Open	n/a	ISC DNS/DHCP
Login/Directory	NT Domain	Closed	n/a	NT

Figure 2: Summary of Protocols/Applications Used

7 Conclusion

We took a solid UNIX environment and layered NT desktop services on top of it and created a rich, tightly integrated environment of NT and UNIX computers. It was critical to base our architecture on open standards for protocols rather than specific applications. We then could choose the server based on our needs of reliability, scalability, and manageability and let customers independently choose their clients based on their needs of platform (NT or UNIX), features and preferences. We selected Sun Solaris (or other UNIX variant) or NetApp for almost all of these back-end platforms. We only resorted to NT Server for PDC's, BDC's and NT Print Services. We provide support for the client applications we have evaluated to be the best business choice and require self-support for all others. This encourages conformity to our recommendations without being fascist. The result is an extremely manageable, reliable and scalable NT desktop service and extremely happy customers.

8 Acknowledgments

System Administration is a collaborative effort. We could not have done this work without the involvement of all of our SA team as well as our SA counterparts at our Holmdel, NJ facilities. The feedback from our users was invaluable. Special thinks to Josh Simon for his excellent editing.

References

[Allman] "Sendmail". Eric Allman. `http://www.sendmail.org/`

[Bashein] *A Credibility Equation for IT Specialists.* Barbara J. Bashein, M. Lynne Markus. Sloan Management Review, 38:4. 1997.

[Berg] `procmail`. S. R. van den Berg. `ftp://ftp.informatik.rwth-aachen.de /pub/packages/procmail`

[BIND] BIND. Internet Software Consortium. `http://www.isc.org/bind.html`

[BudTool] BudTool. IntelliGuard, Inc. `http://www.intelliguard.com/`

[Ches] "Cget, Cput, and Stage". Bill Cheswick. Proceedings of the USENIX 1997 Annual Technical Conference. January, 1997. `http://www.bell-labs.com/topic/swdist/`

[DHCP] DHCP. Internet Software Consortium. `http://www.isc.org/dhcp.html`

[Exceed] Exceed. Hummingbird Communications, Ltd. `http://arctic.www.hummingbird.com /products/exceed/`

[Fair] "Software versus Protocol (or file format) Standards". Erik E. Fair, May 1998. `http://www.clock.org /~fair/opinion/open-standards.html`

[Fulmer] "AutoInstall for NT". Robert Fulmer. Proceedings of the 2nd Usenix Windows NT Symposium. August, 1998.

[Gabriel] *The Rise of Worse Is Better*. Richard Gabriel. reprinted in *The UNIX Haters Handbook* appendix C. IDG. 1994.

[Hitz1] "An NFS File Server Appliance" Whitepaper. Andy Watson. Network Appliance, Inc. `http://www.netapp.com/technology/architecture.html`

[Hitz2] "An Integrated Model for NT and UNIX File Service". Hitz, Allison, Borr, Hawley, Muhlestein. Proceedings of the 2nd Usenix Windows NT Symposium. August, 1998.

[Kirch] "Microsoft Windows NT Server 4.0 versus UNIX". John Kirch. June 1998. `http://www.kirch.net/unix-nt.html`

[Leach] "Common Internet File System (CIFS/1.0) Protocol". Paul J. Leach, Dilip C. Naik. December, 1997. `draft-leach-cifs-v1-spec-01.txt`

[Limoncelli] "Building a network for Bell Labs Research South". Tom Limoncelli, Tom Reingold, Ravi Narayan, and Ralph Loura. Proceedings of the Eleventh Systems Administration Conference (LISA '97). October, 1997.

[LMF] The Lucent Managed Firewall. `http://www.lucent.com/security/`

[NSES] Netscape Enterprise Server. Netscape, Inc. `http://merchant.netscape.com/netstore/servers/enterprise.html`

[NSPS] Netscape Proxy Server. Netscape, Inc. `http://merchant.netscape.com/netstore/servers/proxy.html`

[Petreley] "The new Unix alters NT's orbit". NC World. April 1998. `http://www.ncworldmag.com/ncworld/ncw-04-1998/ncw-04-nextten.html`

[Powell] "LPRng - An Enhanced Printer Spooler System". Patrick Powell, Justin Mason. Proceedings of the Ninth System Administration Conference (LISA '95). September, 1995. `ftp://iona.ie/pub/plp/LPRng/`

[RFC821] "RFC821: Simple Mail Transfer Protocol". J. Postel. August, 1982.

[RFC977] "RFC977: Network News Transfer Protocol". B. Kantor, P. Lapsley. February, 1986.

[RFC1035] "RFC1035: Domain names - implementation and specification". P. V. Mockapetris. November, 1987.

[RFC1179] "RFC 1179: Line printer daemon protocol". L. McLaughlin. August, 1990.

[RFC1730] "RFC1730: Internet Message Access Protocol - Version 4". M. Crispin. December 1994.

[RFC2131] "RFC2131: Dynamic Host Configuration Protocol". R. Droms. March 1997.

[RFC2251] "RFC2251: Lightweight Directory Access Protocol (v3)". M. Wahl, T. Howes, S. Kille. December 1997.

[Salz] "InterNetNews: Usenet Transport for Internet Sites". Rich Salz. USENIX Conference Proceedings. Summer 1992. `http://www.isc.org/inn.html`

[SAMBA] "The SAMBA FAQ". The SAMBA Team. June 1998. `http://samba.anu.edu.au/samba/`

[Smallwood] "Whither The Customer?". Kevin C. Smallwood. ;login: and counterpoint by Rob Kolstad.

[SMB] "CIFS in a Nutshell". Microsoft Corp. January, 1997. `http://www.microsoft.com/workshop/prog/cifs/nutshell.htm`

[Stager] "Network Data Management Protocol". R. Stager, PDC and D. Hitz, Network Appliance. August 1997. `draft-stager-pdc-netapp-backup-04.txt`

[Standish] "SUN Also Rises: Solaris vs. NT". The Standish Group. May 1998. `http://www.standishgroup.com/syst.html`

[Stern] "NFS and NIS". Hal Stern. O'Reilly and Associates. July 1991.

[TAS] TotalNET Advanced Server v5.2. Syntax, Inc. `http://www.syntax.com/totalnet/tasbody2.htm`