Internet Suite Application Protocols

In this report:

Datapro Summary

The Internet suite of protocols is robust and rich with application services. Most users view the Internet protocols in terms of these application services, which usually run over TCP/IP middle layer protocols. Based on the client/server model, Internet services span the basic functions of virtual terminal support, electronic mail, file transfers, and name service. Advanced services include a distributed windowing system, a network file system, and a network management protocol. The demand for these services is fueling the demand for TCP/IP-based enterprise Internets.

The Internet protocol suite continues in great demand in the standards-based networking marketplace. This demand should accelerate into the late 1990s. Once the protocol suite of choice for the U.S. military and a handful of universities, Transmission Control Protocol/Internet Protocol (TCP/IP) is now gaining popularity within the business community. Many businesses have become disillusioned with the excruciatingly slow pace of OSI deployment. Consequently, businesses are turning to TCP/IP to form the building blocks for their enterprise internetworks.

As more organizations embrace TCP/IP, the timetable for OSI ubiquity becomes correspondingly shifted to the right. It is unlikely that an organization building a TCP/IP-based enterprise network today would migrate to OSI before the turn of the century.

Introduction to TCP/IP

TCP, itself, is a transport layer protocol (ISO layer 4) providing a connection-oriented service between host processors. It provides a reliable end-to-end service with provisions for flow control and multiplexing of connections. TCP also provides mechanisms for detecting duplicate, lost, or out-of-sequence packets.

The Internet suite also specifies an optional connectionless-mode transport protocol, User Datagram Protocol (UDP). UDP is used for transaction-based applications where efficiency and low overhead are more important than reliability.

TCP and UDP use the network services of IP (ISO layer 3) as a datagram service. IP is a connectionless-mode network layer service. It is used to route messages between networks and performs any message segmentation and reassembly required. Segmentation and reassembly may be needed if a message must be routed through a network with different packet size restrictions than the source and/or destination network.

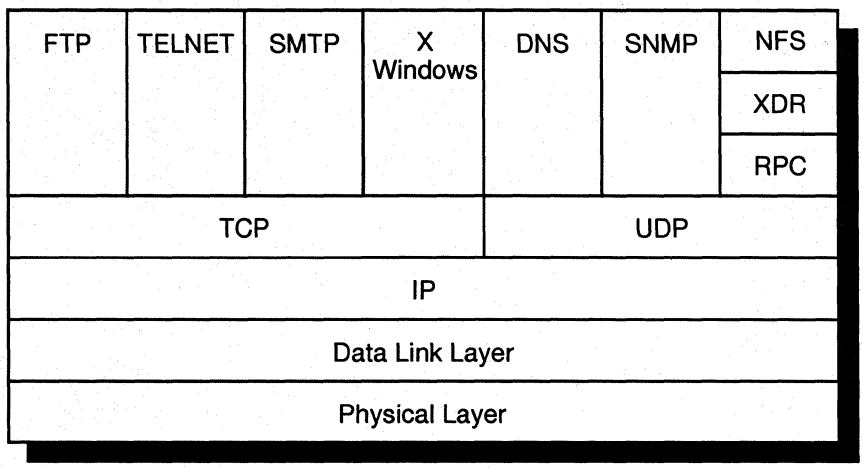
Application protocols are built upon the services of TCP/IP and UDP/IP (see Figure 1). These application protocols form a client/server network computing environment.

Client/Server Computing

Each application layer protocol discussed in this report follows the client/server computing model. This model is very simple. In the client/server model, a client application requests services of a remote service application over a network. Correspondingly, the server performs the requested service of the client and responds with the results according to a well-defined protocol. The client typically executes on the local computer, and the server executes on a remote computer. Except for network delays, a user may be unaware of the interactions between client and server processes in such a distributed computing environment.

—By L. Michael Sabo
U S West Advanced Communications Services

Figure 1.
The Internet Protocol Suite



Application Services

TCP/IP is a suite of communications protocols originally developed under the funding of the United States government in the mid-1970s. Often referred to as the Internet Suite, TCP/IP protocols are open protocols that support interenterprise communications among heterogeneous hosts. As such, each of the application protocols discussed in this report will operate on a mainframe, UNIX workstation, PC, or Macintosh. TCP/IP view each of these hosts as peers.

Most users view the Internet protocols in terms of the services they provide at the application layer. The Internet protocol suite's services are responsible for propelling TCP/IP to its status in enterprise internetworking.

The Internet protocol suite is rich with application services (see Table 1). These services span the basic functions of virtual terminal support, electronic mail, file transfers, and name service. The Internet suite also includes advanced services, such as a distributed windowing system, a network file system, and a network management protocol. This report examines each of these services and describes how a user interacts with them.

Basic Internet Suite Services

The basic Internet suite services are the File Transfer Protocol (FTP), the virtual terminal protocol (Telnet), the Simple Mail Transfer Protocol (SMTP), and the Domain Name Service (DNS). Hosts supporting TCP/IP usually provide at least these basic services.

File Transfer Protocol (FTP)

FTP is a protocol used for the bulk transfer of data between host processors over a TCP datastream. FTP provides a file utility for performing remote file operations such as bidirectional file transfers, deleting files, renaming files, and displaying file directories. Users perform these operations during interactive sessions. To support this on-line interaction between a user and a remote host, FTP provides two connections, as shown in Figure 2. FTP commands and status information exchanges use the control connection. File transfers use a second connection. Common FTP commands and their purpose are listed as follows; for a comprehensive list, refer to your hosts' user manuals:

- ascii—Data will be transferred in ASCII format. This is the default.
- binary—Data will be transferred as binary data.
- bye—Terminate the FTP session with the remote server and exit ftp.
- cd—Change the working directory on the remote host.
- delete—Delete a file on the remote host.
- dir—List the directory contents of the current remote machine directory.
- get—Retrieve a specified file from the remote host and store it on the local host.
- hash—display a hash sign, "#", after each data block is transferred.
- put—Store a specified file from the local host to the remote host.

As does every Internet suite application protocol, FTP follows the client/server model. An FTP client sends commands and interacts with a user or a user program, and the FTP server portion receives and responds to commands. Typically, a host will provide implementations for both an FTP client and an FTP server.

FTP is non-host specific. FTP implementations exist for UNIX systems, mainframes, Macintoshes, and PCs. Thus, FTP allows a user to exchange files between dissimilar hosts, such as a Macintosh and a UNIX workstation, without regard to the particulars of the hosts' file systems. Any system supporting TCP/IP is likely to support FTP. FTP relies on TCP at the transport layer to provide a reliable data path between peer hosts. Thus, users are ensured that files transferred over a network will arrive at their destination error free.

Virtual Terminal Protocol (TELNET)

TELNET is an interactive remote access terminal protocol. It allows a user to log in to a remote computer system, over a network, as though the terminal was attached directly to the remote host. TELNET is based on the client/server model, but DOS-based PCs, Macintoshes, and terminal servers typically support only client TELNET. Thus, users on those devices can initiate an interactive terminal session to a server TELNET but cannot accept a session request from a TELNET client. UNIX hosts and most other multitasking operating systems generally support both client and server TELNET.

TELNET uses a TCP connection to transmit data and TELNET control information. TELNET is based on the concept of a Network Virtual Terminal (NVT) and negotiated options to extend the basic capabilities of NVT.

Network Virtual Terminal (NVT)

When a TELNET connection is initiated, both ends of the connection begin with an NVT profile. NVT is a least common denominator terminal profile used throughout the network, eliminating the need for every host to support the entire range of terminal possibilities (see Figure 3). The NVT is essentially a bidirectional communications facility that uses the seven-bit ASCII character set but encodes them into eight-bit bytes.

Principle of Negotiated Options

TELNET allows services to be supported beyond those defined for the NVT. There are many terminal options not specifically defined within the TELNET Protocol but which can be supported by TELNET. TELNET uses a do/do not, will/will not negotiation process. This scheme allows a client and server to use different conventions for the TELNET session. Specifically, a sender will respond that it will or will not perform some option. Optionally, a sender may request that its peer initiate or not initiate an option. This interaction permits an option request to be refused without knowledge about the requested option. Accepted options take effect immediately. Rejected options require the parameter to remain as defined for an NVT.

Simple Mail Transport Protocol (SMTP)

SMTP is the Internet standard for electronic mail distribution. It is a text-oriented protocol that uses TCP's underlying services to reliably transfer or relay electronic mail. SMTP supports efficient message delivery: if a message has multiple recipients at a particular destination host, SMTP will only send one message to the host listing each recipient. The receiving SMTP will provide each recipient with a copy of the message.

Each mail message contains a header and a body. The header contains elements such as Date, Subject, To, Cc, and From. The client SMTP routine prompts the user for each of these parameters. The body of the message is typically free-form ASCII text. For example, to send a message to user smith at host hosta.xyz.com, a UNIX user Jones on host myhost.xyz.com would perform the following:

% mail smith@hosta.xyz.com <cr>
Subject: Thanks for the information <cr>
Mr. Smith, thanks for sending me the information so promptly.<cr>
-Jones <cr>
<control d>
Cc: <cr>
%

Obtaining RFCs on the Internet

RFCs are available through FTP from Internet host NIC.DDN.MIL.

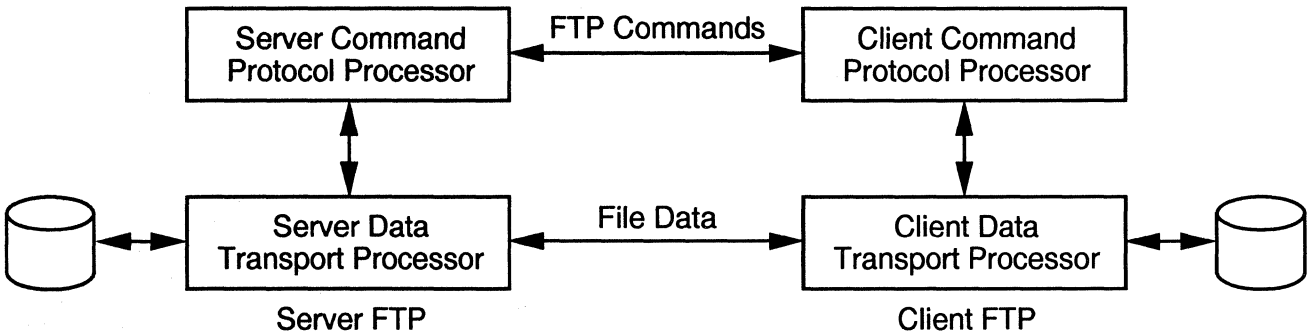
Log in using username "anonymous" and password "guest". Once logged on, type in "get RFC:RFCnnnn.txt", where nnnn is the RFC number. RFCs can also be obtained through electronic mail. Send a message to SERVICE@NIC.DDN.MIL and place the RFC number in the subject field.

To obtain a current index of all RFCs, type "FTP to NIC.DDN.MIL" with anonymous, guest login. Once the session is established, type "dir RFC:RFC-INDEX." A document name will be returned, such as "RFC-INDEX.TXT.nnnn" to fetch the index for review on your local host. To log out of the FTP session, type "quit".

On UNIX systems, a control d or a period on a line by itself is used to indicate the message is complete. Notice the user did not copy—"Cc"—any other recipients. If the user had placed other mailboxes in the "Cc" field, each recipient would have also received the message. To view the actual SMTP protocol interaction, Jones could have used the -v—verbose—mail command option. An example follows:

% mail -v smith@hosta.xyz.com <cr>
Subject: Thanks for the information<cr>
Mr. Smith, thanks for sending me the information so promptly.<cr>
-Jones <cr>
<control d>
Cc:<cr>
% smith@hosta.xyz.com... Connecting to hosta.xyz.com
Trying... connected.
220 HOSTA.XYZ.COM Simple Mail Transfer Server

Figure 2.
FTP Connections



FTP uses two connections, one for commands and replies, and another to support bulk file transfer.

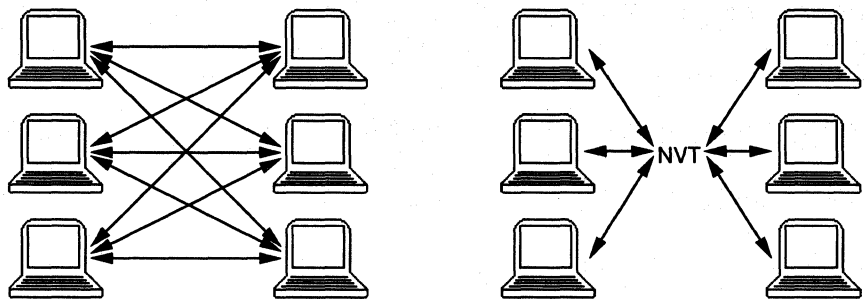


Figure 3.
NVT for Telnet

Telnet's use of network virtual terminal (NVT) reduces the problem of supporting every possible terminal type to only converting between a specific terminal type and NVT.

```
ready
>>> HELO myhost.xyz.com
250 OK
>>> MAIL From:<jones@myhost.xyz.com>
250 OK
>>> RCPT To:<smith@hosta.xyz.com>
250 OK
>>> DATA
354 Input
>>> .
250 OK
>>> QUIT
221 HOSTA.XYZ.COM Simple Mail Transfer server terminated.
```

In the above example, lines beginning with >>> are generated by the sender, and lines beginning with a number are responses from the message recipient. The actual message is sent after the 354 Input response.

Domain Name Service (DNS)

Domain Name Service is the naming protocol used in the Internet suite. DNS provides domain name-to-IP address translation. Names, rather than IP addresses, are much easier for individuals to remember.

DNS allows the administration of domain names to be decentralized. Through DNS, hosts are no longer required to maintain host name-to-IP address configuration tables for every host on the network. Partitioning the domain name into multiple name fields allows the decentralization of name administration. The DDN Network Information Center (NIC) administers the top-level portions of the domain name. A user organization is free to append names in front of the top-level name to define subdomains or specific hosts within the organization. As an example, company XYZ has registered the domain name xyz.com with the NIC. "XYZ" is the organization's name, and "COM" signifies the organization is a commercial venture. Company XYZ may then assign and locally administer the name hosta.xyz.com for one of its hosts.

Currently, the NIC has specified six top-level domain names. They are the following:

- COM—Commercial Organizations
- EDU—Educational Organizations
- GOV—Government Agencies
- MIL—MILNET Hosts
- NET—Networking Organizations
- ORG—Not for Profit Organizations

DNS terms the client portion of the protocol implementation the *resolver*, and the server is called the *name server*. The resolver queries the name server to translate domain names to IP addresses. Resolvers typically cache the results of previous name queries, reducing network traffic and name server interaction.

Advanced Internet Suite Services

The Advanced Internet suite services include the X Window System (X) and the Network File System (NFS). These application protocols are typically implemented on high-performance workstations and used in a LAN environment.

X Window System

The X Window System, typically called simply "X," is a platform-independent, bit-mapped graphical user interface (GUI) that

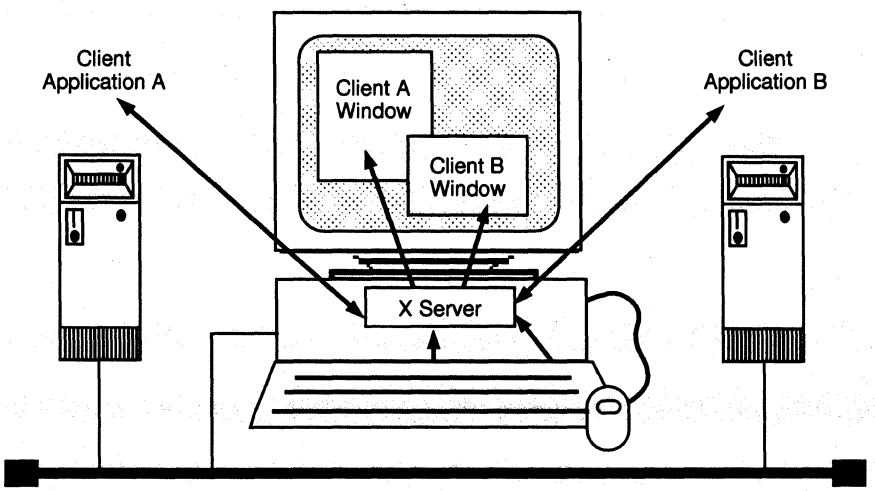


Figure 4.
X Window System

An X user can connect to several X clients simultaneously. The X client can be local to the X terminal or execute remotely over a network.

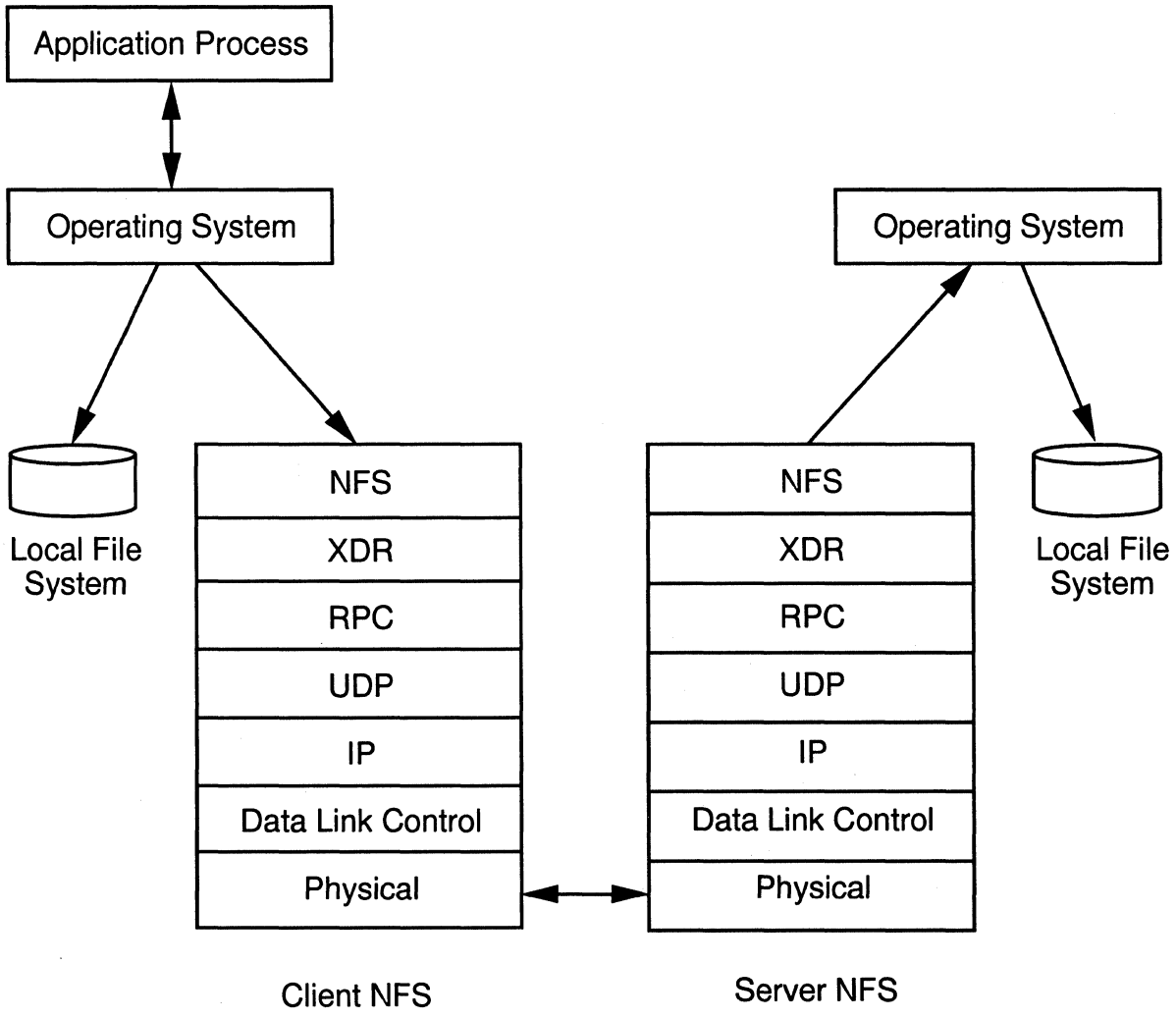
Table 1. Internet Suite Application Layer Protocols

Name	Acronym	Description	RFC Reference	Well-Known Port
Active Users Protocol	USERS	Sends a list of active host users	RFC 866	
Authentication Service	AUTH	Provides TCP authentication mechanism	RFC 931	113
Bootstrap Protocol	BOOTP	Used for booting diskless workstations	RFC 951	67, 68
Character Generator Protocol	CHARGEN	Used for debugging, generates ASCII messages	RFC 864	19
Daytime Protocol	DAYTIME	Provide day and time Information	RFC 867	13
DCNET Time Server Protocol	CLOCK	Provides a mechanism for synchronizing clocks	RFC 778	
Discard Protocol	DISCARD	Used for debugging, discards all messages	RFC 863	9
Domain Name Protocol	DOMAIN	Defines the Domain Name Service	RFCs 881,	53
Echo Protocol	ECHO	Used for debugging, echoes all messages	RFC 862	7
File Transfer Protocol	FTP	Bulk file transfer protocol	RFC 959	20, 21
Finger Protocol	FINGER	Sends information on specified user	RFC 742	79
Graphics Protocol	GRAPHICS	Used for exchanging vector graphics	RFC 493	
Internet Message Protocol	MPM	Provides multimedia mail transfers	RFC 759	46
ISO Electronic Mail	X400	OSI electronic mail standard	RFC 1148	103, 104
Loader Debugger Protocol	LDP	Used for loading, dumping, and debugging hosts	RFC 909	
Line Printer Daemon Protocol	LPR	Print Server protocol	RFC 1179	515
Network File System	SUNRPC	Network File System; Remote Procedure Call	RFC 1094	111
Network News Transfer Protocol	NNTP	Protocol posting and distributing news articles	RFC 977	119

Table 1. Internet Suite Application Layer Protocols (Continued)

Name	Acronym	Description	RFC Reference	Well-Known Port
Network Time Protocol	NTP	Provides a means of synchronizing network clocks	RFC 958	123
Password Generation	PWDGEN	Generates passwords	RFC 972	129
Post Office Protocol	POP3	Allows PC users to access mail from a mail server	RFC 1081,	110
Quote of the Day Protocol	QUOTD	Sends an ASCII message	RFC 865	17
Remote Job Entry	RJE	Used to submit and retrieve batch jobs	RFC 407	77
Remote Telnet Service	RTELNET	Supports special access to user Telnet	RFC 818	
Resource Location Protocol	RLP	Automatically locates a resource	RFC 887	39
Simple File Transfer Protocol	SFTP	Bulk file transfer protocol	RFC 913	115
Simple Mail Transfer Protocol	SMTP	Electronic mail transfer protocol	RFC 821	25
Simple Network Management Protocol	SNMP	Supports the exchange of management information	RFC 1157	161, 162
Statistics Server	STATSRV	Used for sending	RFC 996	95
SUPDUP Protocol	SUPDUP	Telnet-like protocols for display terminals	RFC 734	
TELNET Protocol	TELNET	Defines a remote terminal protocol	RFC 854	23
Time Server Protocol	TIME	Provides time in seconds	RFC 868	37
Trivial File Transfer Protocol	TFTP	Bulk file transfer protocol without access control or parameters	RFC 783	69
Whois Protocol	NICNAME	Sends information on specified user	RFC 954	43
X Window System	X	Network Windowing Protocol	RFC 1198	

Figure 5.
Network File System



The client Network File System (NFS) provides access to the remote file system over a network connection. The remote file access is transparent to the application process.

runs over TCP. X can run over any reliable network, including OSI, but TCP is the most popular implementation.

An X-based application runs in a client/server environment (see Figure 4). In X, the client and server relationship may appear reversed from the traditional interpretation. The X server typically runs on the local workstation, and the X client runs on the remote host. The X client is responsible for application management tasks, and the X server is responsible for providing display services on the graphics terminal.

X Servers and X clients can communicate over a network or through an interprocess control (IPC) connection. Because it can operate on a network, X is considered a network-based windowing system.

The X protocol supports requests and responses between X clients and X servers. It exchanges information necessary to operate the graphical windowing system over a network connection.

X Protocol Internals

Efficiency is an important attribute of the X protocol. It is a necessary requirement when running an application such as a distributed windowing system over a network. Efficiencies have been attained in two primary ways. First, not all X client requests require a corresponding reply from the X server; second, the X Protocol Data Units (PDUs) each have a length that is multiples of four octets. The latter allows the X PDUs to be quickly processed on host processors based on 16- or 32-bit architectures because no alignment is necessary.

The X protocol specifies four PDU types: requests, replies, events, and errors. An X client sends requests to the X server. The X server sends replies, events, and errors to the X client.

X Request PDU

An X request PDU instructs an X server to perform a specific action and may or may not require a reply on behalf of the X server. An X client may, for example, request the X server to create a window, allocate a color, draw a graphic, etc. An X request PDU length must be a multiple of four octets.

X Reply PDU

As mentioned, not all X requests require replies. When the X client requests information, the X server will generate an X reply PDU. An X reply PDU would be sent in response to the "allocate color" request, for example. An X reply PDU's length must be a multiple of four bytes and be a minimum of 32 octets.

X Event PDU

An X event PDU contains information about either a device action or a side effect of a prior request. X clients receive most of their information through X event PDUs. An X server sends an X event expose PDU when a window display action has completed. An X event PDU's length must be a multiple of four bytes and be a minimum of 32 octets.

X Error PDU

An X error PDU is very similar to an X event PDU, with the exception of how the X client handles the PDU. Upon receipt, the X client transfers the error to a special error-handling routine. This facilitates a quick response to error conditions. An X error PDU's length must be a multiple of four bytes and be a minimum of 32 octets.

X appears to have a great future in network computing. Users are demanding windows-based applications because of their ease of use and short learning curves. Software developers find the platform independence of X very appealing. MIS managers realize they can preserve their investment in hardware, such as mainframes, by migrating many applications to the X environment. Each of these forces is forming the foundation for a very large market for X applications.

Network File System (NFS)

NFS provides the services of what is typically called a "network operating system." NFS is a protocol allowing multiple hosts to access each other's file systems as though they were local (see Figure 5). NFS insulates users and applications from the fact that information is being accessed on another over a network connection. Using NFS, a PC can store information on a UNIX host, for example, as easily as storing information to a local hard drive. Additionally, the PC can actually run a program from software stored on the UNIX host.

NFS is host processor and operating system independent. A Remote Procedure Call (RPC) facility accomplishes this. An application can execute RPC functions and receive the results to a local data structure just as if a local procedure had been called. The application is unaware that it is processing in a distributed environment. RPC uses eXternal Data Representation (XDR) as a presentation layer service. XDR is responsible for placing information exchanged between systems in a machine-independent form so that any type of CPU can run NFS.

The demand for NFS is accelerating as network administrators discover the benefits of providing users with a standards-based approach to network operating system capabilities. Client NFS is available on a wide variety of hosts including PCs and Macintoshes, which should spur the demand for this protocol.

Internet Suite Network Management

Managing a TCP/IP-based Internet can be a complex undertaking. A TCP/IP network typically contains multivendor equipment implemented over multiple network media in a network computing environment. The Internet community struggled for many years without protocols and tools to adequately manage these networks. In response to network administrators' demands, a working group was formed to develop a network management protocol for managing TCP/IP-based Internets. The result of this effort was the Simple Network Management Protocol (SNMP).

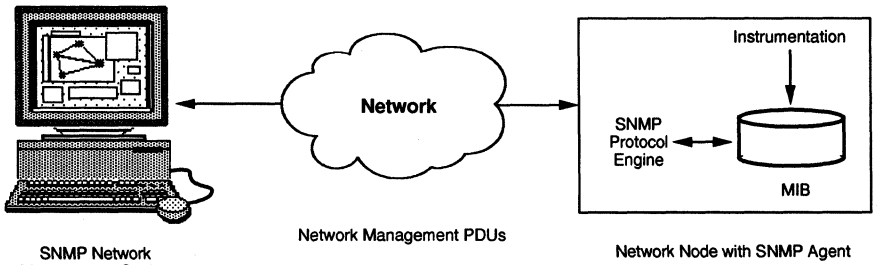
SNMP follows a manager/agent model, similar to the client/server model (see Figure 6). An agent operates on a network device, such as a router, bridge, terminal server, or even host processor. The agent responds to queries directed to it from an SNMP manager. SNMP agent software is responsible for providing the instrumentation required to gather management data. The management data is then stored into a logical database called the Management Information Base (MIB).

The SNMP manager software runs on a network manager station and can perform three operations concerning agent interaction. The SNMP manager can get, get-next, or set specific variables in the agent's MIB. Correspondingly, an agent will provide a response to the manager requests. Additionally, an agent can, on certain events, issue an unsolicited trap message to the SNMP manager.

SNMP's authors designed the protocol with three specific goals. Briefly, SNMP was to be developed such that an agent could be deployed with minimal code, have highly extensible monitoring capabilities, and not depend on the underlying transport protocol. Each of these goals was met: many SNMP agents are under 10KB; the MIB provides private enterprise space for user extensions; and SNMP implementations exist over many transport mechanisms, including raw Ethernet.

Figure 6.
SNMP Management Station

An SNMP Management Station obtains management information from an SNMP agent. The SNMP agent obtains management information through its instrumentation. The MIB is a logical database of management information.



Once the SNMP specification was complete, the standard was quickly embraced by vendors. Virtually every TCP/IP networking device manufactured today supports SNMP management.

Conclusion

The Internet suite of protocols is robust and rich with application services. Most users view the Internet protocols in terms of these application services. The demand for these services is fueling the demand for TCP/IP-based enterprise Internets, which is likely to continue well into the mid-1990s. ■

L. Michael Sabo is a Data Applications Consultant with U S West Advanced Communications Services in Denver, CO. Mr. Sabo designs LAN internetworking solutions for clients using frame-relay, SMDS, high-speed private line, and Transportation LAN Service (LTS) technologies. This includes porting TCP/IP to the emerging ANSI High-Performance Parallel Interface (HIPPI) Gigabit/sec. LAN standard and developing object-oriented and SNMP-based network management architectures. Mr. Sabo has written many technical articles and is a member of Datapro's Board of Technical Advisors for Broadband Communications Services. He participates as a member of the Internet Society, and has been very active in the Internet for eight years. In addition, he is a member of the ATM Forum, Frame Relay Forum, and the SMDS Interest Group.

Mr. Sabo holds a master's degree in Computer Information Management from the University of Denver and a bachelors of science degree in Computer Science from Wright State University.

Internet Suite Application Protocols

In this report:

Datapro Summary

The Internet suite of protocols is robust and rich with application services. Most users view the Internet protocols in terms of these application services, which usually run over TCP/IP middle layer protocols. Based on the client/server model, Internet services span the basic functions of virtual terminal support, electronic mail, file transfers, and name service. Advanced services include a distributed windowing system, a network file system, and a network management protocol. The demand for these services is fueling the demand for TCP/IP-based enterprise internets, which is likely to continue into the mid-1990s.

The Internet protocol suite continues in great demand in the standards-based networking marketplace. This demand should accelerate into the mid-1990s. Once the protocol suite of choice for the U.S. military and a handful of universities, Transmission Control Protocol/Internet Protocol (TCP/IP) is now gaining popularity within the business community. Many businesses have become disillusioned with the excruciatingly slow pace of OSI deployment. Consequently, businesses are turning to TCP/IP to form the building blocks for their enterprise internetworks.

As more organizations embrace TCP/IP, the timetable for OSI ubiquity becomes correspondingly shifted to the right. It is unlikely that an organization building a TCP/IP-based enterprise network today would migrate to OSI before the turn of the century.

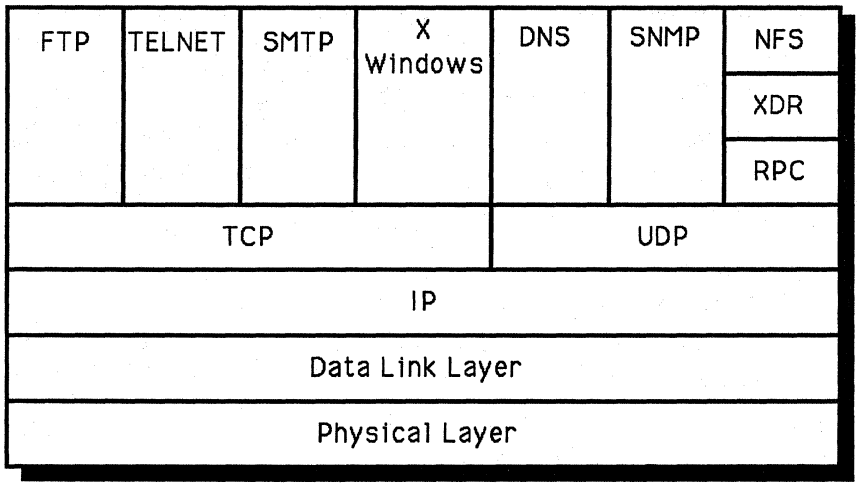
—By L. Michael Sabo
Communications Architect
SSDS, Inc.

Introduction to TCP/IP

TCP is a transport layer protocol (ISO layer 4) providing a connection-oriented service between host processors. It provides a reliable end-to-end service with provisions for flow control and multiplexing of connections. TCP also provides mechanisms for detecting duplicate, lost, or out-of-sequence packets.

The Internet suite also specifies an optional connectionless-mode transport protocol, User Datagram Protocol (UDP). UDP is used for transaction-based applications where efficiency and low overhead are more important than reliability.

Figure 1.
The Internet Protocol Suite



TCP and UDP use the network services of IP (ISO layer 3) as a datagram service. IP is a connectionless-mode network layer service. It is used to route messages between networks and performs any message segmentation and reassembly required. Segmentation and reassembly may be needed if a message must be routed through a network with different packet size restrictions than the source and/or destination network.

Application protocols are built upon the services of TCP/IP and UDP/IP (see Figure 1). These application protocols form a client/server network computing environment.

Client/Server Computing

Each application layer protocol discussed in this report follows the client/server computing model, which is very simple. In the client/server model, a client application requests services of a remote service application over a network. Correspondingly, the server performs the requested service of the client and responds with the results according to a well-defined protocol. The client typically executes on the local computer, and the server executes on a remote computer. Except for network delays, a user is unaware of the interactions between client and server processes in such a distributed computing environment.

Application Services

Once a TCP/IP-based communications infrastructure was in place, research turned quickly toward implementing distributed applications using the reliable services of TCP. Most users view the Internet protocols in terms of the services they provide at the application layer. The Internet protocol suite's services are responsible for propelling TCP/IP to its status in enterprise internetworking.

The Internet protocol suite is rich with application services (see Table 1). These services span the basic functions of virtual terminal support, electronic mail, file transfers, and name service. The Internet suite also includes advanced services, such as a distributed windowing system, a network file system, and a network management protocol. This report examines each of these services and describes how a user interacts with them.

Basic Internet Suite Services

The basic Internet suite services are the File Transfer Protocol (FTP), the virtual terminal protocol (Telnet), the Simple Mail Transfer Protocol (SMTP), and the Domain Name Service (DNS). Hosts supporting TCP/IP are very likely to provide at least these basic services.

File Transfer Protocol (FTP)

FTP is a protocol used for the bulk transfer of data between host processors over a TCP datastream. FTP provides a file utility for performing remote file operations such as bidirectional file transfers, deleting files, renaming files, and displaying file directories. Users perform these operations during interactive sessions. To support this on-line interaction between a user and a remote host, FTP provides two connections, as shown in Figure 2. FTP commands and status information exchanges use the control connection. File transfers use a second connection. Common FTP commands and their purpose are listed as follows; for a comprehensive list, refer to your hosts' user manuals:

- ascii—Data will be transferred in ASCII format.
- binary—Data will be transferred as binary data.
- bye—Terminate the FTP session with the remote server and exit ftp.
- cd—Change the working directory on the remote host.
- delete—Delete a file on the remote host.
- dir—List the directory contents of the current remote machine directory.
- get—Retrieve a specified file from the remote host and store it on the local host.
- hash—display a hash sign, "#", after each data block is transferred.
- put—Store a specified file from the local host to the remote host.

As does every Internet suite application protocol, FTP follows the client/server model. An FTP client sends commands and interacts with a user or a user program, and the FTP server portion receives and responds to commands. Typically, a host will provide implementations for both an FTP client and an FTP server.

FTP is non-host specific. FTP implementations exist for UNIX systems, mainframes, Macintoshes, and PCs. Any system supporting TCP/IP is likely to support FTP. FTP relies on TCP at the transport layer to provide a reliable data path between peer hosts. Thus, users are assured that files transferred over a network will arrive at their destination error free.

Obtaining RFCs on the Internet

RFCs are available through FTP from Internet host NIC.DDN.MIL.

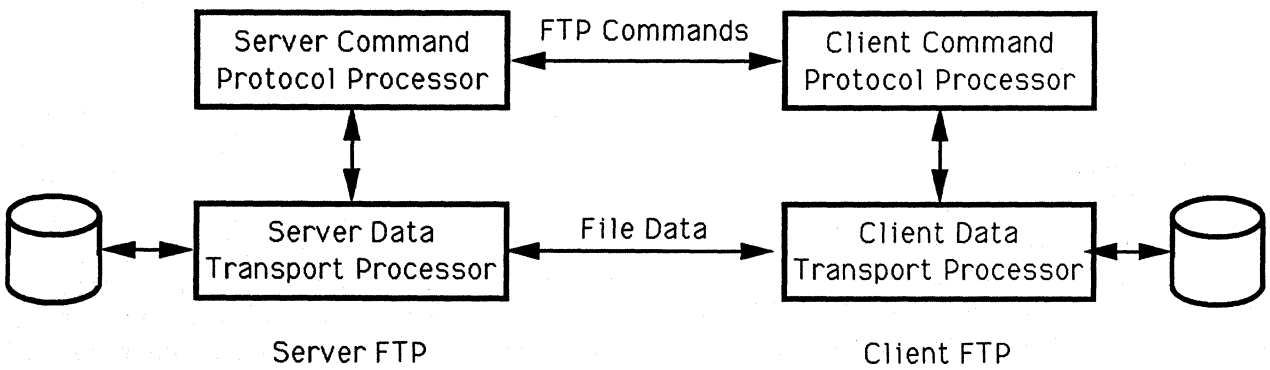
Log in using username "anonymous" and password "guest." Once logged on, type in "get RFC:RFCnnnn.txt", where nnnn is the RFC number. RFCs can also be obtained thorugh electronic mail. Send a message to SERVICE@NIC.DDN.MIL and place the RFC number in the subject field.

To obtain a current index of all RFCs, FTP to NIC.DDN.MIL with anonymous, guest login. Once the session is established, type "dir RFC:RFC-INDEX." A document name will be returned, such as RFC-INDEX.TXT.nnnn" to fetch the index for review on your local host. To log out of the FTP session, type "quit."

Virtual Terminal Protocol (TELNET)

TELNET is an interactive remote access terminal protocol. It allows a user to log in to a remote computer system, over a network, as though the terminal was attached directly to the remote host. TELNET is based on the client/server model, but DOS-based PCs, Macintoshs, and terminal servers typically support only client TELNET. Thus, users on those devices can initiate an interactive terminal session with a server TELNET but cannot accept a session request from a TELNET client.

Figure 2.
FTP Connections



FTP uses two connections, one for commands and replies, and another to support bulk file transfer.

Table 1. Internet Suite Application Layer Protocols

Name	Acronym	Description	RFC Reference	Well-Known Port
Active Users Protocol	USERS	Sends a list of active host users	RFC 866	
Authentication Service	AUTH	Provides TCP authentication mechanism	RFC 931	113
Bootstrap Protocol	BOOTP	Used for booting diskless workstations	RFC 951	67, 68
Character Generator Protocol	CHARGEN	Used for debugging, generates ASCII messages	RFC 864	19
Daytime Protocol	DAYTIME	Provide day and time Information	RFC 867	13
DCNET Time Server Protocol	CLOCK	Provides a mechanism for synchronizing clocks	RFC 778	
Discard Protocol	DISCARD	Used for debugging, discards all messages	RFC 863	9
Domain Name Protocol	DOMAIN	Defines the Domain Name Service	RFCs 881, 882, 883	53
Echo Protocol	ECHO	Used for debugging, echoes all messages	RFC 862	7
File Transfer Protocol	FTP	Bulk file transfer protocol	RFC 959	20, 21
Finger Protocol	FINGER	Sends information on specified user	RFC 742	79
Graphics Protocol	GRAPHICS	Used for exchanging vector graphics	RFC 493	
Internet Message Protocol	MPM	Provides multimedia mail transfers	RFC 759	46
ISO Electronic Mail	X400	OSI electronic mail standard	RFC 1148	103, 104
Loader Debugger Protocol	LDP	Used for loading, dumping, and debugging hosts	RFC 909	
Line Printer Daemon Protocol	LPR	Print Server protocol	RFC 1179	515
Network File System	NFS	Network File System	RFC 1094	2049
Network News Transfer Protocol	NNTP	Protocol posting and distributing news articles	RFC 977	119

UNIX hosts and most other multitasking operating systems generally support both client and server TELNET.

TELNET uses a TCP connection to transmit data and TELNET control information. TELNET is based on the concept of a Network Virtual Terminal (NVT) and negotiated options to extend the basic capabilities of NVT.

Network Virtual Terminal (NVT)

When a TELNET connection is initiated, both ends of the connection begin with an NVT profile. NVT is a least common denominator terminal profile used throughout the network, eliminating the need for every host to support the entire range of terminal possibilities (see Figure 3). The NVT is essentially a bidirectional communications facility that uses the seven-bit ASCII character set but encodes them into eight-bit bytes.

Principle of Negotiated Options

TELNET allows services to be supported beyond those defined for the NVT. There are many terminal options not specifically defined within the TELNET Protocol but which can be supported by TELNET. TELNET uses a do/don't, will/won't negotiation process. This scheme allows a client and server to use different conventions for the TELNET session. Specifically, a sender will respond that it will or will not perform some option.

Table 1. Internet Suite Application Layer Protocols (Continued)

Name	Acronym	Description	RFC Reference	Well-Known Port
Network Time Protocol	NTP	Provides a means of synchronizing network clocks	RFC 958	123
Password Generation	PWDGEN	Generates passwords	RFC 972	129
Post Office Protocol	POP3	Allows PC users to access mail from a mail server	RFC 1081, 1082	110
Quote of the Day Protocol	QUOTE	Sends an ASCII message	RFC 865	
Remote Job Entry	RJE	Used to submit and retrieve batch jobs	RFC 407	
Remote Telnet Service	RTELNET	Supports special access to user Telnet	RFC 818	
Resource Location Protocol	RLP	Automatically locates a resource	RFC 887	39
Simple File Transfer Protocol	SFTP	Bulk file transfer protocol	RFC 913	115
Simple Mail Transfer Protocol	SMTP	Electronic mail transfer protocol	RFC 821	25
Simple Network Management Protocol	SNMP	Supports the exchange of management information	RFC 1157	161, 162
Statistics Server	STATSRV	Used for sending gateway statistics	RFC 996	95
SUPDUP Protocol	SUPDUP	Telnet-like protocols for display terminals	RFC 734	
Telnet Protocol	TELNET	Defines a remote terminal protocol	RFC 854	23
Time Server Protocol	TIME	Provides time in seconds	RFC 868	37
Trivial File Transfer Protocol	TFTP	Bulk file transfer protocol without access control or parameters	RFC 783	69
Whois Protocol	NICNAME	Sends information on specified user	RFC 954	43
X Window System	X	Network Windowing Protocol	RFC 1198	

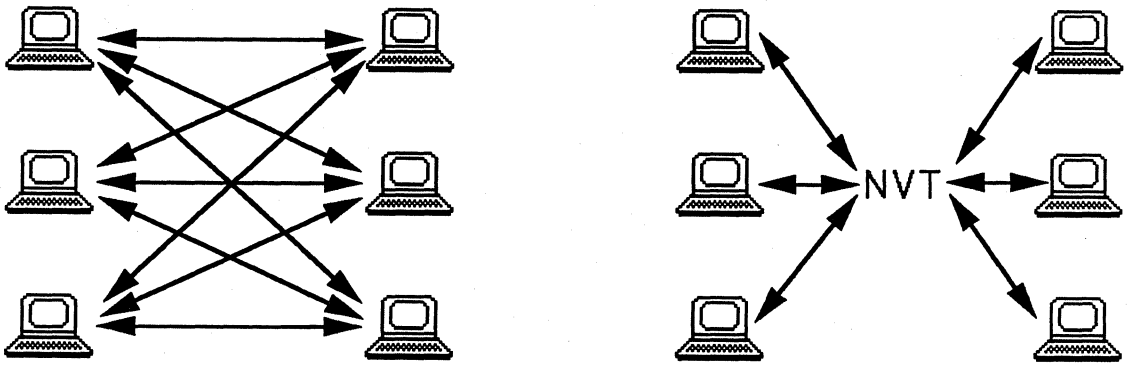
Optionally, a sender may request that its peer initiate or not initiate an option. This interaction permits an option request to be refused without knowledge about the requested option. Accepted options take effect immediately. Rejected options require the parameter to remain as defined for an NVT.

Simple Mail Transport Protocol (SMTP)

SMTP is the Internet standard for electronic mail distribution. It is a text-oriented protocol that uses TCP's underlying services to reliably transfer or relay electronic mail. SMTP supports efficient message delivery: if a message has multiple recipients at a particular destination host, SMTP will only send one message to the host listing each recipient. The receiving SMTP will provide each recipient with a copy of the message.

Each mail message contains a header and a body. The header contains elements such as Date, Subject, To, Cc, and From. The client SMTP routine prompts the user for each of these parameters. The body of the message is typically free-form ASCII text. For example, to send a message to user smith at host hosta.xyz.com, a UNIX user Jones on host myhost.xyz.com would perform the following:

Figure 3.
NVT for Telnet



Telnet's use of network virtual terminal (NVT) reduces the problem of supporting every possible terminal type to only converting between a specific terminal type and NVT.

% mail smith@hosta.xyz.com <cr>
Subject: Thanks for the information <cr>
Mr. Smith, thanks for sending me the information so promptly. <cr> -Jones <cr> <control d>
Cc: <cr> %

On UNIX systems, a control d is used to indicate the message is complete. Notice the user did not copy—"Cc"—any other recipients. If the user had placed other mailboxes in the "Cc" field, each recipient would have also received the message. To view the actual SMTP protocol interaction, Jones could have used the -v—verbose—mail command option. An example follows:

% mail -v smith@hosta.xyz.com <cr>
Subject: Thanks for the information <cr>
Mr. Smith, thanks for sending me the information so promptly. <cr> -Jones <cr> <control d>
Cc: <cr> % smith@hosta.xyz.com... Connecting to hosta.xyz.com
Trying... connected.
220 HOSTA.XYZ.COM Simple Mail Transfer Server ready >>> HELO myhost.xyz.com
250 OK >>> MAIL
From:<jones@myhost.xyz.com>
250 OK >>> RCPT
To:<smith@hosta.xyz.com>
250 OK >>> DATA
354 Input >>> .
250 OK >>> QUIT
221 HOSTA.XYZ.COM Simple Mail Transfer server terminated. %

In the preceding example, lines beginning with >>> are generated by the sender, and lines beginning with a number are responses from the message recipient. The actual message is sent after the 354 Input response.

Domain Name Service

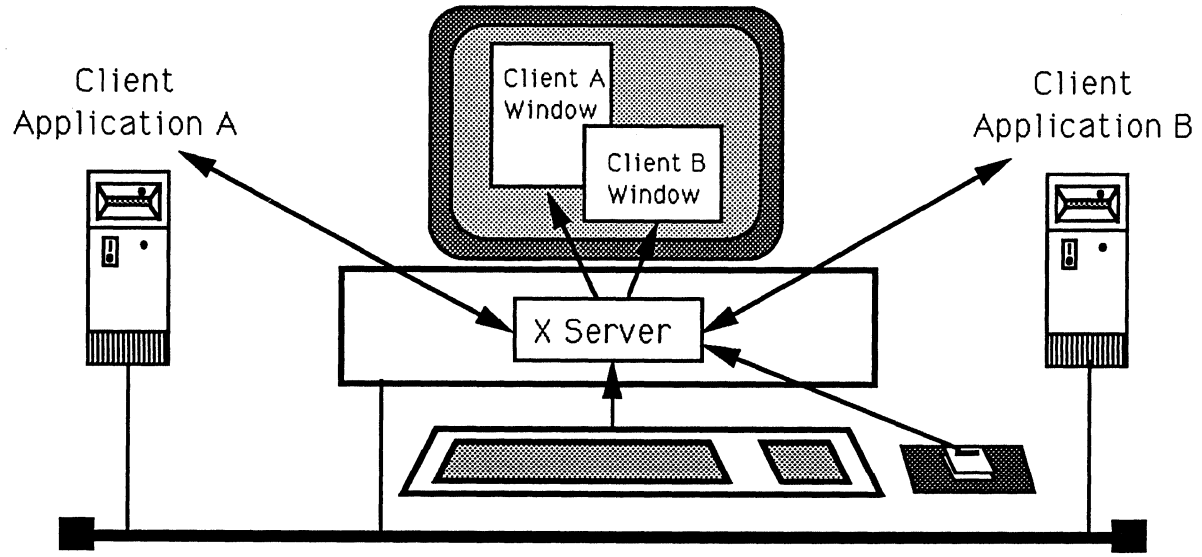
The Domain Name Service (DNS) is the naming protocol used in the Internet suite. DNS provides domain name-to-IP address translation. Names, rather than IP addresses, are much easier for individuals to remember.

DNS allows the administration of domain names to be decentralized. Through DNS, hosts are no longer required to maintain host name-to-IP address configuration tables for every host on the network. Partitioning the domain name into multiple name fields allows the decentralization of name administration. The DDN Network Information Center (NIC) administers the top-level portions of the domain name. A user organization is free to append names in front of the top-level name to define subdomains or specific hosts within the organization. As an example, company XYZ has registered the domain name xyz.com with the NIC. "XYZ" is the organization's name, and "COM" signifies the organization is a commercial venture. Company XYZ may then assign and locally administer the name hosta.xyz.com for one of its hosts.

Currently, the NIC has specified six top-level domain names. They are the following:

- COM—Commercial Organizations
- EDU—Educational Organizations
- GOV—Government Agencies

Figure 4.
X Window System



An X user can connect to several X clients simultaneously. The X client can be local to the X terminal or execute remotely over a network.

- MIL—MILNET Hosts
- NET—Networking Organizations
- ORG—Not for Profit Organizations

DNS terms the client portion of the protocol implementation the *resolver*, and the server is called the *name server*. The resolver queries the name server to translate domain names to IP addresses. Resolvers typically cache the results of previous name queries, reducing network traffic and name server interaction.

Advanced Internet Suite Services

The Advanced Internet suite services include the X Window System (X) and the Network File System (NFS). These application protocols are typically implemented on high-performance workstations and used in a LAN environment.

X Window System

The X Window System, typically called simply "X," is a platform-independent, bit-mapped graphical user interface (GUI) that runs over TCP. X can run over any reliable network, including OSI, but TCP is the most popular implementation.

An X application runs in a client/server environment (see Figure 4). In X, the client and server relationship may appear reversed from the traditional interpretation. The X server typically runs on the local workstation, and the X client runs on the remote host. The X client is responsible for application management tasks, and the X server is responsible for providing display services on the graphics terminal.

X Servers and X clients can communicate over a network or through an interprocess control (IPC) connection. Because it can operate on a network, X is considered a network-based windowing system.

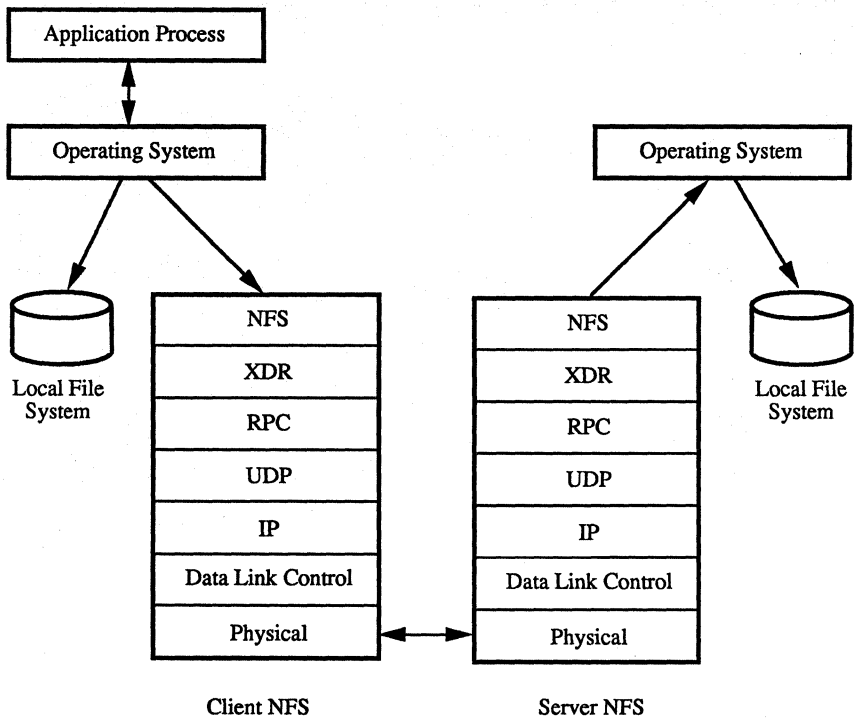
The X protocol supports requests and responses between X clients and X servers. It exchanges information necessary to operate the graphical windowing system over a network connection.

X Protocol Internals

Efficiency is an important attribute of the X protocol. It is a necessary requirement when running an application such as a distributed windowing system over a network. Efficiencies have been attained in two primary ways. First, not all X client requests require a corresponding reply from the X server; second, the X Protocol Data Units (PDUs) each have a length that is multiples of four octets. The latter allows the X PDUs to be quickly processed on host processors based on 16- or 32-bit architectures because no alignment is necessary.

Figure 5.
Network File System

The client Network File System (NFS) provides access to the remote file system over a network connection. The remote file access is transparent to the application process.



The X protocol specifies four PDU types: requests, replies, events, and errors. An X client sends requests to the X server. The X server sends replies, events, and errors to the X client.

X Request PDU
An X request PDU instructs an X server to perform a specific action and may or may not require a reply on behalf of the X server. An X client may, for example, request the X server to create a window, allocate a color, draw a graphic, etc. An X request PDU length must be a multiple of four octets.

X Reply PDU
As mentioned, not all X requests require replies. When the X client requests information, the X server will generate an X reply PDU. An X reply PDU would be sent in response to the "allocate color" request, for example. An X reply PDU's length must be a multiple of four bytes and be a minimum of 32 octets.

X Event PDU
An X event PDU contains information about either a device action or a side effect of a prior request. X clients receive most of their information through X event PDUs. An X server sends an X event expose PDU when a window display action has completed. An X event PDU's length must be a multiple of four bytes and be a minimum of 32 octets.

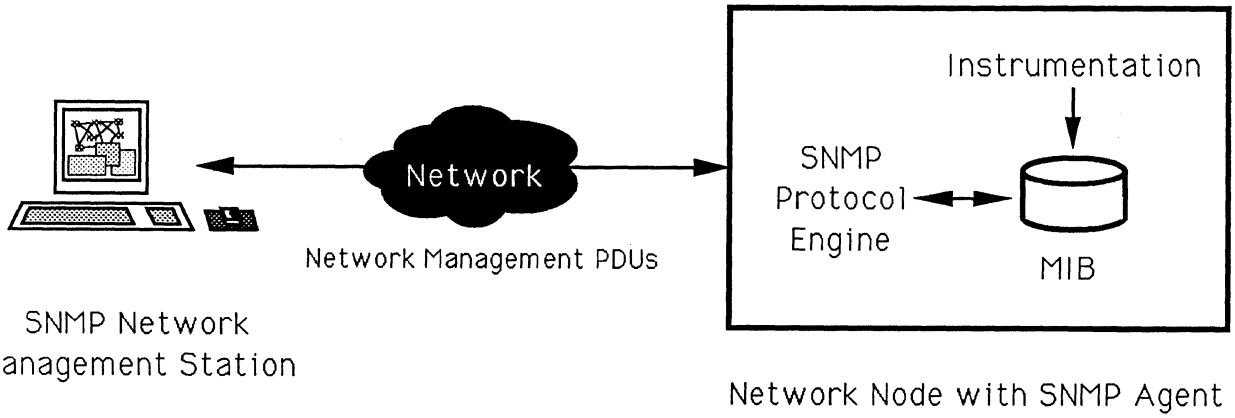
X Error PDU
An X error PDU is very similar to an X event PDU, with the exception of how the X client handles the PDU. Upon receipt, the X client transfers the error to a special error-handling routine. This facilitates a quick response to error conditions. An X error PDU's length must be a multiple of four bytes and be a minimum of 32 octets.

X appears to have a great future in network computing. Users are demanding windows-based applications because of their ease of use and short learning curves. Software developers find the platform independence of X very appealing. MIS managers realize they can preserve their investment in hardware, such as mainframes, by migrating many applications to the X environment. Each of these forces is forming the foundation for a very large market for X applications.

Network File System (NFS)

NFS provides the services of what is typically called a "network operating system." NFS is a protocol allowing multiple hosts to access each other's file systems as though they were local (see Figure 5). NFS insulates users and applications from the fact that information is being accessed

Figure 6.
SNMP Management Station



An SNMP Management Station obtains management information from an SNMP agent. The SNMP agent obtains management information through its instrumentation. The MIB is a logical database of management information.

on another system over a network connection. Using NFS, a PC can store information on a UNIX host, for example, as easily as storing information to a local hard drive. Additionally, the PC can actually run a program from software stored on the UNIX host.

NFS is host processor and operating system independent. A Remote Procedure Call (RPC) facility accomplishes this. An application can execute RPC functions and receive the results to a local data structure just as if a local procedure had been called. The application is unaware that it is processing in a distributed environment. RPC uses eXternal Data Representation (XDR) as a presentation layer service. XDR is responsible for placing information exchanged between systems in a machine-independent form so that any type of CPU can run NFS.

The demand for NFS is accelerating as network administrators discover the benefits of providing users with a standards-based approach to network operating system capabilities. Client NFS has been implemented on PCs and Macintoshes, which should spur the demand for this protocol.

Internet Suite Network Management

Managing a TCP/IP-based internet can be a complex undertaking. A TCP/IP network typically contains multivendor equipment implemented over multiple network media in a network computing environment. The Internet communities struggled for many years without protocols and tools to adequately manage these networks. In response to network administrators' demands, a working group was formed to develop a network management protocol for managing TCP/IP-based internets.

The SNMP follows a manager/agent model, similar to the client/server model (see Figure 6). An agent operates on a network device, such as a router, bridge, terminal server, or even host processor. The agent responds to queries directed to it from an SNMP manager. SNMP agent software is responsible for providing the instrumentation required to gather management data. The management data is then stored into a logical database called the Management Information Base (MIB).

The SNMP manager software runs on a network manager station and can perform three operations concerning agent interaction. The SNMP manager can get, get-next, or set specific variables in the agent's MIB. Correspondingly, an agent will provide a response to the manager requests. Additionally, an agent can, on certain events, issue an unsolicited trap message to the SNMP manager.

SNMP's authors designed the protocol with three specific goals. Briefly, SNMP was to be developed such that an agent could be deployed with minimal code, have highly extensible monitoring capabilities, and not depend on the underlying transport protocol. Each of these goals was met: many SNMP agents are under 10KB, the MIB provides private enterprise space for user extensions, and SNMP implementations exist over many transport mechanisms, including raw Ethernet.

The standard MIB is fairly small, about 100 objects. Even at that, an agent need not support the entire MIB if it does not support a specific protocol. As an example, a host that does not support ICMP need not support the ICMP object group. The Internet community is in the midst of significantly expanding the MIB to include many new objects.

Once the SNMP specification was complete, the standard was quickly embraced by vendors. Virtually every TCP/IP networking device by all manufacturers is sold with SNMP support.

L. Michael Sabo is a communications architect with SSDS, Inc., Littleton, CO, and is currently consulting on various networking projects. Previously, Mr. Sabo participated in porting TCP/IP to the emerging ANSI High-Performance Parallel Interface (HIPPI) Gigabit/sec LAN standard. Mr. Sabo has been active in integrated network management. He participated in developing an object-oriented and SNMP-based network management architecture for Lockheed Integration Services. This effort included defining numerous private enterprise management information base (MIB) objects to support system management functions.

Mr. Sabo is a member of the SNMP working group and has been active in the Internet for six years. He is a member of the board of advisors for Datapro *Network Management*. He holds a master's degree in data processing management from the University of Denver and a bachelor's degree in Computer Science from Wright State University.

Summary

The Internet suite of protocols is robust and rich with application services. Most users view the Internet protocols in terms of these application services. The demand for these services is fueling the demand for TCP/IP-based enterprise internets, which is likely to continue into the mid-1990s. ■