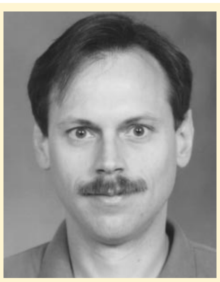
CLIENT/SERVER

AIX Connections

By Bret R. Olszewski and Kay Chang

AIX Connections has established a new standard in UNIX-to-PC connectivity. The combination of services, with product-level robustness and performance, should prove attractive to companies seeking to integrate AIX® applications with PC clients. This article discusses the architecture and performance characteristics of AIX Connections.

Today's complex business environments have many computers interlinked to enterprise data—and often the Internet. Frequently, many different types of computers, both clients and servers, are connected by a variety of protocols and network hardware. This potpourri increasingly challenges those employed to maintain order.



Bret R. Olszewski

Kay Chang

AIX Connections Overview

AIX Connections is a feature of AIX 4.1.4 that was implemented with a simple goal: to provide the premier solution for integrating IBM-compatible and Macintosh® personal computers with UNIX® solutions. Its benefits include access to AIX scalability, consolidated system management, and ease of cooperation and integration with more than 10,000 AIX solutions.

AIX provides unmatched scalability, encompassing hardware from notebook computers to massively parallel systems. Since this is done with binary compatibility, growth is virtually unlimited. With AIX Connections as a PC server, networked PCs can use all of the robust AIX features. For example, printing to a high-speed laser printer can be as simple as redirecting your LPTx.

One thing is certain in the world of PCs—they are all different. Network operating systems implement incompatible protocols, administration, and filesystem structure. Using network operating systems to configure and administer multiple servers is complex and expensive.

But consolidating onto a single AIX Connections solution allows files and printers to be shared by different types of networked PCs. When one user generates a file on a Macintosh, it can be used by another user on a PC running Microsoft® Windows™ 3.1—transparent to both users. This works because AIX's Journaled File System is used directly and without abstraction, allowing files to be controlled through AIX's security apparatus for consistent use by all services. User administration, which is also consistent with AIX, does not require redundant updates as some other solutions.

Since all AIX Connections are implemented consistently, there are no incompatibilities with other AIX solutions. Files and printers are shared. Protocol stacks allow AIX Connections to coexist on the network with other solutions (such as databases) as well as customer-written applications, shown in Figure 1.

AIX Connections Components

AIX Connections includes the following elements:

LSserver: LSserver consists of network protocols, such as NetBEUI, TCP/IP, Server Message Block (SMB), and file, print, and terminal emulation services. LSserver provides services to PC clients running OS/2®, DOS, Windows 3.1, Windows 95, Windows for Workgroups, and Windows NT™ workstations. SMBs are the basic units of the interface, originally developed by IBM, Intel®, and Microsoft, then standardized by X/Open™. AIX Connections provides NetBIOS Control Block (NCB), X/Open Transport Interface (XTI), and Transport Library Interface (TLI)

AIX Connections

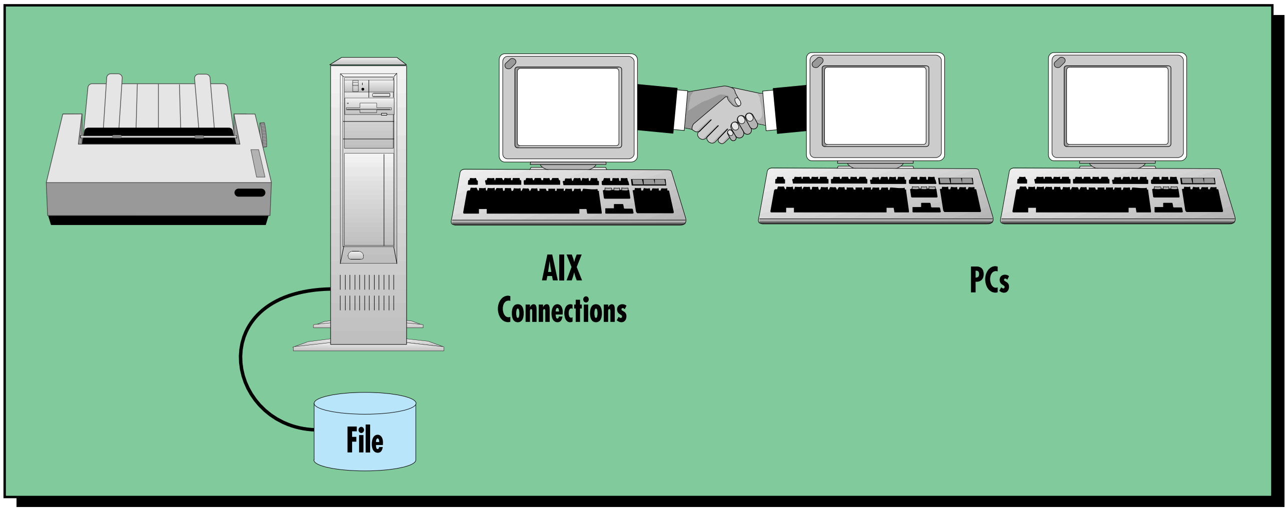


Figure 1. AIX Connections

libraries for program interfaces that map into SMB protocols.

NWserver: NWserver consists of network protocols, such as IPX/SPX, and provides file, print, and terminal emulation services. NWserver provides services to PC clients running OS/2, DOS, Windows 3.1, Windows 95, Windows for Workgroups, and Windows NT workstations. AIX Connections provides XTI and TLI libraries for programming interfaces that are mapped into NetWare Core Protocol (NCP) protocols for file and print.

MACserver: MACserver consists of Apple® Macintosh network protocols such as AppleTalk® Data Stream Protocol (ADSP) and AppleTalk Transaction Protocol (ATP). Apple File Protocol (AFP) and Apple Print Protocol are both based on ATP. MACserver provides file and print services.

TNclient: TNclient gives an AIX host the opportunity to access either SMB or NetWare® network operating systems as a client. For example, an AIX system can mount a filesystem that is actually served from another network operating system. Available services include the following:

- Protocols and requester to SMB servers, such as OS/2 LAN Server, Windows NT Advanced Server, and Microsoft LAN Manager for logon, filesystem mounts, and printer services
- Protocols and requester to NetWare servers, such as the Novell® NetWare Server and NetWare Server for AIX for logon, filesystem mounts, and printer services

The combination of these elements, along with the extensive TCP/IP-based services of AIX, present a wide-ranging solution to the “tower of babel” computing environment we face today.

Architecture

Figure 2 shows the internal architecture of AIX Connections. The diagram shows three sets of entities: protocol services, directory services, and daemons. Interlinked with each of those are AIX services—some explicitly stated such as sockets, and some implicitly invoked, like the filesystem. The architecture effectively consists of four parts:

- **AIX services:** The architecture focuses on using as much of the capability provided by AIX as possible. Instead of creating unique functions, AIX Connections depends on AIX services, such as Data Link Provider Interface (DLPI), Streams, TLI, and security.
- **Communication protocols:** AIX Connections requires that various communication protocol stacks, such as TCP/IP, IPX, and NetBIOS, be implemented.
- **Control processes for file, print, and TTY:** Daemons provide support for service requests initiated by clients via protocols.
- **Programming interfaces:** Control processes use these interfaces for service. Interfaces are also provided for customers to write applications.

The description below illustrates how these components are used by AIX Connections.

AIX Connections

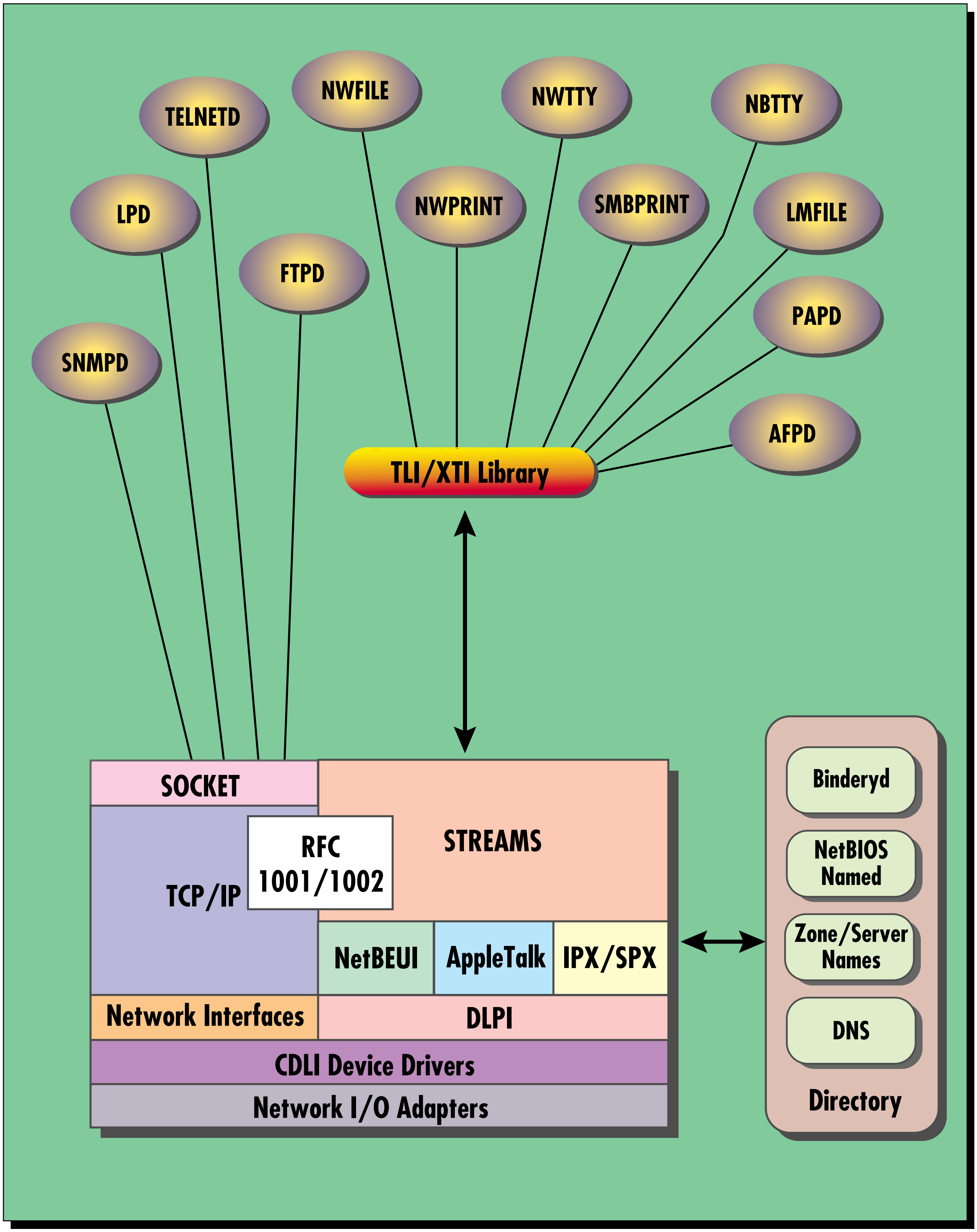


Figure 2. Internal Architecture of AIX Connections

Base AIX Facilities

AIX Connections uses various base I/O services and lower-level protocols, including LAN network device drivers, network interface for TCP/IP, and data link protocol drivers such as DLPI.

The DLPI driver provides two main functions:

- To bridge between network device drivers below and STREAMS protocol drivers above
- To process primitives for implementing IEEE® 802.2 services

It uses both connectionless and connection-oriented services. STREAMS drivers, which require connection-oriented services such as NetBEUI, must depend on the connection-oriented features. Others such as Datagram Delivery Protocol (DDP) and IPX use only the connectionless capability.

Network and Transport Protocols

AIX Connections protocols such as NetBEUI, AppleTalk, and IPX/SPX are implemented primarily via Streams. NetBEUI, which depends primarily on connection-oriented services, ties its session map to the connection-oriented DLPI state, where NetBEUI sessions are brought down when the link-level connections are disconnected.

AppleTalk Datagram Delivery Protocol (ADP), the connectionless service, assumes the logical network connection is always present. AppleTalk Session Protocol (ASP) provides the transport connection for end-to-end file service.

IPX, which originated from the XNS protocol invented by Novell, is connectionless, whereas SPX is connection-oriented. The SPX protocol is implemented by a STREAMS module pushed on top of the IPX driver.

RFC1001/1002, somewhat like TP0 of Open System Interconnection (OSI), implements a null layer using TCP/IP as a transport. This exists primarily for interpreting a mapping that is defined by a NetBIOS service to the TCP/IP service and protocol.

File Services

LMfile, a component of the LSserver, provides a file access service communication over SMB protocols. It requires NetBEUI or TCP/IP transport that is accessible via Transport Library Interface.

NWfile, a component of the NWserver, provides file access service communication over NCP protocols. It requires IPX access via TLI.

AFPD, a component of the MACserver, provides file access service using Apple File Protocol. It requires the Apple Session Protocol via TLI. The File Services components (LMfile, NWfile, and AFP) not only provide file access to the PC clients, but also perform necessary user and group security. They all support the file security scheme used by AIX called Access Control List (ACL).

Print Services

SMBprint provides an interface to the AIX spooler by catching the client's Print SMB requests. It then converts and routes them to the AIX print queue.

NWprint provides an interface to the AIX spooler for catching the client's print NCP requests, then converting and routing them to the AIX print queue.

Papserver provides services that handle print requests from the Macintosh clients and send them to the AIX printer queue.

Terminal Emulation Services

NBtty uses the SMB protocol to provide pseudo-TTY emulation and login service. NWtty is a terminal emulation similar to Novell's Network Virtual Terminal (NVT) function. No terminal emulation for the Macintosh is provided.

Directory Services

Because of the evolutionary nature of various PC protocols, directory services such as naming conventions differ. PC clients expect file names and directory structures on the server to be exactly like their local filesystems. AIX Connections directory services must include the flexibility to handle the PC client assumptions, which results in various ways of handling different namespaces and conventions.

AIX is the common denominator underneath these various resource names. A server name is restricted to a convention, which enables any PC client to read names and understand them. File length can be restricted, or users and group names can be limited. The fundamental underlying element of namespace and security is the AIX user; each PC client session is mapped to an AIX user. This mapping of disparate features of various network operating systems into AIX conventions helps to achieve interoperability.

Performance Aspects

Having explored the extensive capabilities of the architecture, let's explore how it works and performs in a practical setting.

AIX Connections provides the premier solution for integrating IBM-compatible and Macintosh personal computers with UNIX solutions.

About BAPCo

Business Applications Performance Corporation (BAPCo) is a non-profit corporation that develops and distributes a set of objective performance benchmarks based on popular software applications and industry-standard operating systems. It was founded to foster the creation of meaningful benchmarks in an applications environment.

BAPCo benchmarks are unique because they are based on software packages commonly found in retail computer software stores. Using benchmarks based on business applications enables users to conduct evaluations of system handling realistic workloads in environments users might often encounter. This approach provides meaningful data for evaluating systems because the results are relevant to demands they would typically place on systems through day-to-day workloads.

Membership

Membership is open to any organization interested in contributing to the corporation's goals and purposes, while adhering to the corporation's code of conduct. Current members include companies such as Compaq®, Dell®, Hewlett-Packard®, IBM, Intel, InfoWorld®, Microsoft, AT&T® Global Information Solutions®, DEC®, and Unisys®, plus many more. Additional members are invited to contribute to the development of the benchmarks.

Conducting Benchmarks

Since they do not have a lab facility, BAPCo has designed benchmarks so that testing can be done on a company's own systems, following the standards set by BAPCo. BAPCo's method of designing benchmarks focuses on an actual PC user's level of usage. All participating members in the benchmark development groups must follow this guideline. Participation in these groups is open and encouraged. Each participating company receives one vote on the developed benchmark to ensure objectivity.

BAPCo will provide large, long-running, realistic scripts and data that test a wide range of applications and system resources. BAPCo's Workload Characterization methodology is based on modeling user activities at the functional level. The mode is subsequently used to develop scripts similar to the end-users' workloads. To do this, they use the correct functional mix and appropriate data sets (as derived from the models) along with the most popular applications in various categories. This type of characterization produces workloads that closely resemble actual usage patterns of the intended audience.

Benchmark Results

BAPCo will publish its guidelines for executing and reporting benchmark results. It will also clearly define system configurations and formatting and disclose rules for test results.

All BAPCo benchmark licensees (companies that have purchased the benchmark suite) may publish benchmark results obtained on their machines. BAPCo will also publish its own quarterly newsletter, *The BAPCo Report*, which will include informative system-level performance articles and performance benchmark system numbers.

BAPCo Products

SYSmark is a family of benchmark suites from BAPCo continuing from the SYSmark92 and SYSmark93 suite. SYSmark95 for Windows will be used by those interested in evaluating system-level performance of PCs running Windows applications. SYSmark for Windows NT is the industry's first cross-architecture benchmark on Intel, Alpha, and MIPS platforms. SYSmark/FS for Servers is intended as a tool to evaluate the performance of a file server in a network environment servicing requests from network clients.

For more information about BAPCo, call 408-988-7654 or write to:

BAPCo
2200 Mission College Boulevard
M/S RN2-02
Santa Clara, CA 95052-8122

Environments

Knowing the environment or market is key to selecting a benchmark to measure performance. The benchmark must be representative of the work that customers perform, so that the system is tuned to customer requirements.

The traditional environment includes file and print servers that are segmented into three classes:

1. **Servers that provide a wide range of services to PC clients**, including file and print services. Most, if not all, executables reside on the server. This is generally the most performance-sensitive environment.

2. **Servers that provide primarily file access.** These servers function primarily to allow clients to share files. Executables are not usually served and print services are not provided. These servers often support many clients with mission-critical data and have very high reliability requirements.

3. **Servers that provide primarily print services.** Often a site will attempt to centralize print services onto a large server. RISC System/6000® servers are particularly good candidates for such a system.

File Server Benchmarks

Several viable PC-based file server benchmarks exist today. Benchmarking file servers can take place in two dimensions: responsiveness and capacity. A responsiveness benchmark may measure latency of a send/receive request from a client to a server. Another responsiveness benchmark might be to have a client read a file as fast as possible from a server. Both tests are interesting because responsiveness is important for PC productivity applications. Unfortunately, responsiveness tests are frequently not indicative of the capacity of a server since a server may need to service requests from many clients simultaneously. A throughput benchmark is required for capacity measurements.

Three good capacity benchmarks were evaluated for use with AIX Connections: IBM's Claire,

Server Structure

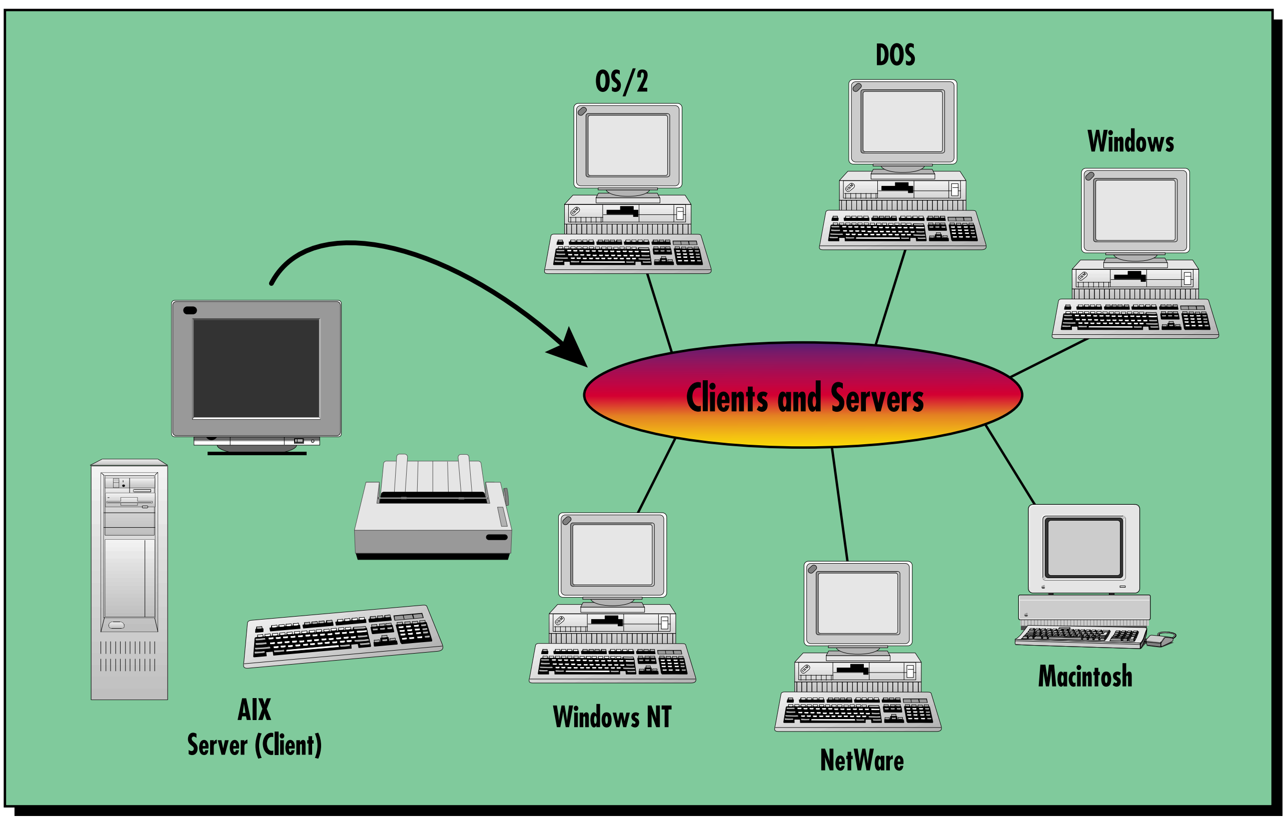


Figure 3. Server Structure

Ziff-Davis' PC Bench 3.0, and the BAPCo file server benchmark. All are based on existing PC applications or their measurements. In this article, when we refer to the BAPCo benchmark, we are referring to the file server benchmark. This is important because BAPCo supports several benchmarks. Currently, the BAPCo file server benchmark is the most recent and sophisticated of those evaluated. Since it is used heavily by the IBM LAN Server team, it was selected for measuring AIX Connections.

As of this writing, the BAPCo benchmark standardization is not complete. Therefore, the results discussed should not be compared with future, authorized results.

BAPCo File Server Benchmark

The BAPCo benchmark is composed of a set of scripted sessions on Lotus 1-2-3® for Windows, Word for Windows, dBASE IV®, Harvard Graphics®, cc:Mail®, Freelance Graphics® for Windows, Paradox® 3.5, WordPerfect® for Windows, and Microsoft Excel. The benchmark, which runs on IBM-compatible personal computers, is currently not applicable to Apple products.

The benchmark is constructed so that each PC client has a directory structure with unique data for the benchmark; the executables are shared on a directory structure. Each client executes the same scripts, but the order is staggered to reduce overlap. So user01 will begin in application Lotus 1-2-3 and end in Microsoft Excel and user02 will begin in Word for Windows and end in Lotus 1-2-3. In spite of the staggering, there are often periods in the benchmark when multiple clients are executing the same application. The DOS executables are local to the client, but the Windows 3.1 executables, drivers, libraries, and configuration files are accessed from the server.

The benchmark includes other common server functions such as printing and locking. Each BAPCo session includes 19 file prints. The dBASE portion of the benchmark includes record locking.

The output metric of BAPCo is scripts per minute—the number of scripts executed (nine per client times the number of clients) divided by the execution time. This is one of BAPCo's major problems: it is not accurate to compare results on dissimilar hardware configurations because client performance matters. Client performance depends on the processor, cache, graphics subsystem, and the efficiency of the network adapters used. Additionally, the amount of memory available on the client can have a significant impact on benchmark results. Response time is not measured.

The benchmark is structured so the PC clients do not emulate user think time or typing-rate delays between key strokes and mouse movements; therefore, each PC runs the applications as fast as it can. This is beneficial to the cost of running the benchmark since fewer PCs are

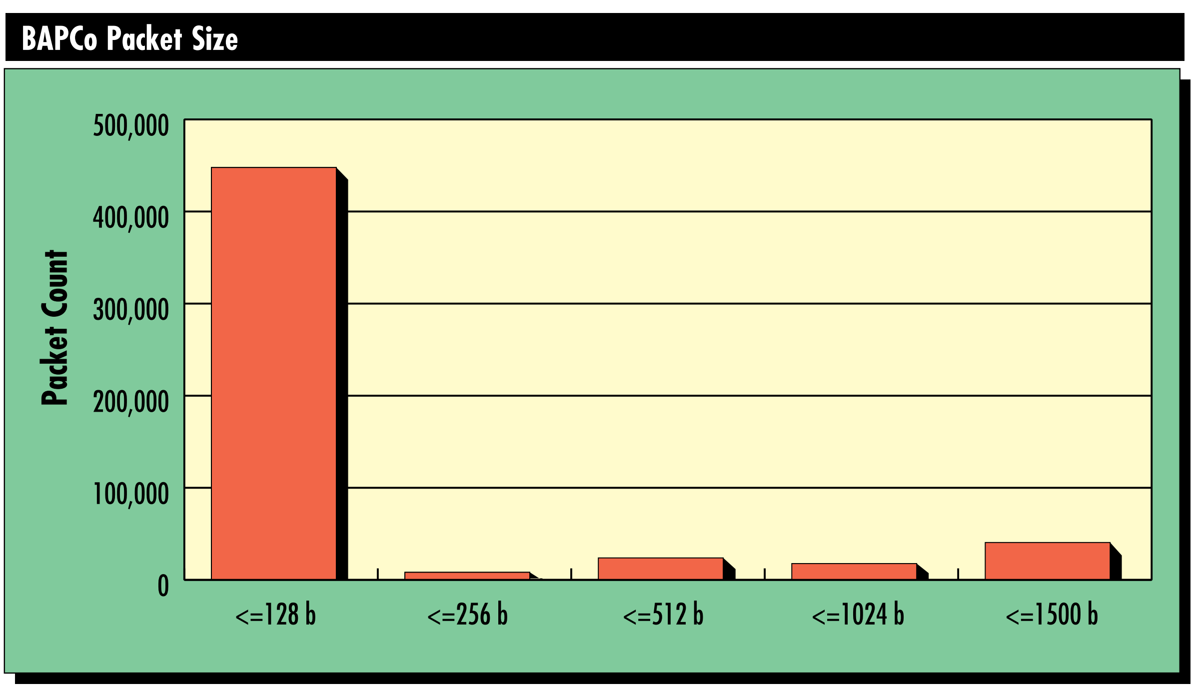


Figure 4. BAPCo packet size measures on NWserver

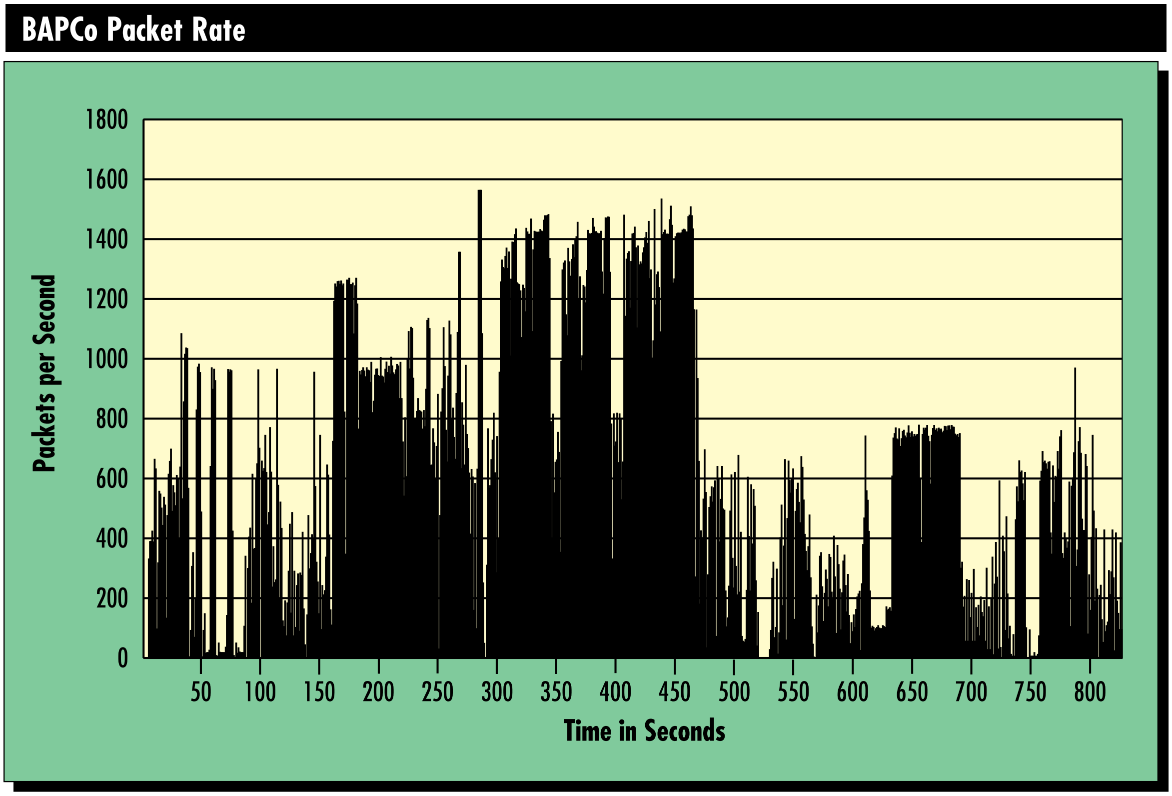


Figure 5. Packet Rate over time for BAPCo using NWserver

needed to saturate a server. It is also a potential problem, since the benchmark does not measure scalability effects that the server might encounter while supporting hundreds or perhaps thousands of real PCs.

BAPCo Benchmark on AIX

The server hardware configuration used for this article included a RISC System/6000 Model E20 (100 MHz 604) with 64 MB of memory, a single 1 GB SCSI disk, and two PCI Ethernet™ adapters. The client configuration included a set of IBM P330 (Pentium™ 90 MHz) systems, each with 8 MB of memory. Clients were uniformly distributed on two Ethernet segments, each segment directly attached to an adapter on the server.

This section will discuss the profile of BAPCo on AIX, using NWserver and NetWare for AIX.

Like most networking benchmarks, the BAPCo benchmark can be viewed as a mix of distinct transactions. Since the transaction types were defined by the communications protocols used, the number of transactions or requests to complete a BAPCo script varies when comparing NetWare for AIX, NWserver, LSserver over NetBEUI, and LSserver over TCP/IP (RFC). At the most atomic level, we can view the number of network packets (it is possible for a protocol transaction to span multiple network packets) as a measure of work. As with NFS, most packets are quite small. Figure 4 shows the distribution of network packet sizes as measured from a single-client execution of the BAPCo benchmark on Ethernet. Remember that Ethernet only allows packets of up to 1500 bytes.

The distribution of network packets over time shows a “bursty” pattern, as expected from end user applications. Figure 5 shows the number of packets sent and received per second during a single-client execution of BAPCo. The area of high packet rates, from approximately 150 seconds to 450 seconds, is where dBASE IV and Harvard Graphics execute. Obviously, these are the most I/O intensive applications in the mix. A single client drives this packet distribution over Ethernet. Since the client in BAPCo does not emulate user think time, it executes as fast as possible. The faster the client—in CPU, graphics, and network adapter efficiency—the more load it can generate on the server. Although fast clients have the horsepower to drive tremendous network load, they rarely do so. Clearly, network bandwidth of faster LANs, such as 100 Mbit Ethernet, are sorely needed as client performance

NetWare Request	Percentage of Total Requests
File Read Requests	77%
File Write Requests	15%
File Open Requests	5%
Directory Requests	3%
Other Requests	<1%

Figure 6. BAPCo script of NetWare request profile

and network-centric applications continue to increase.

Moving further up the food chain of a file server operation, we can measure the basic transaction request types for protocols. Figure 6 shows the NetWare request counts for a single client execution of BAPCo as measured from `sconsole` on NetWare for AIX. Note that the request mix is dominated by file reads and writes.

Figure 7 shows BAPCo's file I/O usage on the server. It indicates the total megabytes of data transferred to and from the filesystem as well as the number of reads and writes for the most frequently used files in the benchmark.

The table data also implies the small sizes of reads and writes used in the benchmark. In particular, consider the 00000008.$$$ file used in Harvard Graphics. The file is opened three times in the benchmark for importing and exporting graphics data. Each file read and write seems to average only eight types. Thus, while the Harvard Graphics script does not move a large amount of data, the CPU consumed is quite high because of the many system calls required to write the data. Since most filesystem reads and writes are cached to memory, the system disk utilization is low.

This BAPCo analysis leads to two conclusions. First, the networked environment is sensitive to the number of clients (and their network bandwidth requirements) on each LAN segment. Normally, it requires at least two Ethernet segments to feed an E20 server. Second, under most circumstances, the CPU will be the limiting factor in server performance. The next section examines how one server actually performs under load testing.

BAPCo Testing on NWserver

This section details measurements performed on an E20 server running NWserver with varying client load. One useful feature of NWserver is that there is no need for complex tuning parameters. In fact, the only two parameters available are the client shell and the enabling or disabling of packet burst on the server (disabled in our case). This makes it less complex to run at high utilization than NetWare for AIX. Under all measures, the I/O wait time was small, even with only a single disk on the server system.

File	MB Transferred	#Filesystem reads	#Filesystem writes	Application
dbase.ovl	9.6	12013	0	dBASE IV
wp.fil	5.7	6343	0	WordPerfect
tmpdnlba.$db	4.3	2839	2658	dBASE IV
customer.mdx	1.8	6176	0	dBASE IV
win386.exe	1.6	1683	0	Windows 3.1
paradox.aux	1.4	3601	0	Paradox 3.5
user.exe	1.2	1528	0	Windows 3.1
flwmain.exe	1.1	1522	0	Freelance
customer.dbf	0.9	4740	734	dBASE IV
00000008.$$$	0.7	43522	43522	Harvard Graphics

Figure 7. Filesystem measure of BAPCo with NWserver

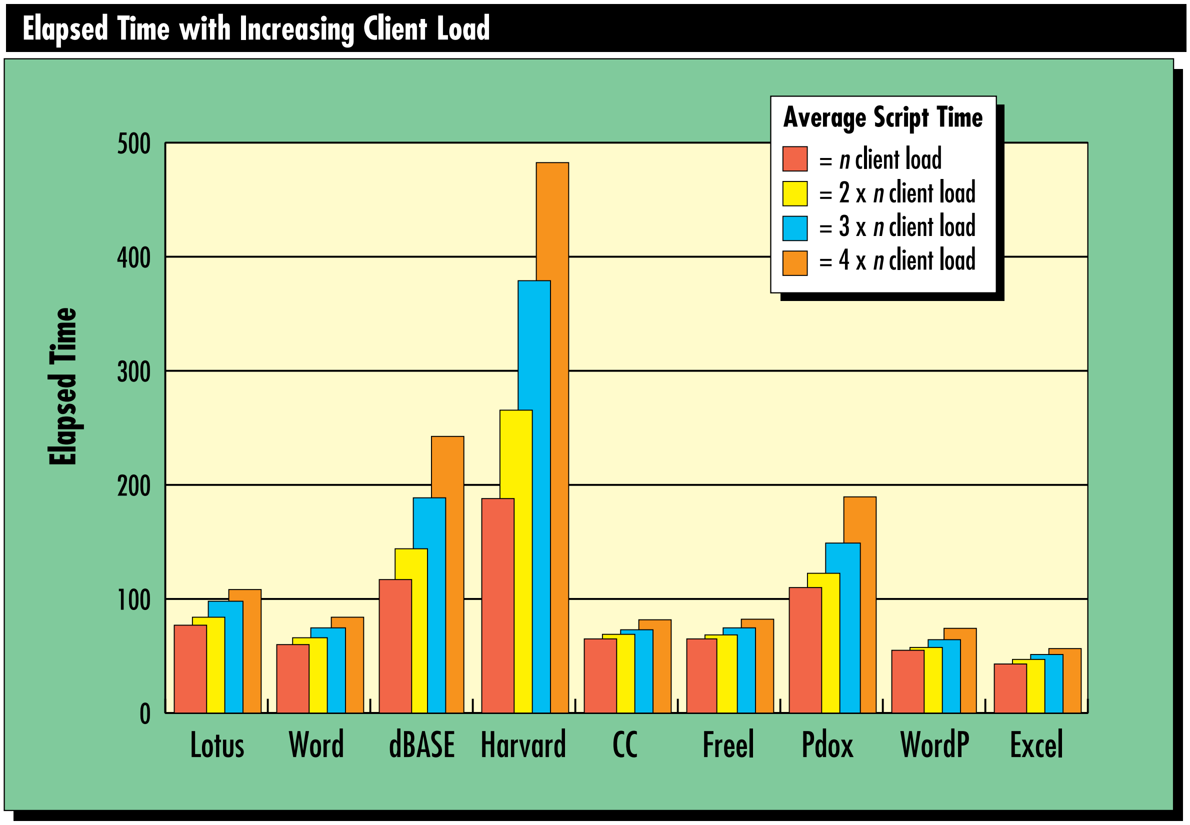


Figure 8. NWserver per script elapsed time with increasing client load

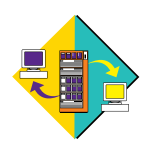
The CPU consumption of the workload is primarily system time, reflecting protocol processing and filesystem activity.

As load on the server is increased by running more clients, the time to complete each script on the client also increases because of queuing for resources, both network and server CPU. Figure 8 shows the increasing elapsed time for the scripts with increasing load.

The CPU utilization increases from 41.5% in the lowest case to 96.8% in the heaviest case. The applications that are most I/O intensive, namely dBASE IV, Harvard Graphics, and Paradox 3.5, suffer the most increase in runtime when the server load is increased. Other applications are less sensitive to server load, obviously employing more of their time executing in the client, drawing screens, and doing calculations.

Clearly, NWserver scales well, running user applications up to very high CPU utilization. However, keystroke-intensive and mouse movement-intensive operations do begin to experience occasional noticeable response time delays at high CPU utilization. Typically, you would not want to run a server at such high utilization, but it is good to know that high-load transient behavior can be handled gracefully.

Although it is not evident from the measurements, we have learned a lot by tuning AIX Connections using the BAPCo benchmark. We have improved the NetBIOS/NetBEUI protocol stack, improved STREAMS services, and removed redundant filesystem operations. The combination of these improvements has almost tripled performance improvement over the initial implementation.



Bret R. Olszewski, IBM Corporation, 11400 Burnet Road, Austin, TX 78758. Mr. Olszewski is a senior programmer working on MP performance. He joined IBM in 1989 and has worked on various aspects of AIX performance. He has a BS in Computer Science from the University of Minnesota.

Kay Chang, IBM Corporation, 11400 Burnet Road, Austin, TX 78758. Ms. Chang is an advisory programmer in AIX communication architecture, currently working on AIX Connections. She has an MS in Computer Science from Wright State University in Dayton, Ohio.