What Is TCP/IP?

Part II

By Stephen Force

This two-part series examines Transmission Control Protocol/Internet Protocol (TCP/IP) and the Internet, which are becoming more prevalent in traditional MVS and VM EDP shops. This article concludes this series by presenting an indepth look at the major features of TCP/IP, and the protocols for monitoring and managing TCP/IP networks.

Major TCP/IP Features

Terminal Emulation Protocol (Telnet): Telnet is a TCP/IP application for remote connection service. It allows a user at one site to gain access to a foreign host as if the user's terminal was connected directly to that foreign host. Several terminal types are supported, with VT100 and IBM 3270 being two of the most popular.

To access MVS-based full-screen applications such as CICS, IMS, TSO or SPF, a user must either have Telnet 3270 emulation on the calling host system or install a protocol converter on the called MVS host system. Once connected to the MVS system, most terminal functions are available.

A MVS user can Telnet to another TCP/IP host. To do this, she/he must log-on to TSO, get into native TSO (or SPF Option 6) and then issue the following command:

```
TELNET host_name
```

where "host_name" is either the nickname of the target host or the internet address of the target host.

Once connected, the MVS TSO user has fullscreen access to the target host. This means, for example, that an MVS-based user can LOGIN a UNIX system without having to exit TSO.

Those of us who must deal with languages other than English are painfully aware of the special characters peculiar to foreign languages. To get around this problem, several vendors provide the so-called national language support that provides character translation tables. For pre-Version 5 MS-DOS users, a possible solution would be to use the SimPC (Simware, Inc., Ottawa, Ontario Canada) terminal emulation software along with the PC/TCP (FTP Software, Inc., Wakefield, Mass.) TCP/IP product, allowing SimPC to handle the character translation. MS/DOS 5.0 users can implement the proper code page to do these translations. TCP/IP translation tables can also be implemented for some character translation if the user feels that these tables apply to all users of this TCP/IP system.

File Transfer Protocol (FTP): You can send data from one system to another easily using the TCP/IP File Transfer Protocol (FTP). FTP is a TCP/IP application used for transferring files to and from foreign hosts. FTP also provides the capability to access directories. Password protection is provided as part of the protocol.

Using FTP is simple. After connecting to the desired host (server), and providing the user and password (if necessary), you can control both the source and target directories (or in MVS, high-level qualifiers) using provided commands, then either GET or PUT the file.

Almost any system can be used as a server. Even MS-DOS can be set up as a server. This means that a MS-DOS PC data can be easily accessed by an MVS system. So those of us with MVS DASD constraints could simply use the PC as a storage medium! Just a thought.

FTP can also be used in batch mode. A MVS user can set up FTP commands in a TSO/E clist (or REXX EXEC), thereby saving keystrokes. Or, FTP can be used in a MVS batch job by simply invoking batch TSO. FTP commands can also be "batched" in MS-DOS (.BAT file), OS/2 (command file) or UNIX (inherent functions).

The same translate tables mentioned in the Telnet segment can also be used for FTP.

Network File Server (NFS): A more powerful data transfer method than FTP is the Network File Server (NFS). The NFS protocol, developed by Sun Microsystems, Inc., allows computers in a network to access each other's file systems. Once accessed, the file systems appears to reside on the local host (client).

NFS can be easily compared to a local area network (LAN) file server. When a PC connects to a LAN file server, it assigns a letter to each logical device. For example, device "C" on a LAN file server might be known to the PC (client) as logical device "V". NFS operates in a similar way. After successful server connection, the client treats the server device as she/he would her/his own, using normal commands.

NFS provides inherent data security by requiring all files that can be accessed be Export files. Only these files can be accessed, no others. Export files can be defined by the server administrator as read-only or full access (read/write).

NFS support software is not always provided in the base TCP/IP product. PC/TCP for MS-DOS users, for example, must obtain FTP Software, Inc.'s InterDrive for DOS to access most NFS servers. NFS is built-in to MVS Data Facility Product (DFP).

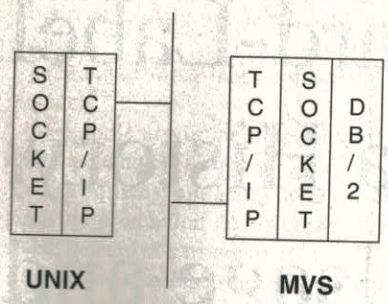
MVS/ESA users, on the other hand, have NFS built-in (DFP 3.2 or higher). This provides MS-DOS, OS/2 and AIX client routines that can be downloaded from MVS using, for example, FTP. This means that an MVS/ESA-TCP/IP V2R1 user with the proper maintenance (PUT) level can start using NFS immediately.

Simple Mail Transfer Protocol (SMTP): TCP/IP provides an built-in electronic mail system. Although primitive, it offers a highly useable means of transferring messages throughout the internet.

A TCP/IP application protocol is used to transfer mail between users on different hosts. SMTP specifies how mail systems interact and the format of control messages they use to transfer mail.

MVS TCP/IP provides a interface between SMTP and JES2 or JES3 NJE. A TSO user can send a message or note to a TCP/IP user the same way as she/he would to another TSO user on either the same or another NJE node. A future article will address this subject in more detail.

•• FIGURE 1: Sample Client/Server Socket



Also, as previously mentioned, a Compuserve user can send messages on the Internet. This is another example of using SMTP as an E-mail system.

Sockets

Most users need to enhance systems to suit their particular needs. TCP/IP offers several possibilities. Sockets, or application programs in TCP/IP terminology, can be written to exploit the full functionality offered by both the host system and its implementation of TCP/IP.

Although MVS TCP/IP sockets can be written in COBOL or PL/I (with IBM assembly subroutines doing the actual TCP/IP interfacing), users are encouraged to use PASCAL or C/370 to make the socket programming standardized and relatively transportable.

SAS and REXX routines can also call the assembler subroutines, thereby making the power of sockets available to the average user.

Normally, sockets consist of two parts: one part executing on the server, the other on the client. For example, a UNIX work station CAD/CAM user might need to retrieve data from a MVS/ESA DB2 data base. A socket can be written on the UNIX work station (client) to request this data and then load the retrieved data onto its disk. The MVS/ESA socket (server) receives this request, issues the proper DB2 calls to extract the data, then sends it to the client. See Figure 1.

TCP/IP sockets can be written to bridge the TCP/IP internet and application program-to-program communication (APPC LU6.2) gap.

A socket could also simplify interaction between a TCP/IP network and a APPC program running anywhere in the SNA network. For example, currently, there is no inexpensive way for a TCP/IP host to get data to and from a CICS application other than by using Telnet and then logging into the CICS system as a terminal.

If the CICS transaction was written using APPC commands, then a (i.e., MVS) socket could be written, which, on one side, issues TCP/IP communication commands, and on the other side, issues APPC communication commands. See Figure 2.

Remote Procedure Call (RPC)

More advanced than sockets are programs written using Remote Procedure Call (RPC) protocol. Developed by SUN Microsystems, the RPC protocol is an application programming interface (API) available for developing and writing distributed applications. The RPC API allows programs to call subroutines that are executed on a remote system.

Packet InterNet Groper (PING)

Often, in data communications we need to know if our system is communicating with others. TCP/IP provides the PING command to do this. PING simply sends out a "Are you there? If so, then please answer!" message to the desired TCP/IP host, then times the response. It is a simple, yet very effective tool.

Monitoring and Managing TCP/IP Networks

Reliability and availability are necessary for successful networking. With the growth in size and complexity of the TCP/IP-based internets, the need for network management is very important to keep it working.

TCP/IP management software runs at the application level. To participate, each host or gateway must have a server program running. Not all implementations of TCP/IP have this server software available, at least not in the base product.

A network manager (client) program contacts all specified servers and sends queries to obtain information from these servers; otherwise it can send commands to alter the conditions of a gateway.

Common Management Information Protocol (CMIP)

Common Management Information Protocol (CMIP) is an evolving network management and control proposal being developed by the International Standards Organization (ISO). The ISO proposals primarily define the functions of network management software, as well as define fault management, configuration management, security and accounting management.

CMIP is the most standard network management software vendors are striving for. However, as CMIP is still quite new and CPU-intensive, most network management software products and philosophies still use the Simple Network Management Protocol (SNMP).

SNMP

SNMP allows network management by elements such as gateways, routers and hosts. This protocol provides a means of communication between network elements regarding network resources. SNMP works well in most large and small TCP/IP networks.

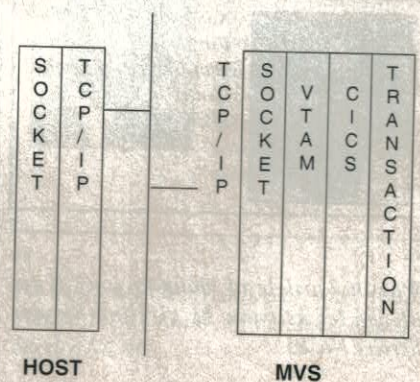
The major components of SNMP are the agents and management stations, with agents reporting to the management stations.

Users can monitor the internet by installing and using tools that interpret SNMP Management Information Base (MIB) standard data. The MIB standard specifies which data items a host or gateway must keep and the operations allowed on each. Several vendors offer products to interpret these MIB standard data. Otherwise, user programs can be written.

NetView Interface

IBM's NetView can be used to monitor the internet by extracting data from the SNMP client on its system using provided clists. These clists are delivered with the MVS TCP/IP product and provide limited SNMP functions. A NetView systems programmer can develop her/his own clists or programs to interpret SNMP data (using well documented MIB objects).

•• FIGURE 2: Sample TCP/IP and APPC Command Socket



Implementing the NetView interface reduces the problems of operating a TCP/IP network. Once the data is made available to NetView, user-selected alerts can be implemented, and perhaps some network outage can be eliminated by automating network recovery procedures.

Summary

TCP/IP is rapidly making in-roads in the traditional EDP shop. Although not a true "open system" architecture, it provides the most widely used and accepted possibilities for linking disparate systems together.

This article intended to introduce you to TCP/IP. It briefly described the history of TCP/IP and some of its key functions. Unfortunately, this series has not covered all aspects of this complicated topic. If you would like to learn more about TCP/IP, the following publications will help:

•*Internetworking With TCP/IP Volume I: Principles, Protocols and Architecture* (Second Edition), Douglas Comer, Prentice-Hall, Inc. 1991, ISBN 0-13-468505-9.

•*Internetworking With TCP/IP Volume II: Design, Implementation and Internals*, Douglas E. Comer and David L. Stevens, Prentice-Hall, Inc. 1991, ISBN 0-13-472242-6.

•*TCP/IP Tutorial and Technical Overview*, IBM ITSC "red" book, GG24-3376-01.

•*TCP/IP V2 for MVS: Installation and Maintenance*, IBM Corp. SC31-6085.

•*TCP/IP V2 for VM: Installation and Maintenance*, IBM Corp. SC31-6081.

•*TCP/IP Tuning Experiences* (MVS), IBM ITSC "red" book, GG24-3838-00.

•*OSI/Communications Subsystem (MVS) Installation, Configuration and Operations Guides*, IBM ITSC "red" book, GG24-3575-00

•*Dvorak's Guide to PC Telecommunications*, Second Edition, John C. Dvorak and Nick Anis Osborne, McGraw-Hill 1992 ISBN 0-07-881787-0.

•*PC Magazine Guide to Connectivity*, Second Edition, Frank J. Derfler, Jr., Ziff-Davis Press, 1992 ISBN 1-56276-047-5.

Other good sources of information are Compuserve forums, UNIX specialists in the organization and IBM Link (or "Dial IBM" in Europe).

NaSPA member Stephen Force is a consultant with Chrysler Technical Center in Auburn Hills, Mich.

/*
Was this article of vlaue to you? If so, please let us know by circling Reader Service No. 85.

VSE Standard-Label Area History, Usage and Performance

By Arthur J. Stipo

To make everyone aware of the VSE performance considerations related to the VSE standard-label area, I have included historical information and usage considerations. Refer to the December 1990 issue of ***Technical Support*** for additional tips and techniques.

History

When the DOS operating system was first introduced in the mid-60s, DOS supported only one batch partition. It was convenient then (and still is) to store the label information required for an installation's most commonly used files in a central place: the standard-label area.

Later, as DOS evolved through DOS/VS, DOS/VSE, DOS/VSE/AF, VSE/SP, VSE/SP2 and now VSE/ESA, VSE was enhanced to handle three, then five, then seven, then 12 and now 200 partitions in VSE/ESA. To maintain compatibility, the standard-label area facility was retained and expanded to support maintaining the ever-growing volume of file labels, but was not enhanced concerning performance considerations.

Originally, the standard-label area was a cylinder within the SYSRES library, then two cylinders on DOS/VSE and now up to 16 cylinders of 3380/3390 DASD can be allowed on VSE/SP3 or VSE/ESA.

Description

The standard-label area is a VSE/SP or VSE/ESA system sequential disk file, consisting of a list of DLBL/Extents and/or TLBLs. The label area is dynamically divided into three areas: STDLABEL, PARSTD and USRLABEL. The STDLABEL option allows labels to be stored and accessed by all jobs in all partitions. The PARSTD option allows labels to be stored and accessed by all subsequent jobs in the specified partition. The USRLABEL option (the system default) stores labels submitted by a job into that partition's temporary label area for use by that job only.

The label area is searched sequentially for a label when a file is opened. The temporary area is searched first; in case a label was submitted with the job, the partition area is searched second and the system area is searched last.

Refer to *VSE System Control Statements, VSE/SP System Management Guide and VSE/ESA Guide to System Functions* for additional information.

Advantages of using the standard-label area include:

•eliminating the time-consuming and error-prone task of duplicating VSE JCL for each job step that uses a common file; and

•allowing a file's location to change