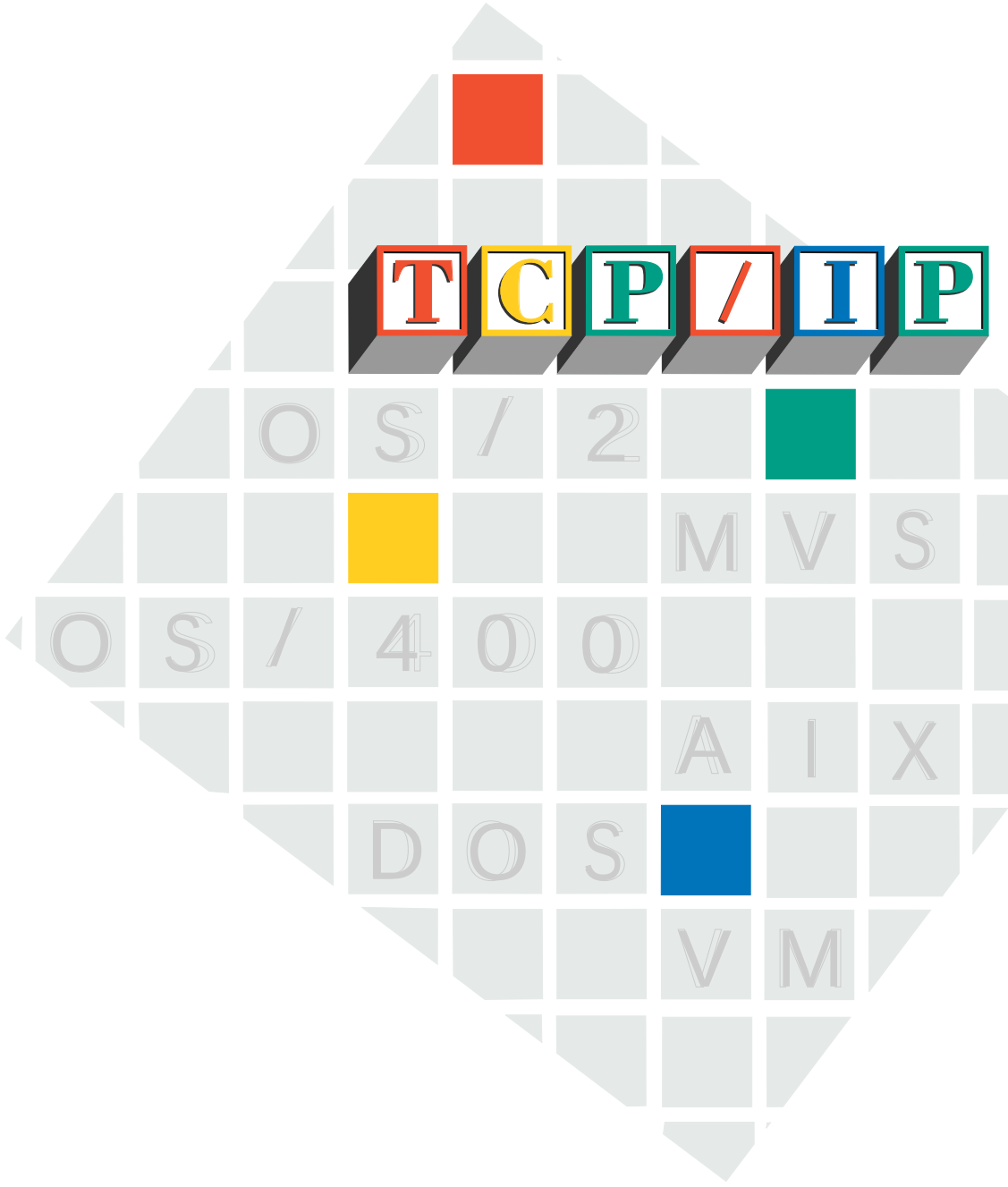
IBM TCP/IP

SC31-7188-02

Performance Tuning Guide


TCP/IP
OS/2
MVS
OS/400
AIX
DOS
VM

IBM TCP/IP

SC31-7188-02

Performance Tuning Guide

Note

Before using this information and the product it supports, be sure to read the general information under Appendix D, “Notices” on page 227.

This book is also available in a softcopy form that can be viewed with IBM BookManager* READ.

Third Edition (March 1997)

This edition applies to OS/390 (5645-002 5645-003) and to the following IBM TCP/IP programs:

- Version 3 Release 2 for MVS
- Version 3 Release 1 for MVS
- Version 2 Release 2.1 for MVS
- Version 2 Release 2 for VM
- Version 3 Release 2.3 AIX for RISC System/6000
- Version 2 for OS/400
- Version 2 Release 0 for OS/2
- Version 2 Release 1.1 for DOS

Publications are not stocked at the address given below. If you want more IBM publications, ask your IBM representative or write to the IBM branch office serving your locality.

A form for your comments is provided at the back of this document. If the form has been removed, you may send comments to:

IBM Corporation
Department CGMD
P.O. Box 12195
Research Triangle Park, North Carolina 27709
U.S.A.

FAX (United States and Canada): **1-800-227-5088**
IBM Mail Exchange: **USIB2HPD at IBMMAIL**
IBMLink: **CIBMORCF at RALVM13**
Internet E-mail: **USIB2HPD@VNET.IBM.COM**
World Wide Web: **http://www.s390.ibm.com/os390**

IBM may use or distribute any of the information you supply in any way or distribute any of the information you supply without incurring any obligation to you.

Contents

Figures

Tables

About This Book

This book describes how to tune IBM TCP/IP for better performance. The book explains how to rank performance objectives in order of priority, the monitoring tools available, and the key tuning parameters.

For comments and suggestions about this book, use the Reader's Comment Form located at the back of this book. This form gives instructions for submitting your comments by mail, by fax, or electronically.

| TCP/IP Version 3 Release 1 and TCP/IP Version 3 Release 2 for MVS are an integral part of the OS/390 family of products. For an overview and mapping of the documentation available for OS/390, see the *OS/390 Information Roadmap*.

Who Should Use This Book

This book is designed for the following audiences:

- System programmers installing TCP/IP to generate the best profile
- Performance analysts to learn the available tools, monitors, and recommended processes for getting the performance data from the system to establish confidence that the system is delivering the best performance and the options for tuning the system if it is not
- Application programmers to know what options are available and their trade-offs for best application performance while minimizing impact on system performance
- System planners for information on capacity planning and hardware and software requirements in comparison to other networks
- Networking specialists to gain insight into how the product works and why different parameters affect performance

Where to Find Related Information on the Internet

You may find the following information helpful.

For current updates to the TCP/IP Version 3 Release 2 for MVS documentation described in "Bibliography" on page 275, check out the TCP/IP for MVS home page:

`http://www.raleigh.ibm.com/tcm/tcmprod.html`

To keep in close touch with OS/390, we suggest you look at the OS/390 home page:

`http://www.s390.ibm.com/os390`

To keep abreast of new products and technologies from IBM Networking, take a look at the IBM Networking home page:

`http://www.raleigh.ibm.com/`

How to Contact IBM Service

For telephone assistance in problem diagnosis and resolution (in the United States or Puerto Rico), call the IBM Software Support Center anytime (1-800-237-5511). You will receive a return call within 8 business hours (Monday – Friday, 8:00 a.m. – 5:00 p.m., local customer time).

Outside of the United States or Puerto Rico, contact your local IBM representative or your authorized IBM supplier.

Summary of Changes

Summary of Changes
for SC31-7188-02
TCP/IP Version 3 Release 2 for MVS

This is the third edition of this book. This book supports TCP/IP Version 3 Release 2 and the OS/390 family of products.

The updates contained in this edition are effective only if the latest level of maintenance has been applied. New and changed information is indicated by a revision bar (|).

New Information: The following enhancements are new for this revision:

- **Performance comparisons of V3R1 and V3R2**
  Performance improvements in V3R2 are compared with V3R1 and V3R1+ for FTP, Telnet, TCP Sockets (MVS Send and MVS Recv), and UDP Sockets CPU (MVS Send and MVS Recv) This information can be found in Appendix C, "IBM's TCP/IP V3R2 Performance Improvements" on page 217.
- **Performance Tuning Checklist**
  If you are experiencing performance problems or want to ensure that your TCP/IP environment is properly tuned, check parameters provided in the checklist "Performance Tuning Checklist" on page 105. Also given are a summary of guidelines to determine the source of your TCP/IP performance problems.
- **Guideline to determine TCP/IP performance problems**
  Guidelines are provided to determine the source of your TCP/IP problem.

Changed Information: Various minor editorial and technical updates have been applied to this edition.

Summary of Changes
for SC31-7188-01
OS/390 Release 1:
TCP/IP Version 3 Release 1 for MVS

This is the second edition of this book. This book supports TCP/IP Version 3 Release 1 and the OS/390 family of products.

The updates contained in this edition are effective only if the latest level of maintenance has been applied. New and changed information is indicated by a revision bar (|).

New Information: The following enhancements are new for this revision:

- **General performance tuning tips and techniques**
  The latest tips and techniques for tuning the performance of TCP/IP Version 3 Release 1 and TCP/IP Version 3 Release 2 are provided in Appendix A, "Performance Tuning Tips for MVS" on page 183
- **Performance PTFs**
  Enhancements boost internet capacity for S/390, increase efficiency and reduce overhead of TCP/IP, especially in high– volume Telnet, TN3270, and FTP envi-

ronments. These PTFs are already installed on the level of TCP/IP Version 3 Release 1 that comes with OS/390.

Changed Information: Various minor editorial and technical updates have been applied to this edition.

Part 1. Performance and Tuning

Chapter 1. Understand the Performance and Tuning Environment for TCP/IP

This chapter explains performance tuning, giving guidelines and discussing the limits of what you can expect. The performance tuning process is explained, as are some basic TCP/IP concepts.

What Is Performance Tuning?

Performance is the way a computer or network behaves given a particular work load. It can be measured by monitoring response time, throughput, and availability; and it is affected by:

- The resources available
- How well they are used and shared

In general, you should do performance tuning when you want to improve the price/performance ratio of your system. Possible specific goals might be:

- To send more files across the network in a given day without buying more hardware or using more processor time
- To pass data more quickly between two specific host computers
- To lower the overhead of TCP/IP on an application

Translating performance from technical terms to economic terms is difficult. Performance tuning itself certainly costs money through people's time and through processor time spent monitoring. So before you undertake a tuning project, weigh the costs of the tuning against the possible benefits. Some of these benefits might be measurable, such as more efficient use of resources and the ability to send more files across the network, while others such as happier users because of quicker response time are harder to quantify. All possible benefits should be considered.

Guidelines for Performance Tuning

The following guidelines should help you to develop your overall approach to performance tuning for TCP/IP.

Performance Tuning Is a Reallocation of Resources: Carefully choose your tuning objectives because the way you tune to improve one resource might affect the way one or more other resources are used. For example, when you reduce CPU utilization, storage consumption might increase. Before you begin tuning, have a clear idea of which of your resources you are running out of and in which there is still room for growth.

Remember the Law of Diminishing Returns: Your greatest performance benefits usually come from your initial efforts. Further changes generally produce smaller and smaller benefits and require more and more effort.

Do Not Tune Just for the Sake of Tuning: Tune to relieve identified constraints. If you tune resources that are not the primary cause of performance problems, this has little or no effect on response time until you have relieved the major constraints, and it can actually make subsequent tuning work more difficult. If there is

any significant improvement potential, it lies in improving the performance of the resources that are major factors in the response time.

Consider the Network As a Whole: Network performance is constrained by how far and how fast information can travel and by how many times you have to move the data. Consider network performance as a whole, not just TCP/IP.

Change One Parameter at a Time: Do not change more than one performance tuning parameter at a time. Even if you are sure that all the changes will be beneficial, you will have no way of evaluating how much each change contributed. You also cannot effectively judge the trade-off you have made by changing each parameter. Every time you adjust a parameter to improve one area, you almost always affect at least one other area.

Understand the Problem before You Upgrade Your Hardware: Even if it seems that more storage, processor power, or a new router could immediately improve performance, take the time to understand where your bottlenecks are. You might spend the money on more storage only to find that you do not have the processing power to exploit it or the limits of the network you are crossing cannot move data any faster anyway.

Put Fallback Procedures in Place before You Start Tuning: Sometimes tuning can cause unexpected performance results. If what you have changed leads to poorer performance, you should reverse the change and try something else. If you save the former setup in such a manner that it can be recalled, then backing out of the incorrect change will be much simpler.

Limits to Performance Tuning

There are limits to how much you can improve the efficiency of a system. Consider how much time and money you should spend on improving TCP/IP performance, and how much the spending of additional time and money will help the users of TCP/IP.

Your installation of TCP/IP might perform adequately without any tuning at all, but it probably will not perform to its potential. Unfortunately, using the default tuning parameters is often not a good solution. Each network and set of applications is unique. Given a clear knowledge of your network and applications, investigate the tuning parameters available and learn how you can customize their settings to reflect your situation. In some circumstances, there will only be a small benefit from tuning TCP/IP. In others, the benefit will be significant.

As your network approaches a performance bottleneck, it is more likely that tuning will be effective. If you are close to this and you increase the number of users on the network by, for example, 10 percent, the response time is likely to rise by much more than 10 percent. However, there is a point beyond which tuning cannot help you. At that point, the only thing to do, other than adding new hardware, is to change your objectives.

How to Improve Performance

There is a logical step-by-step process to follow when tuning performance.

1. **Know Your Current Network Environment**

   Before you begin to tune performance, you must know what your needs are given your current hardware, software, and applications.

2. **Establish Performance Objectives**

   Once you know what you are working with, you must define your objectives and their priority.

3. **Monitor and Collect Performance Data**

   Find and use the right tools to help you monitor the resources that you have defined as the highest priorities.

4. **Analyze Your Performance Data**

   Next you should analyze the results of the monitoring tools you chose to use.

5. **Tune Your Network Environment**

   You should make incremental tuning changes to your configuration, one at a time.

6. **Test the Changes to Performance**

   You should test the changes you have made to see if they have had the results you anticipated, while paying special attention to sensitive resources.

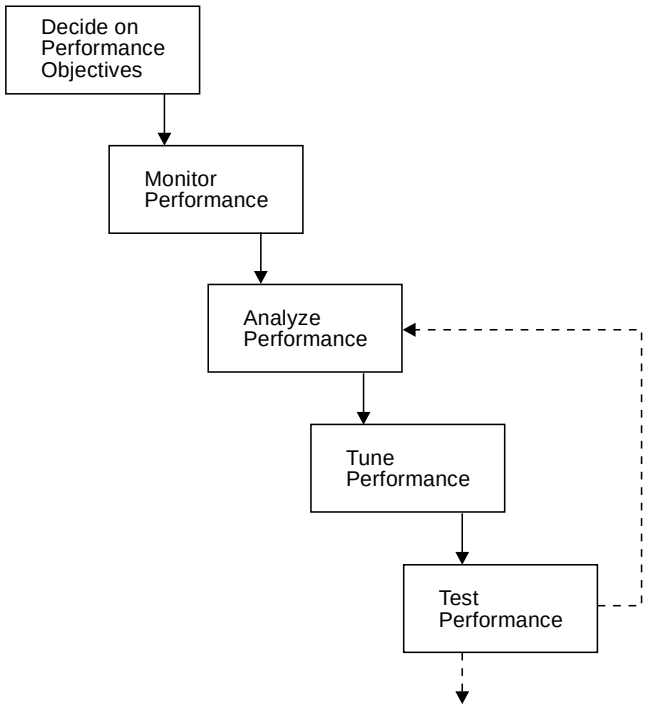


Figure 1. The Iterative Nature of Tuning

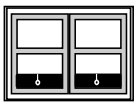
The whole performance tuning process is iterative as you make a change, test it, make another, test that, and so on until you feel you have reached the best performance you can achieve without spending too much time or money.

Periodically after you have made any significant changes to either the work load, system, or network, re-examine your objectives and refine your monitoring and tuning strategy.

Understand Two Basic TCP/IP Concepts

Before tuning, you should understand some basic concepts of TCP/IP that are true on all operating systems. The two concepts explained here are windowing and fragmentation. The window (sending or receiving buffer) size can have a major effect on throughput. Fragmentation and reassembly of packets can be expensive in bandwidth and CPU utilization.

Understanding TCP/IP Send and Receive Buffers



The TCP/IP suite includes two major protocols: Transmission Control Protocol (TCP) and User Datagram Protocol (UDP).

TCP Send and Receive Buffers

TCP/IP groups the TCP data it transmits into segments and uses multiple data buffers also called windows to send and receive the segments.

Each segment is sent onto the network with an IP header and a TCP header. Together they are called a packet. The maximum segment size (MSS) will vary by host and network media and therefore, the packet size will also vary. When the packet reaches the destination, the headers are removed and the segment is stored in the receiving window.

The best buffer size is generally one that allows for maximum throughput. The receiver determines how rapidly to send data by telling the sender how much receive buffer space is available. If there is not enough receive buffer space, the sender will have to slow down how many packets it sends, decreasing throughput.

The sender tries to send as much data as it can before receiving an acknowledgment. The sending buffer size needs to be large enough to transmit data quickly. Windowing allows the sender to maximize throughput by keeping the network constantly busy rather than waiting for acknowledgments.

The send and receive buffers can also be thought of as sliding windows. Figure 2 shows an example of how TCP/IP windows can be used by the sender to group its segments for transmission.

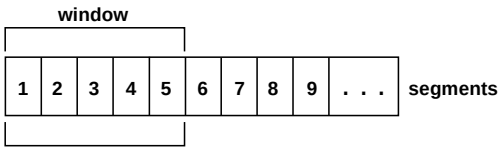


Figure 2. Segments in a Sending Window

The sender will send out all the segments within the window, whether or not it has received an acknowledgment (ACK) for any of the bytes in the segments, as shown in Figure 3. It also starts a time-out timer for each packet sent.

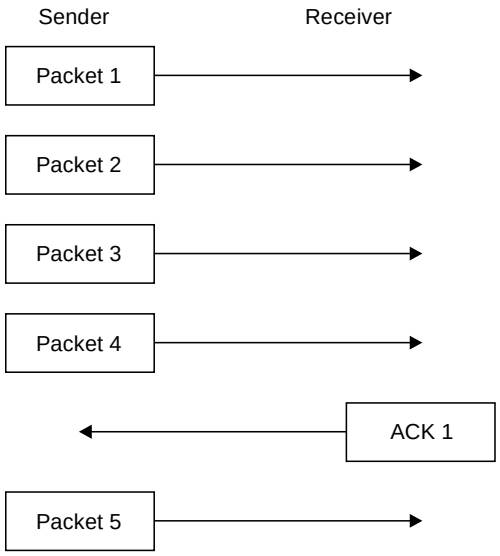


Figure 3. Packets and Acknowledgments

The receiver sends back an ACK indicating the sequence number.

At the moment the sender receives the ACK for the last byte in segment 1 (ACK 1), it slides its window to exclude segment 1, as shown in Figure 4.

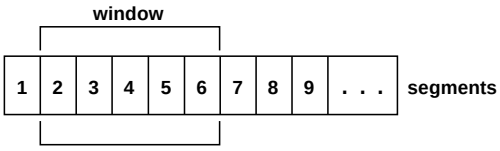


Figure 4. Sliding the Sending Window

In this example, the sender would now transmit segment 6.

The sender stops transmitting when it has sent all the segments in the window. If the sender notices the time-out for a packet, it will retransmit that packet.

TCP window size is expressed as a number of bytes (rather than a number of packets as in other protocols). It is determined by the receiver when the connection is established, and can vary during the data transfer. Each ACK message will include the window size that the receiver is able to process at that particular time.

As shown in Figure 5, the receiver in this example can only hold 4 segments in its window, while the sender can hold 5.

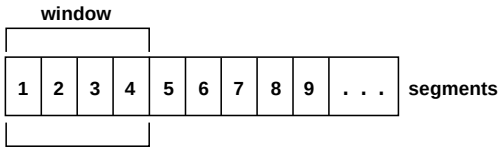


Figure 5. The Receiving Window

Because of this implementation, using a larger window size on both the sending and receiving ends will usually result in better throughput. However, the bigger the window, the bigger the exposure of how much the sender might have to re-transmit.

UDP Send and Receive Buffers

Instead of sending and receiving data as multiple packets of a byte stream, UDP sends and receives datagrams.

A datagram is an arbitrary unit of data. It can be part of a larger set of data. For example, if an application has to send 4KB of data to another application on another host, it can use UDP to send it as one 4KB datagram or as multiple datagrams, like four 1KB datagrams, as shown in Figure 6.

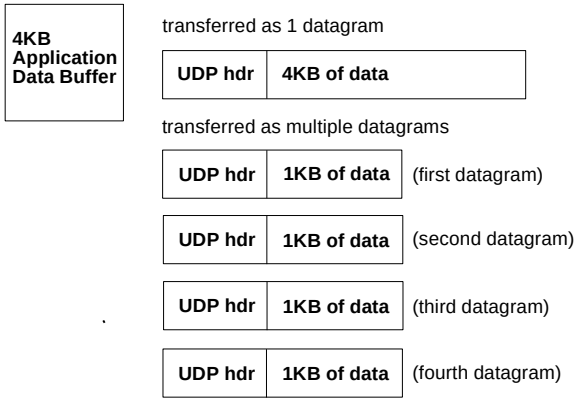


Figure 6. Example of How UDP Transfers Datagrams

Each datagram is treated separately by UDP. No checking is done to see that the datagram is delivered successfully, and no acknowledgments are returned from the receiver. So in our example, UDP on the sending end will send each of the four 1K datagrams to the destination without checking to see if the other datagrams have been delivered successfully.

UDP on the receiving end will direct each datagram separately to the application on the other host. It is up to the application programs to detect lost datagrams, datagrams received out of order, and other reliability problems. Figure 7 shows the process followed.

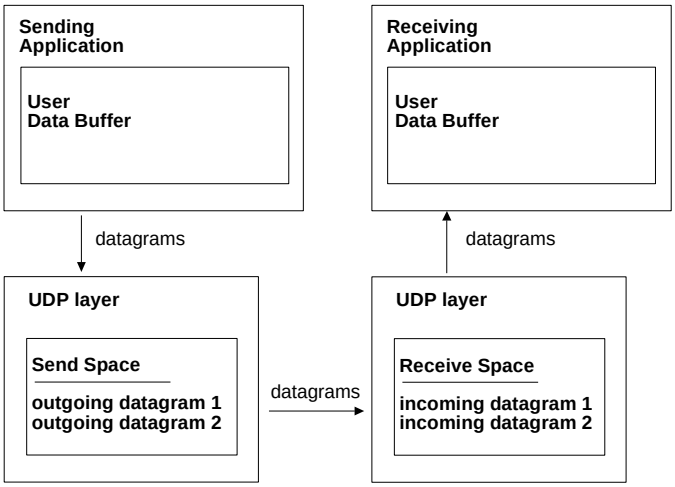


Figure 7. Example of How UDP Sends Datagrams to an Application

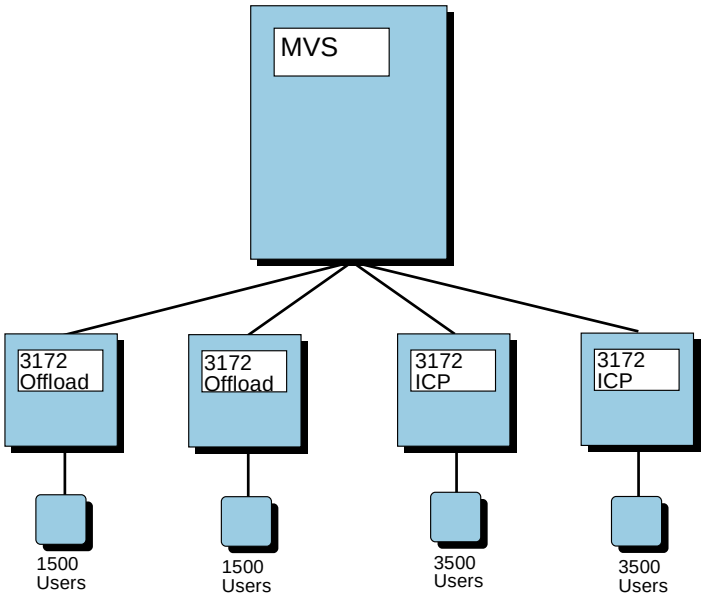
Like TCP, UDP requires buffer space for its data transfers. The receiving space is used to store incoming datagrams until the target application is ready to receive them. The sending space is used to store outgoing datagrams that have been received from a local application but have not yet been sent out to the network.

When the receiving buffer space for UDP is filled with incoming datagrams, it cannot accept any more datagrams from the network. When this happens with a TCP application, the sender is informed and stops sending packets until there is more room in the buffer. With UDP, the sender does not know when the receiving buffer is full, so it continues to send more data. The receiver discards the extra incoming data.

Similarly, UDP cannot accept any more data from local applications when the UDP sending buffer space is full. The sending application has to wait until there is room in the sending buffer.

As a result, it is important that you define large enough sending and receiving buffer spaces so that UDP does not discard incoming data or delay outgoing data. Also, you should establish a way of asking UDP for resends, to control the sending rate, and so on, as needed for reliable reception of the completed data. Network File System** (NFS) has a built-in way to make reception more reliable.

Understanding Fragmentation



As part of performance tuning, you can change the maximum transmission unit (MTU) for the packet size. Some networks limit the packet size to a smaller value. For example, the largest packet size for an Ethernet Version 2 IEEE is 1500 bytes and for Ethernet 802.3 it is 1492 bytes.

In some circumstances, you can change the MSS, indirectly changing the MTU. The MSS plus the IP header of 20 bytes and the TCP header of 20 bytes equals the MTU or packet size. For example, an MSS of 536 plus the 40-byte header would make an MTU of 576 bytes.

Each packet contains at least one header. The size of the packet headers are independent of the amount of data included, so the larger the packets sent, the less relative bandwidth is consumed by protocol headers.

TCP/IP consumes a fixed amount of CPU overhead time for each packet, independent of the packet size. There is also a variable amount of CPU time depending on the packet size. Overall, less CPU time is consumed as the number of packets transmitted decreases, and as each packet size increases.

Large packets can be fragmented, each with its own header, by intervening gateways as the packet passes from sender to receiver. Each time the packets are fragmented, the proportion of header to data increases, as shown in Figure 8.

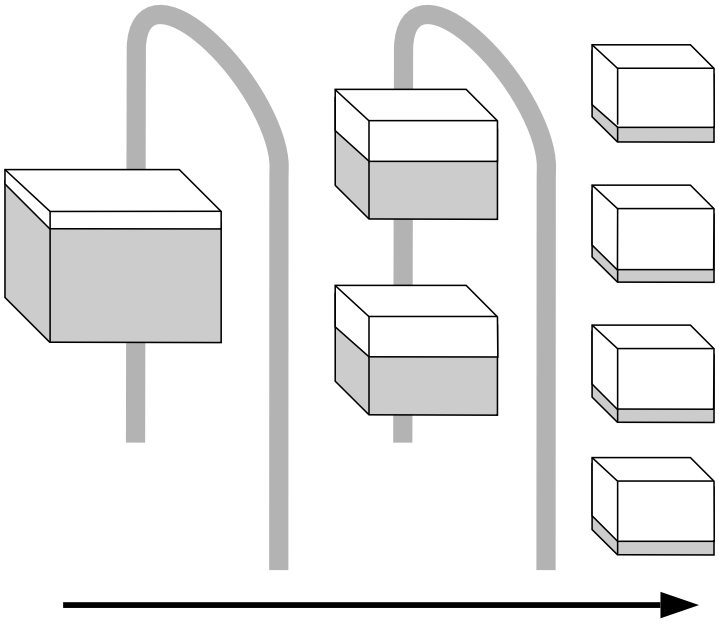


Figure 8. Packet Fragmentation

To reassemble the fragments, the receiving host allocates a buffer in storage as soon as the first fragment arrives. When subsequent fragments of the data arrive, they are copied into the buffer storage. As soon as all fragments have arrived, the complete datagram is restored.

The resource expenses of fragmentation for the sender are:

- CPU overhead of creating and transmitting the additional packets
- Retransmission of **all** the packets in the datagram if a packet is lost

The resource expenses of fragmentation for the receiver are:

- CPU overhead of re-assembling the packets
- Memory overhead of maintaining the buffer to re-assemble the packets
- Delays (longer response time) if a packet is lost

Chapter 2. Know Your Current Environment

Your environment includes anything that contributes to the performance of TCP/IP.

This book reflects the performance enhancements made in both TCP/IP Version 3 Release 1 and Version 3 Release 2 for MVS.

The first part of this chapter describes some of the applications, functions, and protocols that you might be using at your TCP/IP installation.

The second part of this chapter describes some of the hardware that you might be using and how it might affect performance of TCP/IP.

What Applications, Functions, and Protocols Do You Use?

To help you to tune TCP/IP performance, you must understand:

1. What version and level of TCP/IP are you using?

   Periodic performance enhancements are delivered by means of new releases and PTFs. This book was originally published when Version 3 Release 1 of TCP/IP for MVS was the most current version. At that time, many customers were still on V2R2.1; consequently, this book contains examples and tuning parameters for all versions (V2R2.1, V3R1 and V3R2)

   Since the original publication date of this book, a number of performance PTFs have been released. Many of these were for V3R1. In addition, new tuning techniques have been developed based on those PTFs.

   Version 3 Release 2 of TCP/IP for MVS is the most current version. In addition to many performance enhancements, V3R1 Performance PTFs are integrated into V3R2. V3R2 provides significant reduction in CPU cycles and also improves throughput. Appendix A, “Performance Tuning Tips for MVS” on page 183 provides information about tuning recommendations.

2. What applications, functions, and protocols are being used?
3. What is the volume of work for each?

   Does the mix change over the course of the day, week, or month? For example, is there a spurt of file transfers using File Transfer Protocol (FTP) at 8:30 each morning, or are file transfers spread out more or less evenly through the day?

4. What is the priority of work in the installation?

   For example, Telnet is usually a high priority application because its users are sensitive to performance degradation. As a Telnet user, you will notice if there is a delay between the time you press a key and when the system responds.

   On the other hand, Simple Mail Transfer Protocol (SMTP) might be a lower priority, with its users requiring that mail be delivered within minutes, not seconds. This is usually because mail is sent expecting recipients to open it at their convenience.

   FTP could be more difficult to judge. You might have users who are transferring large or small files throughout the day where each transfer must occur as quickly as possible because they (or another person or application) must

wait for the file transfer to complete before they can continue working. Or you might have an automated program that transfers files outside of normal working hours, where the elapsed time is not as important. Most likely you will have a combination of various requirements for FTP.

5. What is the priority of the TCP/IP work to the rest of the system? Are most of your users using TCP/IP or is TCP/IP co-existing with other kinds of network products? Which product has top priority? You need to know whether TCP/IP is critical or merely allowed space as a courtesy.
6. For each application, function or protocol, where is the traffic going?

   Do you have knowledge of or control over each host that is exchanging data using TCP/IP, that is, is all the data being transferred within your installation? Or are your users using TCP/IP to communicate with hosts on a larger network, such as the Internet or a private inter- or intra-enterprise network? You need to find out as much as possible about the possible and then probable sources, destinations, and intermediate hosts.

   Keep in mind that you will want to tune first for high-priority traffic; second for high-volume traffic; last for low-volume and low-priority traffic.

You need to know the answers to the previous questions for all applications, functions, and protocols. The following sections give a brief explanation of the most common applications and their possible impact on performance. The more you know about your TCP/IP environment before you begin tuning, the better.

Using FTP

FTP is used to transfer files between hosts. You initiate the transfer on one host by using the FTP command to login to the other host and then requesting that a file or files be transferred to or from the other host.

FTP is usually a high-visibility item in terms of performance. Throughput (KBps) is calculated and displayed to the end user when the transfer is complete. Often you watch your display while the transfer is taking place.

Shown in Figure 9 is an example of the response returned when using FTP on VM.

```
Command:
put rfc1178.txt
>>>SITE VARrecfm
500 'SITE VARRECFM': command not understood.
>>>PORT 9,67,22,1,11,18
200 PORT command successful.
>>>STOR rfc1178.txt
150 Opening data connection for rfc1178.txt.
226 Transfer complete.
18626 bytes transferred. Transfer rate 57.84 Kbytes/sec.
```

Figure 9. Example of an FTP Response on VM

FTP performance is influenced by:

- Whether any data translation has to take place (ASCII to EBCDIC, for example). Binary (no translation) is faster than translation.

  Figure 10 shows amount of time spent reading, writing, and completing other processes with the transfer of a binary file.

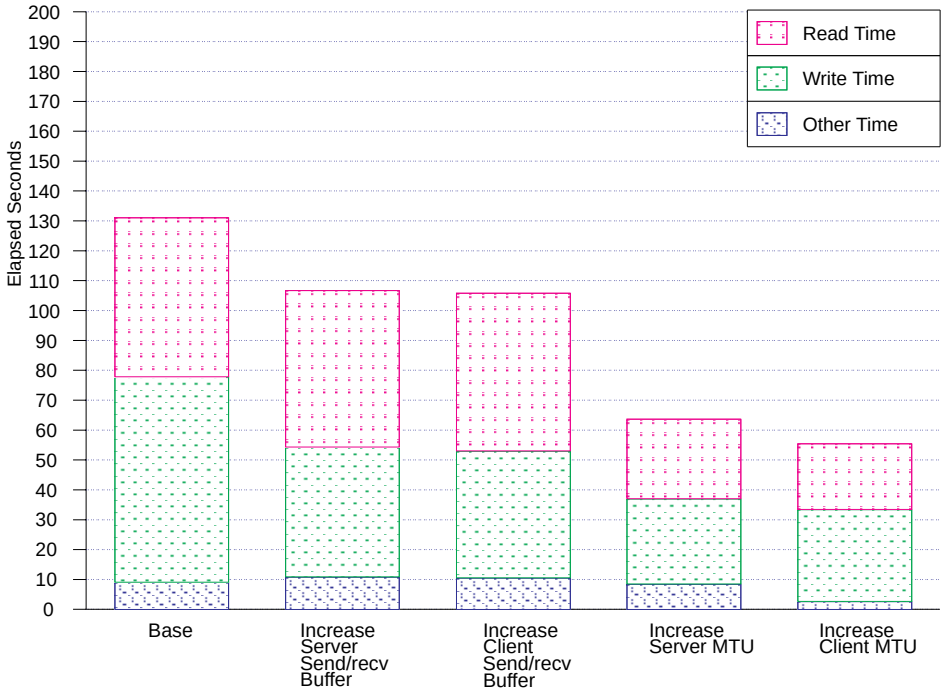


Figure 10. Where the Time Is Spent During the Transfer of Binary Files

Figure 11 shows amount of time spent reading, writing, and completing other processes (including translation) with the transfer of an ASCII file.

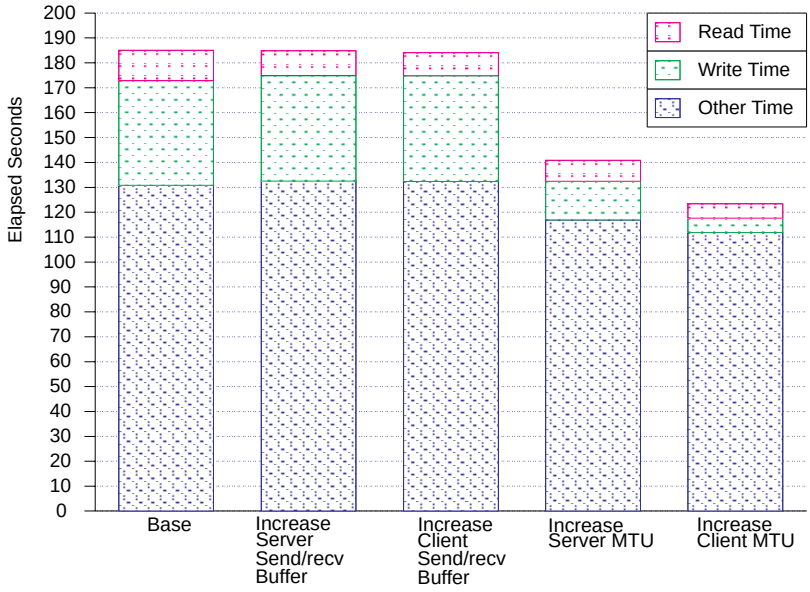


Figure 11. Where the Time Is Spent During the Transfer of ASCII Files

- **Source and destination file system characteristics.** Larger block size is faster than smaller; fixed record size is faster than variable.
- **File size.** Smaller files take less time per file than large files; larger files take less time per byte than small files.

  Startup time is fairly constant. Therefore the percentage of time spent on startup during a data transfer for a small file is more than for a large file, as shown in Figure 12.

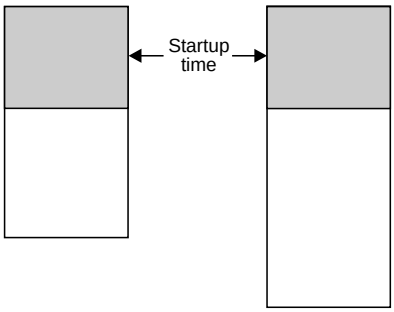


Figure 12. Startup Time Ratio

Possible Impact on Performance:

1. Which machines are servers and which are clients (some might be both)?
2. How often are your users transferring files (per minute/hour/day)?
3. What is the average size of a file that is transferred?
4. What are the file system characteristics of the client?
5. What are the file system characteristics of the server?
6. Do the files need to be translated? (It might be easier to ask if the files are executable programs, meaning they are binary, or if the files are documents or text, which might require translation.)
7. How critical is throughput? Is someone waiting for the transfer to complete?
8. Where is the usual source/destination?

On MVS and VM systems, you might also want to find out how many FTP servers are active and what their names are. On MVS, whether you are using the Pascal or C server may affect performance.

Using Telnet

Telnet is used to allow a user on one host to login to another host. It provides terminal emulation, which might include ASCII-to-EBCDIC character translation. You can safely assume that response time is critical for Telnet users.

Possible Impact on Performance:

1. Which machines are servers and which are clients (some might be both)?
2. How many connections need to be supported by each server (this will vary from one host to another)?
3. What terminal type will be emulated for each connection?
4. What is the approximate transaction rate that the server needs to support?
5. How much data is sent and received for each transaction?

Using SMTP

SMTP is used to provide electronic mail delivery. Reasonable response time is always required; however, it is often not as critical as FTP, Telnet, and NFS** throughput.

Possible Impact on Performance:

1. What is the volume of mail that gets sent and received on the network?
2. Where does the SMTP server reside?
3. Are there predictable, high-priority requests for mail delivery?
4. Is mail exchanged with another network outside your control?

Using NDB

Network DataBase (NDB) system allows access to DB2* or SQL/DS* databases using SQL.

Possible Impact on Performance:

1. Which databases are accessed using NDB?
2. Which machines are clients and which are servers?
3. Who is the database administrator for each of the databases?
4. What is the expected transaction rate to each database?
5. Does each database administrator have performance guidelines or requirements for the database users?

Using NFS

Network File System (NFS) is a distributed file system that allows a set of computers to cooperatively access each other's files in a way that is not apparent to users. Unlike FTP, NFS sends data records, not files or data sets.

Possible Impact on Performance:

1. Which machines are servers and which are clients (some might be both)?
2. How often are your users accessing files (per minute/hour/day)?
3. How many remote file systems are mounted and active?
4. Are most files being read, written, or both?
5. What is the average size of a file that is accessed?
6. What are the file system characteristics of the client?
7. What are the file system characteristics of the server?
8. Do the files need to be translated? (It might be easier to ask if the files are executable programs, meaning they are binary, or if the files are documents or text, which might require translation.)
9. Where is the usual source/destination?

Using NCS

Network Computing System** (NCS**) is a programmer tool kit that allows programmers to distribute processing power to other hosts.

Possible Impact on Performance:

1. What NCS applications are being used?
2. Who wrote the applications?
3. Who maintains the applications?

4. Which machines are servers and which are clients (some might be both)?
5. What are the performance requirements for the applications?

Using SNMP

Simple Network Management Protocol (SNMP) provides a means for managing an internet environment by elements, such as gateways, routers, and hosts. Network elements act as servers and contain management agents, which perform the management functions requested. Network management stations act as clients; they run the management applications, which monitor and control the network. SNMP provides a means of communicating between these elements and stations to send and receive information about network resources.

SNMP does not specify details such as how often data should be collected. That will depend on the application. Applications that exchange data more frequently than others will put a higher load on both the CPU and the network.

Some of the information items collected by SNMP applications are performance-related; others can be used by applications to compute performance statistics such as kilobytes transferred per second over some specified period of time.

Possible Impact on Performance:

1. What SNMP application are you running?
2. Which machines are your network management stations?
3. Which machines are agents?
4. How often will the network management station request data from the agents?
5. How much data will be sent in response to each request?
6. Does the SNMP application have performance requirements?
7. Will the SNMP application provide you with reliable and useful performance statistics?

Using Socket Applications

Socket interfaces allow you to write your own applications to communicate using TCP/IP. Data can be transmitted and received simultaneously from any other network device using sockets. There are also general-purpose APIs for languages such as Pascal, C, and assembler and for protocols including TCP, IP, and UDP.

Possible Impact on Performance:

1. Which machines are servers and which are clients (some might be both)?
2. What are the performance requirements for each?
3. How many connections will be active at any given time?
4. Can you influence the development of new applications in order to try to meet network and system performance objectives, for example, by using options such as UDP bulkmode on MVS and VM?

Using DNS

Domain Name System (DNS) uses a hierarchical system for naming hosts. A name server can be used to provide mapping from host names to IP addresses. Performance of the name server can affect the performance of other applications if those applications refer to other hosts by name instead of IP address. It is usually a good idea to use the Domain Name System and a name server; however, it is important to keep the host that the name server is running on available and well-tuned.

When running performance benchmarks it might be desirable to stop using the name server and use a HOSTS file or IP address instead, in order to exclude the delays or overhead associated with the name server. Another time this would be appropriate is when your system is experiencing severe performance degradation and you are trying to resolve the problem. If you stop using the name server before you start issuing commands that refer to other hosts, you will avoid more delays from the degraded network.

Possible Impact on Performance:

1. Which machine is running the name server?

Using NPF

Network Print Facility (NPF) lets you route VTAM, CICS, IMS, JES2, or JES3 printer output to local and remote printers supported by TCP/IP.

You use the NPF routing file to specify local and remote printer parameters and destinations. The NPF options file is used to specify the LPR print options used during printing.

NPF also provides a queue manager to manage the NPF print data sets that print successfully or that require further processing. This function deletes data sets, retains data sets, retries operations if unsuccessful, and so on. For more information on NPF, see *TCP/IP for MVS: Network Print Facility*

Possible Impact on Performance:

1. How many NPF VTAM applications are you running?
2. How many NPF threads do you have for each NPF VTAM or JES application?
3. What is the size of the files you are printing?
4. How many NPF print transactions or jobs are you submitting per minute?
5. How many logical printers are you using?
6. How many print jobs are queued on a particular printer at one time?
7. How fast are your printers?
8. How long are files retained?
9. How many retries have you specified in the NPF routing parms?
10. How many FSS writers have you specified for JES?

Using IP PrintWay

IP PrintWay has numerous advantages over NPF. It excels in virtually all areas of performance, usability, capacity, reliability, and function. NPF customers currently constrained by LPR port limitations should benefit from IP PrintWay. For additional information on IP PrintWay, see *IP PrintWay: Automatic Network Printing Using MVS TCP/IP*, 296-328.

What Hardware Do You Use?

This second part of the chapter describes some of the hardware that you might be using and how it might affect the performance of TCP/IP.

What Hosts Do You Use?

What are the machine types of the hosts that you are tuning? The machines used for the performance testing to support this book are listed in “Hardware Used in our Telnet Benchmarks” on page 136 and “Hardware Used in Our FTP Benchmarks” on page 146.

Some questions to know the answers to are:

- What is the relative power of the central processing unit? For example:
  1. Speed rating
  2. Architecture
  3. Number of CPUs
- How much memory (main storage) does the system have?
- How much disk space and how many paths to the disks does it have?

Any of these three items, if constrained, can affect the performance of TCP/IP. If possible, find out if other subsystems or applications are already creating a constraint that TCP/IP will need to adapt to when you are tuning it.

What Type of Media Do You Use?

What media and speed are you using on your network? Some examples are:

- Token Ring
- Ethernet
- Fiber Distribution Data Interface (FDDI)
- X.25
- Parallel channel
- ESCON* channel

For example, Ethernet has collision detection which affects the timing of the packets. As a result, you cannot get 100% utilization on an Ethernet or there would be too many lost packets due to collisions. At most, an Ethernet should run at 70% utilization and will saturate before the host.

Workstation Hardware

This section covers communication hardware necessary for the workstation.

What Type of Adapter Do You Use?

What adapters are you using to access the network? For some hosts, the hardware interface to the network is an adapter card. For example on a RISC System/6000* computer, you might have a High Performance Token Ring Adapter, a FDDI adapter, an Ethernet Adapter or some combination of the three, as shown in Figure 13.

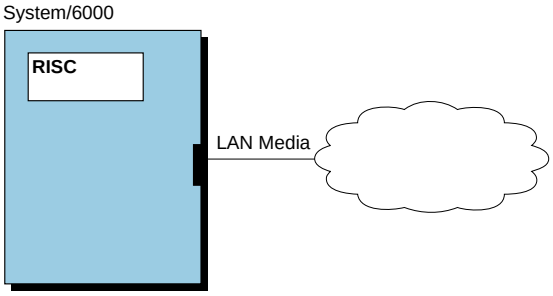


Figure 13. Example of a LAN Adapter in a Workstation

It is important to know, if possible, the limitations of the adapters that you are using. Knowing the maximum packet size is vital to tuning TCP/IP. Be aware that although the limit of the Token Ring might be 18,000 bytes, the adapter might only support 4000 bytes.

Some examples of maximum packet size are:

- Ethernet 802.3 at 1492 bytes
- Ethernet Version 2 IEEE at 1500 bytes
- Frame Relay at 2048 bytes
- FDDI at 4352 bytes
- Channel-to-channel (CTC) at 65 527 bytes

LAN and Workstation Adapters

Your TCP/IP performance will vary depending on the LAN adapter you are using to communicate over the network. The adapters used in workstations are usually for Token Ring, Ethernet, or FDDI.

There are various types of adapters available. Each adapter has parameters that you can set. The range and the default value for each parameter is usually provided in a network information file (NIF) supplied with the adapter. The values to which you set these parameters can have an impact on performance.

Device drivers can also affect performance. A device driver is the program by which an application communicates with an adapter. Some examples are: NDIS, ODI, and ASI.

SLIP: Serial Line Interface Protocol (SLIP) enables you to connect two workstation hosts over a serial line. The serial line sets up point-to-point links. The connection uses a telephone line through a modem or over a serial line using a null-modem cable. SLIP supports speeds from 1200 to 38,400 bps (bits per second).

Any valid workstation serial communication port (COM port) can be used. With TCP/IP for OS/2, you can use up to 8 ports at a time. With TCP/IP for DOS, you can use only one COM port at a time. Therefore, if you have one established SLIP connection, you cannot accept or originate another SLIP connection.

System/390* and System/370* Hardware

This section covers communication hardware necessary for the mainframe computer.

What Type of Controller Do You Use?

Some of the products designed to connect the System/390 or System/370 host to the network include:

- IBM 3172-3 Interconnect Controller
- IBM RISC System/6000 computer used as a controller
- IBM Channel-to-Channel (CTC) Adapter
- IBM 8232 LAN Channel Station
- IBM 37XX Communication Controller using channel DLC protocol, SNALINK or X.25
- High Performance Parallel Interface (HIPPI)
- Network Systems Corporation HYPERchannel** A220 Processor Adapter
- Network attachment device using the Continuously Executing Transfer Interface (CETI)

(There are other commercially available products not mentioned here.)

The System/390 or System/370 hosts running MVS or VM perform I/O operations using special-purpose processors called channels. Channels connect to another type of special-purpose processor called a control unit or they can connect to an adapter in another host.

We will discuss three of the possible ways to connect the mainframe to the network:

Using the 3172-3 Interconnect Controller: One way these hosts can connect to the network is through an IBM 3172-3 Interconnect Controller, which contains hardware for the connection to the channel as well as adapters to connect to the network using, for example, Token Ring, Ethernet, or FDDI adapters, as shown in Figure 14.

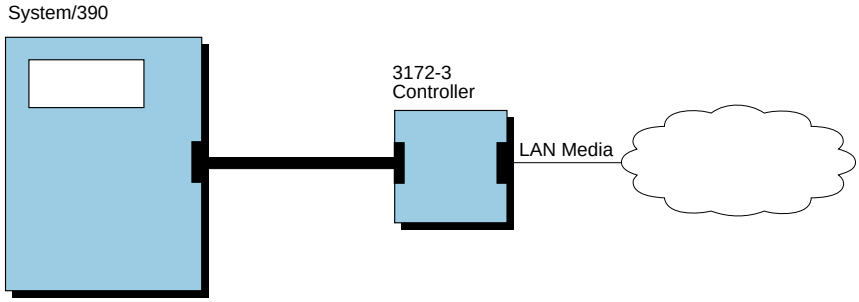


Figure 14. Example of a System/390 Connection to LAN via 3172-3

Using the RISC System/6000 Computer As a Controller: Another way the System/390 or System/370 host can connect to the network is through a RISC System/6000 host that has been equipped with a channel adapter. The System/390 or System/370 host and the RISC System/6000 host can then communicate with each other using TCP/IP; however, then the RISC System/6000 host will most likely also have Token Ring, Ethernet, or FDDI adapters, through which the System/390 host can communicate with other hosts on the LAN, as shown in Figure 15. In this case, the RISC System/6000 computer is performing the functions that the 3172-3 controller performed in the case above.

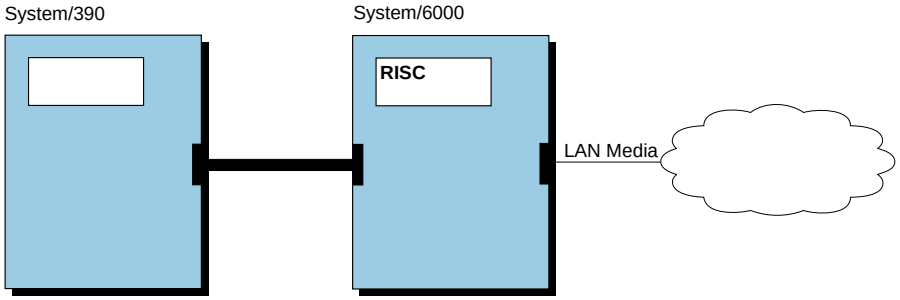


Figure 15. Example of a System/390 Connection to LAN via a RISC System/6000 Computer

Using the Channel-to-Channel Adapter: A third way the System/390 or System/370 host can connect to the network is through a Channel-to-Channel adapter connection to another System/390 or System/370 host. Again, this would primarily allow the two hosts to communicate with each other using TCP/IP; however, if either host has another network connection (such as a 3172-3 or a channel-attached RISC System/6000 host), then the other host can access the hosts on that network via the intermediate System/390 or System/370 host, as shown in Figure 16.

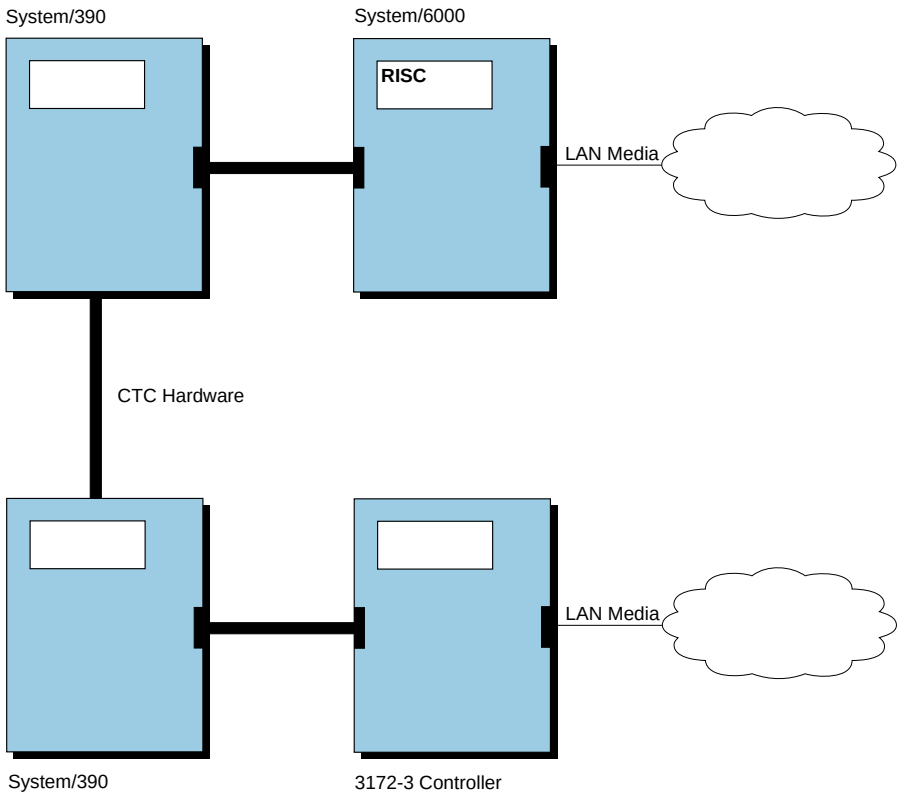


Figure 16. Example of a Channel-to-Channel Connection between System/390 Hosts

What Type of Channel Do You Use?

There are two types of channels:

- Parallel channel at a speed of up to 4.5MBps
- ESCON channel at a speed of up to 18MBps

Parallel channels use copper-wire-based technology. They use multiple copper wires sending signals in parallel.

ESCON channels use serial, fiber-based technology. They use single fiber issuing signals in a series. ESCON channels provide higher potential throughput, require less CPU processing by the System/390 host, and permit control units to be installed farther away from the host. All System/370 and System/390 hosts support parallel channels, but only newer processors support ESCON channels.

The 3172-3, RISC System/6000 computer, and CTC adapter can be used with either an ESCON or parallel channel:

- 3172-3s can be purchased with either ESCON or parallel channel connections, plus various LAN adapters.
- Both ESCON and parallel channel adapters can be purchased for the RISC System/6000 computer (certain models only).
- CTC connections can use ESCON or parallel channels, although they must be the same on both ends of any one connection.

What Type of Controller Software Do You Use?

The 3172-3 Interconnect Controller has two modes of operation with TCP/IP:

1. In the first mode, the software in the 3172-3 is the IBM Interconnect Controller Program (ICP) (5601-400 or 5601-433).
2. In the second mode, the software in the 3172-3 is the OS/2 operating system with the Offload Feature for TCP/IP for MVS or VM. The Offload feature moves some TCP/IP processing from the System/390 or System/370 host to the 3172-3, reducing the host CPU requirements for TCP/IP. The trade-off for using Offload is a reduction in throughput. It is important that you realize that you will be offloading CPU cycles at the cost of throughput. There is also a limit of 2040 total connections for a System/390 or System/370 host using one Offload device.

Identify Your Critical Resources

What are you going to do with all this information? Once you have answered all the questions that pertained to your TCP/IP environment, what is next?

You might want to make a record of all the information you take in. For example, you might want to make a table like Table 1 for each host you need to tune, marking under critical resources if there is a constraint (—) or surplus (+):

Table 1. Hosts to Tune and Their Critical Resources

Type of host	Throughput	Response Time	CPU Utilization	Storage	Disk access
OS/2	–	+	–	–	–
MVS	–	+	–	+	+
AIX	+	–	+	–	–

When talking about work load, it is good to know the answers to the following questions:

- What part of the work load is most important?
- What is the bulk of the work?
- What part of the work load is least important?

Chapter 3. Establish Performance Objectives

Now that you know what your critical resources are, what are reasonable performance objectives?

How you define good performance depends on your particular needs and priorities. Performance objectives should be realistic, in line with your budget, understandable, and measurable.

Before you put time into tuning the performance of how TCP/IP runs, you should have already spent time tuning your operating system.

Transaction Rate and Throughput

Transaction rate is defined as the number of units of work accomplished in a given amount of time. It is often looked at when planning a network. Does your application need to complete a certain amount of work each day, in an hour, or every Monday morning?

The transaction rate of your installation of TCP/IP is important if your applications use sockets, NDB, or to some degree, Telnet. For example, transaction rate is a important if some of the orders received by your company cannot be processed each day and business is lost.

Throughput is defined as the amount of data received in a given amount of time. It is a common concern when the applications used include FTP or NFS.

When the throughput is unusually low, there can be unacceptable delays before data is delivered. Resources might be unused while people or applications are waiting for work to do.

Response Time

Response time is the amount of time a user waits between pressing a key and the response from the system.

When judging response time, it is helpful to separate the execution and travel time using TCP/IP from the execution time of the application. TCP/IP affects the travel part of the response time and only part of the execution time.

The idea when tuning TCP/IP for response time is to put the thinnest layer of TCP/IP or Telnet time between the user and the host. Although it can be hard to measure where the TCP/IP execution time ends and the application execution time begins, it can be done.

Research has proven that there is a serious effect on productivity whenever the response time is greater than 1/2 second. You might have other response time requirements, like 3 or 6 seconds.

An example of a situation where response time is a concern is when customers call in to place catalog orders using an 800 number. The length of response time as the sales representative moves through the application, keying in the order, must

be balanced against the cost of the telephone call. If there are long waits when the sales representative is getting or sending data as part of the data entry application, the cost of the call goes up.

CPU Utilization

CPU utilization is the amount of CPU busy time that a system takes to process work. You might think of CPU utilization in terms of path length, CPU busy time, system overhead, processor utilization, percent busy, or MIPS. A computer can only do a certain number of units of work per second.

The goal when you are tuning for the lowest CPU utilization is to decrease the amount of overhead caused by TCP/IP on top of other applications. You want to make sure that the system is spending most of its time on the user applications and not on processing TCP/IP commands.

The impact of running out of CPU as a resource is longer response time overall and less system capacity. Options when you are in this situation include upgrading hardware, which can be expensive, or moving some of the processing to another machine, such as 3172 Offload for OS/2 if you are running TCP/IP on MVS or VM.

Offload is a TCP/IP feature that can be used to move the TCP/IP execution to a 3172-3. As shown in Figure 17, using Offload reduces CPU busy time for FTP.

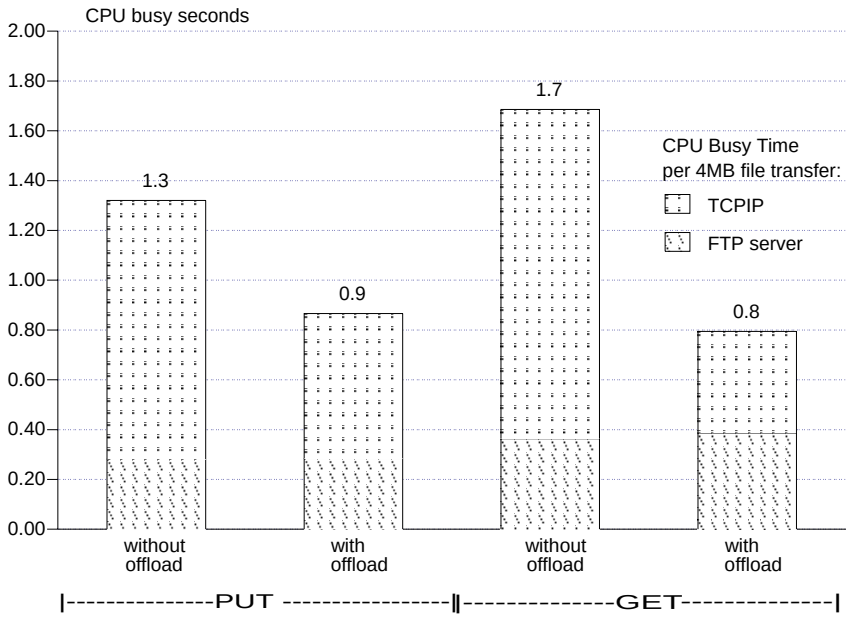


Figure 17. 3172 Offload for OS/2 Reduces CPU Utilization

Storage Consumption

Storage or active memory is the work space of the computer, where data and instructions are buffered as needed. With MVS and VM, if TCP/IP is affecting your system performance by taking so much storage, you will want to lower the impact of running TCP/IP.

Your options when you are running out of storage include buying more memory or changing the way data is buffered. Data buffers too large in size or too many data buffers can cause storage contention. The effect of storage contention is paging. Paging costs in CPU seconds and I/O delays while the paging is being done. Choosing the right amount and size of buffers is the key to tuning TCP/IP if improving storage consumption is your main objective. With TCP/IP on VM for example, virtual storage must increase by the same amount as the amount your data buffers increase.

Disk I/O Rate

The disk I/O rate is the speed at which the computer is able to read and write data to and from disk storage. When too much time is spent on disk I/O, the CPU utilization goes down because the system has to wait for the I/O to be done, and response time goes up.

Your options for improving the disk I/O rate when using TCP/IP include:

- File organization (for example, high performance file system (HPFS) versus file allocation table (FAT) when using OS/2)
- Speed of the disk (for example, access time of 10 versus 100 milliseconds has a large impact)
- Multiple disks, channels, and controllers on a multi-user system
- Cached controllers (VM and MVS)
- Disk caching options (for example, minidisk cache (MDC) on the VM/ESA* operating system)
- Buffer size (more data per disk I/O might improve TCP/IP performance)

Be careful when choosing your disk I/O rate objectives. Sometimes the host itself has disk I/O constraints that cannot be improved.

You should also be aware of the limits of your file system configuration. For example, using AIX you cannot currently save a file larger than 2 gigabytes.

Trade-Offs in the Cost of Performance

Your performance objectives are defined by the:

- Applications you are running
- Resources available
- Needs of your users

You need to know which applications and users are most important. For example, your objectives will be different if you are using Telnet more than FTP. Do you want to focus on the amount of work being done by the system or how long a user has to wait for a response?

The next decision you need to make is what trade-offs you are prepared to make in the area of performance. Where can you afford to sacrifice?

Are you looking at performance tuning because you do not have enough throughput or because response time is too slow? Or are you studying tuning because you need to know “how fast this thing will go?” Or are you trying to find out how to get

the best performance given your current system without buying any new hardware? (Sometimes fixing the system by buying more hardware is easier than getting along with what you have.) Or is TCP/IP being accused of exhausting resources (CPU utilization, disk I/O, storage, LAN storage, etc.), impacting someone else's applications?

For each application or function, have the users provide you with their goals. The users might not be able to quantify the target range for response time, but they might know how many transactions an hour they need. Even so, they might be able to rank goals in order of importance to them, along with which applications they use the most and which ones they rarely use.

You need to understand the limitations of your current system and network (CPU, storage, disk I/O, and communication devices). You will usually know if you have reached the limits of your hardware. TCP/IP is rarely going to put you over the edge of those limits. You also need to find out if there are any artificial system constraints that you have to work within. You might have been allocated a subset of the CPU utilization or disk storage for example.

Limit your performance goals to the size of the network where most of your processing is done. Keep in mind that if you are not trying to send and receive information with the whole world, then you should not try to tune TCP/IP as though you were working with the whole world.

It is also important to make sure you balance the amount of time you spend tuning the performance of TCP/IP with the possible payoff. If you can quantify this in terms of dollars, it will be easier for you to see if the effort and time will be worthwhile. Tuning performance is somewhat like bringing your car into the shop to be tuned up. The differences in performance the mechanic makes are not always huge, although the cost might be high.

It is just as important to know which resources are not constrained as it is to know which are limited. You need to know what your goal is or what you need to fix. If you do not think that you are constrained on a particular resource, you have the opportunity to trade-off that resource for the one you are short of.

An example of a trade-off is when using 3172 Offload for OS/2. While 3172 Offload for OS/2 will help move CPU cycles off a mainframe, decreasing its CPU utilization, Offload has been shown to reduce the throughput and increase response time of TCP/IP. While Figure 17 on page 28 showed the decreased CPU utilization possible with 3172 Offload for OS/2, Figure 18 shows a bar chart example of the change in throughput when using 3172 Offload for OS/2.

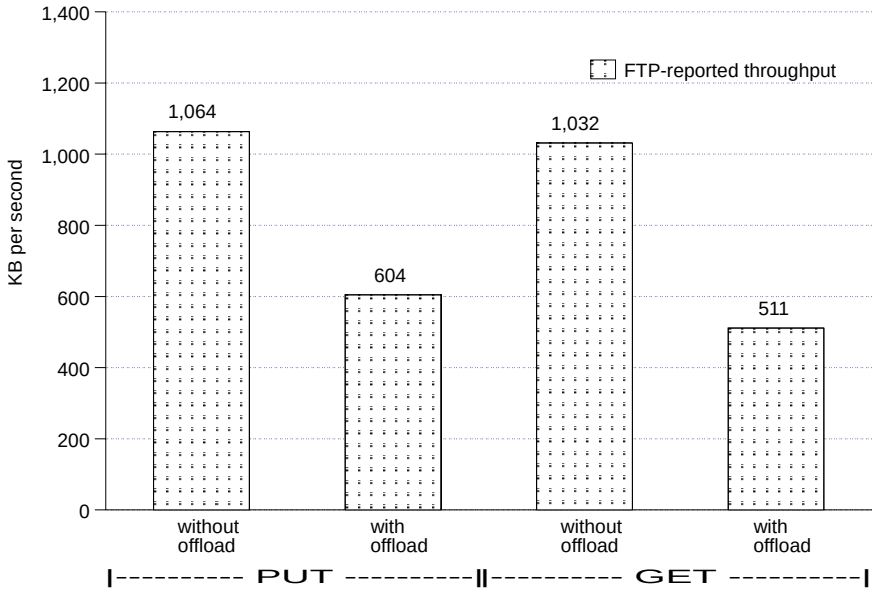
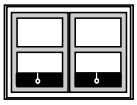


Figure 18. 3172 Offload for OS/2 Reduces Throughput

Changing the Send and Receive Buffer Size



You can make an impact on performance by improving the efficiency of what you are already using. For example, you could change the size of the send and receive buffers TCP/IP is using. (For an introduction to this topic, see “Understanding TCP/IP Send and Receive Buffers” on page 6.) Although all versions of TCP/IP use send and receive buffers, the size of the buffers varies by host and implementation.

Usually you can increase the send and receive buffer size. With an increased buffer size, you will have better throughput when you are using large data moves (such as with FTP) because the data buffer is bigger. Since the cost of handling a buffer is not necessarily going to change whether the buffer is large or small, you will probably also have lower CPU path length. A bigger buffer will also allow the sending machine to send data faster, but it is important to be able to receive it as quickly so that there will not be a bottleneck.

The trade-off is that both real and virtual storage will be impacted. If you increase the size of the buffers and decrease the amount of buffers, you are constraining the number of applications that can use a buffer at a given time.

Increasing the send and receive buffer size can also lead to waste if most of the transactions do not take advantage of the larger buffer size. Therefore a larger buffer size is not the best choice when most of your users are sending small amounts of data most of the time, and only occasionally sending large amounts.

Figure 19 shows a bar chart example of what happens to FTP throughput when you change the data buffers between 8KB and 32KB on MVS.

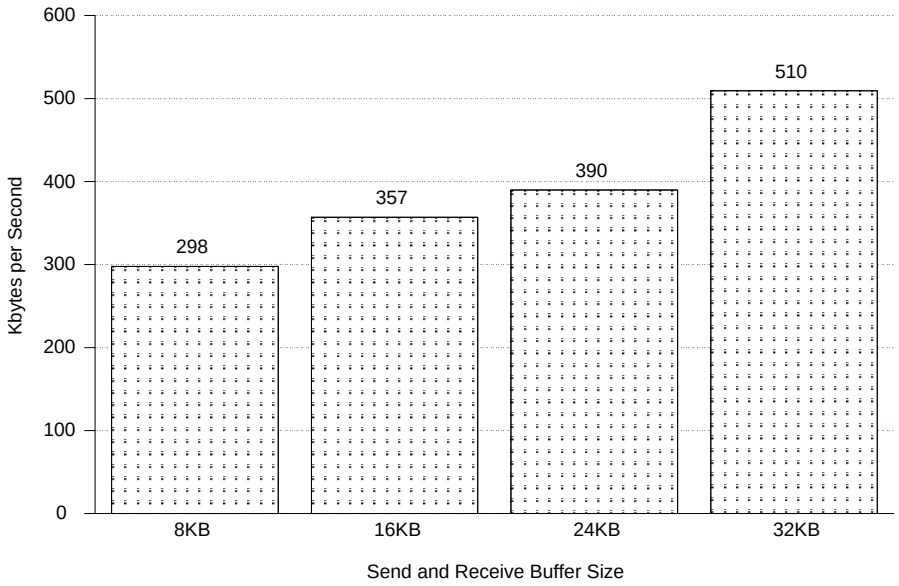
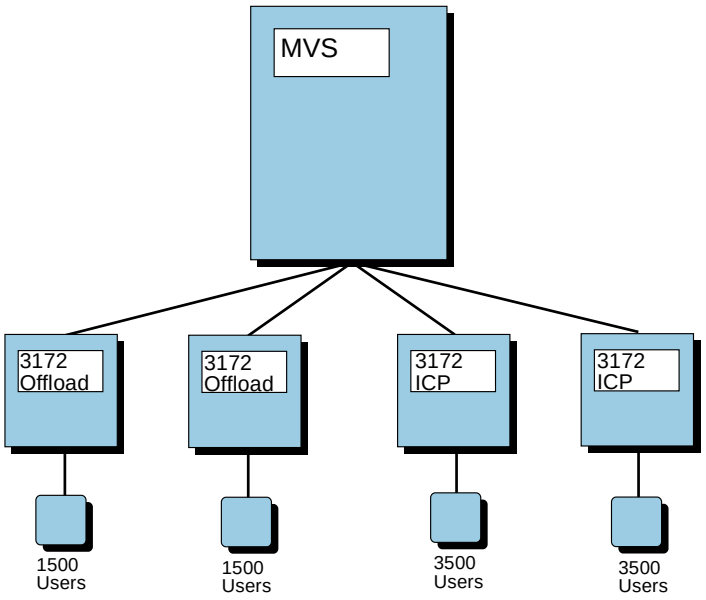


Figure 19. Larger Data Buffers Increase Throughput

Also, if you decide to change the send and receive size and you exchange data most often with one particular machine, make sure you also change the buffer size at the other end (if you have influence over it).

Changing the Packet Size



As part of performance tuning, you can alter the MTU or the packet size according to your needs. (For an introduction to packet size, see “Understanding Fragmentation” on page 10.) If you know the packets will not be further broken down between the sender and the receiver, you can send them at the maximum size for that known route. For example, if the majority of the packets you are sending using TCP/IP are going via channel-to-channel connection to one specific machine, you should send packets of 65 527 bytes.

On the other hand, you might not have a definite idea of how many network interfaces will have to pass on the data before it reaches the receiver. It might even be going through the big Internet. If you do not know how the datagram will get from you to the receiver, you should experiment with MTU to find the optimum size. You might find that the best size for your users is the minimum amount of 576 bytes. However, if it is a reliable route, fragmentation is not much of a problem and you could use a larger MTU.

Here is a scenario to illustrate how the network affects fragmentation. Suppose you send out packets of 5000 bytes across the network. They must first cross a FDDI network. The FDDI network has a packet size of 4352 bytes so it will have to break down the packets into fragments, adding another header to the original one so that the receiver can put the pieces back together when it gets all the packets.

Then the data crosses an Ethernet. The Ethernet will have to break down the packets into smaller fragments since it has a packet size of 1500, adding yet another header onto the data, so that the percentage of data to header in the packet grows smaller and smaller each time it crosses into another type of network.

The receiver then spends even more time receiving the data because the number of packets has increased and then more time to piece the fragments together. The sender will also have more acknowledgements to receive and sort out.

To summarize, when you are sending small packets and there is no fragmentation, then you have:

- More packets to send, so a higher cost for the same amount of data
- Increased number of I/O operations to the network (which adds overhead)
- Increased TCP/IP path length
- Lower throughput

A bigger MTU increases throughput in most cases. When you are sending large packets and there is no fragmentation, then you have:

- Fewer packets to send, so a lower cost for the same amount of data
- Fewer I/O operations to the network (which lowers overhead)
- Higher throughput

When you are sending large packets that become fragmented, then you have:

- More packets to send overall than if the MTU were smaller, so a higher cost for the same amount of data
- Increased number of I/O operations to the network (which adds overhead)
- Lower throughput
- More packets must be sent to recover if data packets are lost

If you look only at the throughput rate for a packet, you might be led to incorrectly conclude that a smaller datagram is better than a large datagram. Figure 20 shows the throughput results when sending varying amounts of data with PING.

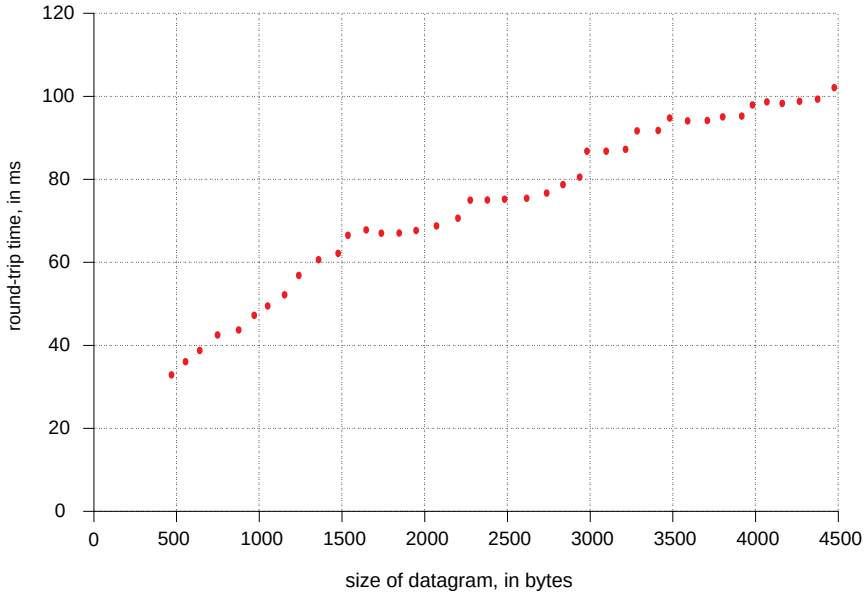


Figure 20. Throughput Time Results from PING

If you then take the information for any packet size and calculate the amount of time required to send 16KB of datagrams using each specific size, the results are likely to look like Figure 21.

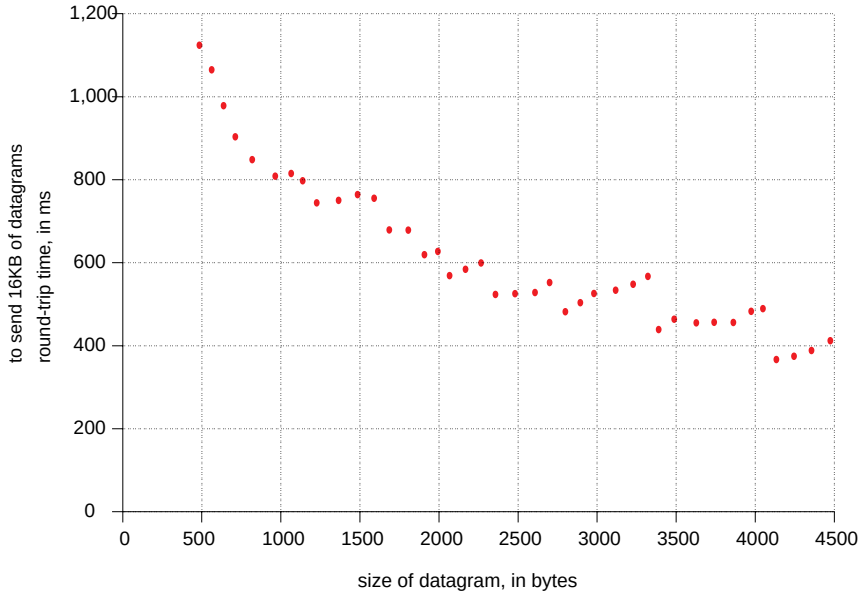


Figure 21. Time Spent Sending 16KB of Datagrams Using PING

As a result, the benefit of using the larger datagram is obvious.

Figure 22 shows a bar chart example of what happens to MVS CPU overhead for a FTP file transfer when you change the MTU between 576 and 2000 bytes. The CPU cost to transfer the file is lower for the larger MTU.

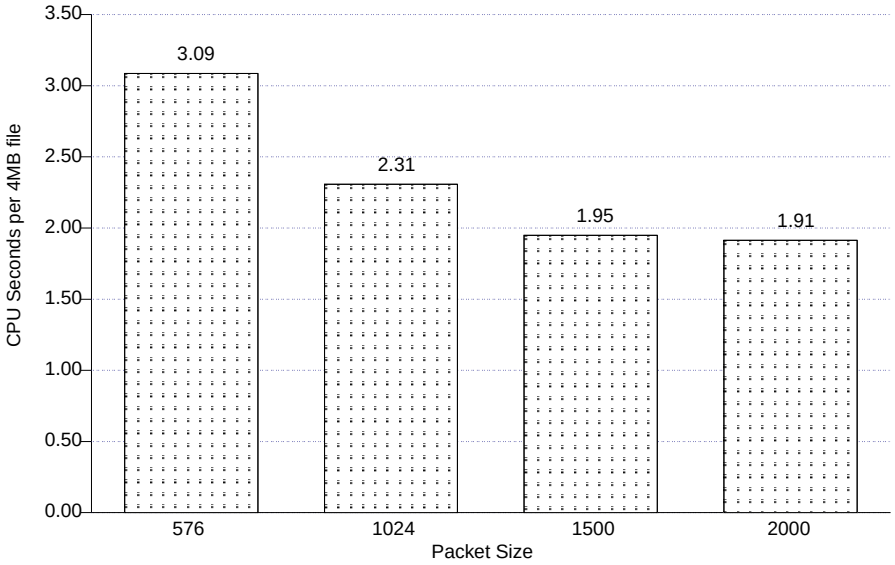


Figure 22. Larger MTU Decreases CPU Busy Time

As with window size, if you decide to change the packet size and you exchange data most often with one particular machine, make sure you also change the packet size at the other end (if you have influence over it).

Chapter 4. Monitor Performance

Now that you have set your performance objectives, you need to take a practical look at what you can measure.

The first rule of tuning is:

If you can't measure it, you can't control it.

Performance measurements are relative: they tell how a system behaves for a particular work load. A system is considered to perform well if it can complete a particular work load faster with fewer resources.

In a test system, you can control the work load by running the same tasks many times. During each iteration you can measure how fast your system completed the tasks and how much of a particular resource it used.

However, in a production system, it is difficult to compare measurements taken at different times, because the work load is constantly changing. To obtain a performance measurement, you must compare the ***average*** performance of the system measured over a period of time to the work load it processed during that time.

You need to plan how you will monitor your TCP/IP environment and how you will analyze the data that results. First, determine the kinds of analysis that you will perform and the tools that you will use. Then document the data you have extracted from each monitoring tool. Some of these tools will provide reports that help to organize data, but in addition, it is helpful to create work sheets to help you extract and organize the performance indicators that are specific to your environment.

Take into account that the monitor you choose might have an impact on the system you are testing. Figure 23 shows the influence that the AIX netpmon monitor can have on total CPU time.

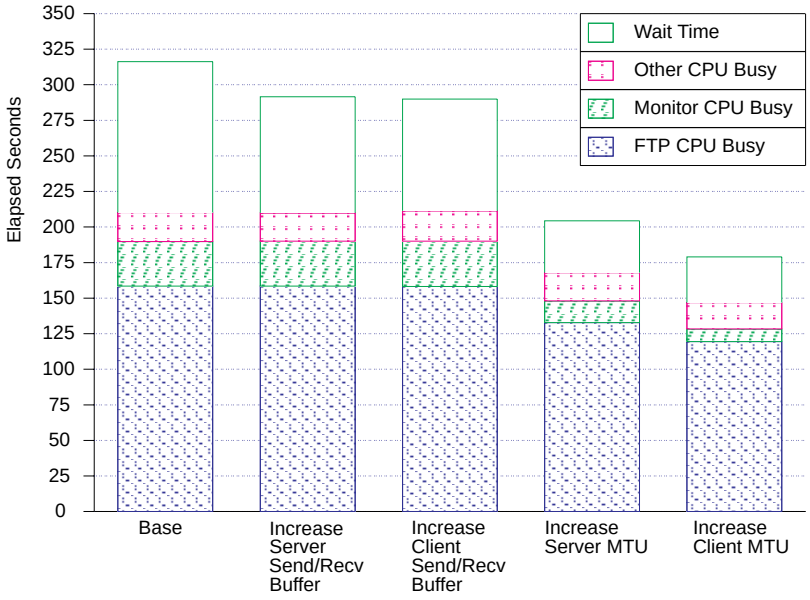


Figure 23. Impact of the netpmon Monitor on Total CPU Time

Disclaimer: No attempt has been made to fully explain how to use each tool, only how it pertains to TCP/IP performance tuning.

Network Tools

The tools listed in this section for monitoring your network are:

- PING
- DatagLANce* Network Analyzer
- NETSTAT (described later in this chapter under each operating system)

Be aware that there are many other commercially available products for network monitoring that are not listed here.

PING

Packet InterNet Groper (PING) sends an echo to a host to determine if the host is accessible.

The round-trip time is only as accurate as the clock accuracy on your host. For example, on OS/2, the clock increments in 32 millisecond jumps, so it is only accurate to 32 milliseconds.

Using PING on Different Operating Systems

This section shows the use of PING to send a 462-byte packet 5 times on the different operating systems.

- Figure 24 shows an AIX example
- Figure 25 shows an DOS example
- Figure 26 shows an MVS example
- Figure 28 shows an VM example
- Figure 27 shows an OS/2 example

On the AIX Operating System

```
perf@Wimpy!8% ping -c5 -s462 9.67.238.3
PING 9.67.238.3: (9.67.113.1): 462 data bytes
470 bytes from 9.67.113.1: icmp_seq=0 ttl=255 time=80 ms
470 bytes from 9.67.113.1: icmp_seq=1 ttl=255 time=60 ms
470 bytes from 9.67.113.1: icmp_seq=2 ttl=255 time=49 ms
470 bytes from 9.67.113.1: icmp_seq=3 ttl=255 time=61 ms
470 bytes from 9.67.113.1: icmp_seq=4 ttl=255 time=59 ms
----9.67.238.3 PING Statistics----
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 49/62/80 ms
```

Figure 24. Example of Using PING to Send 462 Bytes 5 Times on the AIX Operating System

On the DOS Operating System

```
C:\>ping 9.67.29.11 -c 5 -s 462
PING 9.67.29.11: 462 data bytes
470 bytes from 9.67.29.11: icmp_seq=0 ttl=59 time=60 ms
470 bytes from 9.67.29.11: icmp_seq=1 ttl=59 time=50 ms
470 bytes from 9.67.29.11: icmp_seq=2 ttl=59 time=60 ms
470 bytes from 9.67.29.11: icmp_seq=3 ttl=59 time=60 ms
470 bytes from 9.67.29.11: icmp_seq=4 ttl=59 time=0 ms
----9.67.29.11 PING Statistics----
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 0/46/60 ms
C:\>
```

Figure 25. Example of Using PING to Send 462 Bytes 5 Times on DOS

On the MVS Operating System

```
 READY
ping 9.67.238.3 (length 462 count 5
 TCPPNG015I Ping V2R2.1: Pinging host 9.67.238.3.  Use PA1 to interrupt.
 TCPPNG020I PING: Ping #1 response took 0.063 seconds. Successes so far 1.
 TCPPNG020I PING: Ping #2 response took 0.074 seconds. Successes so far 2.
 TCPPNG020I PING: Ping #3 response took 0.070 seconds. Successes so far 3.
 TCPPNG020I PING: Ping #4 response took 0.078 seconds. Successes so far 4.
 TCPPNG020I PING: Ping #5 response took 0.069 seconds. Successes so far 5.
 READY
```

Figure 26. Example of Using PING to Send 462 Bytes 5 Times on MVS

On the OS/2 Operating System

```
C:\“ping 9.67.238.3 462 5
PING 9.67.238.3: 462 data bytes
470 bytes from 9.67.238.3: icmp_seq=0. time=94. ms
470 bytes from 9.67.238.3: icmp_seq=1. time=62. ms
470 bytes from 9.67.238.3: icmp_seq=2. time=63. ms
470 bytes from 9.67.238.3: icmp_seq=3. time=63. ms
470 bytes from 9.67.238.3: icmp_seq=4. time=62. ms

----9.67.238.3 PING Statistics----
5 packets transmitted, 5 packets received, 0% packet loss
round-trip (ms)  min/avg/max = 62/68/94

C:\“
```

Figure 27. Example of Using PING to Send 462 Bytes 5 Times on the OS/2 Operating System

On the VM Operating System

```
ping 9.67.238.3 (length 462 count 5
Ping V2R2: Pinging host 9.67.238.3.  Enter #CP EXT to interrupt.
PING: Ping #1 response took 0.085 seconds. Successes so far 1.
PING: Ping #2 response took 0.054 seconds. Successes so far 2.
PING: Ping #3 response took 0.054 seconds. Successes so far 3.
PING: Ping #4 response took 0.083 seconds. Successes so far 4.
PING: Ping #5 response took 0.083 seconds. Successes so far 5.
Ready;
```

Figure 28. Example of Using PING to Send 462 Bytes 5 Times on VM

Using PING to Learn About MTU Size Along a Route

Part of the response you receive from PING is the round-trip time. By varying the amount of data you send with PING and by issuing PING to intermediate hosts, you can get an idea of the performance of the underlying networks.

To illustrate, we varied the amount of data sent with the PING command. By default, PING sends 56 bytes of data, along with an 8-byte header, for a total datagram of 64 bytes. In this example, the IP header added another 20 bytes to the packet. Figure 24 on page 38 shows an AIX example of varying the amount of data sent with PING.

We began by issuing PING to the IP address of our default route, 9.67.113.1. We recorded the minimum round-trip time returned of 49ms. Notice that 470 bytes (462 + 8) were sent. The packets sent also included the 20-byte IP header.

Next we increased the number of bytes from 562 to 1962 in increments of 100 bytes. The minimum round-trip time to increased as the amount of data increased. The increase in time was small because we used a fast LAN (16Mbps), but it was still observable. Figure 29 shows the results.

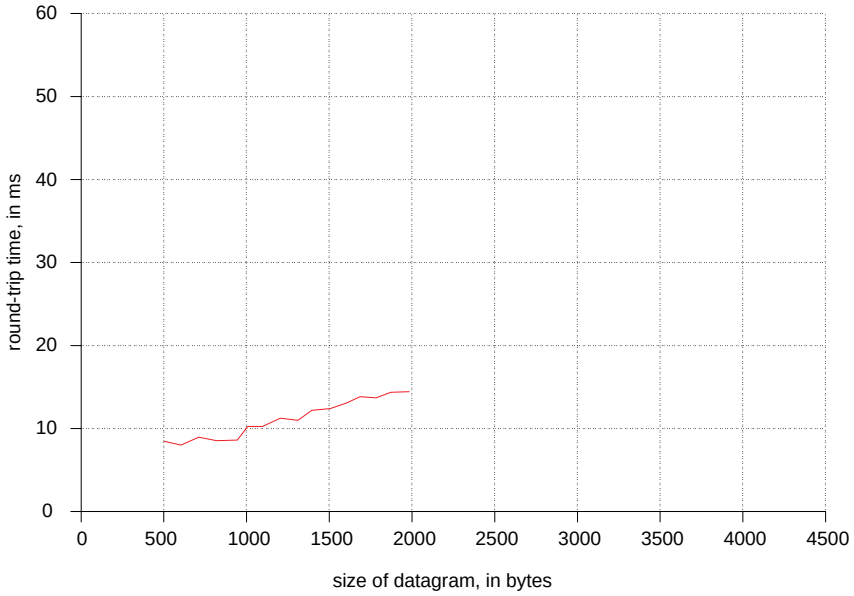


Figure 29. Using PING to Send Datagrams of 562 to 1962 Bytes

Then we increased the number of bytes by 100, sending 2062 data bytes and the 8-byte header, or 2070 bytes, plus the 20-byte IP header, for a total of 2090 bytes. Since our MTU is 2000 bytes, PING had to send the datagram in 2 packets, and the response was returned in 2 packets. This change shows up as a sharper increase in the round-trip time, accounting for additional delays in processing extra packets. Figure 30 shows the results.

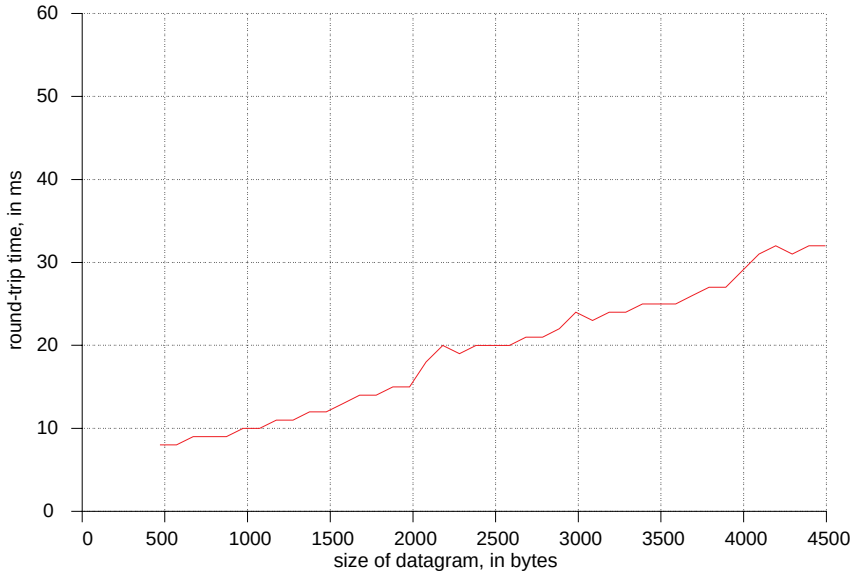


Figure 30. Using PING to Send Datagrams of 562 to 4470 Bytes

We then issued PING to an address beyond the default route, 9.67.96.1. Notice in Figure 31 the sharper increase in round-trip time between sending 1462 bytes and 1562 bytes.

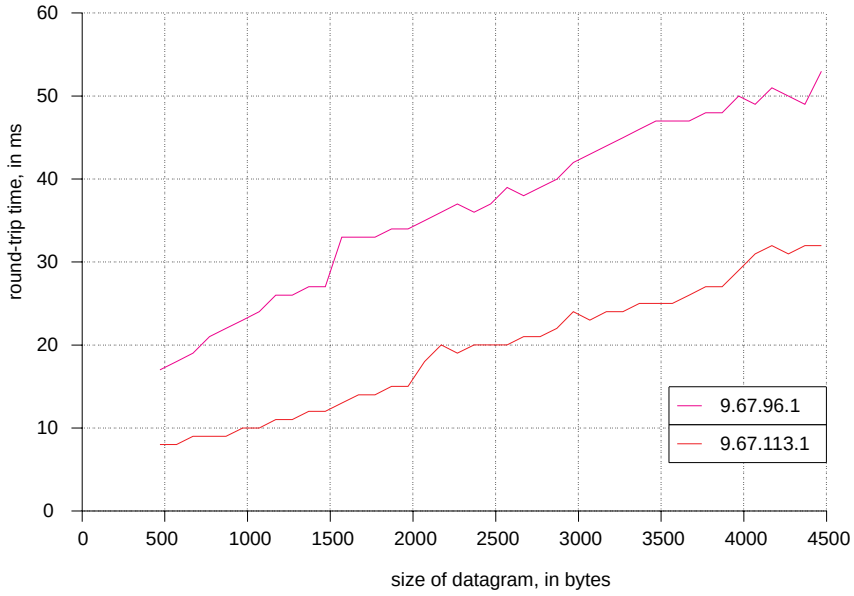


Figure 31. Using PING to Send Datagrams of 562 to 4470 Bytes to 2 Addresses

The sharp increase in time is an indication that an interface somewhere in the route to the destination host has an MTU approximately 1500 bytes. In this case, PING sent the datagram as 1 packet, but the packet was fragmented because of the smaller MTU, so the response was returned as 2 packets. This fragmentation effect can become more pronounced as the number of intermediate hosts increases or the speed of the intermediate links decreases.

If you have an idea what your most commonly used routes are, you might want to force the MTU for your interface to that route to be smaller than you would normally choose. This decrease would allow most traffic to travel without fragmenting.

Using PING to Learn About Routes

You can also use PING to monitor intermediate hosts. If you own your network, you probably already know through which intermediate hosts your data passes. If not, you can use PING to request the route information.

PING specifies the record route option in the IP header of the packet. As a result, the IP header is significantly increased. Figure 32 shows an example of how PING on the AIX operating system reports the route information it receives.

```
perf@Wimpy!5% ping -Rn 9.67.238.3
PING 9.67.238.3: (9.67.238.3): 56 data bytes
64 bytes from 9.67.238.3: icmp_seq=0 ttl=57 time=62 ms
RR:     9.67.96.4
        9.67.11.153
        9.67.238.1
        9.67.1.223
        9.67.96.1
        9.67.113.1
        9.67.113.18
64 bytes from 9.67.238.3: icmp_seq=1 ttl=57 time=51 ms  (same route)
64 bytes from 9.67.238.3: icmp_seq=2 ttl=57 time=30 ms  (same route)
64 bytes from 9.67.238.3: icmp_seq=3 ttl=57 time=33 ms  (same route)
64 bytes from 9.67.238.3: icmp_seq=4 ttl=57 time=34 ms  (same route)
64 bytes from 9.67.238.3: icmp_seq=5 ttl=57 time=33 ms  (same route)
¬C
----9.67.238.3 PING Statistics----
6 packets transmitted, 6 packets received, 0% packet loss
round-trip min/avg/max = 30/40/62 ms
perf@Wimpy!6%
```

Figure 32. Using PING to Do Route Tracing

The starting point where the PING command was issued was at IP address 9.67.113.18 on a RISC System/600 running the AIX operating system. The destination is a VM system at 9.67.238.3.

-R is an option on the DOS and AIX ping commands, but is not available on MVS, VM, or OS/2. On MVS, you can use the TRACERTE command instead. For more information on TRACERTE, see “Using TRACERTE Instead of PING on MVS” on page 45.

The ttl value is the ***time-to-live*** value from the IP header. It is decremented every time a datagram is processed by an intermediate host. This value is also known as the hop count.

Notes on the ttl value:

- The primary purpose of the ttl value is to prevent a packet from being routed infinitely through a network. If ttl=0, the routing host discards the packet.
- AIX, DOS, and OS/2 hosts set the ttl on outgoing datagrams to be 255. So if a PING response from any of these operating systems comes in with a ttl of 255, there were no intervening hosts. If it is 254, then there was 1 intervening host and so on.
- VM and MVS set the ttl on outgoing datagrams to be 60 for 3172 ICP network and RISC/System 6000 channel interfaces.
- For Offload, the 3172 responds to all PINGs, so it uses 255.

In our example, since the ttl is 57 for the packets ***AND*** the destination is a VM system, we know there must have been 3 hops, not 198.

The following route was used in our example:

1. The packet originated from our local IP address of 9.67.113.18 (the last address in the list)
2. The packet was sent from 9.67.113.18 to 9.67.113.1 (the default route set up for our local machine), with ttl=255
3. The packet was sent to IP address 9.67.96.1 with ttl=254
4. The packet was sent to IP address 9.67.1.223 with ttl=253

The next hop was our third, so we know that the packet is sent from 9.67.1.223 to 9.67.238.3 (our destination). (The destination does not appear in the route information because of the way data gets recorded in the packet.)

Next the destination sent a response to PING, with a ttl=60. The destination VM system sent the packet from 9.67.238.3 to 9.67.238.1.

Then the following route was used:

1. The packet was sent to IP address 9.67.11.153 with ttl=59
2. The packet was sent to IP address 9.67.96.4 with ttl=58
3. The packet was sent to the local host, IP address 9.67.113.18 with ttl=57

Why does the packet appear to have returned by a different route? Remember that the intermediate hosts are routing packets to multiple physical networks. Each host has to have at least 2 network interfaces, and therefore each has at least 2 IP addresses. Figure 33 shows the logical networks (we have subnetted class A addresses).

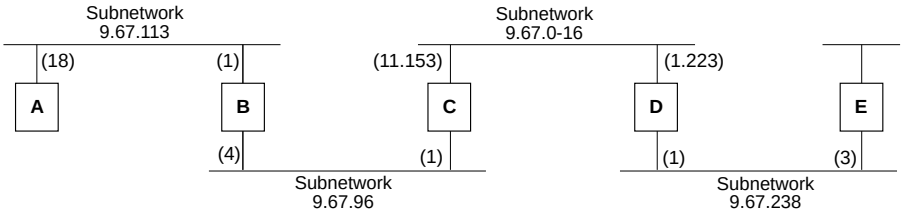


Figure 33. Subnetwork Route Used in the PING Routing Example

So the route we are using is:

```
Host A to
Host B to
Host C to
Host D to
Host E (destination) to
Host D to
Host C to
Host B to
Host A
```

However, the routing information gives us the IP address of the interface to which the packet is sent, which gives us the following route:

```
9.67.113.18 (Host A) to
9.67.113.1 (Host B) to
9.67.96.1 (Host C) to
9.67.1.223 (Host D) to
9.67.238.3 (Host E) to  (not included)
9.67.238.1 (Host D) to
9.67.11.153 (Host C) to
9.67.96.4 (Host B) to
9.67.113.18 (Host A) (not included)
```

Note: All intermediate hosts have the option of ignoring the request for routing information, so the routing information might not be complete. We can deduce whether this is true from the ttl value. In addition, the PING response could return via a different route.

Once you know the route, you can PING the intermediate hosts with varying data sizes and plot the results. Figure 34 shows a graph of our results, using straight lines to make it easier to read the information:

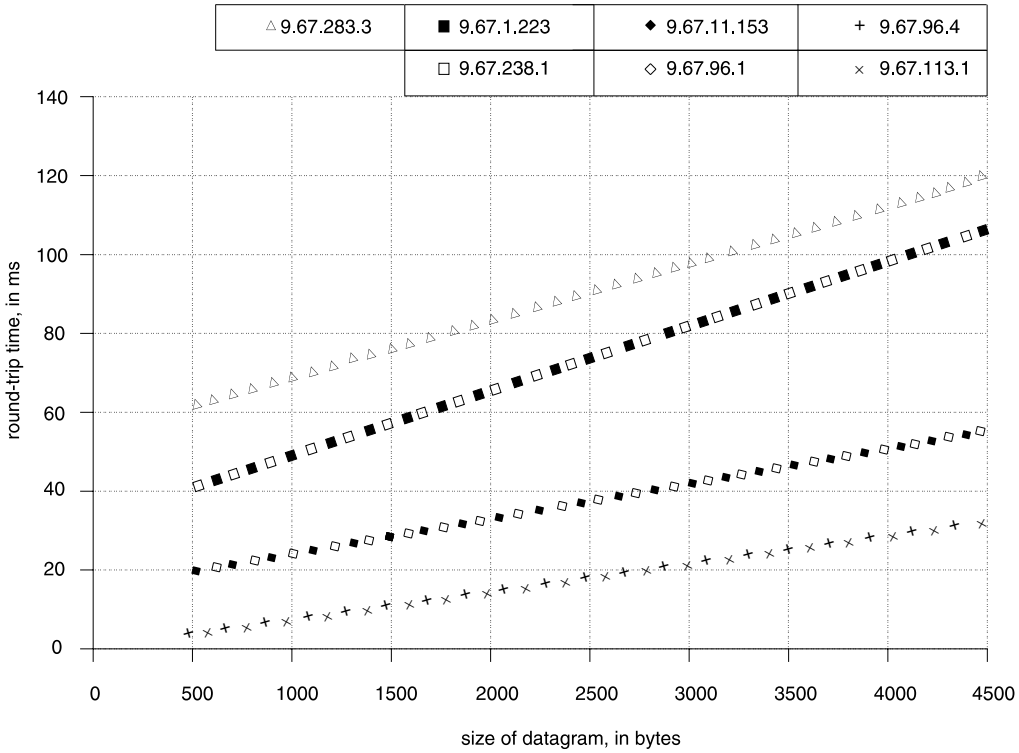


Figure 34. Using PING to Monitor Intermediate Hosts

Here are our conclusions based on the graph:

- Not surprisingly, the minimum round-trip time to each intermediate host is less than the minimum round-trip time to our original destination. Also, the minimum round-trip time to the interfaces that are on the same intermediate hosts are nearly identical. For example, there is no appreciable difference between issuing PING to 9.67.11.153 and 9.67.96.1 on host C.
- After looking at the relative distance between the lines on the graph,
  - The link between host A and host B appears to be about the same speed as the link between host B and host C.

- The link between host C and host D is significantly slower than the links between host A and host B and between host B and host C. This might be because of the LAN media being used, or, it might be because host D has some other resource constraint, such as CPU or adapter speed.
- The link between host D and host E appears to be faster than the link between host C and host D but slower than the other links. Again, this could be due to a resource constraint on host D or on host E, or both.

So what could we do with this information? If we have control over all the hosts in the network, we will probably start by tuning TCP/IP on host D. If not, we might try to find an alternate route to the host E, one that does not go through host D. Another alternative might be to recommend to our users that they try to limit the number or size of file transfers they request from the host E (and host D) during peak times.

Using TRACERTE Instead of PING on MVS

TRACERTE is an equivalent MVS command to the -R option on PING for other platforms. To use this command, your user ID must be on the OBEY statement in the TCP/IP profile. Figure 35 shows an example of using the TRACERTE command to trace the route.

```
-------------------------  TSO COMMAND PROCESSOR
ENTER TSO COMMAND, CLIST, OR REXX EXEC BELOW:

===> tracerte 9.67.35.3




Trace route to 9.67.35.3 (9.67.35.3)
1  IBM (9.67.113.1) 9 ms 22 ms 9 ms
2  IBM (9.67.96.1) 22 ms 20 ms 19 ms
3  IBM (9.67.2.111) 29 ms 35 ms 40 ms
4  IBM (9.67.24.1) 36 ms 43 ms 37 ms
5  IBM (9.67.35.3) 213 ms 178 ms 197 ms

***
```

Figure 35. Using TRACERTE to Do Route Tracing

For more information on the TRACERTE command, see the *IBM TCP/IP for MVS: User's Guide*.

DatagLANce Network Analyzer

DatagLANce Network Analyzer is a network monitoring and analyzing program that you can use with Ethernet and Token Ring networks.

DatagLANce is installed onto a PS/2 running OS/2. We recommend you also add a second LAN adapter when setting up DatagLANce because then you can use Telnet and FTP on TCP/IP for OS/2 for remote access and control. (The second LAN adapter does not have to be one that DatagLANce supports. Use LAPS only for the adapter TCP/IP will use. Also check your CONFIG.SYS to make sure the DatagLANce device drivers are loaded **before** the other device drivers.)

To get started with DatagLANce you need to have some Presentation Manager* skills. To make full use of its features, however, you need to be a network specialist.

Here are some of the ways you can use DatagLANce to monitor TCP/IP performance:

- Monitor network utilization:
  - What is the overall percent utilization of the LAN itself?
  - What percent of the traffic on the LAN is TCP/IP vs. other protocols?
  - What hosts are generating the most network traffic?
  - What are the trends for each of these items?
- Monitor key TCP/IP characteristics that can affect performance:
  - What is the average packet size on the network?
  - What is the average TCP window size being advertised?
  - How much fragmentation is occurring?
- Monitor a performance test:
  - Verify the size of the packets and the TCP window by capturing the first 88 bytes of every frame transferred between the client and server. You can also identify if delays occurred. (This is how our performance test team uses DatagLANce.)
- Monitor and tune your TCP/IP sockets application for performance:
  - Does the data flow the way you expected? You can capture the packets generated by your application, then identify delays so that you can adjust your application code to avoid them.
- Use the protocol decoder to see how these key TCP/IP concepts get implemented:
  - Windowing
  - Fragmentation
  - Acknowledgements
  - Data translation

MVS Tools

The following tables list tools that can be used to monitor the performance of TCP/IP on MVS, IMS, and CICS*. The tables include:

- Name of the tool
- Form of output the tool provides
- Data provided by the tool
- Standard reduction program to help analyze the data, if there is one

Tools to Monitor Transaction Rate and Throughput

Table 2 lists the tools on MVS that can be used to monitor transaction rate and throughput.

Table 2. Monitoring Tools for Transaction Rate and Throughput on MVS, IMS, and CICS

Tool	Output	Data	Reduction
FTP	Messages	KBytes per second	None
TPNS utility	Resp report	Transactions per second	Reduces TPNS log data
RMF* WKLD	SMF records	Transactions per second by group	RMF Reportwriter
NETSTAT	Screen data		None
TCPSTATISTICS	Report	tcpRetransSegs, tcpOutSegs	None

Tools to Monitor Response Time

Table 3 lists the tools on MVS that can be used to monitor internal and external response time. Tools that monitor response time internally look only at the response of the system, not the network.

Table 3. Monitoring Tools for Internal Response Time on MVS, IMS, and CICS

Tool	Output	Data	Reduction
RMF WKLD	SMF records	Transactions per second by group	RMF Reportwriter
CICS Monitor	SMF records	Time stamp start and stop	DFH$MOLS or user program

Table 4 lists the tools that monitor response time end-to-end or externally and look at the response from sender to receiver, including the system and the network.

Table 4. Monitoring Tools for External Response Time on MVS, IMS, and CICS

Tool	Output	Data	Reduction
TPNS	Log data records	Time-stamped screens	TPNS utilities

Tools to Monitor CPU Utilization

Table 5 lists the tools on MVS that can be used to monitor CPU utilization.

Table 5. Monitoring Tools for CPU Utilization on MVS, IMS, and CICS

Tool	Output	Data	Reduction
RMF ARD	SMF records	TCB and CPU seconds	RMF Reportwriter
SDSF DA	Screen/Print	Percentage of CPU seconds	Simple REXX, Shells
CICS Monitor	SMF records		DFH$MOLS or user program
cbsample	Real-time graphs	Internal TCP/IP loads	None

Tools to Monitor Storage Consumption

Table 6 lists the tools on MVS that can be used to monitor storage consumption.

Table 6. Monitoring Tools for Storage Consumption on MVS, IMS, and CICS

Tool	Output	Data	Reduction
RMF VSTOR	SMF records	CSA and ECSA	RMF Reportwriter
RMF ASD	SMF records	CS and ES	RMF Reportwriter
SDSF DA	Screen/Print	Active pages	Simple REXX, Shells
cbsample	Real-time graphs	Buffer and control block usage	None

Tools to Monitor Disk I/O Rate

Table 7 lists the tools on MVS that can be used to monitor disk I/O rate.

Table 7. Monitoring Tools for Disk I/O Rate on MVS, IMS, and CICS

Tool	Output	Data	Reduction
RMF DASD	SMF records	Util, Que, Resp	RMF Reportwriter
SDSF DA	Screen/Print	SIO, EXCP	Simple REXX, Shells

Tools to Monitor Communication Device Utilization

Table 8 lists the tools on MVS that can be used to monitor communication device utilization.

Table 8. Monitoring Tools for Communication Device Utilization on MVS

Tool	Output	Data	Reduction
RMF DEV	SMF records	Util, Resp	RMF Reportwriter

cbsample Program

A sample monitoring program, called cbsample, is provided with TCP/IP for MVS (and also for VM) to monitor internal TCP/IP loads and buffer and control block usage. You can tell the monitor which thresholds to watch and whether to look for minimums, maximums and so on. You can then print the results, take a snapshot, or just log the event.

The cbsample program is run from either an AIX X Windows client on a RISC/System 6000 machine or a Sun 4 workstation running UNIX base Athena non-Motif. It is intended to be used on a color graphics display. Each plot shows the last 200 seconds of activity. Each variable is displayed in a different color. The name of each variable is shown below the plot (using the corresponding color).

The first number to the right of the variable name is the value of that variable for the previous 1-second interval. For the pool size buffers, the second number is the ***minimum*** value for the previous 200-second interval. For the other variables, the

second number is the ***average*** for the variable over the previous 200-second interval.

There are two levels of monitoring that you can select:

- Simple exception monitor
- Comprehensive TCP/IP load monitor

There is a button you can click on to switch between the 2 monitors.

A README file is provided in *hlq*.SEZAINST(CBSAMP) that describes how to install the sample program, options that can be used when starting cbsample, and the meaning of the variables displayed by these monitors.

Simple Exception Monitor

The simple exception monitor shows some indicators of (possible) exceptional events being processed by MVS TCP/IP. TCP/IP pool size buffers are also shown. To select this monitor, start cbsample with the following options on the RISC System/6000 computer:

```
cbsample -xmonexcp -h<hostname_or_dotted_address>
```

Figure 36 is a sample display of the simple monitor.

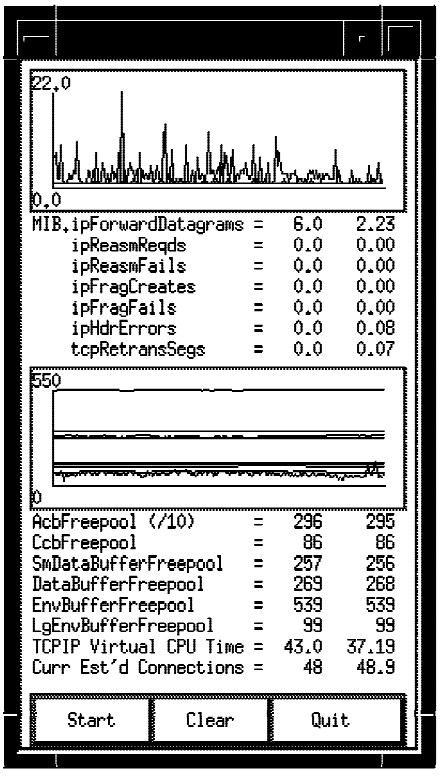


Figure 36. Example of the Simple Exception Monitor

TCP/IP Load Monitor

A more comprehensive monitor includes more internal variables from MVS TCP/IP. The values shown for “CPU Time” are in milliseconds. To select this monitor, start cbsample with the following options on the RISC System/6000 computer:

```
cbsample -xmon -h<hostname_or_dotted_address>
```

Figure 37 is a sample display of the TCP/IP internal load monitor.

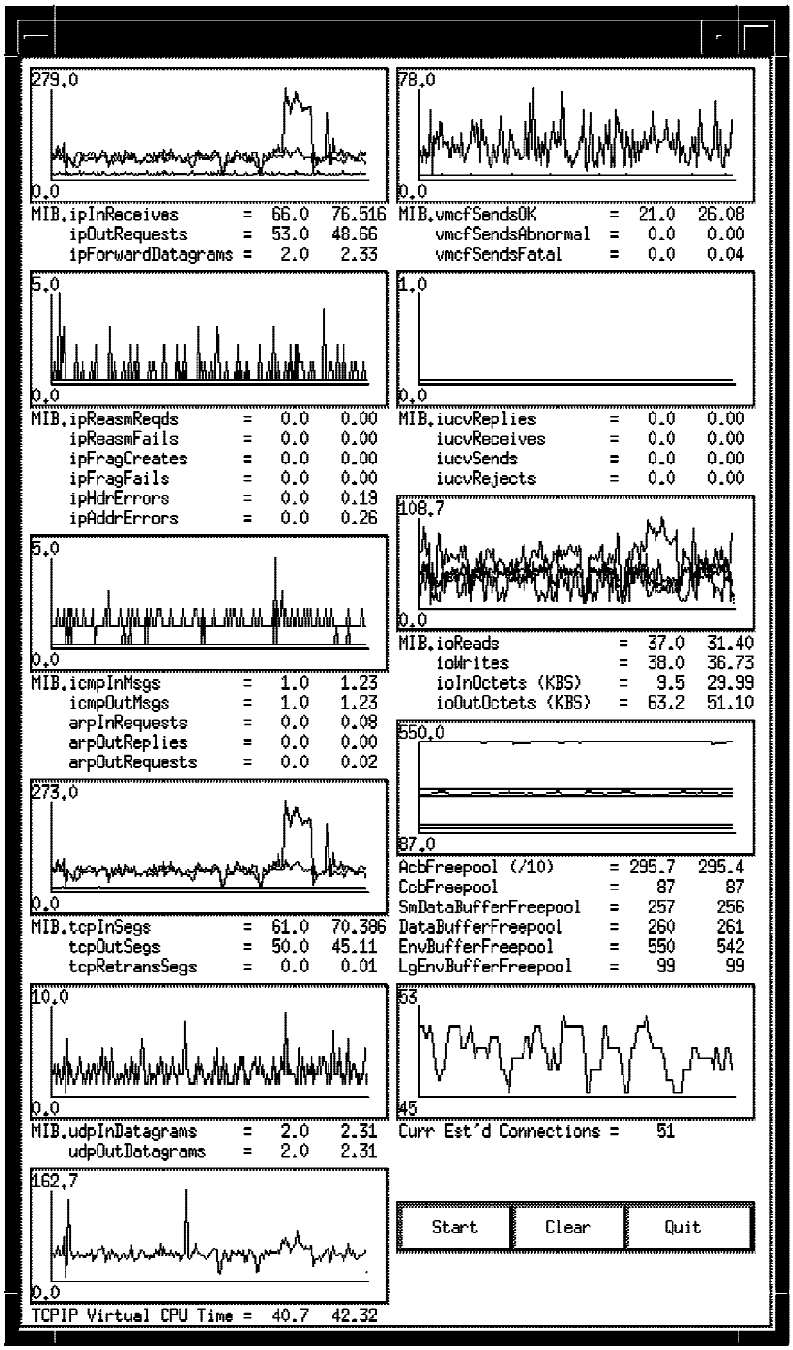


Figure 37. Example of TCP/IP Internal Load Monitor

CICS Monitor

The costs of using the tool involve some CPU overhead, and the level of expertise necessary to use the tool is that of a systems analyst with knowledge of CICS. For more information about the CICS Monitor, see the *CICS/ESA Performance Guide*.

FTP

As shown in “Using FTP” on page 14, the response time and throughput are shown every time a user transfers a file using FTP.

NETSTAT

This command that can be used to monitor TCP/IP. The following list is a subset of the NETSTAT parameters that you might want to use:

- ALL
- GATE
- POOLSIZE

For examples of NETSTAT, see “NETSTAT ALL Command” on page 59. For more information on the NETSTAT command, see the *TCP/IP for MVS: User's Guide*.

RMF

The Resource Measurement Facility* (RMF) program is a monitoring tool. Some commands can be issued in batch or from TSO. Reportwriter is used to print the data you get with the RMF commands. The costs of using the tool involve some CPU overhead, and the level of expertise necessary to use the tool is that of a systems analyst.

RMF data can be viewed online as it is collected or it can be saved for later processing.

To see an example of how RMF commands can be used to monitor CPU usage, see “Performance Example” on page 109. In that example, RMF data is collected and saved for later processing using the RMF Reportwriter post processor.

Alternatively, you can use RMF commands to view data online as it is collected. This interactive approach is explained in the *RMF User's Guide*.

From to TSO/E, enter the following command:

```
RMFMON
```

RMF will display a primary menu, shown in Figure 38.

```
                    RMF DISPLAY MENU
NAME      PFK#  DESCRIPTION
                                                        IPS = IEAIPS00
ARD        1    ADDRESS SPACE RESOURCE DATA
ASD        2    ADDRESS SPACE STATE DATA
ASRM       3    ADDRESS SPACE SRM DATA
CHANNEL    4    CHANNEL PATH DISPLAY
DDMN       5    SYSTEM DOMAIN DISPLAY
DEV        6    SYSTEM DEVICE DATA
PGSP       7    SYSTEM PAGING SPACE DATA
SENQ       8    SYSTEM ENQUEUE CONTENTION
SENQR      9    SYSTEM ENQUEUE RESERVE
SPAG       10   PAGING DATA
SRCS       11   CENTRAL STORAGE / CPU / SRM DATA
TRX        12   TRANSACTION ACTIVITY DATA
ARDJ            RESOURCE DATA FOR SPECIFIC JOBNAME
ASDJ            STATE DATA FOR SPECIFIC JOBNAME
ASRMJ           SRM DATA FOR SPECIFIC JOBNAME
DEVV            SYSTEM DEVICE DATA FOR A SPECIFIC VOL/NUMBER
IOQUEUE         I/O QUEUING ACTIVITY DISPLAY
USER            USER PICTURE
```

Figure 38. RMF Monitor Primary Menu

You can then enter RMFMON session commands by typing them in the input area (the top left corner) and pressing the ENTER key. To exit the RMFMON session, enter Z.

The primary menu lists the names of reports that can be requested during the RMFMON session. You can request the report by entering the name in the input area or by pressing the corresponding function key (if there is one). Most of the reports have options that can be selected by entering them after the report name.

The reports are either row reports or table reports. Row reports have only one line of reported data. When you request a row report repeatedly, each request adds one new line of data to the report. When the screen is full, the next request overlays the first line of data, and the original line of data is lost.

Table reports have a variable number of data lines. You can use the F (frame) command to scroll forward through the data.

Here are some other (non-report) commands that control the RMFMON session:

M Display the menu

D ON Set delta mode reporting (reports will reflect only the differences since the previous request for the report)

D OFF Set total mode reporting (reports will reflect the total values for the session)

T x,y Have the report updated automatically, x times at y seconds between each report

Example of Getting CPU Busy Time and Disk I/O Rates for TCPIP Address Spaces

1. From TSO, enter `RMFMON`.
2. Enter `ARD B,,5` in the input area on the primary menu. A report will be created on batch, started task, and mount task address spaces in domain 5.
3. Enter `D ON` in the input area. This causes each report to include only the data since the last report, not a cumulative total.

4. Press the ENTER key to begin. Record the time stamp. Notice there is no CPU time for the TCP22R address space or the FTP servers, as shown in Figure 39.

```
                                CPU=  8 UIC=255 PFR=   0 ARD      D
13:33:43 DEV    FF PRIV LSQA LSQA X SRM  TCB    CPU   EXCP SWAP LPA CSA NVI V&H
JOBNAME  CONN  BEL   FF  CSF  ESF M ABS  TIME   TIME  RATE RATE  RT  RT  RT  RT
PCAUTH   0.000   0    2   21    0 X 0.0 ------ ------ 0.00 0.00 0.0 0.0 0.0 0.0
RASP     0.000 --- ---- ---- ---- X 0.0 ------ ------ 0.00 0.00 0.0 0.0 0.0 0.0
TRACE    0.000   0    3  273    0 X 0.0 ------ ------ 0.00 0.00 0.0 0.0 0.0 0.0
XCFAS    0.000   0   21  213    0 X 0.0   0.00   0.00 0.00 0.00 0.0 0.0 0.0 0.0
SMXC     0.000   0    2   16    0   0.0 ------ ------ 0.00 0.00 0.0 0.0 0.0 0.0
SYSBMAS  0.000   0    7   73    0   0.0 ------ ------ 0.00 0.00 0.0 0.0 0.0 0.0
CONSOLE  0.000   0    6   33    0 X 0.0 ------ ------ 0.00 0.00 0.0 0.0 0.0 0.0
ALLOCAS  0.000   0    2   65    0 X 0.0 ------ ------ 0.00 0.00 0.0 0.0 0.0 0.0
LLA      0.000  22    8   48    0 X 0.0 ------ ------ 0.00 0.00 0.0 0.0 0.0 0.0
JES2     0.025  21   27   61    0   323   0.00   0.00 4.00 0.00 0.0 0.0 0.0 0.0
NETTCP   0.000  24   12   49    0    35 ------ ------ 0.00 0.00 0.0 0.0 0.0 0.0
IOSAS    0.000   0    3   24    0   0.0 ------ ------ 0.00 0.00 0.0 0.0 0.0 0.0
TNF      0.000   0    2   16    0 X 0.0 ------ ------ 0.00 0.00 0.0 0.0 0.0 0.0
VMCF     0.000   0    2   16    0 X 0.0 ------ ------ 0.00 0.00 0.0 0.0 0.0 0.0
VLF      0.000   4    9   32    0 X 0.0 ------ ------ 0.00 0.00 0.0 0.0 0.0 0.0
FTPSR35  0.000   0    2   31    0   0.0 ------ ------ 0.00 0.00 0.0 0.0 0.0 0.0
FTPSR36  0.000   0    2   31    0   0.0 ------ ------ 0.00 0.00 0.0 0.0 0.0 0.0
FTPSR37  0.000   0    2   31    0   0.0 ------ ------ 0.00 0.00 0.0 0.0 0.0 0.0
FTPSR38  0.000   0    2   34    0   0.0 ------ ------ 0.00 0.00 0.0 0.0 0.0 0.0
TCP22R   0.000  13   13   51    0   0.0 ------ ------ 0.00 0.00 0.0 0.0 0.0 0.0
```

Figure 39. Using RMFMON to Get CPU Busy Time and Disk I/O Rates

5. Start the test on the client. For our example, we transferred a single 4MB file from the MVS server to an AIX client across the FDDI network interface.
6. Press the ENTER key at the RMFMON session to get an updated report, as shown in Figure 40.

```
                                CPU=  2 UIC=255 PFR=   0 ARD      D
13:34:08 DEV    FF PRIV LSQA LSQA X SRM  TCB    CPU   EXCP SWAP LPA CSA NVI V&H
JOBNAME  CONN  BEL   FF  CSF  ESF M ABS  TIME   TIME  RATE RATE  RT  RT  RT  RT
PCAUTH   0.000   0    2   21    0 X 0.0 ------ ------ 0.00 0.00 0.0 0.0 0.0 0.0
RASP     0.000 --- ---- ---- ---- X 0.0 ------ ------ 0.00 0.00 0.0 0.0 0.0 0.0
TRACE    0.000   0    3  273    0 X 0.0 ------ ------ 0.00 0.00 0.0 0.0 0.0 0.0
XCFAS    0.000   0   21  213    0 X 8.3   0.01   0.01 0.00 0.00 0.0 0.0 0.0 0.0
SMXC     0.000   0    2   16    0   0.0 ------ ------ 0.00 0.00 0.0 0.0 0.0 0.0
SYSBMAS  0.000   0    7   73    0   0.0 ------ ------ 0.00 0.00 0.0 0.0 0.0 0.0
CONSOLE  0.000   0    6   33    0 X 0.0 ------ ------ 0.00 0.00 0.0 0.0 0.0 0.0
ALLOCAS  0.000   0    2   65    0 X 0.0 ------ ------ 0.00 0.00 0.0 0.0 0.0 0.0
LLA      0.000  22    8   48    0 X 0.0 ------ ------ 0.00 0.00 0.0 0.0 0.0 0.0
JES2     0.476  21   27   61    0   153   0.11   0.13 3.20 0.00 0.0 0.0 0.0 0.0
NETTCP   0.000  24   12   49    0   8.8   0.01   0.01 0.00 0.00 0.0 0.0 0.0 0.0
IOSAS    0.000   0    3   24    0   0.0 ------ ------ 0.00 0.00 0.0 0.0 0.0 0.0
TNF      0.000   0    2   16    0 X 0.0 ------ ------ 0.00 0.00 0.0 0.0 0.0 0.0
VMCF     0.000   0    2   16    0 X 0.0 ------ ------ 0.00 0.00 0.0 0.0 0.0 0.0
VLF      0.000   4    9   32    0 X 0.8   0.00   0.00 0.00 0.00 0.0 0.0 0.0 0.0
FTPSR35  0.000   0    2   31    0   0.0 ------ ------ 0.00 0.00 0.0 0.0 0.0 0.0
FTPSR36  0.000   0    2   31    0   0.0 ------ ------ 0.00 0.00 0.0 0.0 0.0 0.0
FTPSR37  0.000   0    2   31    0   0.0 ------ ------ 0.00 0.00 0.0 0.0 0.0 0.0
FTPSR38  1.223   0    2   34    0   897   0.48   0.52 7.76 0.00 0.0 0.0 0.0 0.0
TCP22R   1.080  13   13   51    0    4K   1.21   1.31 0.00 0.00 0.0 0.0 0.0 0.0
```

Figure 40. Using RMFMON to Get Updated CPU Busy Time and Disk I/O Rates

   Record the time stamp.

7. Analyze the results for CPU busy time:
   - The elapsed time is 25 seconds (13::34:08 – 13:33:43).
   - The CPU TIME for the TCPIP address space (TCP22R) is 1.31 seconds.

- The FTP server (FTPSR38) CPU TIME is 0.52 seconds.
- Total CPU TIME is 1.83 seconds.

8. Analyze the results for disk I/O rate:
   - The EXCP RATE is the disk I/O rate.
   - The EXCP RATE for the FTP server (FTPSR38) is 7.76 per second.
   - The EXCP RATE for the TCPIP address space (TCP22R) is 0.
   - The elapsed time was 25 seconds.
   - The total number of EXCPs is 194 (25 * 7.76). Since a 4MB file was transferred, this means approximately 21KB per EXCP (4MB / 194 EXCPs).

Example of How to Get Communications I/O Rates for TCP/IP Address Spaces

1. Start Monitor I data collection (installation-dependent)
2. From TSO, enter `RMFMON`.
3. Enter `DEV COMM` in the input area on the primary menu. This creates a report on I/O activity to communications devices.
4. Enter `D ON` in the input area. This causes each report to include only the data since the last report, not a cumulative total.
5. Press the ENTER key to begin. Record the time stamp. Our 3172 is at addresses B78 and B79. (One of these addresses will match the addresses on the DEVICE statement in the PROFILE.TCPIP data set.) The resulting screen is shown in Figure 41.

```
                                CPU=  2 UIC=254 PFR=   0 DEV      T
13:14:57        DEV      ACTV RESP IOSQ ---DELAY--- PEND DISC CONN %DEV %D
STG GRP  VOLSER NUM LCU  RATE TIME TIME DPB CUB DB  TIME TIME TIME UTIL RV I%
                00F --- 0.000    0    0     0.0 0.0  0.0  0.0  0.0 0.00  0 27
                158 --- 0.000    0    0     0.0 0.0  0.0  0.0  0.0 0.00  0 27
                159 --- 0.000    0    0     0.0 0.0  0.0  0.0  0.0 0.00  0 27
                9CE --- 0.000    0    0     0.0 0.0  0.0  0.0  0.0 0.00  0 27
                9CF --- 0.000    0    0     0.0 0.0  0.0  0.0  0.0 0.00  0 27
                B78 --- 0.000    0    0     0.0 0.0  0.0  0.0  0.0 0.00  0 27
                B79 --- 0.000    0    0     0.0 0.0  0.0  0.0  0.0 0.00  0 27
                DAA --- 0.000    0    0     0.0 0.0  0.0  0.0  0.0 0.00  0 27
```

Figure 41. Using RMFMON to Get Communications I/O Rates

6. Start the test on the client. For our example, we transferred a single 4MB file from the MVS server to an AIX client across the FDDI network interface.
7. Press the ENTER key at the RMFMON session to get an updated report, as shown in Figure 42.

```
                                CPU= 13 UIC=254 PFR=   0 DEV      T
13:15:17        DEV      ACTV RESP IOSQ ---DELAY--- PEND DISC CONN %DEV %D
STG GRP  VOLSER NUM LCU  RATE TIME TIME DPB CUB DB  TIME TIME TIME UTIL RV I%
                00F --- 0.000    0    0     0.0 0.0  0.0  0.0  0.0 0.00  0 61
                158 --- 0.000    0    0     0.0 0.0  0.0  0.0  0.0 0.00  0 61
                159 --- 0.000    0    0     0.0 0.0  0.0  0.0  0.0 0.00  0 61
                9CE --- 0.000    0    0     0.0 0.0  0.0  0.0  0.0 0.00  0 61
                9CF --- 0.000    0    0     0.0 0.0  0.0  0.0  0.0 0.00  0 61
                B78 --- 15.21   64    0     0.0 0.0  0.2 63.5  0.1 96.8  0 61
                B79 --- 12.68    2    0     0.0 0.0  0.0  0.1  2.2 2.91  0 61
                DAA --- 0.000    0    0     0.0 0.0  0.0  0.0  0.0 0.00  0 61
```

Figure 42. Using RMFMON to Get Updated Communications I/O Rates

Record the time stamp.

8. Analyze the results for CPU busy time:
   - The elapsed time is 20 seconds (13:15:17 – 13:14:57).
   - The ACTV RATE to B78 is 15.21 per second, for a total of 304 I/Os (15.21 * 20).
   - The ACTV RATE to B79 is 12.68 per second, for a total of 254 I/Os (12.68 * 20).
   - The total number of I/Os is 558 (304 + 254). Since a 4MB file was transferred, this means approximately 7KB per communications I/O (4MB / 558 I/Os).

For more information on the RMF tool, see the *RMF User's Guide* and the *RMF Analyzing RMF Monitor I & II Reports* manuals.

SDSF

System Display Service Facility (SDSF) is a system management aid that you can use to analyze the operation of an MVS/JES2 system. The costs of using the tool involve some CPU overhead, and the level of expertise necessary to use the tool is that of an end user. You can customize which SDSF fields are displayed and in which order so that not all users will see all the fields.

Figure 43 shows the response from SDSF when we entered D.DA from the main ISPF panel:

```
SDSF DA MVSO PAGING  0.00 SIO   6.70 CPU  55.64%   LINE 16-28 (28)
COMMAND INPUT ===>                                    SCROLL ===>
NP JOBNAME   REAL  PAGING   SIO   CPU%  EXCP-CNT   CPU-TIME
   SOF1       208    0.00   0.00   0.00        86      10.60
   NETTCPP   1704    0.00   0.00   0.00       828    5583.67
   IOSAS      112    0.00   0.00   0.00        79     415.13
   TNF         96    0.00   0.00   0.00         9       0.03
   VMCF       168    0.00   0.00   0.00        29       0.05
   CATALOG   2284    0.00   0.00   0.00       551    1271.45
   FTPSR35   1776    0.00   0.00   0.00     1,694      12.08
   USER3     1248    0.00   0.00  31.16       753       2.43
   TSO        172    0.00   0.00   0.00        54       0.76
   TCP31R     10T    0.00   0.00  17.04       549      66.19
   FTPSR37   1776    0.00   0.00   0.00     1,724      12.23
   FTPSR36   1776    0.00   0.00   0.00     1,723      12.27
   FTPSR38   1776    0.00   0.00   0.00     1,722      12.26
```

Figure 43. Example of the Display Active Users Panel Using SDSF

The title line of the DA panel displays the following information for the whole system:

- `CPU` for the total percentage of time the CPU is busy
- `SIO` for the total system start I/O rate
- `PAGING` for the total demand paging rate

From the Display Active Users (DA) panel, the following fields might useful in monitoring TCP/IP:

- `CPU%` for the CPU time expressed as a percentage of the total CPU by address space
- `CPU-TIME` for the accumulated CPU time for the current job step by address space

- `EXCP-CNT` for the total I/O rate by the current job step
- `PAGING` for the demand paging rate (only present if the address space was swapped in for the entire interval)
- `REAL` for working set size for the address space (current utilization of real storage)

For more information on using SDSF, see the *SDSF Guide and Reference* manual.

TPNS Utility

Teleprocessing Network Simulator (TPNS) is a terminal and network simulation tool that you can use to determine system performance and response time and to evaluate network design. The costs of using the tool involve CPU overhead only after the measurement period. For more information on using TPNS, see the *TPNS General Utilities* manual.

TCPIPSTATISTICS

You can specify TCPIPSTATISTICS on the ASSORTEDPARMS statement to get information about TCP/IP The catch is that you have to shut TCP/IP down to get the report.

When TCP/IP is shutdown, the values of several counters are printed in TCP.OUTPUT, as shown in Figure 44.

See the section on MIB objects in the *TCP/IP for MVS: User's Guide* for a complete description of the counters.

```
ipInReceives      =         622   vmcfSendsOK       =         188
ipOutRequests     =         371   vmcfSendsAbnormal =           0
ipForwDatagrams   =           0   vmcfSendsFatal    =           5
ipReasmReqds      =           0   iucvReplies       =           0
ipReasmFails      =           0   iucvReceives      =           0
ipFragCreates     =           0   iucvSends         =           0
ipFragFails       =           0   iucvRejects       =           0
ipInHdrErrors     =           0   ioReads           =         320
ipInAddrErrors    =           9   ioWrites          =         224
icmpInMsgs        =           1   ioInOctets        =      234860
icmpOutMsgs       =           4   ioOutOctets       =      134482
arpInRequests     =          58
arpOutReplies     =           0
arpOutRequests    =           3
tcpInSegs         =         449
tcpOutSegs        =         360
tcpRetransSegs    =          72
udpInDatagrams    =           0
udpOutDatagrams   =           0

ShutDown at 133.679 seconds
```

Figure 44. TCPIPSTATISTICS Counters in the TCP.OUTPUT Data Set

For example, to get the percentage of re-transmitted packets, compare the tcpRetransSegs value to the tcpOutSegs value. The tcpRetransSegs value is the number of TCP re-transmissions that occurred since TCP/IP was started. The tcpOutSegs value is the total number of TCP segments sent out from the MVS TCP/IP system.

```
72 / 360 = 0.20 = 20%
```

VM Tools

The following tables list tools that can be used to monitor the performance of TCP/IP on VM. The tables include:

- Name of the tool
- Form of output the tool provides
- Data provided by the tool
- Standard reduction program to help analyze the data, if there is one

Tools to Monitor Transaction Rate and Throughput

Table 9 lists the tools on VM that can be used to monitor transaction rate and throughput.

Table 9. Monitoring Tools for Transaction Rate and Throughput on VM

Tool	Output	Data	Reduction
VM Monitor	Disk or tape	Average, trivial, nontrivial command rates	VMPRF
FTP client	Screen data	Bytes per second	None
RTM VM/ESA*	Screen data	Transaction rate	None
TPNS utility	Resp report	transactions per second	Reduces TPNS log data
NETSTAT	Screen data		None

Tools to Monitor Response Time

Table 10 lists the tools on VM that can be used to monitor response time.

Table 10. Monitoring Tools for Internal Response Time on VM

Tool	Output	Data	Reduction
VM Monitor	Disk or tape	Trivial response time, average, major, minor	VMPRF
RTM VM/ESA	Screen data	Response time	None

Tools to Monitor CPU Utilization

Table 11 lists the tools on VM that can be used to monitor CPU utilization.

Table 11. Monitoring Tools for CPU Utilization on VM

Tool	Output	Data	Reduction
VM Monitor	Disk or tape	CPU busy time % by virtual machine	VMPRF
CP INDICATE LOAD	Screen data	CPU % utilization	None
RTM VM/ESA	Screen data	CPU % utilization	None
cbsample	Real-time graphs	Internal TCP/IP loads	None

Tools to Monitor Storage Consumption

Table 12 lists the tools on VM that can be used to monitor storage consumption.

Table 12. Monitoring Tools for Storage Consumption on VM

Tool	Output	Data	Reduction
VM Monitor	Disk or tape	Paging rate, working set size, queues	VMPRF
CP INDICATE LOAD	Screen data	Paging rate, expanded storage activity	None
CP QUERY XSTORE	Screen data	Expanded storage assignment	None
RTM VM/ESA	Screen data	Storage % utilization, paging rate working set size	None
cbsample	Real-time graphs	Buffer and control block usage	None

Tools to Monitor Disk I/O Rate

Table 13 lists the tools on VM that can be used to monitor disk I/O rate.

Table 13. Monitoring Tools for Disk I/O Rate on VM

Tool	Output	Data	Reduction
VM Monitor	Disk or tape	Disk I/O rate	VMPRF
RTM VM/ESA	Screen data	I/O rate, % utilization	None

Tools to Monitor Communication Device Utilization

Table 14 lists the tools on VM that can be used to monitor communication device utilization.

Table 14. Monitoring Tools for Communication Device Utilization on VM

Tool	Output	Data	Reduction
VM Monitor	Disk or tape	Channel utilization, device utilization	VMPRF
RTM VM/ESA	Screen data	I/O rate, % utilization	None

cbsample Program

A sample monitoring program, called cbsample, can be used on VM to monitor internal TCP/IP loads and buffer and control block usage. See “cbsample Program” on page 48 for more information.

CP INDICATE LOAD

INDICATE LOAD is a CP command that displays statistics about the operating load on the VM system. It is available to privilege class E (systems analyst) or G (general) users. More data is displayed to class E users. For more information, see the *CP Command and Utility Reference* manual.

CP QUERY XSTORE

QUERY XSTORE is a CP command that displays information about real expanded storage. It is available to privilege class B (system resource operator) users. For more information, see the *CP Command and Utility Reference* manual.

FTP

As shown in “Using FTP” on page 14, the response time and throughput are shown every time a user transfers a file using FTP.

NETSTAT

This command can be used to monitor TCP/IP. The following list is a subset of the NETSTAT parameters that you might want to use:

- ALL
- GATE
- POOLSIZE

For more examples of the NETSTAT command, including NETSTAT GATE, see “Example Environment” on page 109. For more information on the NETSTAT command, see the *TCP/IP for VM: User's Guide*.

NETSTAT ALL Command

You can use the NETSTAT ALL command to view information about all the TCP/IP connections. In the following example, the output for only one TCP/IP user is displayed. The output from this command can be enormous if there are many users, especially Telnet, on your system.

In Figure 45, notice that the amount shown for the `MaxSndWnd`, or send window, is 28KB. This amount is the default size of the OS/2 receive buffer.

```
netstat all
VM TCP/IP Netstat V2R2

...
Client: FTPSERVE                        Last Touched:   0:00:10
Local Socket: RALVM12.RALEIGH.IBM.COM..FTP-C
Foreign Socket: DAGGER.RALEIGH.IBM.COM..1025
     BackoffCount: 0
     ClientRcvNxt: 3474309660
     ClientSndNxt: 1017879133
     CongestionWindow: 7260
     Local connection name: 1170
     Sender frustration level: Contented
     Incoming window number: 3474317826
     Initial receive sequence number: 3474309633
     Initial send sequence number: 1017878848
     Maximum segment size: 1452
     Outgoing window number: 1017907691
     Precedence: Routine
     RcvNxt: 3474309660
     Round-trip information:
        Max number unacked: 1
        Smooth trip time:   0.070
        Smooth trip variance:   0.079
        Total acked: 4
        Acks not counted: 0
        Average trip time:   0.142
     SlowStartThreshold: 14336
     SndNxt: 1017879133
     SndUna: 1017879133
     SndWl1: 3474309660
     SndWl2: 1017879133
     SndWnd: 28558
     MaxSndWnd: 28672
     State: Established
     Pending TCP-receive buffer: 8192
...
```

Figure 45. Example of NETSTAT ALL Command on VM

NETSTAT POOLSIZE Command

You should use the NETSTAT POOLSIZE command to monitor the TCP/IP buffer pools. When you use this command, the following information is displayed:

- Number of buffers allocated at startup time
- Current number of free buffers
- Low-water number, indicating the lowest number of free buffers reached since startup
- Permit size

When the number of free buffers reaches the permit size, a warning message is sent to the user IDs defined in the INFORM list, and TCPIP will slow down.

Depending on the observed variations of the low-water numbers in comparison to the permit sizes, you might decide to change the values associated with the statements in your configuration file.

Figure 46 shows an example of the response you could receive with the NETSTAT POOLSIZE command.

```
netstat poolsize

VM TCP/IP Netstat V2R2
TCPIP Free pool status:
Object     # alloc     # free      Lo-water    Permit size
======     =======     ======      ========    ===========
ACB          2000        1987        1846         200
CCB           750         716         706          50
Dat buf       300         280         204          60
Sm dat buf   1200        1078        1030         120
Env           750         750         721          75
Lrg env        25          24          23           5
RCB            50          50          50           3
SCB           512         490         363          34
SKCB          256         251         234          17
TCB           800         665         544          53
UCB           100          95          93           6
```

Figure 46. Example of NETSTAT POOLSIZE on VM

RTM VM/ESA

RTM VM/ESA is used for monitoring, analysis, and problem solving. The costs of using the tool involve some CPU overhead, and the level of expertise necessary to use the tool is that of a system programmer.

Specific commands that can be used for monitoring TCP/IP are:

- DISPLAY DEVICE
- DISPLAY GENERAL
- DISPLAY SRC USER
- DISPLAY USER
- DISPLAY VMX USER
- DISPLAY XUSER

For more information, refer to the *Realtime Monitor VM/ESA Program Description/Operations Manual*.

TPNS Utility

Teleprocessing Network Simulator (TPNS) is a terminal and network simulation tool that you can use to determine system performance and response time and to evaluate network design. The costs of using the tool involve CPU overhead only after the measurement period. For more information on using TPNS, see the *TPNS General Utilities* manual.

VM Monitor

VM Monitor is a collection facility that comes with the VM operating system. The costs of using the tool involve some CPU overhead on monitoring and data reduction, and the level of expertise necessary to use the tool is that of a system programmer.

For more information on the VM Monitor, see the *VM/ESA Performance* book.

To reduce the data that you get from VM Monitor, use VMPRF. VMPRF detects and diagnoses performance problems, analyzes system performance, and then gives you printed reports and trend data showing performance analysis.

Example of the Use of VM Monitor and VMPRF

These are commands set up in the PROFILE EXEC of the user ID you will be monitoring from. These commands are not really specific to TCP/IP but the high sample and interval rate are useful when benchmarking TCP/IP.

```
'CP SET PF12 RET'
'CP MONITOR SAMPLE ENABLE I/O ALL'
'CP MONITOR SAMPLE ENABLE USER ALL'
'CP MONITOR SAMPLE ENABLE PROCESSOR'
'CP MONITOR SAMPLE ENABLE STORAGE'
'CP MONITOR SAMPLE INTERVAL 10 SECONDS'
'CP MONITOR SAMPLE RATE 2 SECONDS'
'SET LDRTBLS 25'
```

Figure 47. VM Profile Commands for Monitoring

Figure 48 shows the commands that are in an EXEC on another user ID. The EXEC is invoked when everything is set up to begin testing. (You might find it convenient to have function keys set up for querying the time.)

```
'SET PF1 IMMED Q TIME '
'SET PF2 IMMED Q MONITOR '
'CP SEND MONWRITE MONWRITE MONDCSS * MONITOR DISK'
'CP SLEEP 5 SEC'
'CP MONITOR START'
```

Figure 48. EXEC to Set Up and Start Monitoring with VM Monitor

Notice that the first MONWRITE in the third line is the user ID we are monitoring from. The second MONWRITE is the command we are sending to the user ID.

On the same user ID, we have an EXEC to stop the monitor when testing is completed as shown in Figure 49:

```
'CP MONITOR STOP'
'CP SLEEP 10 SEC'
'CP SEND MONWRITE STOP'
```

Figure 49. EXEC to Stop Monitoring with VM Monitor

Figure 50 shows the file created by the monitor that has the date as the file name and the time as the file type. You might want to rename the file to have a more meaningful name.

```
 MONWRITE FILELIST A0  V 108  Trunc=108 Size=29 Line=1 Col=1 Alt=0
Cmd   Filename Filetype Fm Format Lrecl    Records     Blocks   Date
      D120193  T103129  A1 F       4096        148        148 12/01/93 10
```

Figure 50. File Created by VM Monitor

Figure 51 shows an example of the use of the VMPRF command to reduce the data:

```
vmprf sshxpn disk nmy0993e monwdata (settings(stime=23:01:54
etime=23:02:45 system=nmy0993e)
2 Dec 1993 09:06:48 VMPRF Version 1.2.1
2 Dec 1993 09:06:49 VMPRF Command Line appears on following line:
2 Dec 1993 09:06:49 VMPRF vmprf sshxpn disk nmy0993e monwdata
(settings(stime=2
:01:54 etime=23:02:45 system=nmy0993e)
2 Dec 1993 09:06:55 VMPRF Reduction Started.
DMSLIO740I Execution begins...
2 Dec 1993 09:08:03 VMPRF Vtime=13.02, Ttime=13.98, Connect time=   45
2 Dec 1993 09:08:04 VMPRF Reduction Completed with Return Code: 4
2 Dec 1993 09:08:13 VMPRF Completed with Return Code 4
Ready(00004);
```

Figure 51. Using VMPRF to Reduce the VM Monitor Data

The second word in the VMPRF command, sshxpn, is the name of the file SSHXPN MASTER. NMY0993E MONWDATA is the data produced by VM Monitor.

The EXEC uses the following file called SSHXPN MASTER, as shown in Figure 52.

```
*** Input Parameter Files ***

SETTINGS      SSHXPN   SETTINGS *
REPORTS       SHANIS   REPORTS  *
INCLUSER      SHANIS   INCLUSER *
UCLASS        SHANIS   UCLASS   *

*** Output Files ***

LISTING       SSHXPN   LISTING A
LOG           VMPRF    LOG     A
RUNFILE       VMPRF    RUNFILE A
TREND         VMPRF
SUMMARY       VMPRF    SUMMARY A
```

Figure 52. Sample Master File

The input files to SSHXPN MASTER, including usage notes, are shown in “Input Files for the VM Monitor Example” on page 201. The INCLUSER file makes sure that certain users are always included in the reports.

To find out the CPU utilization for the time period monitored, look at the USER_RESOURCE_UTIL report, as shown in Figure 53:

```
          <---------CPU--------->  ...
               <-Seconds->         ...
0                                  ...
                               T/V  ...
 Userid    Pct  Total  Virt Ratio  ...

 TCPIP3    0.6    22    14   1.5  ...
 FTPD32    0.1     2     1   2.1  ...
 FTPD31    0.0     1     1   2.1  ...
 MONWRITE  0.0     1     1   2.5  ...
 FTPSERV3  0.0     1     0   2.1  ...
 FTPD3     0.0     1     0   2.1  ...
...
...
...
 Sum/Mean  0.9    32    19   1.7  ...
```

Figure 53. Part of Sample USER_RESOURCE_UTIL Report

Notice that for our example the TCPIP3 user ID used 22 seconds of CPU time.

You can also track the CPU utilization on the UCLASS_VMCOMM_ACTIVITY report, as shown in Figure 54. CPU seconds is the number of CPU seconds divided by the number of messages that were sent.

```
                    <-Message Rate-> ... <-Per Message->
                    IUCV+VMCF ...
           Log-
 User       ged  Total          ...           CPU  ...
 Class    Users   Msgs  Total   ...          Secs  ...

 TCPIP Sv     1   3744  4.274   ...        0.0057  ...
 FTP Serv     4   3400  3.882   ...        0.0013  ...
 Monitors     1    585  0.668   ...        0.0021  ...
 OurUsers     1    344  0.393   ...        0.0120  ...
 ...
 ...
 ...
 Sum/Mean    14   8073  9.216   ...        0.0039  ...
```

Figure 54. Part of Sample UCLASS_VMCOMM_ACTIVITY Report

For this example, the FTP servers used 0.0013 * 3400 or 4.4 seconds of CPU time. The TCP/IP user ID used 0.0057 * 3744 or 21.3 seconds of CPU time.

UCLASS reports group users as specified in the UCLASS file. Our UCLASS file (shown in Figure 99 on page 203) groups the FTP users together.

You might want to group all the default users together, as shown in Figure 100 on page 204. This can be an initial way to start studying the resources that TCP/IP is using overall.

To look at disk I/O (DASD), use the UCLASS_RESOUCE_UTIL (Figure 55) or UCLASS_VMCOMM_ACTIVITY report (Figure 56) to get the number of disk I/Os.

```
                <---------CPU--------->  ... <-DASD->
                      <-Seconds->        ...
0          Log-                         ...   Rate
 User       ged                     T/V ...  While
 Class    Users   Pct Total  Virt Ratio ... Logged

 TCPIP Sv     1   0.6    22    14   1.5 ...      0
 FTP Serv     4   0.1     4     2   2.1 ...   2.37
 OurUsers     1   0.1     4     2   1.9 ...   1.18
 Monitors     1   0.0     1     1   2.5 ...   2.39
 CMS User     3   0.0     0     0   3.4 ...      0
...
...
...
 Sum/Mean    14   0.9    32    19   1.7 ...   5.95
```

Figure 55. Part of Sample UCLASS_RESOURCE_UTIL Report

Because TCP sends the information to FTP and FTP writes or reads to disk, the disk I/O for TCP should always be zero, as shown in Figure 56.

```
                     <-Message Rate-> ... <-Per Message->
                     IUCV+VMCF ...
           Log-                          DASD
 User       ged  Total           ...  SSCH   ...
 Class    Users   Msgs  Total    ... +RSCH   ...

 TCPIP Sv     1   3744  4.274    ...     0   ...
 FTP Serv     4   3400  3.882    ...  0.61   ...
 Monitors     1    585  0.668    ...  3.58   ...
 OurUsers     1    344  0.393    ...  3.00   ...
 ...
 ...
 ...
 Sum/Mean    14   8073  9.216    ...  0.64   ...
```

Figure 56. Part of Sample UCLASS_VMCOMM_ACTIVITY Report

When you look at the DASD SSCH +RSCH column, it is equal to the total number of disk I/O operations divided by the number of messages. In this example, the FTP servers issued 3400 * 0.61 or 2074 disk I/Os.

Network I/O is shown in the DIAG 98 column in Figure 57 and has the count for TCP/IP communication with the Interconnect Controller on its way to and from the network (in this case through an ESCON channel and the AIX operating system).

```
                    <-------Message Rate--->  ...
                    IUCV+VMCF ...
           Log-
User        ged  Total          ...   DIAG  ...
Class     Users   Msgs  Total   ...     98  ...

TCPIP Sv      1   3744  4.274   ... 14.897  ...
FTP Serv      4   3400  3.882   ...      0  ...
Monitors      1    585  0.668   ...      0  ...
OurUsers      1    344  0.393   ...      0  ...
...
...
...
Sum/Mean     14   8073  9.216   ... 14.897  ...
```

Figure 57. Part of Sample UCLASS_VMCOMM_ACTIVITY Report

If your installation is not using DIAGNOSE X'98' for TCP/IP I/O, then you can view I/O counts for network devices using the RTM/VM ESA commands, DISPLAY I/O or DISPLAY DEV.

AIX Tools

The following tables list tools that can be used to monitor the performance of TCP/IP on the AIX operating system. The tables include:

- Name of the tool
- Form of output the tool provides
- Data provided by the tool
- Standard reduction program to help analyze the data, if there is one

Tools to Monitor Transaction Rate and Throughput

Table 15 lists the AIX tools that can be used to monitor transaction rate and throughput.

Table 15. AIX Monitoring Tools for Transaction Rate and Throughput

Tool	Output	Data	Reduction
ftp client	Screen data	KB per second	None
netpmon	Standard output	Packets, reads, writes, and bytes per second	None

Tools to Monitor Response Time

Table 16 lists the AIX tools that can be used to monitor response time.

Table 16. AIX Monitoring Tools for Internal Response Time

Tool	Output	Data	Reduction
netpmon	Standard output	Response time for sending, receiving, and processing packets, reads, writes, and rpcs	None

Tools to Monitor CPU Utilization

Table 17 lists the AIX tools that can be used to monitor CPU utilization.

Table 17. AIX Monitoring Tools for CPU Utilization

Tool	Output	Data	Reduction
iostat -t	Standard output	% user busy, % systems busy	None
sar -u	Standard output	% user busy, % systems busy	Other forms of sar command
time/timex	Standard output	% user busy, % systems busy	None
ps u or v	Standard output		None
netpmon	Standard output	CPU use for network activity	None

Tools to Monitor Storage Consumption

Table 18 lists the AIX tools that can be used to monitor storage consumption.

Table 18. AIX Monitoring Tools for Storage Consumption

Tool	Output	Data	Reduction
sar -r	Standard output	Page faults per second	None
ps v	Standard output		None
vmstat	Standard output		None

Tools to Monitor Disk I/O Rate

Table 19 lists the AIX tools that can be used to monitor disk I/O rate.

Table 19. AIX Monitoring Tools for Disk I/O Rate

Tool	Output	Data	Reduction
iostat -d	Standard output	% active KB per second, transfers per second	None
filemon	Standard output	Reads, writes, response times, sizes	filemon

Tools to Monitor Communication Device Utilization

Table 20 lists the AIX tools that can be used to monitor communication device utilization.

Table 20. AIX Monitoring Tools for Communication Device Utilization

Tool	Output	Data	Reduction
netstat	Screen data		None
nfsstat	Screen data		None

The commands to monitor TCP/IP on the AIX operating system are standard commands that come with the operating system. They are covered very thoroughly in the *IBM AIX Version 3.2 for RISC System/6000 Performance Monitoring and Tuning Guide*. The level of expertise necessary to use the commands varies from end user to system programmer.

filemon Command

This command monitors the performance of the file system and reports the I/O activity. There is some trace analysis overhead associated with it.

ftp

As shown in "Using FTP" on page 14, the response time and throughput are shown every time a user transfers a file using FTP.

iostat Command

This command monitors terminal and disk I/O activity. You can run it to get cumulative statistics or statistics for successive intervals. Specifically, iostat:

- -d limits the report to the physical volume data
- -t limits the report to the controlling workstation and CPU data

netpmon Command

This command monitors activity and reports statistics on network I/O and network-related CPU usage. There is some CPU overhead associated with using it.

netstat Command

This command that can be used to monitor TCP/IP. The following list is a subset of the netstat parameters that you might want to use:

Coll Displays the number of input or output collisions occurring on SLIP interfaces

Mtu Displays the largest packet allowed on a particular interface

Packets Displays the number of incoming and outgoing packets for an interface

-i Displays statistics about configured interfaces

-m Displays statistics recorded by the network memory management routines, including the number of assigned mbuf structures

-v Displays statistics about network devices

-I Displays statistics about the network interface

nfsstat Command

This command displays information about the Network File System (NFS) and remote procedure calls (RPCs).

ps Command

This command displays information about active processes. Specifically, ps:

- u displays the following columns in the report: USER, PID, %CPU, %MEM, SZ, RSS, TTY, STAT, STIME, TIME, COMMAND
- v displays the following columns in the report: PID, TTY, STAT, TIME, PGIN, SIZE, RSS, LIM, TSIZ, TRS, %CPU, %MEM, COMMAND

sar Command

This command produces reports on system activity and resource usage. Specifically, sar:

- -r reports on paging statistics
- -u reports on CPU utilization

time and timex Commands

These commands report the elapsed time, user CPU time, and system CPU time for a command. The timex command has options that request accounting and system activity statistics.

vmstat Command

This command monitors real and virtual memory statistics. It also provides statistics on disk transfers, system traps, and CPU activity.

OS/2 Tools

The following tables list tools that can be used to monitor the performance of TCP/IP on the OS/2 operating system. The tables include:

- Name of the tool
- Form of output the tool provides
- Data provided by the tool
- Standard reduction program to help analyze the data, if there is one

Tools to Monitor Transaction Rate, Throughput, and Response Time

Table 21 lists the OS/2 tools that can be used to monitor transaction rate and throughput.

Table 21. OS/2 Monitoring Tools for Transaction Rate and Throughput

Tool	Output	Data	Reduction
FTP	Messages	KB/second	None
NETSTAT	Screen data		None

Tools to Monitor CPU Utilization

Table 22 lists the OS/2 tools that can be used to monitor CPU utilization.

Table 22. OS/2 Monitoring Tools for CPU Utilization

Tool	Output	Data	Reduction
PULSE	Interactive graphing	% CPU utilization	None
SPM/2	Screen print, File	CPU seconds	SPM/2

Tools to Monitor Storage Consumption

Table 23 lists the OS/2 tools that can be used to monitor storage consumption.

Table 23. OS/2 Monitoring Tools for Storage Consumption

Tool	Output	Data	Reduction
SPM/2	Screen, File	Memory analysis	SPM/2
Theseus/2	Screen, File	Memory Analysis	SPM/2

Tools to Monitor Disk I/O Rate

Table 24 lists the OS/2 tools that can be used to monitor disk I/O rate.

Table 24. OS/2 Monitoring Tools for Disk I/O Rate

Tool	Output	Data	Reduction
SPM/2	Screen, File	Disk I/O analysis	SPM/2

FTP

As shown in “Using FTP” on page 14, the response time and throughput are shown every time a user transfers a file using FTP.

NETSTAT

This command that can be used to monitor TCP/IP. The following list is a subset of the NETSTAT parameters that you might want to use:

-i Displays statistics about the IP layer

-n Displays statistics about LAN interfaces

-m Displays statistics about memory buffer usage

-r Displays statistics about routing tables and corresponding network interfaces

-t Displays statistics about the TCP layer

-u Displays statistics about the UDP layer

The following example shows the `netstat -i` command:

```
¼C:\‘netstat -i
total packets received 8087
checksum bad 0
packet too short 0
not enough data 0
ip header length < data size 0
ip length < ip header length 0
fragments received 0
frags dropped (dups, out of space) 0
fragments timed out 0
packets forwarded 0
packets rcvd for unreachable dest 75
packets forwarded on same net 0
requests for transmission 590
output packets discarded because no route could be found 0
input packets delivered successfully to user-protocols 8012
input packets with an unknown protocol 0
output packets successfully fragmented 0
output fragments created 0
fragmentation failed 0
successfully assembled packets 0
```

For more information on the NETSTAT command, see the *TCP/IP for OS/2: Command Reference*.

PULSE

Pulse is a graphic representation of the use of the system. It shows how different activities affect the system and how much processor power is available for other programs. The system use is displayed in a dynamically updated graph in the window. Although you can opt to have the graph filled in with color, we do not recommend that you use this while measuring CPU utilization.

Since Pulse will rescale if the utilization is high, it will not always represent 0–100, but more like 90–100.

Pulse comes with OS/2. To use it:

1. Select OS/2 System
2. Select Productivity
3. Select Pulse

SPM/2

System Performance Monitor/2 (SPM/2) is used to analyze the performance of the hardware and software on an OS/2 system. You can use SPM/2 commands through either:

- The Presentation Manager interface
- An OS/2 prompt
- A C language program written to the application programming interface (API).

You can use SPM/2 to:

- Collect performance data about the OS/2 system and the applications running on it
- Display performance data via graphs in real time as well as in playback mode
- Create performance reports
- Calculate directory and file sizes
- Analyze memory at several levels, including the working set

Be aware that the more statistics you collect using SPM/2, the larger the impact on your system overhead, especially when graphing in real time.

THESEUS2*

THESEUS2 is supplied with SPM/2 and it monitors system usage in much more detail. Using THESEUS2's memory analysis, you start with a list of all active processes and navigate from one control block to another via the use of hyperblocks.

You could use THESEUS2 to find out:

- What is the memory usage or working set of the entire OS/2 system?
- What is the kernal memory usage?
- What is the memory usage or working set of a single process?

You can also use THESEUS2 to do program analysis functions. Using program analysis you could find out:

- What are the contents of control registers?
- What is at an address?
- What are the valid contexts for a shared linear address?
- Where is the code loaded?
- Is a process accumulating memory?

You can use THESEUS2 through the Presentation Manager interface or with a program written in either C or REXX to the API.

We recommend you use SPM/2 to monitor your OS/2 system, then using THESEUS2 only if you need a more detailed and complex analysis.

The SPM/2 and THESEUS2 manuals are delivered softcopy with the product.

DOS Tools

The following tables list tools used to monitor the performance of TCP/IP on DOS. The tables include:

- Name of the tool
- Form of output the tool provides
- Data provided by the tool
- Standard reduction program to help analyze the data, if there is one

Tools to Monitor Transaction Rate, Throughput, and Response Time

Table 25 lists the tools on DOS that can be used to monitor transaction rate and throughput.

Table 25. Monitoring Tools for Transaction Rate and Throughput on DOS

Tool	Output	Data	Reduction
FTP	Messages	KB/second	None
NETSTAT	Screen data		None

Tools to Monitor Storage Consumption

Table 26 lists the tools on DOS that can be used to monitor storage consumption.

Table 26. Monitoring Tools for Storage Consumption on DOS

Tool	Output	Data	Reduction
MEM or MEM/C	Screen	Memory analysis	None

FTP

As shown in "Using FTP" on page 14, the response time and throughput are shown every time a user transfers a file using FTP.

MEM Command

Figure 58 shows an example of the MEM command:

```
C:\>mem


    655360 bytes total memory
    654336 bytes available for DOS
    610512 largest executable program size

   2473984 bytes total EMS memory
   2097152 bytes free EMS memory

   2031616 bytes total XMS memory
   2031616 bytes available XMS memory
         0 bytes available contiguous extended memory
           64Kb High Memory Area available
```

Figure 58. Example of the MEM Command on DOS

Figure 59 shows an example of the MEM /C command:

```
C:\>mem /c

Memory below 640K (Conventional Memory)

  Name                Size in Decimal       Size in Hex
-------------      ---------------------   -------------
  DOS                 32176      ( 31.4K)       7DB0
  COMMAND              4880      (  4.8K)       1310
  APPEND               6480      (  6.3K)       1950
  FREE                  192      (  0.2K)         C0
  FREE                610512      (596.2K)      950D0
       Total FREE:     610704  (596.4K)



Total bytes avail to programs              :     610704  (596.4K)
Largest executable program size            :     610512  (596.2K)

   2473984 bytes total EMS memory
   2097152 bytes free EMS memory

   2031616 bytes total XMS memory
   2031616 bytes available XMS memory
         0 bytes available contiguous extended memory
           64Kb High Memory Area available
```

Figure 59. Example of the MEM /C Command on DOS

NETSTAT Command

This command that can be used to monitor TCP/IP. The following list is a subset of the NETSTAT parameters that you might want to use:

-i Displays statistics regarding configured interfaces
-m Displays statistics for the network's private buffer pool

For more information on the NETSTAT command, see the *TCP/IP for DOS: Command Reference*.

OS/400* Tools

For more information on performance monitoring on the OS/400 operating system, see the *AS/400 Programming: Work Management Guide* and the *AS/400 Programming: Performance Tools/400 Guide*.

Chapter 5. Tune Your Network Environment

In this chapter, tuning changes that you can make are listed by operating system within each topic to improve performance in the following areas:

- Transaction rate, throughput, and response time
- CPU utilization
- Storage consumption
- Disk I/O rate
- Communication device utilization
- Specific software applications

There are also guidelines for SNALINK performance.

Improving Transaction Rate, Throughput, and Response Time

Throughput can be tuned across TCP/IP by working with the send and receive buffer size and packet size. You can also increase throughput by relieving constraints on storage and disk I/O.

Table 27 shows the commands to view the buffer and packet sizes across the operating systems.

Table 27. How to View Send/Receive Buffer Size and MTU by Operating System

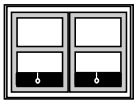
Host	Send/Receive Buffer Size	MTU/Packet Size
MVS	Look at TCP/IP profile or use the NETSTAT ALL command	Use the NETSTAT GATE command
VM	Look at TCP/IP profile or use the NETSTAT ALL command	Use the NETSTAT GATE command
AIX	Use the NO -A command	Use the NETSTAT -I command
OS/2	(Always set to 28KB)	Use the NETSTAT -N command
DOS	Look at the TCPDOS.INI file	Look at the TCPDOS.INI file
OS/400	Use the Display Connection Status (DSPCNNSTS) command (screen 3 of 4)	Use Work with TCP/IP Network Status (WRKTCPSTS) command or the NETSTAT command, then display TCP/IP route information

You might want to make another table like this for any non-IBM operating systems where you are running TCP/IP to use for quick reference.

After tuning TCP/IP using send and receive buffer size and MTU size, consider relieving another constraint with storage or disk I/O to further improve response time.

You should also look at which host is doing any data translation. Make sure that the translation is being done on the faster or less constrained computer.

Work with Send and Receive and Other Key Buffer Sizes



As discussed in “Understanding TCP/IP Send and Receive Buffers” on page 6 and “Changing the Send and Receive Buffer Size” on page 31, you can change the send and receive buffer sizes to alter performance. You affect the window size that TCP uses by changing the size and amount of send and receive buffers.

Before changing the size and number of the various types of data buffers, you should carefully consider the possible impact on the system.

Determining the Optimum Buffer Size

You can determine the theoretical optimum size of your send and receive buffers.

1. Monitor the round-trip time for a packet to a typical host using PING. (We recommend that you do a series of PINGs to get a better survey of the possible round-trip times.) Be sure to use the average packet size for the length of the PING, not the default size.
2. Determine the theoretical bandwidth of the transmission medium you are using. For example, a 10Mb Ethernet has a bandwidth of 1.2MBps, and a 16Mb token ring has a bandwidth of 2MBps.

   If you have multiple links of different LAN media on your network, use the bandwidth of the ***slowest*** link to perform the calculation.
3. Use the following formula:

   ```
   round-trip time * maximum network bandwidth = optimum window
   ```

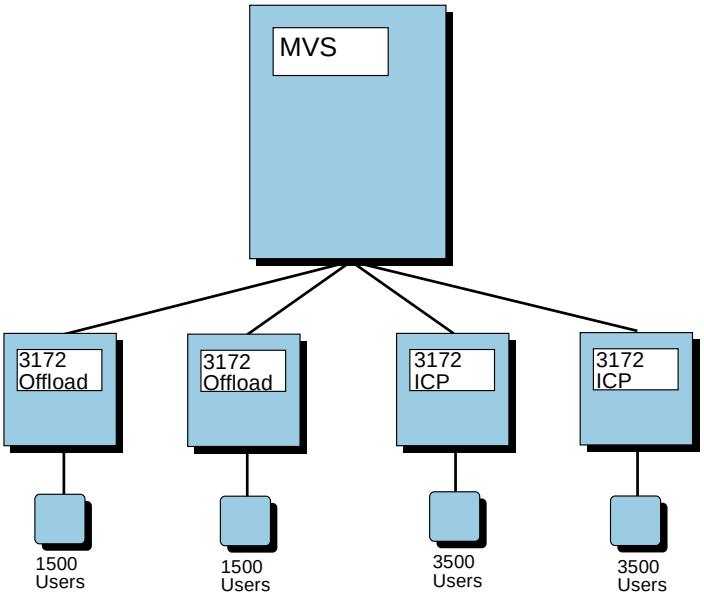
   For example, if you determine a round-trip time of 15 milliseconds and a token-ring bandwidth of 2MB per second, then:

   ```
   0.015 seconds * 2,000,000 bytes per second = 30,000 bytes or 30KB
   ```

Using a smaller buffer size than the optimum window size will decrease throughput; using a larger size will not improve throughput.

This calculation is only a guess at the optimum buffer size since not all your traffic will be going to the same host.

Work with Packet Size



As discussed in “Understanding Fragmentation” on page 10 and “Changing the Packet Size” on page 32, the MTU is the number of bytes that can be handled by the host or network. Some networks limit the packet size to a smaller value. Recommended values for packet size are:

- 1492 bytes for Ethernet 802.3
- 1500 bytes for Ethernet Version 2 IEEE
- 1500 bytes or 2KB or 4KB for token ring
- 2046 bytes for frame relay
- 4352 bytes or 2KB or 4KB for FDDI
- 65 515 bytes for HPPI (for MVS)
- 32 756 bytes for HPPI (for VM)
- 65 520 bytes for CTC (for MVS)
- 32 768 bytes for CTC (for VM)

Note: Packet size is determined during the connection establishment between client and server. The packet size for each is compared, and the smaller size is used. Because TCP/IP for OS/2 attempts to optimize the combination of values used for segment size and TCP window size, and you may get unexpected results, see “Getting Unexpected Results After Changing the MTU Size” on page 84 for more information.

Determining the Optimum Packet Size

Throughout this guide, our advice on packet size and send and receive buffer size has been “bigger is better.” But bigger is not ***always*** better. Figure 60 shows throughput plotted against packet size.

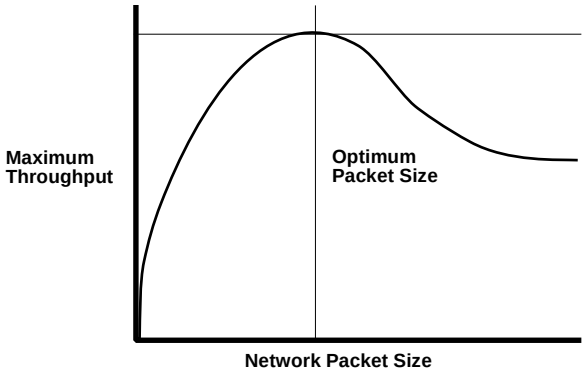


Figure 60. Maximum Throughput As a Function of Packet Size

As packet size is increased, throughput increases, but only up to a certain point. After that point, throughput decreases as packet size is increased. One reason is that the TCP window has a fixed limit to its size. If we could keep increasing the size of the window as we continued to increase the packet size, we would not see this drop-off in throughput.

If you have a system where the TCP window size is kept small and the packet size is very large, you might see this effect. However, most adapters and software implementations limit the packet size to a small value relative to the TCP window size, so you might never see this happen.

Bigger is not always better for send and receive buffer size either. If the packet size is kept very small and the number of hops between systems is small, then a bigger TCP window will not always improve throughput. At some point there will be enough capacity in the TCP window so that throughput cannot be increased by increasing the window size.

On the MVS and VM Operating Systems

The following key areas affect the transaction rate and throughput:

- Send and receive buffer amount and size
- Packet size

These areas can be tuned using statements in the configuration data set or file. For more detailed information about these statements, see the *TCP/IP for MVS: Customization and Administration Guide* for the MVS configuration data set (*hlq*.PROFILE.TCPIP) or *TCP/IP for VM: Planning and Customization* for the VM configuration file (PROFILE TCPIP).

Work with Send and Receive Buffers

There are several kinds of data buffers that influence window size.

- Data buffers
- Small data buffers
- Tiny data buffers
- Envelopes
- Large envelopes

The number of data buffers is limited only by the amount of virtual storage.

For more information on how to determine how many you need of each type of data buffer, see “Example of Analyzing Buffers Used by FTP on MVS” on page 175.

Using the DATABUFFERPOOLSIZE Statement: Use this statement to specify the size of the buffers in the free pool that are used to hold data during TCP/IP address space processing. You can also use it to specify the number of data buffers.

Here is an example of a DATABUFFERPOOLSIZE statement to create 100 data buffers of 32KB each:

```
DATABUFFERPOOLSIZE          100 32768
```

The size of the data buffer can be any of the following byte amounts:

- 8 192 (default for MVS V3R1 without performance PTFs)
- 12 288
- 16 384 (default for to MVS V3R1 with performance PTFs and also V3R2)
- 24 576
- 28 672
- 32 768
- 49 152

If you are tuning your MVS system, you can also use the following sizes:

- 65 536
- 98 304
- 131 072
- 196 608
- 262 144

Using larger data buffers will often increase throughput for a file transfer. We recommend that you use 32 768 bytes. If you are only using IBM's TCP/IP with IBM's Offload, then you can decrease the size to 28 672 bytes.

The default number of regular data buffers is 160. Running out of data buffers causes the abnormal ending of active connections.

Using the SMALLDATABUFFERPOOLSIZE Statement: Use this statement to specify the number of small data buffers. Small data buffers hold 2048 bytes of data, in contrast to regular data buffers, which hold 8192 bytes or more. They are used by TELNET server TCP connections. Each Telnet server connection holds one buffer while idle. Small data buffers are also required for any 3172 Offload feature connection.

If there is no small data buffer pool defined, Telnet will use regular data buffers. Using small data buffers in the place of regular data buffers saves at least 6144 (8192 – 2048) bytes of virtual storage. More virtual storage is saved if the buffer size specified on the DATABUFFERPOOLSIZE statement is larger than 8192.

The default number of small data buffers is 0.

Here is an example of the SMALLDATABUFFERPOOLSIZE statement:

```
SMALLDATABUFFERPOOLSIZE     1200
```

Using the TINYDATABUFFERPOOLSIZE Statement: Use this statement to specify the number of tiny data buffers. Tiny data buffers hold 256 bytes of 3172 Offload for OS/2 control information. You must have enough tiny data buffers for TCP/IP to work correctly with 3172 Offload for OS/2. If you run out of tiny data buffers, TCP/IP will not be able to communicate with the Offload host. The default number of tiny data buffers is 0.

Here is an example of the TINYDATABUFFERPOOLSIZE statement:

```
TINYDATABUFFERPOOLSIZE      500
```

Using the ENVELOPEPOOLSIZE Statement: Use this statement to specify the number of small envelopes. Envelopes hold datagrams and fragments during TCP/IP processing. They can be small or large. The smaller envelopes hold 2048 bytes of data. The default number of envelopes is 750.

Here is an example of the ENVELOPEPOOLSIZE statement:

```
ENVELOPEPOOLSIZE            750
```

Using the LARGEENVELOPEPOOLSIZE Statement: Use this statement to specify the size and number of large envelopes. A large envelope is used only if a packet does not fit in a small envelope. Also, large envelopes are used to hold IP datagram fragments during reassembly because the datagram size is not known until reassembly is complete.

You can change the number of large envelopes from the default of 50. Running out of large envelopes causes TCP/IP to drop outgoing and incoming packets. The resulting retransmission of lost packets lowers performance.

Here is an example of the LARGEENVELOPEPOOLSIZE statement:

```
LARGEENVELOPEPOOLSIZE       50 8192
```

The size of the large envelope can be any of the following byte amounts:

- 8 192 (default)
- 16 384
- 32 768
- 65 535 (MVS only)

Work with Packet Size

The MTU can be no larger than the large envelope size. You can change the MTU using the GATEWAY statement entry pertaining to the network connection.

If you are using ROUTED, you will need to change the MTU using the BSDROUTINGPARMS statement instead of the GATEWAY statement.

Do not assume that just because you can modify the MTU size on MVS or VM, that the adapter, controller software, or client software will support the same size packet.

Using the GATEWAY Statement: Use this statement to specify the MTU size.

Here is an example of a GATEWAY statement to set the MTU to 16 384 for the LINK1 interface:

```
GATEWAY
  * Network  first-hop   driver  packet-size  subnet-mask
    192.8.4  14.0.0.10    LINK1   16384          0
```

Using the BSDROUTINGPARMS Statement: Use this statement to specify the MTU size if you are using ROUTED. You can say whether or not the packet size you specify for the interface is always used, regardless of the final destination host, or if the default maximum packet size of 576 is used when sending to networks that are not locally attached.

Here is an example of a BSDROUTINGPARMS statement to set the MTU to 16 384 for the LINK1 interface:

```
BSDROUTINGPARMS false
  ; link   maxmtu  metric  subnet mask    dest addr
    LINK1  16384    0      255.255.255.0   0
ENDBSDROUTINGPARMS
```

To Take Advantage of a 64K Large Envelope Size: Using a CTC or HPPI connection between to MVS systems, you can take advantage of the 64K large envelope size by specifying packet size on the GATEWAY or BSDROUTINGPARMS statement to be:

- 65 520 for CTC, is the maximum value
- 65 515 for HPPI, which allows for the 40-byte HPPI FP/LE header

On the AIX Operating System

The following key areas affect the transaction rate and throughput in any AIX environment:

- Send and receive buffer size
- MTU (packet) size

The mbuf pool size can also affect response time in the AIX environment. Refer to the section on that subject in the *AIX for RISC System/6000 Performance Monitoring and Tuning Guide*.

Work with Send and Receive Buffer Size

Higher throughput occurs with the maximum possible send and receive buffer size because fewer acknowledgements are needed. When running on the AIX operating system on a PS/2, send and receive buffer sizes are fixed at 16KB. On the RISC System/6000, you can make the send and receive buffer size equal to 4, 16, 32 or 64KB. The default is 16KB.

If you have root authority, the NO command can also be used to change the size of the send and receive buffers. Use NO -A to view the current setting. To make these buffers as large as possible, issue the following commands:

```
no -o tcp_sendspace=32768
no -o tcp_recvspace=32768
no -o udp_sendspace=32768
no -o udp_recvspace=32768
```

To permanently change the size of these buffers, you can edit the /etc/rc.net file.

Work with MTU (Packet) Size

If you have root authority, you can change the maximum packet size you can use with TCP/IP for AIX using the smit tcpip or ifconfig commands.

1. Type smit tcpip.
2. Choose the ***Further Configuration*** option.
3. From that menu, go to the ***Network Interfaces***.
4. At the ***Network Interface Drivers*** screen, select an interface.
5. Change the maximum IP datagram size for this device.

Before changing the MTU using the ifconfig command, use the netstat command to get the existing value, as shown in Figure 61.

```
root@fester!16# netstat -in -I fi0
Name  Mtu   Network     Address         Ipkts    Ierrs Opkts    Oerrs Col
fi0   1500  <Link>                         10231     0    17612     0
fi0   1500  9.67.115    9.67.115.24        10231     0    17612     0
root@fester!17# ifconfig fi0 down
root@fester!18# ifconfig fi0 mtu 4352
root@fester!19# ifconfig fi0 up
root@fester!20# netstat -in -I fi0
Name  Mtu   Network     Address         Ipkts    Ierrs Opkts    Oerrs Col
fi0   4352  <Link>                         10231     0    17612     0
fi0   4352  9.67.115    9.67.115.24        10231     0    17612     0
root@fester!21#
```

Figure 61. Example of the ifconfig Command to Change the MTU on the AIX Operating System

On the OS/2 Operating System

Work with Send and Receive and Other Key Buffer Sizes

On OS/2 the send and receive buffer size cannot be changed; it is set at 28KB.

Work with Packet Size

The MTU can be changed on the Configure Network Interfaces panel. The default number is 1500 bytes. To get to the Configure Network Interfaces panel:

1. Select the ***TCP/IP*** icon.
2. Select the ***Configuration*** icon.
3. Select ***Configure Network Interfaces***.

You can also change the packet size on OS/2 using the IFCONFIG command. In Figure 62 we used the NETSTAT -n command to get the LAN interface statistics before changing the MTU.

```
C:\>netstat -n
Interface 0: 802.5
physical address    10005a8dbce2      MTU 1500

speed 4000000 bits/sec
unicast packets received 31621
broadcast packets received 5052340
total bytes received 380194360
unicast packets sent 41108
broadcast packets sent 13
total bytes sent 3606713
packets discarded on transmission 1
packets discarded on reception 1673
received packets in error 0
errors trying to send 0
packets received in unsupported protocols 0

C:\>ifconfig lan0 down

C:\>ifconfig lan0 mtu 2000
```

Figure 62. Example of the NETSTAT -n and IFCONFIG Commands on OS/2

Then we brought the LAN interface back up and used the NETSTAT -n command to get the LAN interface statistics, as shown in Figure 63.

```
C:\>ifconfig lan0 up
C:\>netstat -n
Interface 0: 802.5
physical address    10005a8dbce2      MTU 2000

speed 4000000 bits/sec
unicast packets received 31623
broadcast packets received 5052681
total bytes received 380219685
unicast packets sent 41110
broadcast packets sent 13
total bytes sent 3606883
packets discarded on transmission 1
packets discarded on reception 1673
received packets in error 0
errors trying to send 0
packets received in unsupported protocols 0
```

Figure 63. Example of the IFCONFIG and NETSTAT -n Commands on OS/2

You might also need to configure the adapter to allow a larger packet size. For example, you have to configure your token-ring adapter to support the transmission of buffers that are at least as large as the new MTU.

Configuring the Hardware Adapter: You might also need to configure the hardware adapter to allow a larger packet size. For example, you will need to configure your token-ring adapter to support the transmission of buffers that are at least as large as the new MTU.

For IBM token-ring network adapters, you can:

1. Invoke LAPS.
2. Choose the edit option after highlighting your type of adapter.
3. Change the TRANSMIT BUFFER SIZE field.

Or you can:

1. Edit the PROTOCOL.INI file.
2. Add or change the XmitBufSize= parameter under IBMTOK_nif.

For the IBM LANStreamer* adapter, you can:

1. Invoke LAPS.
2. Choose the edit option after highlighting the LANStreamer adapter NDIS device driver.
3. Change the SIZE OF ADAPTER DRIVER RECEIVE BUFFER field.

Or you can:

1. Edit the PROTOCOL.INI file.
2. Add or change the SizWorkBuf= parameter under IBMTRDB_nif.

Note: We recommend that you use LAPS unless you are an experienced OS/2 system programmer.

Getting Unexpected Results After Changing the MTU Size: TCP/IP for OS/2 attempts to optimize the combination of values used for segment size and TCP window size for best performance of TCP applications such as FTP. This optimization can cause unexpected results, so read the following information before changing the MTU. (To review how the TCP windows work, see "Understanding TCP/IP Send and Receive Buffers" on page 6.)

- TCP/IP always sets the maximum window size to be a multiple of the maximum segment size (MSS). For example, the default MSS is 1460 bytes and the corresponding maximum window size is 27740 bytes, avoiding wasted space.

  The TCP segment is the data portion of the packet (excluding IP and TCP headers), and is the part of the packet that gets stored in the TCP window. If the maximum window size is not a multiple of the MSS, then it is possible for the window to have some room available but not enough to hold a segment of data.

- When you increase the MTU beyond 2088 bytes, TCP/IP sets the MSS to be a multiple of 2048. In order to set the MSS, TCP/IP starts with the MTU size and subtracts 40 (the combined length of the IP and TCP headers). The remaining value is decreased to the next lowest multiple of 2048.

The following example shows how setting the MTU might produce unexpected results:

1. You set the MTU for your token-ring adapter to 4096 bytes.
2. You use FTP to transfer a file to your host from another host that is connected to the same token ring and also has the MTU set to 4096.
3. The MTU remains 4096, since both hosts agree to it.
4. When the FTP connection is established, your host sets the MSS to a multiple of 2048 that is less than or equal to 4096 – 40. Since the only multiple of 2048 that is less than or equal to 4056 is 2048, the MSS is set to 2048.
5. TCP/IP transfers the file. The file is sent in packets no larger than the MSS plus 40 bytes. Therefore the largest packets sent are 2088 bytes, even though the MTU is 4096.

Unless you are using a network analyzer and keeping track of the packet sizes, the only evidence you will have of this unexpected result is lower-than-expected throughput.

The solution is to ***specify an MTU at least 40 bytes larger than a multiple of 2048***. If you want to send the data in 4KB segments, you need to specify an MTU of at least 4136. Please note that it is still necessary that your adapter support the MTU size you choose, not just the MSS being used.

On the DOS Operating System

Work with Send and Receive Buffer Size

The default send and receive buffer size is 16KB. To change the size of these buffers, you can change the following fields in the TCPDOS.INI file:

- tcp.sendspace
- tcp.recvspace

Work with Packet Size

For DOS (unlike other platforms), the MTU value determines the size of the packet within your network, while the MSS value determines the size of the packet when the packet leaves your network.

- To change the size of the MTU, change the DEFAULT.MTU field in the TCPDOS.INI file. The default number is 1496 bytes.
- To change the size of the MSS, change the TCP.MSSDFLT field in the TCPDOS.INI file. The default number is 512 bytes.

In Figure 64 shows the use of the NETSTAT -ian command to get the LAN statistics showing the MTU size.

```
C:\>netstat -ian

Name Mtu   Network    Address           Ipkts Ierrs  Opkts Oerrs Coll
nd0  12888 <Link>                         3     0      1     0    0
      Media Type       = Token Ring
      Media Speed      = 16000000 (bps)
      Hardware Address = 10:00:5a:cc:0b:b0
nd0  12888 9          9.67.113.41          3     0      1     0    0
```

Figure 64. Example of Using the NETSTAT -ian Command to Get LAN Statistics

Configuring the Hardware Adapter: You might also need to configure the hardware adapter to allow a larger packet size. For example, you will need to configure your token-ring adapter to support the transmission of buffers that are at least as large as the new MTU size.

For IBM token-ring network adapters, you can:

1. Invoke the CUSTOM EXEC.
2. Choose the NDIS PROTOCOL MANAGER under CONFIGURE.
3. Edit the appropriate token-ring adapter type.
4. Add or change the XmitBufSize parameter.

The default value is 2040, which is fine for most older token-ring adapters. If you are using the 16/4 Adapter or the 16/4 Adapter/A, then change the default to the largest frame possible, which is 4096 for 4Mbps networks and 17952 for a 16Mbps networks. (For the Token-Ring adapter, the frame include the data portion only, with no headers.) Keep in mind, however, that any increase in the XmitBufSize value will require more workstation memory.

For the IBM LANStreamer adapter, you can:

1. Invoke the CUSTOM EXEC.
2. Choose the NDIS PROTOCOL MANAGER under CONFIGURE.
3. Edit the LANStreamer adapter.
4. Add or change the SizWorkBuf= parameter.

For optimum performance, the size of the adapter driver receive buffer (SizWorkBuf) value multiplied by the minimum adapter driver receive buffers (MinRcvBuffs) value must be greater than the size of the largest possible frame on your network. (For the LANStreamer adapter, the frame includes the data portion plus the IP header and the DLC header.)

The largest possible frame is 4500 bytes on a 4Mbps network and 18000 bytes on a 16Mbps network. Since the default value for SizWorkBuf (1120 bytes) multiplied by the default number of 6 for MinRcvBuffs totals equals 6720 bytes, you may want to change these values on a 16Mbps network. Keep in mind however that any increase in the SizWorkbuf or the MinRcvBuffs values will require more workstation memory.

On the OS/400 Operating System

The following key areas affect the transaction rate and throughput:

- Number of send and receive buffers
- Maximum datagram and frame size

For more information on tuning TCP/IP for the AS/400*, see the *Application System/400 Transmission Control Protocol/ Internet Protocol Guide*.

Work with the Number of Send and Receive Buffers

With OS/400, you can change the tuning values to influence performance. The tuning values that can affect performance (as opposed to function) are:

- Data buffers
- Data envelopes

Changes to the tuning values take effect the next time the QTCP subsystem starts.

To change the tuning values, use the **CFGTCP** command to go to the CONFIGURE TCP/IP screen. Select **14** for "Change TCP/IP tuning values" to go to the corresponding screen.

Changing the Amount of Data Buffers: You can change the number of data buffers but not the size. The default number of data buffers is 160, but you can specify the amount to be any number between 10 and 160. Each connection requires at least 2 data buffers. Too few buffers prevent new TCP/IP connections from being opened.

Changing the Amount of Data Envelopes: You can change the number of data envelopes but not the size. The default number of data envelopes is 750, but you can specify the amount to be any number between 40 and 1500. The number of data envelopes necessary depends on many factors including the amount of network traffic and the level of activity on the AS/400 system.

Running out of data envelopes causes TCP/IP to drop outgoing and incoming packets. The resulting retransmission of lost packets lowers performance.

Work with Maximum Datagram and Frame Size

To change the size of MTU, you should change both the maximum datagram size and the maximum frame size.

Changing the Maximum Datagram Size: To change the maximum datagram size, you need to either:

- Add a routing entry using the Add TCP/IP Route Entry (ADDTCPRTE) command
- Change an existing routing entry using the Change TCP/IP Route Entry (CHGTCPRTE) command

The value you fill in for the MAXDTGSIZE parameter should be influenced by the maximum frame size of the type of line that you are using. TCP/IP will use whichever value is lower as the maximum size of the datagrams sent over that line.

Changing the Maximum Frame Size: To change the maximum frame size, you need to create a line description for the physical line you are using.

- For Token Ring, use the Create Line Description (Token-Ring) (CRTLINTRN) command
- For Ethernet, use the Create Line Description (Ethernet) (CRTLINETH) command
- For X.25, use the Create Line Description (X.25) (CRTLINX25) command

These are the largest maximum frame sizes you can use:

- 1994 bytes for Token Ring
- 1496 bytes for Ethernet (802.3)
- 1500 bytes for Ethernet Version 2 IEEE
- 1024 bytes for X.25

Improving CPU Utilization

After tuning TCP/IP using packet size, consider decreasing CPU utilization. A faster CPU will improve your CPU utilization.

On the MVS and VM Operating Systems

Some ways to reduce CPU utilization on MVS and VM are to use:

- Larger buffer sizes
- Larger MTU size
- Offload
- DIAGNOSE X'98' for I/O on VM

If CPU is a constraint because other (non-TCP/IP) applications are competing for it, you might want to grant favored status to TCP/IP virtual machines or address spaces.

- For VM/ESA, use the SET QUICKDSP and SET SHARE commands.
- For VM/ESA 370 feature or VM/SP, use the SET PRIORITY, SET QDROP OFF, and SET FAVOR commands.
- Dispatching priority for address spaces on MVS

For more information, see the *CP Command and Utility Reference*.

Using Offload

Performance benchmarking has shown that 3172 Offload for OS/2 reduces mainframe host cycles. Here are some examples of host CPU utilization reduction:

- 30–55% with TCP/IP for MVS using FTP
- 25–50% with TCP/IP for VM using FTP
- 12–15% with TCP/IP for MVS and VM using Telnet

However, it is important to note that there is a decrease in throughput when using 3172 Offload for OS/2, up to 30% using FTP.

Each Telnet, FTP, or socket application session counts as a connection. There is a limit of 2040 connections with 3172 Offload for OS/2.

On the AIX Operating System

On AIX, you can reduce CPU utilization using a larger MTU size.

Improving Storage Use

After tuning TCP/IP using send and receive buffer size, consider acquiring more storage to support the increased use of TCP/IP without paging.

On the MVS Operating Systems

Using fewer data buffers reduces storage use.

Tuning for Telnet

When using Telnet, increase the number of small data buffers and decrease the number of regular data buffers. Monitor your buffer usage using the NETSTAT POOLSIZE command and adjust (increase or decrease) your buffer sizes accordingly.

Tuning for Pascal FTP

If you are running FTP on MVS, you can improve your storage usage by running a single C server instead of multiple copies of the Pascal server.

The Pascal FTP server is no longer supported under TCP/IP Version 3 Release 2. If you are using the Pascal FTP server (TCP/IP for MVS Version 3.1 or the FTP of any earlier version) you can decrease the disk I/O buffers (number of channel programs or NCP) to be used during data transfer by decreasing the value of either:

- NCP statement in the FTP.DATA data set
- NCP parameter on the FTP SITE subcommand
- NCP parameter on the FTP LOCSITE subcommand

Although having less buffers lowers your storage requirements, it also lowers the overlap of I/O operations.

When using the FTP server, use the following formula to estimate the storage used:

```
# of sessions * NCP value * BLKSIZE = storage
                ---------   -------
                 session    buffer
```

For example, if you have specified NCP=20 and if you have 10 simultaneous FTP sessions transferring files with 32K block sizes, TCP/IP needs over 6 megabytes of I/O storage below the line.

```
10 *   20    *  32KB  = 6.25MB
     -------   ------
     session   buffer
```

When using the FTP client, use the following formula to estimate the storage used:

```
# of sessions * NCP value * BLKSIZE = storage
```

The FTP client has no problem with BLKSIZE=32K and NCP=20 when REGION=2M or more.

```
1 * 20 * 32KB = .625MB
```

Tuning for the C-FTP Server

The disk I/O can be improved by increasing the value for BUFNO in either one of the following ways:

- BUFNO statement in the FTP.DATA data set
- BUFNO parameter using the FTP SITE subcommand

The default value for BUFNO is 5. The recommended value for BUFNO is 35. The valid range for a BUFNO value is 1–255.

BUFNO value will help in improving throughput for PUT. The storage size can be estimated using the following formula:

```
Number of sessions * BUFNO value * Record length = Storage
```

"Storage" is the I/O storage below the 16M line. An EXTRATASK statement defines the number of FTP tasks. Allowed value for EXTRATASKSj is 0–254.

On the VM Operating Systems

As with MVS, using fewer data buffers reduces storage use. With Telnet, you can increase the number of small data buffers and decrease the number of regular data buffers. Monitor your buffer usage using the NETSTAT POOLSIZE command and adjust (increase or decrease) your buffer sizes accordingly.

Use the CP SET RESERVE command to reduce paging in the TCP/IP virtual machine. For more information, see the *CP Command and Utility Reference*.

On the AIX Operating System

Using smaller send and receive buffers will reduce storage use.

Improving Disk I/O Rate

Along with increasing the size or amount of the data buffers to improve the disk I/O rate, you can:

- Buy more disks
- Create more paths to the disks
- Buy faster disks
- Use memory for disk caching

On the MVS Operating System

To improve the disk I/O rate on MVS, you can use:

- Larger data buffers (this might help)
- Larger block data set size
- Number of channel programs used by Pascal FTP server and client
- Number of buffers used by C-FTP server (BUFNO)
- Use of cached DASD controllers

Work with Data Set Block Size

The attributes of the data set to which a member is copied play an important role in effective throughput during a file transfer. The recommended block size for MVS is the number of records that fill a 1/2 track or about 23 440 on 3380s. This amount is a good trade-off between the largest number of bytes per read and the least unused space on a track.

If you can change the record format of your files, fixed block gives the best performance. Sequential data sets are recommended for performance rather than partitioned data sets (PDS).

Work with the Number of Channel Programs (Pascal FTP Server/Client)

If you are using the Pascal FTP server, (TCP/IP for MVS Version 3.1 or the FTP of any earlier version) you can increase the disk I/O buffers to be used during data transfer by increasing the value of either:

- NCP statement in the FTP.DATA data set
- NCP parameter on the FTP SITE subcommand
- NCP parameter on the FTP LOCSITE subcommand

Although having more buffers allows for a better overlap of I/O operations, it also increases your storage requirements.

Before using the SITE NCP command, use the STAT command to get the list of all the settings that can be changed and their current values. Figure 65 shows an example of the STAT command.

```
ftp> quote stat
211-Server FTP talking to host 9.67.113.18, port 3053
211-User: USER1    Working directory: USER1.
211-The control connection has transferred 321 bytes.
211-There is no current data connection.
211-The next data connection will be actively opened
211-to host 9.67.113.18, port 3053, using
211-mode Stream, structure File, type ASCII, byte-size 8.
211-Automatic recall of migrated data sets.
211-Automatic mount of direct access volumes.
211-Data set mode. (Do not treat each qualifier as a directory.)
211-Primary allocation 50 tracks. Secondary allocation 50 tracks.
211-Partitioned data sets will be created with 27 directory blocks.
211-FileType SEQ (Sequential - default).
211-Records in Parallel I/O buffer is 8
211-Number of Channel Programs is 3 1
211-RDW's from VB/VBS files are discarded.
211-Retention period is 0
211-DB2 subsystem name is DB2
211-SQL results sent in report (NOSPREAD) format.
211-SQLCOL (column headings) use NAMES
211-JESLRECL is 80.
211 Record format VB. Lrecl: 256, Blocksize: 6233
```

Figure 65. Example of the STAT Command to Show Current Status on MVS

Notice that **1** the number of channel programs was 3. Figure 66 shows an example of the SITE NCP command used to change the number of channel programs to 20, then the STAT command to display that the change was made.

```
ftp> quote site ncp=20
200 Site command was accepted
ftp> quote stat
211-Server FTP talking to host 9.67.113.18, port 3053
211-User: USER1    Working directory: USER1.
211-The control connection has transferred 1405 bytes.
211-There is no current data connection.
211-The next data connection will be actively opened
211-to host 9.67.113.18, port 3053, using
211-mode Stream, structure File, type ASCII, byte-size 8.
211-Automatic recall of migrated data sets.
211-Automatic mount of direct access volumes.
211-Data set mode. (Do not treat each qualifier as a directory.)
211-Primary allocation 50 tracks. Secondary allocation 50 tracks.
211-Partitioned data sets will be created with 27 directory blocks.
211-FileType SEQ (Sequential - default).
211-Records in Parallel I/O buffer is 8
211-Number of Channel Programs is 20 1
211-RDW's from VB/VBS files are discarded.
211-Retention period is 0
211-DB2 subsystem name is DB2
211-SQL results sent in report (NOSPREAD) format.
211-SQLCOL (column headings) use NAMES
211-JESLRECL is 80.
211 Record format VB. Lrecl: 256, Blocksize: 6233
```

Figure 66. Example of the SITE NCP command

C-FTP Server Performance

C-FTP server uses BUFNO instead of NCP (number of channel programs used by Pascal FTP server and client). BUFNO helps in improving the throughput for the C-FTP server for the PUT initiated by remote FTP client. BUFNO uses record length for determining the buffer size. The default value for BUFNO is 5. The recommended value for BUFNO is 35. The valid range for a BUFNO value is 1–255. BUFNO value can be changed in one of the following ways:

- BUFNO statement in the FTP.DATA data set
- BUFNO parameter on the FTP SITE subcommand

On the VM Operating System

To improve the disk I/O rate on VM, you can:

- Use larger data buffers
- Move minidisks across multiple paths
- Use minidisk cache (MDC)
- Use cached controllers

Moving Minidisks Across Multiple Paths

If you have control over where user files accessed by TCP/IP are stored on disk, spreading the files across multiple paths can give you substantial performance savings. The access time is longer when all the files are stored on minidisks in the same disk drive than across several disk drives.

On the AIX Operating System

To improve the disk I/O rate, you can change the way noncritical data is stored to disk. With noncritical data, do not use:

- Mirror Write consistency with COPIES > 1
- Write Verify

Not using these settings will make your disk I/O faster, but riskier.

Improving Communication Device Utilization

After tuning TCP/IP for the other objectives, you might want to look at tuning your communication device utilization to improve performance.

With LAN Adapters

The type of LAN media you use contributes to performance. It is important to define sufficiently large buffers when configuring network cards in workstations. For example, if you are using a token-ring adapter in a PS/2, you should set the buffer to 32KB. Using the Reference Diskette, opt to change (or set) the configuration. Look for the adapter name (in this case IBMTR Network 16/4 Adapter/A), then change the RAM Size and Address Range parameter from the default of 16KB to 32KB.

With 3172s Running ICP

When configuring the 3172 in ICP mode, there are parameters that can be set to affect performance. These parameters can be changed using the Operator Facility when configuring or updating the 3172 LAN gateway definition. The 3172 puts received LAN frames into blocks of up to 20KB before sending them over the channel to the host. Data is sent over the channel when the block is filled at 20KB or when the block delay timer interval has expired.

On the 3172-1

For the Token Ring 16/4 adapter, a record size can be set for a maximum outbound and inbound record length. These parameters define the largest record size that can be transmitted or received by the token-ring adapter. A record size larger than this value would be logged and then discarded by the 3172. For best performance, you should set both the maximum outbound and inbound record length to 2048.

For the LAN Gateway function, a block delay time and maximum response length can be set for each LAN adapter. The block delay time is the amount of delay permitted while received frames are blocked for retransmission. For best performance, you should set the block delay time to 10 milliseconds.

The maximum response length is the length of the longest frame to be sent to the host computer without blocking. Frames smaller than the maximum response length are sent to the channel without delay. Frames larger than the maximum response length are held for the block delay time before they are blocked and sent on. For best performance, you should set the maximum response length to 500 bytes.

On the 3172-3

For the LAN Gateway function, a block delay time and maximum response length can be set for each LAN adapter. For best performance, you should set the block delay time to 10 milliseconds and the maximum response length to 500 bytes.

MVS Data Compression for Network Bottlenecks

On MVS, you can use data compression for data that must pass through a node that is a network bottleneck. Compression lowers the amount of packets sent, although it does raise the CPU utilization when compressing or de-compressing the file.

Since only the end nodes have to do the compression and de-compression, the more intermediate nodes that the file will pass through, the better the cost justification for compression.

Use the MODE C subcommand for FTP to compress the data. For more information, see the *TCP/IP for MVS: User's Guide*.

TCP Timeout and Retransmission

X.25 networks often have relatively slow links in the path, producing long network delays. TCP/IP will time out, causing an increase in re-transmissions and the acknowledgments associated with them. The network can become clogged with extra packets and acknowledgments.

With TCP/IP for MVS, you can delay the time-outs so that packets are not re-transmitted so often. Also you can lower congestion in the network by delaying acknowledgments. Lowering the amount of packets is doubly important if you are being billed for all packets transmitted over the network, including the re-transmitted packets,

TCP typically follows certain rules in deciding when to retransmit packets. TCP uses the round trip time from the time when data was sent until the ACK was

received (RTT) to calculate the smoothed round trip time (SmoothTime) and the smoothed variance (SmoothVar) in the following formula:

```
Err = abs (SmoothTime - RTT)

SmoothTime = SmoothTime + RoundTripGain * (RTT - SmoothTime)

SmoothVar = SmoothVar + VarianceGain * (Err - SmoothVar)
```

When a data packet is sent, the values from these formulas are used to estimate the retransmission time-out (RTO) that is used to set the time-out value:

```
Interval = round (SmoothTime + VarianceMultiplier * SmoothVariance)

RTO = min (MaximumRetransmitTime, max (MinimumRetransmitTime, Interval))
```

TCP then sends a packet and sets the time-out to expire after RTO amount of time. If an ACK has not been received when the time-out expires, then the data is retransmitted.

Changing the Way TCP/IP Times Out

You can specify re-transmission parameters on the GATEWAY statement so that you can control when TCP/IP will time out and re-transmit a packet.

First, you monitor to see whether your system is re-transmitting too often. You can:

- Use an SNMP monitor, such as NetView
- Use cbsample
- Specify TCPIPSTATISTICS on the ASSORTEDPARMS statement, comparing the tcpRetransSegs value to the tcpOutSegs value to get the percentage of re-transmitted packets

If you decide that you need to change the time-out value, you can supply a new time-out range and variance on the GATEWAY statement. You can adjust the way TCP calculates its time-outs based on the following parameters on the GATEWAY statement:

- MINIMUMRETRANSMITTIME (default is 0.75 seconds)
- MAXIMUMRETRANSMITTIME (default is 60 seconds)
- ROUNDTRIPGAIN (default is 0.12)
- VARIANCEGAIN (default is 0.25)
- VARIANCEMULTIPLIER (default is 2.00)

TCP uses the following rules in deciding when to transmit:

- Use the minimum round-trip time as long as it is less than the MINIMUMTRANSMITTIME
- Use the average round-trip time if round-trip time is greater than or equal to the MINIMUMTRANSMITTIME
- Use the maximum round-trip time if round-trip time is greater than the MAXIMUMTRANSMITTIME.

You can use ROUNDTRIPGAIN, VARIANCEGAIN, and VARIANCEMULTIPLIER to tell TCP how heavily it should weight the most recent behavior of the network versus the long term behavior. If you use smaller values for these parameters, then TCP will attempt to correct for congestion only if the congestion is sustained.

With larger values, TCP will correct for congestion more quickly and the system will be more sensitive to variations in network performance.

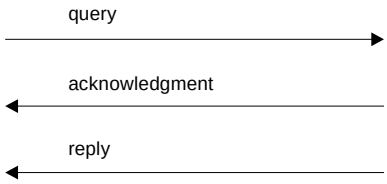
We recommend using the default values unless you find your re-transmission rate is too high.

Here is an example of a GATEWAY statement to set network 193.9.200 at LINK1 to use a MINIMUMRETRANSMITTIME of 2 seconds and a VARIANCEMULTIPLIER of 4:

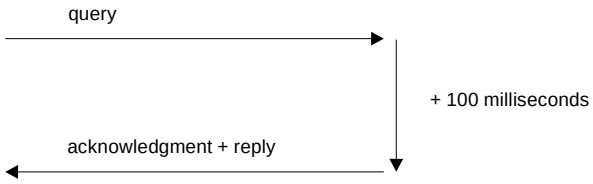
```
GATEWAY
; net_number first_hop link_name packet_size subnet_mask subnet_value
  193.9.200          =     LINK1 1500              0
             MINIMUMRETRANSMITTIME 2.0
             VARIANCEMULTIPLIER 4.00
 DEFAULTNET  193.0.2.3     LINK2 DEFAULTSIZE       0
```

Delaying Acknowledgments

You can delay the acknowledgments that TCP sends. Normally, when TCP/IP receives a TCP packet with the PSH bit on in the, header, it sends an acknowledgment (ACK packet) immediately.



Using the DELAYACKS parameter, TCP/IP will set a 100 millisecond timer. If the application has data to send to the other host within the 100 milliseconds, the acknowledgment will be combined with the data packet, resulting in one packet to the other host instead of two packets.



Delaying the acknowledgment is useful in applications that predominantly use a 2-way protocol in the passing messages, but may slow down bulk data transfers. For example, if you are sending lots of big files using FTP, you would not want to delay the acknowledgments.

You can specify the delay on either the PORT statement or the GATEWAY statement. Use the PORT statement if you will be specifying a particular port in the application. Here is an example of using the PORT statement to delay acknowledgments for any TCP connections on port 999:

```
PORT
     21 TCP FTPSERVE
     20 TCP FTPSERVE NOAUTOLOG
     23 TCP INTCLIEN
    999 TCP MYSERVER NOAUTOLOG DELAYACKS
```

If you want every application to use delayed acknowledgments, use the GATEWAY statement to specify them. Here is an example of a GATEWAY statement to delay acknowledgments for network 193.9.200 at LINK1:

```
GATEWAY
; net_number first_hop link_name packet_size subnet_mask subnet_value
  193.9.200          =     LINK1 576                  0
             DELAYACKS
 DEFAULTNET  193.0.2.3     LINK2 DEFAULTSIZE          0
```

Improving Performance of Specific Software Applications

After tuning TCP/IP for the other objectives, you might want to look at tuning your specific software applications to improve performance.

NPF

While running benchmarks, we have found some ways to tune NPF performance:

- Whether you use one or multiple NPF VTAM applications can make a difference. Significant throughput increases has been seen when using multiple NPF VTAM applications with CICS and NPF. CPU transaction time decreases with multiple NPF VTAM applications. Assign a different LUCLASS to each NPF VTAM application.

  Throughput increases at a higher rate than CPU utilization when using multiple NPF VTAM applications.
- Throughput may increase when you increase the number of NPF threads within an NPF VTAM and JES applications. Change the NPFVTAMTHREAD or NPFJESTHREAD parms specified in TCPIP.DATA data set.
- You can define a single FSS writer with multiple printers or multiple FSS writers with one printer each.
- Using NPF gives slightly lower throughput than file printing from a job or TSO command with LPR commands.
- As you increase the number of NPF transactions, throughput may decrease slightly.
- The longer the retain times are, the more DASD space will be used. Change the RETAINS (successful) and RETAINU (unsuccessful) parameters in the NPF routing file.
- Although increasing the number of retries will ensure printing reliability, it also increases the CPU utilization. Change the RETRYT (retry time) and RETRYL (# of retries) parameters in NPF routing file.

For more information about NPF, see the *TCP/IP for MVS: Network Print Facility* and see the *TCP/IP for MVS: Customization and Administration Guide* for the TCPIP.DATA data set.

Sockets

As the message size gets larger, MVS throughput increases and MVS CPU decreases compared to non UDP bulkmode.

- UDP Bulkmode may reduce CPU time by 63%

Socket Tuning Options:

- Small Partial Packets
  - TCP_NODELAY option using `setsockopt()` call
- Bulkmode
  - SO_BULKMODE option using SETIBMSOCKOPT() call
- Bulkmode
  - SOCKBULKMODE option in HLQ.TCPIP.DATA file
- Sockets
  - SOCKNOTESTSTOR option in HLQ.TCPIP.DATA file
- Sockets
  - NOUDPQUEUELIMIT option under assorted parameters in HLQ.xxxxxxxx.TCPIP file

Use of bulkmode can improve performance. The amount of improvement depends on the system load and the arrival pattern of the datagram messages at the socket. As the system load increases, the reduction in CPU usage because of bulkmode should also increase. When datagrams for the socket are processed, there should be an even greater reduction in CPU usage.

You can use the SOCKBULKMODE statement in a TCPIP.DATA data set for all sockets that access that data set or use the sock_do_bulkmode() socket call in specific socket applications.

To improve response time, you can disable checking for calls that attempt storage outside the caller's address space. You can use the SOCKNOTESTSTOR statement in a TCPIP.DATA data set for all sockets that access that data set or use the sock_do_teststor() socket call in specific socket applications.

See the *TCP/IP for MVS: Customization and Administration Guide* for the TCPIP.DATA data set and *TCP/IP for MVS: Application Programming Interface Reference* for the socket calls.

CICS Sockets

There are several adjustments that you can make to improve the ability of CICS sockets to handle many open transactions. They are:

1. You can set the priority of the listener transaction, CSKL, high enough so that it can dispatch incoming requests. CSKL has a limited queue for incoming requests. If it get too full, it will drop incoming requests. We recommend a priority of 255.
2. The priority of the enabling and disabling transactions, CSKE and CSKD, should also be high to allow for control of the socket interface under high loads.

3. When you are sending long running batch transactions, you need to specify correct parameters for CICS system input table in the SIT module or in SYSIN.
   - You may need to increase the MXT (maximum number of running tasks) parameter to allow room for short transactions.
   - Increase the AMXT (maximun number of active tasks) parameter so that both batch and interactive tasks can be running.
   - After monitoring your storage use with the stats records, you may find that you need to adjust the storage parameters and cushions to allow the long transactions to run. Look at the CDSA, UDSA, ECDSA, and EUDSA parameters.

In order to monitor the CICS transactions, you should enable monitoring in the monitor control table (MCT). Create the MCT with the monitor options you want, then specify the MCT in your SIT or SYSIN. Then, enable monitoring using the CEMT transaction or with SYSIN. If you use SYSIN, specify:

```
MN=ON
MNEXC=ON
MNENV=ON
```

For more information about CICS sockets, see the *TCP/IP for MVS: CICS TCP/IP Socket Interface Guide and Reference*.

ADSM with V3R2

ADSM automatically archives data from designated files on LAN attached hard disks. It is used in situations where companies cannot risk the loss of valuable data due to hard disk problems.

ADSM can use TCP/IP as a transport mechanism. ADSM users are, therefore, very interested in how TCP/IP will perform with this product.

A set of measurements were performed using ADSM with TCP/IP V3R1 and V3R2. Some of the improvements noted in V3R2 are as follows:

- Backup Function
  - 33.6% reduction CPU cycles
  - 12% improvement in throughput
- Restore Function
  - 48% reduction CPU cycles
  - Equivalent throughput

For additional information on tuning performance, see Appendix A, “Performance Tuning Tips for MVS” on page 183.

Improving SNALINK Performance

SNALINK can be used on MVS, VM, and OS/2 operating systems.

On the MVS Operating System

This section gives some guidelines to follow when improving the performance of SNALINK on MVS.

1. Use the largest MaxRuCode possible that VTAM* and NCP will support. You can use several SNALINK virtual machine tasks for each MaxRuCode to maximize your connection.

   For NCP, the parameters are MAXDATA and MAXBFRU. For VTAM, the parameter is RACMLUBF in both ISTRACON and MAXBFRU. For more information about the maximum request or response unit (RU) size used, see the *VTAM Programming*.

2. If you use SNALINK across a CTC connection, use a value other then zero for the retry delay parameter on the VTAM CTC definition.

When in doubt, consult your VTAM network administrator. For more information, see the *TCP/IP for MVS: Customization and Administration Guide*.

Using SNALINK LU6.2

This section gives some guidelines to follow when improving the performance of SNALINK LU6.2 on MVS.

Using the INIT option: The overhead in establishing an SNA LU type 6.2 session can be high. In fact, the delay in establishing a connection could be long enough to cause a Telnet or FTP command to time out during initialization before the connection can be established.

Here is an example of the use of INIT option:

```
MODIFY SNALNK62 , RESTART INIT
```

The INIT option in the SNALINK LU6.2 configuration data set can be used to establish the connection to a specific destination during the initialization of the SNALINK LU6.2 address space, rather than when the first packet is sent or received. The performance will then be consistent for all users of the connection.

If you use the INIT option to end the connection, using zero for the time-out value will not cause inactive time.

You should use the INIT option for all destinations that will be connected using dependent logical units (LUs). Since transaction programs using dependent LUs cannot initiate connections, the connection must be initiated by the host. The most convenient way to do this is with the INIT option. Also, you should specify a zero time-out value for all destinations using dependent LU connections.

Changing the Region Size: The SNALINK LU6.2 address space typically requires a region size of at least 132KB for a minimal configuration. The region size might increase significantly depending on the MTU and the number of destinations and links defined.

The SNALINK LU6.2 interface has been designed to isolate established connections from errors caused by insufficient storage available to establish new connections or links. If insufficient storage is available to establish either an IUCV link with the TCP/IP address space, or a VTAM connection with a destination node, the

connection attempt is abandoned and error messages are written to the SYSPRINT data set. Already established connections will continue unaffected. This way the impact to users is reduced, and you can increase the region size at a convenient time.

You can use the MODIFY subcommand RESTART ALL to establish connections with all destinations defined in the SNALINK LU6.2 configuration. This is a convenient way to verify that a sufficiently large region size has been defined to allow all connections to be established concurrently.

Here is an example of using the MODIFY command to restart:

```
MODIFY SNALNK62 , RESTART ALL
```

Increasing the Size of the MTU: To improve the performance of applications such as FTP, you should specify the MTU to be as large as possible (maximum of 32KB). Be aware that increasing the MTU will also increase the amount of virtual storage required for the SNALINK LU6.2 address space.

For common destination, you must specify the same value for the max_packet_size parameter in the BUFFERS statement of the SNALINK LU6.2 configuration that you specified for the max_packet_size in the GATEWAY statement in your TCP/IP profile. This value should also be the same as the corresponding value defined on all of the directly connected destination nodes.

Here is an example of using the BUFFERS statement to set the max_packet_size parameter to 32759:

```
BUFFERS 32759
```

If packets larger than the value you specify are received (either by the IUCV from the local TCP/IP address space or by VTAM from a destination node), then they will be discarded and a message will be written to the SYSPRINT data set.

Using a Larger RU Size for the VTAM Connection: You should use a larger RU size for the VTAM connection. You can set the RU size when configuring, when generating the NCP deck, and in the host VTAMLST.

Using Additional Send Buffers: Send buffers are used to store datagrams that have been extracted from messages sent by the local TCP/IP by IUCV, and are waiting to be passed to VTAM for transmission to the destination nodes. The required number of additional send buffers therefore depends on factors such as the maximum packet size, the amount of IP traffic for a destination, and VTAM performance.

You should set the number of additional buffers to an initial values, changing it afterwards if necessary. If you do not specify a value for this parameter, no additional buffers will be allocated.

Typically, the smaller the maximum packet size value, the more additional send buffers are required. When you use a smaller maximum packet size, more datagrams can be packed into a single IUCV message.

As a starting point, set the number of additional send buffers to 10. The contents of the SYSPRINT data set written by the SNALINK LU6.2 data set should then be regularly monitored to check for messages indicating that datagrams have been discarded due to a shortage of available send buffers.

There will not be a noticeable difference in performance when there are only a small amount of these messages across several hours of operation. However, when the number of messages becomes high, increase the number of additional send buffers and restart the SNALINK LU6.2 address space.

Here is an example of using the BUFFERS statement to set the additional send buffers to 10:

```
BUFFERS 32759 10
```

Try not to specify an excess number of additional buffers because they will increase the region size for the SNALINK LU6.2 address space and lower the amount of possible connections.

Changing the Send Queue Limit: To provide a consistent level of send buffer availability for each connection, you should set the send queue limit to be 1 more than the number of additional send buffers. The correct send queue limit will prevent a single connection from using more than its share of the available send buffers. To help in processing peak loads, however, you might decide to set the send queue limit value higher than the number of send buffers.

Using Multiple SNALINK LU6.2 Address Spaces: Multiple SNALINK LU6.2 address spaces should only be used where different MTUs are required for destinations in the directly connected network or networks.

Using Multiple IUCV Links: Because of the buffering performed for data being passed to or from the TCP/IP address space by IUCV links, you should define multiple links only when different network identifiers, log mode tables, or time-out values are required.

For more information, see the *TCP/IP for MVS: Customization and Administration Guide*.

On the VM Operating System

The following performance consideration exists for SNALINK LU Type 0:

- You should use the largest MaxRuCode available to increase performance. The maximum RU size is 32KB. Before changing the MaxRuCode, make sure that VTAM and NCP will support the size.

  For VTAM, look at the MAXBFRU and RACMLUBF fields in the ISTRACON parameter. For NCP, look at the MAXDATA, TRANSFER, BFRS, and MAXBFRU parameters. You can use several SNALINK virtual machine tasks for each MaxRuCode to maximize your connection.

  For more information about the maximum request or response unit (RU) size used, see the *VTAM Programming*.

- If you use SNALINK across a CTC connection, use a value other then zero for the retry delay parameter on the VTAM CTC definition. Also, you should check

to make sure the delay parameter on the 37x5 channel definition is set properly.

For more information, see the *TCP/IP for VM: Planning and Customization* book.

On the OS/2 Operating System

A guideline to follow when using SNALINK with OS/2 TCP/IP concerns the INIT option. For best performance, if you specify the INIT option for a specific destination, the remote node should also specify the option. Otherwise, there might be excess time spent waiting between the establishment of the sending session and that of the receiving session before the data can be transmitted.

Using a Larger RU Size for the VTAM Connection: You should use a larger RU size for the VTAM connection. For OS/2 to OS/2 LU6.2 sessions, use an RU size of 8192. You can set the RU size when configuring Communications Manager/2.

For more information, see the *TCP/IP for OS/2: Command Reference* book.

Native IP over Channel (3745) Versus SNALINK

Native IP over Channel for 3745/46 allows IP datagrams to be sent and received natively on the channel, rather than encapsulated in SNA frames, as is done with SNALINK.

To use Native IP over Channel for 3745/46, you need MVS TCP/IP V3R1 PTF UN84059 (and prerequisites) and NCP V7R3.

When compared to SNALINK using the TCP/IP FTP application, the new Native IP over Channel (3745) increases throughput from 6 to 87% and reduces MVS CPU by 75% to 89%.

Note: Better performance would be achieved if you used the 3746-9x0 instead of the 3745. See Appendix A, “Performance Tuning Tips for MVS” on page 183 for more information on the 3746-9x0.

Performance Summary

Table 28. Native IP over channel versus SNALINK (3745) for TCP/IP FTP.

FTP Type	Throughput Improvement	CPU Reduction (TCP/IP, FTP, SNALINK and VTAM)
Bin Put	71 to 87%	87 to 89%
Bin Get	6 to 31%	75 to 79%

The following shows the performance parameters used for the Native IP Over Channel (3745) and SNALINK (3745) performance tests.

Clients 4 RS/6000 WS, WS TR MTU = 2000

Server 3090-200J (MVS 5.1, MVS TCP/IP 3.1 and PTFs, 4 Pascal FTP Servers)

TCP/IP Profile (Packet size = 1000)

FTP Char	Binary Put/Get; users = 1,2,4,8; filesize = 4 MB; PUT: WS --> MVS Get: WS (/dev/null) <-- MVS
LAN	16Mbyte Token Ring
LAN Packets	512 bytes (Due to multiple networks and mssdefault = 512)
CCntrl	3745 (Single CCU)
NCP	V7R3; Buffer Size = 1K; 400 Read and 400 Write Buffers; SNALINK MTU: 32768, Delay = 0

MVS OE TCP/IP Performance Tuning

This section discusses performance tuning recommendations for OpenEdition MVS.

General OpenEdition TCP/IP Performance Tuning

The following is a list of suggestions for performance tuning when using the TCP/IP OpenEdition feature.

1. Follow the OpenEdition (OE) performance tuning guidelines in the chapter "Monitoring and Tuning the OpenEdition MVS Environment" in *Planning: OpenEdition MVS* (SC23-3015). See the following home page for more information.

   `http://www.S390.ibm.com/products/oe/bpxa1tun.html`

2. Follow the current MVS TCP/IP performance tuning guidelines. See the *IBM TCP/IP for MVS Performance Tuning Guide* (SC31-7188).
3. Update your MVS TCPIP Profile, TCPIP.DATA and FTP.DATA files.
4. Estimate how many OpenEdition MVS users, processes, ptys, sockets, and threads are needed for your OpenEdition MVS installation. Update your MVS BPXPRMxx member in SYS1.PARMLIB.
5. Estimate how many ASCH initiators are needed for your OpenEdition MVS installation. For more information, see "ASCH Initiator Guidelines" on page 104. Update your MVS ASCHPMxx member in SYS1.PARMLIB.
6. Spread OpenEdition MVS user HFS datasets among many DASD volumes for optimal performance.
7. Monitor your OpenEdition MVS resources with RMF and/or system commands (DISPLAY ACTIVE, DISPLAY OMVS, DISPLAY ASCH, DISPLAY APPC, and so on).
8. Adjust OpenEdition MVS system parameters to improve performance.

BPXPRMxx (SYS1.PARMLIB) Tuning

The BPXPRMxx member in SYS1.PARMLIB is used to set up general OpenEdition parameters. The xx part of BPXPRMxx is used in the start OMVS command (S OMVSRST,OMVS=xx). The following are guidelines for setting BPXPRMxx parmeters.

1. Make sure the MAXPROCSYS, MAXPROCUSER, MAXUIDS, MAXFILEPROC, MAXPTYS, MAXTHREADSTASKS, and MAXTHREADS are optimally set. If these parameters are not optimally set, your OpenEdition MVS performance

may be degraded. For more information about these parameters, see *MVS/ESA Initialization and Tuning Reference* (SC28-1452).

2. Make sure the MAXSOCKETS(n) parameter for the AF_INET domain is set high enough so that one does not run out of OE sockets. As an example, each OE Telnet session would require 2 OE sockets and each FTP session would require 1 OE socket. Once the MAXSOCKETS limit is reached, no more Telnet sessions, FTP sessions, or other applications that require OE sockets would be allowed to start.

ASCHPMxx (SYS1.PARMLIB) Tuning

The ASCHPMxx member in SYS1.PARMLIB is used to set up the mininum and maximum ASSCH initiator parameters. The following are guidelines for setting ASCHPMxx parms.

1. Set the MIN parameter for the CLASSADD statement to the average or typical number of OpenEdition MVS Address spaces used. Each OpenEdition MVS logon, shell command, OpenEdition Telnet session, FTP session, and so on requires a new OpenEdition MVS address space. Each OpenEdition MVS address space requires one ASCH initiator. Setting the ASCH MIN value high enough improves performance because new requests for an ASCH initiator can be taken from the free pool of ASCH's initiators initially generated (set by the MIN parameter). This is true as long as available ASCH initiators are in the free pool. If an ASCH initiator is not available, a new one must be created; however, creating a new initiator can degrade performance, especially interactive response time. The proper number for MIN should be the average or typical number of ASCH initiators used. Setting this parameter too high will waste storage.

2. Set the MAX parameter for the CLASSADD statement high enough so that all requests for ASCH initiators can be satisfied without waiting. If the system reaches the maximum number of ASCH initiators, new requests for ASCH initiators will be delayed until other ASCH initiators are freed up. This can severely hurt interactive performance if set too low.

For more information about these parameters, see the *MVS/ESA Initialization and Tuning Reference* (SC28-1452).

ASCH Initiator Guidelines

The following table should be used as a guide for estimating your OMVS address space/ASCH initiator usage by server or application. The minimum number required about 5 (OMVS Init, SYSLOGD, INETD, TCPIP, OE FTP Server) but you might want to increase this if more servers are started. The proper number for ASCH MIN should be the average or typical number of ASCH initiators used for optimal performance. Remember, setting this parameter too high will waste storage.

Table 29. ASCH Initiator Guidelines

Server/Application Name	# of ASCH Initiators Required
OMVS Init	1
SYSLOGD	1

Table 29. ASCH Initiator Guidelines

Server/Application Name	# of ASCH Initiators Required
INETD	1
TCPIP	1
OE FTP Server	1
each TSO OMVS logon	1
each logon (via otelnetd)	2
each OMVS shell command	1
each c89 compile	1
each OE FTP session	1
each REXEC command	2

Performance Tuning Checklist

MVS TCP/IP Performance Tuning Checklist

Check the following TCP/IP Performance parameters if you are experiencing performance problems or to ensure that your TCPIP environment is properly tuned:

- MVS dispatching priority of TCP/IP, VMCF, ADSM, FTP, or other TCP/IP servers.
  Recommendation: Set TCPIP priority approximately the same as VTAM; set the priority for the other TCP/IP servers slightly below that of TCP/IP.
- FTP: Specify a large DATABUFFERPOOLSIZE for better performance.
  Recommendation: Set the DATABUFFERPOOLSIZE parameter in the MVS TCP/IP Profile to 16384, 32768, or 65536.
- Make sure that the client and server TCP Window sizes are equal.
  Recommendation: Set TCP Window size equal to 16384, 32768, or 65536.
- Make sure that the client and server MTU packet sizes are equal.
  Recommendation: use the following packet sizes:

LAN type	MTU packet size
Ethernet	1500
Token Ring	2000
FDDI	4352

- 3172: Make sure that the Delay timer and Maximum response length are set correctly for each LAN adapter.
  Recommendation: Delay timer = 10 ms; Maximum response length = 500
- FTP: Make sure the BUFNO and EXTRATASKS parameters are set for MVS C-FTP server and the NCP parameter is set for MVS PASCAL FTP client/server. These parameters are set in the MVS HLQ.FTP.DATA file.
  Recommendation: BUFNO=5 or greater; EXTRATASKS=10 to 30; NCP=3 or greater. Larger values will improve performance, but will use more virtual storage.

- FTP: Use large file blocksizes on MVS.
  Recommendation:
  - Blocksize = 1/2 DASD track
  - 3380 DASD: approximately 23424
  - 3390/9334 DASD: approximately 29,000.
- TELNET: Check TIMEMARK, SCANINTERVAL, INACTIVE, DISABLESGA parameters (MVS TCPIP Profile).
  Recommendation: TIMEMARK = 1200-1800 (20-30 min.); SCANINTERVAL= 20-30 (sec.); INACTIVE = 3600 (60 min.); DISABLESGA.
- Monitor MVS TCP/IP buffer usage with NETSTAT POOLSIZE command. Increase the number of buffers, if needed.
  Recommendation: See Appendix A for guidance.
- Sockets: Use large message sizes (> 1KB) for better performance. Other socket recommendations:
  - To avoid small, partial packets, use TCP_NODELAY option on SETSOCKOPT() socket call.
    (Example: rc = setsockopt(s, IPPROTO_TCP, TCP_NODELAY, (char *) &delay, sizeof(delay));)
  - MVS CPU time may be reduced by adding the SOCKNOTESTSTOR. parameter in your HLQ.TCPIP.DATA file.
  - For better UDP socket performance, add the NOUDPQUEUELIMIT parameter under assorted parameters in your HLQ.xxxxxxxx.TCPIP file (TCPIP Profile).
  - For bulk UDP data transfers, using UDP Bulkmode will reduce MVS CPU time significantly. This can be done by setting the SO_BULKMODE option on the SETIBMSOCKOPT() socket call or by adding the SOCKBULKMODE parameter in your HLQ.TCPIP.DATA file.

MVS Performance Problem Determination

If you are having a performance problem with TCP/IP, use the following guidelines to determine the source of the problem:

- Make sure the tuning recommendations listed in this chapter under MVS TCP/IP Performance Tuning Checklist, and those listed in Appendix A of this book are followed.
- TCP/IP configuration data sets, such as TCP/IP profile, FTP.DATA and TCP/IP, will be required for problem determination.
- Define the problem accurately. Is it related to CPU utilization or throughput?
- MVS TCP/IP traces, such as, IP packet trace, socktr and moretrace may also be needed for problem determination. Check the *TCP/IP for MVS: Diagnosis Guide* and the *TCP/IP for MVS: Customization and Administration Guide* for information on the above.
- The following information will be helpful in problem determination.
  - TCP/IP Version and Release and level of maintenance
  - Type and model of mainframe; if using multiple CP LPARs, understand how they are setup(number of processors, storage)
  - Network configuration diagram

- Type of channel attached device (ESCON/Parallel channel attached 3172-3 ICP/Offload or OSA, CISCO datagram/offload or other)
- Type of LAN media used; if a router is used, obtain information on the router TR, FDDI, Ethernet
- Workstation (type of workstation, model), and Window size, MTU/Packet size used
- Key applications used (e.g. FTP, Telnet, CICS) and workload characteristics
- Workload characteristics and number of sessions per application
- IP packet traces may also help with some problems.

Chapter 6. Test Changes to Performance

This chapter includes an example of how the performance tuning process works. Also included is a section on common pitfalls to avoid in testing.

Performance Example

In this example, we want to improve the throughput of TCP/IP using FTP with an MVS server and an AIX client.

Table 30 shows the hosts in the example and the critical resources in which there is a constraint (—) or surplus (+).

Table 30. Example Hosts to Tune and Their Critical Resources

Type of host	Throughput	Response Time	CPU Utilization	Storage	Disk access
AIX	–	–	+	+	+
MVS	–	–	–	+	+

Note: Throughout the example, we assumed that all file transfers were approximately half ASCII and half binary, and approximately half PUTs and half GETs.

Example Environment

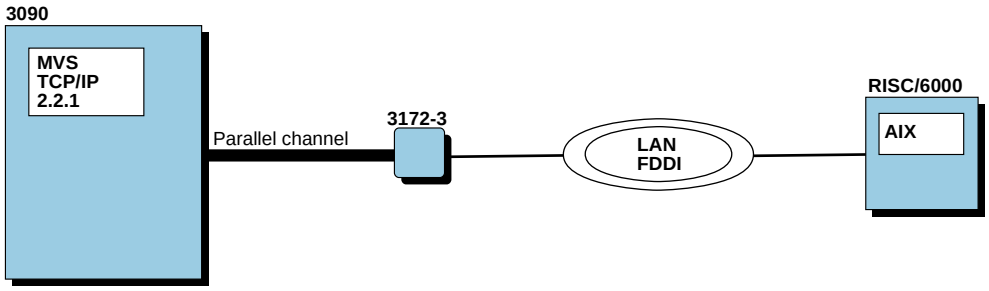


Figure 67. Environment of the Example

Table 31. Sample Environment

	Server	ICC	Client
CPU	3090* 400J	3172-3 50 MHz	RISC/6000 Model 530h
Memory	512MB central 3072MB expanded	8MB	64MB
Disks	3990/3390 cached controller	N/A	SCSI
File system	Partitioned data sets	N/A	jfs
Software	MVS/ESA 4.2.2 TCP/IP for MVS 2.2.1	ICP 3.2	AIX 3.2.3

Before we began testing, we found out the current values and configuration for TCP/IP and some of our key hardware devices. We reviewed the settings for our 3172-3 using the 3172 Operator Facility and saw that the block delay timer was set to 10ms and the maximum response length was set to 500 bytes.

Next we checked the status of caching on the disk controllers we planned to use. We wanted to be sure caching was active, to avoid any unnecessary delays during our file transfers. To check the status, we entered the DEVSERV command at the MVS console. The syntax for invoking this command is described in the *MVS/ESA System Commands* manual.Figure 68 shows the command we issued and the response received.

```
DEVSERV PATHS,5C0,8
IEE459I 22.00.12 DEVSERV PATHS 341
UNIT DTYPE  M CNT VOLSER  CHPID=PATH STATUS
             SSID CFW TC   DFW   PIN  DC-STATE CCA  DDC   ALT  CU-TYPE
05C0,3380K ,A,002,TCPSP1,49=R 4D=R
             0008  Y  YY.  NY.    N   SIMPLEX   C0   00        3990-3
05C1,3380K ,O,000,WRKLB5,49=+ 4D=+
             0008  Y  YY.  NY.    N   SIMPLEX   C1   01        3990-3
05C2,3380K ,A,000,WRKLB4,49=+ 4D=+
             0008  Y  YY.  NY.    N   SIMPLEX   C2   02        3990-3
05C3,3380K ,O,000,ES2RES,49=+ 4D=+
             0008  Y  YY.  NY.    N   SIMPLEX   C3   03        3990-3
05C4,3380K ,O,000,VMPRF9,49=+ 4D=+
             0008  Y  YY.  NY.    N   SIMPLEX   C4   04        3990-3
05C5,3380K ,A,002,TCPIP4,49=+ 4D=+
             0008  Y  YY.  NY.    N   SIMPLEX   C5   05        3990-3
05C6,3380K ,O,000,TCPIP2,49=+ 4D=+
             0008  Y  YY.  NY.    N   SIMPLEX   C6   06        3990-3
05C7,3380K ,F,000,      ,49=+ 4D=+
             0008  Y  YY.  NY.    N   SIMPLEX   C7   07        3990-3
********************** SYMBOL DEFINITIONS ************************
A = ALLOCATED                      F = OFFLINE
O = ONLINE                         + = PATH AVAILABLE
```

Figure 68. Example of the DEVSERV Command to Check If Caching Was Active

The field we are interested in is labeled TC. It is a 2-character field. If both characters are Y, then caching is active on the controller. Otherwise, caching is not active. There is a detailed explanation of all the output fields for DEVSERV under message IEE459I in the *MVS/ESA System Messages Volume 4 IEC-IFD* manual.

Next we used the NETSTAT ALL command on MVS from the initial state, with no active sessions. We got information on all the current clients to the TCP engine, including our FTP server, as shown in Figure 69.

```
...
Client: FTPSRV9                          Last Touched:   0:01:12
Local Socket: *..FTP-C                   Foreign Socket: *..*
   BackoffCount: 0
   ClientRcvNxt: 0
   ClientSndNxt: 867761301
   CongestionWindow: 65535  1
   Local connection name: 1074
   Sender frustration level: Contented
   Incoming window number: 0
   Initial receive sequence number: 0
   Initial send sequence number: 867761300
   Maximum segment size: 536  2
   Outgoing window number: 0
   Precedence: Routine
   RcvNxt: 0
   Round-trip information:
      Smooth trip time:   0.000
      Smooth trip variance:   1.500
   SlowStartThreshold: 65535
   SndNxt: 867761300
   SndUna: 867761300
   SndWl1: 0
   SndWl2: 0
   SndWnd: 0
   MaxSndWnd: 0  3
   State: Listen
   No pending TCP-receive
...
```

Figure 69. Using the NETSTAT ALL Command to Show Example Environment

1 The CongestionWindow is always set to 65 535 before any connection is established.

2 The maximum segment or packet size is 576 minus the 40 bytes used for TCP and IP headers.

3 The maximum send window is always set to 0 before any connection is established.

After establishing our FTP session, Figure 70 shows the response from the NETSTAT ALL command. (Notice that all the default values are still intact.)

```
...
Client: FTPSRV9
...
   CongestionWindow: 2680  1
...
   Maximum segment size: 536
...
   MaxSndWnd: 16384  2
   State: Established  3
   Pending TCP-receive buffer: 8192  4
...
```

Figure 70. NETSTAT ALL Command after Establishing the FTP Session

1 The CongestionWindow has shrunk from 64KB to 2680 bytes. This is the amount that the sender will advertise to the receiver. It will grow throughout the example.

2 The maximum send window now shows the size of the receive buffer on the AIX client.

3 The connection is now established.

4 The receive buffer on the server shows the default value of 8KB.

To get the current window size on the AIX operating system, you can also use the no -a command. It lists all the current values of the options, including tcp_sendspace and tcp_recvspace.Figure 71 shows the output from the no -a command.

```
               dog_ticks = 60
                lowclust = 60
                 lowmbuf = 119
                 thewall = 2048
              mb_cl_hiwat = 120
                compat_43 = 1
                   sb_max = 65536
           detach_route = 1
        subnetsarelocal = 1
                 maxttl = 255
              ipfragttl = 60
        ipsendredirects = 1
           ipforwarding = 1
                udp_ttl = 30
                tcp_ttl = 60
             arpt_killc = 20
          tcp_sendspace = 16384
          tcp_recvspace = 16384
          udp_sendspace = 16384
          udp_recvspace = 16384
         loop_check_sum = 1
         rfc1122addrchk = 0
         nonlocsrcroute = 1
          tcp_keepintvl = 150
           tcp_keepidle = 14400
              ipqmaxlen = 50
```

Figure 71. Using the no -a Command to View Current Send and Receive Buffers

We also checked the current MTU size on MVS using the NETSTAT GATE command., as shown in Figure 72.

```
netstat gate
MVS TCP/IP Netstat V2R2.1
Known gateways:
NetAddress    FirstHop      Link     Pkt Sz  Subnet Mask     Subnet Value
----------    --------      ------   ------  -----------     ------------
Default       9.67.113.11   TR1      Default <none>
9.0.0.0       <direct>      FDDI     Default 0.255.255.128   0.67.115.0
...
```

Figure 72. Using the NETSTAT GATE Command to Show the Current Packet Size

Since the response shows the packet size as default, we know that the current packet size is 576.

To get the current MTU size on the AIX operating system, you can also use the netstat -i -I fi0 command.

Step 1. Measure the Current System

The steps that are part of measuring the current system are:

1. Start monitors
2. Run the test from the client
3. Record throughput and time stamps
4. Stop monitors
5. Reduce data and generate reports
6. Summarize data

1. Start Monitors

We started the AIX client monitor with the following command:

```
netpmon -o basecase.mon -O cpu,dd,so
```

We started monitoring the MVS server. To start RMF, we used a job called RMFL which was started by entering S RMFL on the console.

Then we used the NETSTAT POOLSIZE command and recorded the data, as shown in Figure 73.

```
netstat poolsize
MVS TCP/IP Netstat V2R2.1
TCPIP Free pool status:
Object       # alloc     # free      Lo-water    Permit size
======       =======     ======      ========    ===========
ACB            1000         997          993         100
CCB             150         135          135          10
Dat buf         160         153          150          32
Sm dat buf        0           0            0           0
Tiny dat buf      0           0            0           0
Env             750         750          744          75
Lrg env          50          49           49          10
RCB              50          50           50           3
SCB             256         246          246          17
SKCB            256         256          256          17
TCB             256         250          250          17
UCB             100         100          100           6
```

Figure 73. NETSTAT POOLSIZE Command before Making Changes in the Example

2. Run Test from the Client

We ran the test from the client, using FTP to the server. We did a PUT of a known 10MB binary file 3 times for repeatability. Then we did a GET of the same binary file 3 times, followed by 3 PUTs and 3 GETs of a 10MB ASCII file.

Using the following formula, the total data to be transferred was 122 880KB (120MB).

```
3 iterations * 4 measurements * 10MB = 122,880KB
```

Here is the sequence of commands used to send the binary file:

```
ftp 9.67.115.16
USER2
<password>
binary  1
quote site lrecl=64 blocksize=23424 recfm=fb
cd TEST
delete M10
put m10 M10
delete M10
put m10 M10
delete M10
put m10 M10
quit
```

Here is the sequence of commands used to get the binary file:

```
ftp 9.67.115.16
USER2
<password>
binary  1
cd TEST
get M10 /dev/null  2
get M10 /dev/null
get M10 /dev/null
quit
```

1 We tested using an ASCII file with the same commands, leaving out the change to binary, since ASCII is the default.

2 The /dev/null file is used because it acts as a bit bucket to avoid the possible disk delays caused by writing to disk.

3. Record Throughput and Time Stamps

We determined the start times by entering `D T` on the MVS/ESA* console at the same time we entered PUT or GET on the workstation. Stop times were determined by entering `D T` on the MVS/ESA console when the `Transfer completed successfully` message was observed on the AIX screen. It is necessary to have time stamps from the MVS/ESA host in order to correctly analyze the RMF data. (We could not use the AIX clock or wall clock because we could not be sure the clocks were synchronized with the MVS clock.)

4. Stop Monitors

We stopped the AIX client monitor with the following command:

```
trcstop
```

We stopped RMF on the MVS server (we entered `P RMFL` at the console), then we entered `netstat poolsize` from the TSO user.

5. Reduce Data and Generate Reports

We collected data from both the client and the server.

Client Reports: The netpmon output was in basecase.mon, the file we specified. Although there are 4 reports generated for each test we ran, as an example, we have included only the 2 in which we are interested, in this case from the ASCII PUT test. The first report (shown in Figure 74) contains the CPU usage statistics.

```
Process CPU Usage Statistics:
-----------------------------

                                                        Network
Process (top 20)             PID  CPU Time   CPU %   CPU %
------------------------------------------------------------
ftp                        12957   64.8857  23.037   6.324
netpmon                    17052    9.6820   3.438   0.000
X                          12331    6.4929   2.305   0.000
aixterm                    15157    0.9312   0.331   0.000
trace                      11670    0.8000   0.284   0.000
vi                         13431    0.5486   0.195   0.000
cmon                        8265    0.5436   0.193   0.000
aixterm                    13620    0.4481   0.159   0.000
xclock                      7219    0.2834   0.101   0.000
netw                         771    0.2435   0.086   0.086
sm00                        4278    0.1968   0.070   0.000
init                           1    0.1452   0.052   0.000
swapper                        0    0.1062   0.038   0.000
syncd                       2570    0.0481   0.017   0.000
mwm                        10544    0.0448   0.016   0.000
cron                        3951    0.0375   0.013   0.000
snmpd                       6248    0.0263   0.009   0.000
trcstop                    12958    0.0202   0.007   0.000
csh                        17235    0.0133   0.005   0.000
wplmd                       5448    0.0105   0.004   0.000
------------------------------------------------------------
Total (all processes)               85.5218  30.364   6.411
Idle time                          175.4921  62.307
```

Figure 74. Process CPU Usage Statistics Report from Base Measurement of ASCII PUT

The second report (shown in Figure 75) contains the information on the reads and writes.

```
Detailed TCP Socket Call Statistics (by Process):
-------------------------------------------------

PROCESS: ftp   PID: 12957
reads:                  18
  read sizes (bytes):   avg 4096.0  min 4096    max 4096    sdev 0.0
  read times (msec):    avg 128.116 min 0.195   max 385.275 sdev 117.414
writes:                 7814
  write sizes (bytes):  avg 4088.7  min 6       max 4096    sdev 172.6
  write times (msec):   avg 5.289   min 0.258   max 37.106  sdev 5.831

PROTOCOL: TCP (All Processes)
reads:                  18
  read sizes (bytes):   avg 4096.0  min 4096    max 4096    sdev 0.0
  read times (msec):    avg 128.116 min 0.195   max 385.275 sdev 117.414
writes:                 7814
  write sizes (bytes):  avg 4088.7  min 6       max 4096    sdev 172.6
  write times (msec):   avg 5.289   min 0.258   max 37.106  sdev 5.831
```

Figure 75. Detailed TCP Statistics Report from Base Measurement of ASCII PUT

Server Reports: After our measurement, we produced a report using the RMF post processor. See the *RMF User's Guide* for information on how to use the RMF post processor to produce reports.

We used the following control statements as input:

```
RTOD(2253,2256)
REPORTS(ARD)
```

These requested an Address Space Resource Data report for data collected between 22:53:00 and 22:56:00. We used this report to determine the CPU time used on the MVS/ESA system during our measurement. See *Analyzing RMF Monitor I and II Reports* for more information on how to read the report.

RMF only accepts hhmm as the time stamp, even though data is recorded at much smaller intervals.

There are 2 ways data is reported in the interval reports, as a rate or as a cumulative total. Some of the data is reported as a rate, such as the EXCP (execute channel program) RATE.

If we wanted to know the total number of EXCP events that occurred during our measurement, we would have to process this information for each one of the interval reports.

If the data is reported as a cumulative total, such as CPU TIME, we need to take the value on the ending report and subtract the value from the starting report.

Our starting time stamp was 22:53:45, so we ignored all the interval reports before that time, except the report right before that time and the one just following our ending time. These reports are those at 22:53:42 and 22:56:23. We kept these 2 reports and deleted all the other interval reports.

Figure 76 shows the part of the interval report from 22:53:42 that we are interested in:

```
...
 22:53:42 DEV    FF PRIV LSQA LSQA X SRM  TCB    CPU   EXCP SWAP LPA CSA NVI V&H
 JOBNAME  CONN  BEL   FF  CSF  ESF M ABS  TIME   TIME  RATE RATE  RT  RT  RT  RT
 ...
 TCP22R   32.22   0   13   43    2   0.0  63.99  68.42 0.00 0.00 0.0 0.0 0.0 0.0
 FTPSRV6  0.648   0    2   30    2   0.0   0.20   0.21 0.00 0.00 0.0 0.0 0.0 0.0
 FTPSRV7  0.720   0    2   30    2   0.0   0.19   0.20 0.00 0.00 0.0 0.0 0.0 0.0
 FTPSRV8  0.598   0    2   30    2   0.0   0.19   0.20 0.00 0.00 0.0 0.0 0.0 0.0
 FTPSRV9  50.46   0    2   31    2   0.0  17.46  18.25 3.40 0.00 0.0 0.0 0.0 0.0
...
```

Figure 76. Partial RMF Interval Report from 22:53:42

Figure 77 shows the part of the interval report from 22:56:23 that we are interested in:

```
...
 22:56:23 DEV    FF PRIV LSQA LSQA X SRM  TCB    CPU   EXCP SWAP LPA CSA NVI V&H
 JOBNAME  CONN  BEL   FF  CSF  ESF M ABS  TIME   TIME  RATE RATE  RT  RT  RT  RT
 ...
 TCP22R   42.51   0   13   43    2   0.0  87.14  93.06 0.00 0.00 0.0 0.0 0.0 0.0
 FTPSRV6  0.648   0    2   30    2   0.0   0.20   0.21 0.00 0.00 0.0 0.0 0.0 0.0
 FTPSRV7  0.720   0    2   30    2   0.0   0.19   0.20 0.00 0.00 0.0 0.0 0.0 0.0
 FTPSRV8  0.598   0    2   30    2   0.0   0.19   0.20 0.00 0.00 0.0 0.0 0.0 0.0
 FTPSRV9  70.04   0    2   31    2   0.0  27.29  28.35 11.6 0.00 0.0 0.0 0.0 0.0
...
```

Figure 77. Partial RMF Interval Report from 22:56:23

The complete interval reports are shown in “RMF Interval Reports” on page 215.

Now we compared the CPU time information from the address spaces that are associated with the TCP/IP product between the 2 reports. The difference between the CPU time for TCP22R (the main TCP/IP address space) in the first report and the last report was 24.64 seconds (93.06 – 68.42 = 24.64).

We followed the same procedure to get the CPU time for the 4 FTP servers:

```
FTPSRV6  0.21 -  0.21 =  0.00
FTPSRV7  0.20 -  0.20 =  0.00
FTPSRV8  0.20 -  0.20 =  0.00
FTPSRV9 28.35 - 18.25 = 10.10
```

As you can see, 3 out of the 4 FTP servers had no CPU busy time. This is because we never had more than one FTP connection open. Multiple FTP servers can improve system performance by sharing the work generated by multiple users. However, any single connection can only be serviced by one FTP server, so you will not see the improvement when only one connection is open.

The CPU time for the TCPIP address space (TCP22R) was 24.64 seconds and the FTP server was 10.10 seconds, the total TCP/IP time for these 2 address spaces in this measurement was 34.74 seconds.

Test Results: Table 32 and Table 33 contain the reduced data from our base measurement.

Table 32. Test Results from the Base Measurement

Test Type	Start Time	Stop Time	Test #1	Test #2	Test #3
Binary PUT	22.38.39	22.40.53	23.70 seconds (432.1KB/s)	23.72 seconds (431.6KB/s)	23.89 seconds (428.7KB/s)
ASCII PUT	22.53.45	22.56.22	31.21 seconds (333.2KB/s)	31.23 seconds (333.0KB/s)	31.25 seconds (332.8KB/s)
Binary GET	22.46.18	22.48.20	19.75 seconds (518.6KB/s)	19.65 seconds (521.1KB/s)	19.60 seconds (522.6KB/s)
ASCII GET	22.58.35	23.00.57	30.03 seconds (351.7KB/s)	29.86 seconds (353.7KB/s)	29.99 seconds (352.1KB/s)

Table 33. CPU Time from Monitors on Base Measurement

	From netpmon:	***From RMF:***			
Test Type	**FTP**	**TCP/IP Address Space**	**FTP Server Address Space**	**Sum of Address Spaces**	**Total MVS**
Binary PUT	6.218	20.97	6.02	26.99	34.22
ASCII PUT	64.886	24.64	10.10	34.74	43.15
Binary GET	12.979	23.62	5.35	28.97	35.19
ASCII GET	59.554	18.74	6.81	25.55	33.59
Total	143.64	87.97	28.28	116.25	146.15

6. Summarize Data

The data included:

- CPU load for both client and server
- Throughput for client

Table 34 contains the data from our base measurement.

Table 34. Total Throughput and CPU Time after the Base Measurement

	Base
Total Time	313.88
Total Throughput	391.5
AIX FTP Busy	143.64
MVS TCP/IP Busy	87.97
MVS FTP Busy	28.28
MVS TCP/IP + FTP	116.25
MVS Total Busy	146.15

Step 2. Change the MTU Size on the Server

We changed the MTU size on the server using OBEYFILE.

Size	***Server***
Default	576
New	4352

First we created a data set similar to the TCP/IP profile but containing only the GATEWAY statement. Then we entered the following TSO command:

```
OBEYFILE 'data set name'
```

This change is temporary until TCP/IP goes down again. Then when TCP/IP comes back up, it will look to the values in PROFILE.TCPIP, so you should change the profile permanently once you have confirmed that the change was worthwhile.

We reestablished an FTP session so that we could use the NETSTAT ALL command on MVS.Figure 78 shows the new server packet size on the **1** `Maximum segment size` parameter.

```
...
Client: FTPSRV9
...
   CongestionWindow: 3752
...
   Maximum segment size: 1460 1
...
   MaxSndWnd: 16384
   State: Established
   Pending TCP-receive buffer: 8192
...
```

Figure 78. NETSTAT ALL Command after the Second Change in the Example

The MTU on the server is now set to 1500, and the transmitted packet size is 1500 bytes, but the TCP data portion (or maximum segment size) is 1460.

Even though we have changed the MTU to 4KB, the client is still set to a packet size of 1500. The server and client negotiated the packet size to the lower amount.

Step 3. Measure the Change

We repeated the steps we followed in “Step 1. Measure the Current System” on page 113. Table 35 and Table 36 contain the data after the first change.

Table 35. Test Results after the First Change

Test Type	Start Time	Stop Time	Test #1	Test #2	Test #3
Binary PUT	00.38.05	00.40.03	17.37 seconds (589.5KB/s)	17.18 seconds (596.2KB/s)	17.31 seconds (591.7KB/s)
ASCII PUT	00.58.42	01.00.37	22.84 seconds (455.3KB/s)	22.83 seconds (455.6KB/s)	22.85 seconds (455.1KB/s)
Binary GET	00.43.13	00.44.34	10.46 seconds (978.9KB/s)	10.36 seconds (988.0KB/s)	10.51 seconds (974.4KB/s)
ASCII GET	01.03.12	01.05.18	24.50 seconds (431.1KB/s)	24.54 seconds (430.3KB/s)	24.47 seconds (431.6KB/s)

Table 36. CPU Time from Monitors after the First Change

	From netpmon:	***From RMF:***			
Test Type	**FTP**	**TCP/IP Address Space**	**FTP Server Address Space**	**Sum of Address Spaces**	**Total MVS**
Binary PUT	4.858	15.01	6.21	21.22	27.35
ASCII PUT	52.443	18.11	10.25	28.36	34.37
Binary GET	10.347	18.82	5.48	24.30	27.99
ASCII GET	57.792	17.83	6.98	24.81	31.12
Total	125.44	69.77	28.92	98.69	120.83

Then we used the NETSTAT POOLSIZE command, as shown in Figure 79.

```
netstat poolsize
MVS TCP/IP Netstat V2R2.1
TCPIP Free pool status:
Object       # alloc     # free      Lo-water    Permit size
======       =======     ======      ========    ===========
ACB             1000        996           993          100
CCB              150        135           135           10
Dat buf          160        152           150           32
Sm dat buf         0          0             0            0
Tiny dat buf       0          0             0            0
Env              750        750           744           75
Lrg env           50         49            49           10
RCB               50         50            50            3
SCB              256        246           246           17
SKCB             256        256           256           17
TCB              256        250           250           17
UCB              100        100           100            6
```

Figure 79. NETSTAT POOLSIZE Command after the First Change in the Example

Step 4. Compare the Results

Table 37 shows the comparison between the base measurement and the first change.

Table 37. Total Throughput and CPU Time after the First Change

	Base	First Change
Total Time	313.88	225.22
Total Throughput	391.5	545.60
AIX FTP Busy	143.64	125.44
MVS TCP/IP Busy	87.97	69.77
MVS FTP Busy	28.28	28.92
MVS TCP/IP + FTP	116.25	98.69
MVS Total Busy	146.15	120.83

To summarize the results from the first change:

- Total throughput was 545.6KB per second, an improvement of 39% over the base.
- The sum of AIX FTP CPU busy time was 125.44 seconds, a decrease of 13%.
- The sum of MVS FTP and TCP/IP busy time was 98.69 seconds, a decrease of 15% (FTP time was 28.92, no appreciable change, and TCP/IP time was 69.77, down 21%).

Step 5. Change the Window Size on the Server

Next we changed the window size on the server.

Size	***Server***
Default	8KB
New	32KB

This change required stopping TCP/IP on MVS.

1. Stop TCP/IP

   We entered the following command from the MVS console:

   ```
   P TCP22R
   ```

2. Edit the PROFILE.TCPIP data set
   - Change DATABUFFERPOOLSIZE from 160 8192 to 160 32768
3. Start TCP/IP with the new size
4. Verify the new size

   We reestablished an FTP session so that we could use the NETSTAT ALL command on MVS.Figure 80 shows the new receive buffer value on the **1** `Pending TCP-receive buffer` parameter.

```
...
Client: FTPSRV9
...
   CongestionWindow: 7300
...
   Maximum segment size: 1460
...
   MaxSndWnd: 16384
   State: Established
   Pending TCP-receive buffer: 32768 1
...
```

Figure 80. NETSTAT ALL Command after the First Change in the Example

Step 6. Measure the Second Change

We repeated the steps we followed in “Step 1. Measure the Current System” on page 113. Table 38 and Table 39 contain the data after the second change.

Table 38. Test Results after the Second Change

Test Type	Start Time	Stop Time	Test #1	Test #2	Test #3
Binary PUT	01.42.17	01.44.11	13.35 seconds (767.0KB/s)	13.41 seconds (763.5KB/s)	13.37 seconds (765.7KB/s)
ASCII PUT	01.49.50	01.52.24	22.61 seconds (460.0KB/s)	22.59 seconds (460.3KB/s)	22.61 seconds (460.1KB/s)
Binary GET	01.46.04	01.47.24	10.56 seconds (969.4KB/s)	11.47 seconds (893.1KB/s)	10.63 seconds (963.4KB/s)
ASCII GET	01.53.59	01.55.58	24.65 seconds (428.3KB/s)	24.49 seconds (431.1KB/s)	24.47 seconds (431.6KB/s)

Table 39. CPU Time from Monitors after the Second Change

	From netpmon:	***From RMF:***			
Test Type	**FTP**	**TCP/IP Address Space**	**FTP Server Address Space**	**Sum of Address Spaces**	**Total MVS**
Binary PUT	11.092	10.79	4.45	15.24	21.19
ASCII PUT	55.309	15.40	8.07	23.47	30.53
Binary GET	10.320	16.66	3.49	20.15	23.80
ASCII GET	57.778	15.27	4.90	20.17	26.26
Total	134.50	58.12	20.91	79.03	101.78

Then we used the NETSTAT POOLSIZE command, as shown inFigure 81 .

```
netstat poolsize
MVS TCP/IP Netstat V2R2.1
TCPIP Free pool status:
Object       # alloc     # free      Lo-water    Permit size
======       =======     ======      ========    ===========
ACB            1000        997         989         100
CCB             150        135         135          10
Dat buf         160        153         144          32
Sm dat buf        0          0           0           0
Tiny dat buf      0          0           0           0
Env             750        750         743          75
Lrg env          50         49          49          10
RCB              50         50          50           3
SCB             256        246         244          17
SKCB            256        256         256          17
TCB             256        250         246          17
UCB             100        100         100           6
```

Figure 81. NETSTAT POOLSIZE Command after the Second Change in the Example

Step 7. Compare the Results

Table 40 shows the comparison between the base measurement and the 2 changes:

Table 40. Total Throughput and CPU Time after the Second Change

	Base	First Change	Second Change
Total Time	313.88	225.22	214.21
Total Throughput	391.5	545.60	573.64
AIX FTP Busy	143.64	125.44	134.50
MVS TCP/IP Busy	87.97	69.77	58.12
MVS FTP Busy	28.28	28.92	20.91
MVS TCP/IP + FTP	116.25	98.69	79.03
MVS Total Busy	146.15	120.83	101.78

To summarize the results from the second change:

- Total throughput was 573.64KB per second, a 5% increase from the previous measurement.
- The sum of AIX FTP CPU busy time was 134.50 seconds, a 7% increase. (This serves as an example of a tuning trade-off. If CPU time was at a premium on AIX, you might want to reverse this change.)
- The sum of MVS FTP and TCP/IP busy time was 79.03 seconds, a 20% decrease (FTP time was 20.91, 28% less, and TCP/IP time was 58.12, 17% less).

Step 8. Change the MTU Size on the Client

Next we changed the MTU size on the client to 4352 (the recommendation from the *AIX 3.2 for RISC System/6000 Performance Monitoring and Tuning Guide*).

Size	***Client***
Default	1500
New	4352

We changed the TCP/IP MTU size on AIX with the following commands:

```
ifconfig fi0 down
ifconfig fi0 mtu 4352
ifconfig fi0 up
netstat -i -I fi0
```

This change is temporary until the system is rebooted. If the results of the change are proven to be positive, we will make the change permanent by using the smit command.

We reestablished an FTP session so that we could use the NETSTAT ALL command on MVS.Figure 82 shows the new client packet size on the **1** `Maximum segment size` parameter.

```
...
Client: FTPSRV9
...
   CongestionWindow: 14881
...
   Maximum segment size: 4096  1
...
   MaxSndWnd: 16384
   State: Established
   Pending TCP-receive buffer: 32768
...
```

Figure 82. NETSTAT ALL Command after the Fourth Change in the Example

The MTU is now 4096 bytes. Although the AIX client is set to 4352, it is the parallel-channel-attached 3172 that influences the amount because 4096 bytes is its MTU. This illustrates the concept that it is not only the client and the server that determine performance, but the other network components between them.

Step 9. Measure the Third Change

We repeated the steps we followed in “Step 1. Measure the Current System” on page 113. Table 41 and Table 42 contain the data after the third change.

Table 41. Test Results after the Third Change

Test Type	Start Time	Stop Time	Test #1	Test #2	Test #3
Binary PUT	02.01.20	02.03.06	13.19 seconds (776.2KB/s)	13.13 seconds (780.0KB/s)	13.17 seconds (777.6KB/s)
ASCII PUT	02.08.56	02.10.40	19.48 seconds (533.8KB/s)	19.46 seconds (534.4KB/s)	19.45 seconds (534.8KB/s)
Binary GET	02.04.33	02.05.49	9.402 seconds (1089KB/s)	9.423 seconds (1087KB/s)	9.374 seconds (1092KB/s)
ASCII GET	02.12.05	02.14.04	21.48 seconds (491.6KB/s)	21.53 seconds (490.4KB/s)	21.6 seconds (488.9KB/s)

Table 42. CPU Time from Monitors after the Third Change

	From netpmon:	***From RMF:***			
Test Type	**FTP**	**TCP/IP Address Space**	**FTP Server Address Space**	**Sum of Address Spaces**	**Total MVS**
Binary PUT	7.055	7.68	4.73	12.41	18.49
ASCII PUT	51.140	15.92	11.10	27.02	32.81
Binary GET	4.872	8.52	3.46	11.98	17.06
ASCII GET	56.020	9.47	4.91	14.38	20.45
Total	119.09	41.59	24.20	65.79	88.81

Then we used the NETSTAT POOLSIZE command, as shown inFigure 83 .

```
netstat poolsize
MVS TCP/IP Netstat V2R2.1
TCPIP Free pool status:
Object       # alloc     # free      Lo-water    Permit size
======       =======     ======      ========    ===========
ACB            1000         993         988         100
CCB             150         135         135          10
Dat buf         160         152         144          32
Sm dat buf        0           0           0           0
Tiny dat buf      0           0           0           0
Env             750         750         743          75
Lrg env          50          49          49          10
RCB              50          50          50           3
SCB             256         244         244          17
SKCB            256         256         256          17
TCB             256         248         246          17
UCB             100         100         100           6
```

Figure 83. NETSTAT POOLSIZE Command after the Third Change in the Example

Step 10. Compare the Results

Table 43 shows the comparison between the base measurement and the 3 changes:

Table 43. Total Throughput and CPU Time after the Third Change

	Base	First Change	Second Change	Third Change
Total Time	313.88	225.22	214.21	190.69
Total Throughput	391.5	545.60	573.64	644.40
AIX FTP Busy	143.64	125.44	134.50	119.09
MVS TCP/IP Busy	87.97	69.77	58.12	41.59
MVS FTP Busy	28.28	28.92	20.91	24.20
MVS TCP/IP + FTP	116.25	98.69	79.03	65.79
MVS Total Busy	146.15	120.83	101.78	88.81

To summarize the results from the third change:

- Total throughput was 644.40KB per second, an improvement of 12% over the previous measurement.
- The sum of AIX FTP CPU busy time was 119.09 seconds, a decrease of 11%.
- The sum of MVS FTP and TCP/IP busy time was 65.79 seconds, a decrease of 17% (FTP time was 24.20, 16% more, and TCP/IP time was 41.59, down 28%).

Step 11. Change the Window Size on the Client

The next change was to increase the window size of the AIX client.

Size	***Client***
Default	16KB
New	32KB

We changed the TCP/IP window size on AIX with the following commands:

```
no -o tcp_sendspace=32768
no -o tcp_recvspace=32768
```

This change is temporary until the system is rebooted. If the results of the change are proven to be positive, we will make the change permanent by editing the /etc/rc.net file.

Since this example is concerned with FTP, we did not need to change the udp_sendspace and udp_recvspace in addition to tcp_sendspace and tcp_recvspace. If you are running NFS, you might want to try changing those values also.

We reestablished an FTP session so that we could use the NETSTAT ALL command on MVS.Figure 84 shows the new send buffer value on the **1** `MaxSndWnd` parameter.

```
...
Client: FTPSRV9
...
   CongestionWindow: 20480
...
   Maximum segment size: 4096
...
   MaxSndWnd: 32768 1
   State: Established
   Pending TCP-receive buffer: 32768
...
```

Figure 84. NETSTAT ALL Command after the Second Change in the Example

An alternative way to view the current window size on the AIX operating system is to use the no -a command.

Step 12. Measure the Fourth Change

We repeated the steps we followed in “Step 1. Measure the Current System” on page 113. Table 44 and Table 45 contain the data after the fourth change.

Table 44. Test Results after the Fourth Change

Test Type	Start Time	Stop Time	Test #1	Test #2	Test #3
Binary PUT	02.18.00	02.20.17	11.06 seconds (925.7KB/s)	10.97 seconds (933.1KB/s)	11.06 seconds (925.7KB/s)
ASCII PUT	02.24.39	02.26.33	19.52 seconds (532.8KB/s)	19.46 seconds (534.3KB/s)	19.47 seconds (534.2KB/s)
Binary GET	02.21.32	02.22.36	7.935 seconds (1290KB/s)	7.226 seconds (1417KB/s)	7.290 seconds (1405KB/s)
ASCII GET	02.27.49	02.29.35	21.60 seconds (489.0KB/s)	21.60 seconds (489.0KB/s)	21.51 seconds (490.9KB/s)

Table 45. CPU Time from Monitors after the Fourth Change

	From netpmon:	***From RMF:***			
Test Type	**FTP**	**TCP/IP Address Space**	**FTP Server Address Space**	**Sum of Address Spaces**	**Total MVS**
Binary PUT	6.642	7.46	4.04	11.50	17.92
ASCII PUT	51.163	15.92	11.17	27.09	33.04
Binary GET	4.829	8.42	3.46	11.88	16.78
ASCII GET	56.023	9.48	4.93	14.41	19.75
Total	118.66	41.28	23.60	64.88	87.49

Then we used the NETSTAT POOLSIZE command, as shown inFigure 85 .

```
netstat poolsize
MVS TCP/IP Netstat V2R2.1
TCPIP Free pool status:
Object       # alloc     # free      Lo-water    Permit size
======       =======     ======      ========    ===========
ACB            1000         993         988         100
CCB             150         135         135          10
Dat buf         160         153         144          32
Sm dat buf        0           0           0           0
Tiny dat buf      0           0           0           0
Env             750         750         743          75
Lrg env          50          49          41          10
RCB              50          50          50           3
SCB             256         244         244          17
SKCB            256         256         256          17
TCB             256         247         246          17
UCB             100         100         100           6
```

Figure 85. NETSTAT POOLSIZE Command after the Fourth Change in the Example

Step 13. Compare the Results

Table 46 shows the comparison between the base measurement and the 4 changes:

Table 46. Total Throughput and CPU Time after the Fourth Change

	Base	First Change	Second Change	Third Change	Fourth Change
Total Time	313.88	225.22	214.21	190.69	178.70
Total Throughput	391.5	545.60	573.64	644.40	687.63
AIX FTP Busy	143.64	125.44	134.50	119.09	118.66
MVS TCP/IP Busy	87.97	69.77	58.12	41.59	41.28
MVS FTP Busy	28.28	28.92	20.91	24.20	23.60
MVS TCP/IP + FTP	116.25	98.69	79.03	65.79	64.88
MVS Total Busy	146.15	120.83	101.78	88.81	87.49

To summarize the results from the fourth change:

- Total throughput was 687.63KB per second, an improvement of 7% over the previous measurement.
- The sum of AIX FTP CPU busy time was 118.66 seconds, no appreciable change.
- The sum of MVS FTP and TCP/IP busy time was 64.88 seconds, no appreciable change (FTP time was 23.60, no appreciable change, and TCP/IP time was 41.28, no appreciable change).

Analyze for Overall Changes

The next step is to compare the results of the changes overall.

- Total testing time went from 313.88 seconds to 178.70 seconds, a decrease of 43% overall.
- Overall throughput was went from 391.5KB per second to 687.63KB per second, an improvement of 76% overall.
- Client FTP CPU busy time went from 143.64 seconds to 118.66 seconds, a decrease of 17% overall.
- Total server FTP and TCP/IP busy time went from 116.25 seconds to 64.88 seconds, a decrease of 44% overall (FTP time decreased 17% and TCP/IP time decreased 53%).

Figure 86 shows the progressive improvements made in this example.

- The minimum throughput used in the chart is the lowest throughput, whether it was for ASCII or binary, PUT or GET.
- The total throughput used in the chart is the sum of all the throughput bytes divided by all the throughput times.
- The maximum throughput used in the chart is the highest throughput, whether it was for ASCII or binary, PUT or GET.

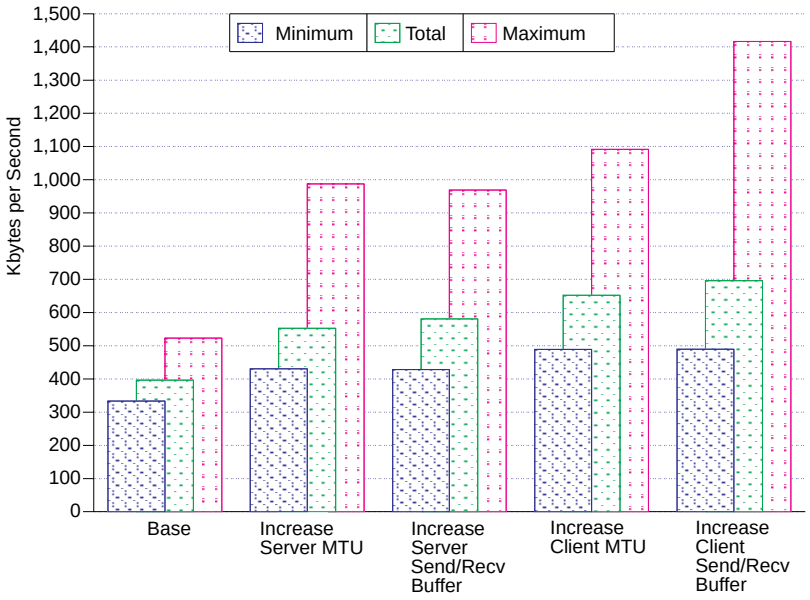


Figure 86. Overall Changes to Throughput

Figure 87 and Figure 88 show the contrast of how the time was spent on FTP, CPU busy time, and other travel and waiting time.

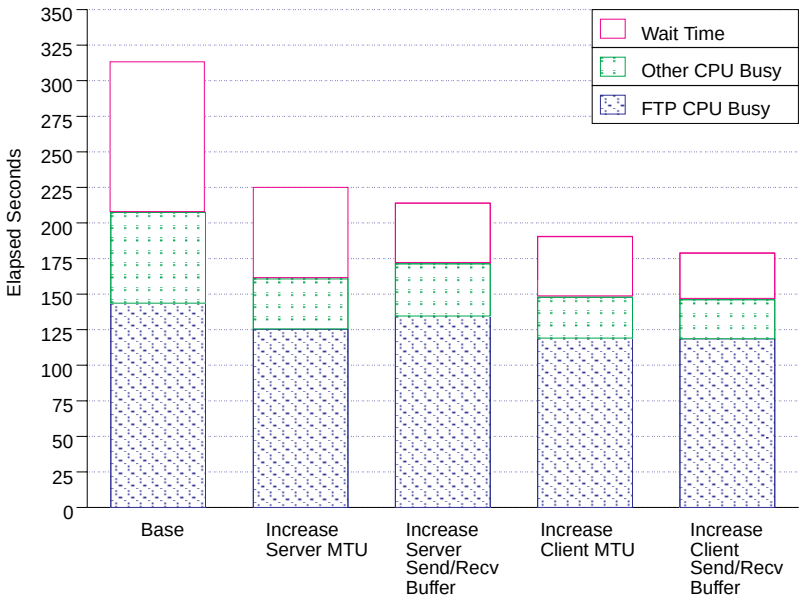


Figure 87. Client View of Changes in FTP, Total CPU, and Other Time

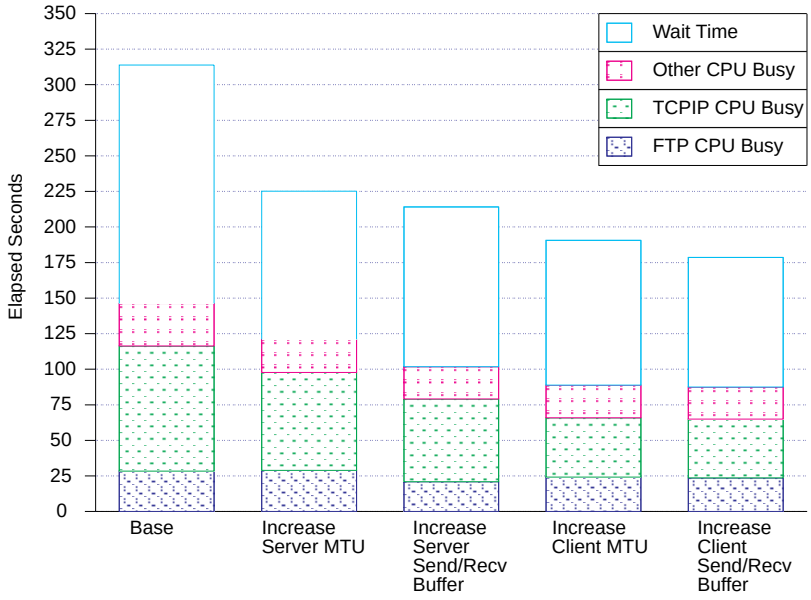


Figure 88. Server View of Changes in FTP, Total CPU, and Other Time

Make Temporary Changes Permanent

We now ensured the temporary changes made earlier became permanent.

- We edited the /etc/rc.net file to increase the window size of the AIX client.
- We edited the MVS PROFILE.TCPIP data set to change the server MTU.
- We used the smit command to change the client MTU.

Common Pitfalls to Avoid

This section covers some common pitfalls to avoid when testing changes to performance. Most pitfalls involve over-simplifying the measurement environment and then drawing conclusions about real systems based on those measurements.

Measuring Single-Session Performance and Extrapolating to Multisession Behavior: Multiprogramming systems are complex environments whose characteristics change as the number of independent processes increases. Single-session measurements are much easier to perform and can be useful, but you will not usually be able to predict multisession performance using only single session data.

Measuring the TCP/IP System with a Varying Non-TCP/IP Load on the Same System: Accurate performance testing requires a controlled environment. If you attempt to measure changes on a system while other users are active on that system, you are likely to get inconsistent results. Sometimes it is appropriate to have other non-TCP/IP activity on the system during measurements but only if you can re-create the same activity so that you can compare the measurements to each other. Creating the same activity is often difficult and might require sophisticated tools.

Measuring the TCP/IP Network with a Varying Load from Other Hosts: The volume of data being transferred across the LAN can impact the resources used by the hosts you are measuring even if the data has a completely different destination. Ideally you should either isolate the measured network from all other systems or use a tool that will generate a consistent load during each set of measurements.

It might be tempting to believe that if you perform your testing outside the prime work shift for your location, you will avoid interference from a large, varying network load. Nevertheless, many users and systems might have set up their work so that large data transfers are delayed until after the prime work shift. In that case, beware because you might have a heavier network load off-shift than during prime shift.

Measuring in Loopback Mode: Loopback mode means that the client and the server are on the same host. No network traffic is generated by data transfer across a loopback connection. The primary useful purpose of loopback is functional testing. It is tempting to test loopback performance and use the results to draw conclusions about real network connections, but there are several reasons why these conclusions might not be valid. They are:

- Some implementations of TCP/IP do not use the same code to process loopback connections as they do for real network connections. Consequently, differences could include:
  - Path length
  - Storage management
  - Inter-process communication
- The client and the server might impact each other's performance on the loopback system if there is a resource constraint. For example, on a CPU-constrained server system, the addition of a client for loopback testing will increase wait time for the server because the additional client is also competing for the CPU.

- The loopback test might create a resource constraint where there was none when the client and the server were on separate hosts. An example of this situation would be when testing FTP on a host where the target file and the source file are on the same disk drive but are physically far enough apart to cause significant SEEK activity.
- If the real network connection will be between heterogeneous hosts, then either the client or the server being measured in loopback mode is executing code that will never be executed in the real network environment. If there are significant differences between the 2 client implementations or between the 2 server implementations, then loopback testing is unlikely to be useful. This is especially important when testing file server applications such as FTP, where the code that handles the file system has a significant impact on performance.

Therefore, if you want to use loopback to test your system, all of the following statements should be true:

- The loopback path and the network path through the TCP/IP implementation are identical except for the generation of network data traffic, which does not occur on loopback.
- The loopback system is not suffering from any resource constraints that might be aggravated by having both the client and the server on the same system.
- The client and the server on the loopback system are not attempting to share an otherwise unconstrained resource such as a disk drive.
- The real network system is between homogenous hosts; that is, the client implementation on the loopback system is the same as the client implementation on the remote host, and the file system on the loopback system is the same as the file system on the remote host.

Not Understanding the Margin of Error in Testing the Measured System: Even if your environment is highly controlled, you are likely to see variations of at least 5% when executing the same operation multiple times. If there are several uncontrollable variables, your results will vary more widely. You need to take this into account when interpreting the results of performance testing. If a test results in a 4% improvement in throughput, you are not safe in concluding with any certainty that the improvement was a result of a change that you made and not just normal run variation.

Some fields are more sensitive to variation than others. In our experience, the MVS and VM host CPU times are highly consistent fields, while FTP throughput is more variable.

Using Clock Times from One Source When Filtering Data from Another Source: Using clock times from one source to filter performance data collected from another source is likely to give you incorrect results. An example of this situation would be if you used the AIX clock to collect time stamps for a test and then used those time stamps to process MVS RMF data. Unless you had already confirmed that the AIX and MVS clocks were synchronized at the beginning of the test and were still synchronized at the end of the test, you will probably have some missing data or some extraneous data in your RMF reports.

Part 2. Capacity Planning

Our View of Capacity Planning

This part of the book covers capacity planning. The majority of the information is in reference to MVS and VM hosts and applications with a focus on CPU utilization, storage consumption, and disk I/O rate.

The applications discussed separately are:

- Telnet
- FTP

Notes about Our Measurement Environment

The following list explains some of the principles we followed in doing the measurements:

- We used a single instance of TCP/IP in all cases.
- We did not use logical partitioning on the 3090. We used LP for ES9021–982.
- We tried to be conservative because it is better to have a surplus of a resource than a shortage.
- Our measurements are made in a ***controlled*** system and network environment. It is unlikely that you will achieve the same level of throughput because you will have other activity on your system and network. Since your network environment is not the same as ours, the benchmarks that you do on your network will probably not exactly match the benchmarks we have done on our network.
- Our measurements were all made within a ***single*** LAN, which is essentially a direct connection. Real-life environments will usually include one or more routers. The fact that traffic has to be forwarded through a router means lower throughput. Therefore, you will probably see lower throughput than we report here.
- We calculate throughput by taking the user data being transferred and dividing it by the elapsed time. The actual amount of bytes sent across the network is higher because of headers, acknowledgements, and so forth. As a result, our throughput numbers are far from the limits of the hardware (controllers, media, and so on).

  For example, to transfer a 2017-byte file from VM to AIX can take 15 packets exchanged for a total of 3057 bytes. We divide 2017 by the elapsed time to get the throughput. If we used the total byte count of 3057, throughput would appear 50% higher.

Note: Because our IBM products are constantly under development, we have new benchmarks all the time. For information see “Where to Find Related Information on the Internet” on page ix

Chapter 7. Capacity Planning for Telnet

This chapter covers capacity planning for Telnet on MVS or VM. Included are sections on computing the CPU, storage, and disk I/O capacity needed.

Review Your Knowledge about Telnet and Your Hardware

The steps for a typical Telnet transaction works this way:

1. The user types a command.
2. The command line becomes the data portion of the Telnet operation (it can include data translation).
3. The Telnet client code sends the data to the Telnet server (this includes network delays).
4. The Telnet server sends the data to the application.
5. The application interprets the data as a command and executes the command, which results in output to the screen.
6. The screen output becomes the data portion of the Telnet operation.
7. The Telnet server code sends the data to the Telnet client (this includes network delays).
8. The Telnet client code displays the data on the user's screen (it can include data translation).

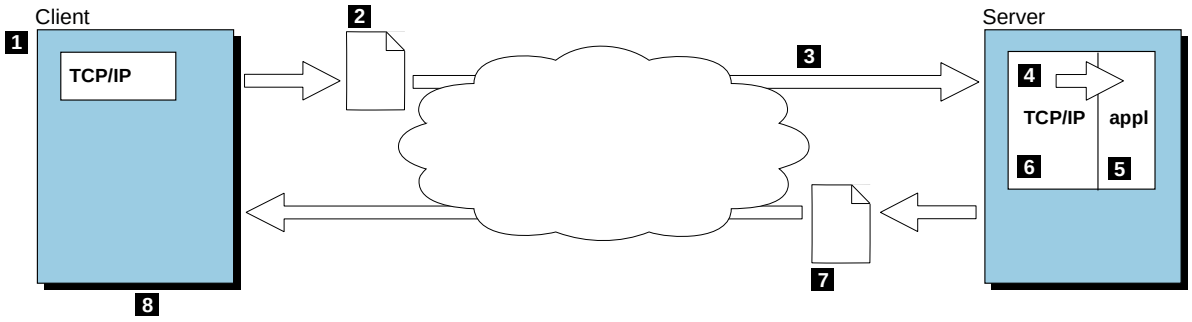


Figure 89. Diagram of the Telnet Transaction Process

Knowledge about Telnet

In Chapter 2, “Know Your Current Environment” on page 13, you were asked the following questions about Telnet:

- Which machines are servers and which are clients?
- How many connections need to be supported by each server?
- What terminal type will be emulated for each connection?
- What is the approximate transaction rate that the server needs to support?
- How much data is sent and received for each transaction?

If you are going to size the requirements for TCP/IP on the client, you need to consider steps 1, 2, 3, 7, and 8.

If you are going to size the requirements for TCP/IP on the server, you need to consider steps 3, 4, 6, and 7.

Notice that you do not need to consider step 5 when you size the TCP/IP requirements for a Telnet transaction. Step 5 includes the processing and other requirements for the application that are not part of TCP/IP processing. As a result you might want to disregard the non-TCP/IP part of the transaction when sizing the impact of TCP/IP. Even if the application were using a different protocol (SNA for example), the amount of time taken by step 5 would remain constant.

Knowledge about Your Hardware

In Chapter 2, "Know Your Current Environment" on page 13, you were asked the following questions about the hardware you use:

- What types of machines do you use?
  - What is the relative power of the CPU?
  - How much memory does it have?
  - How much disk space and how many paths to the disks does it have?
- What hardware and software do you use for your network connection?

Hardware Used in our Telnet Benchmarks

To produce the information in this chapter, different servers were used for different tests. The clients were a variety of single and multi-user workstations. The terminal type each client used was a 3270 (transparent, not line mode).

The primary focus of the tests was sizing the TCP/IP requirements for the server.

Computing the CPU Capacity Needed for MVS and VM TCP/IP

The following formula was used to estimate the CPU requirements for MVS Telnet server:

```
# transactions per user * # users * CPU seconds per transaction
-------------------------------------------------------------
                     # of elapsed seconds
```

To use this formula, you need to have an idea of the CPU seconds per transaction. We have measured CPU seconds per transaction for some network combinations, as shown in Table 48, Table 49, and Table 47. You can use our measurement data, but you need to adjust it based on the following variables, which impact the CPU seconds per transaction:

- ***Speed and architecture of the local and remote host***

  To compare your processor to ours, use the LSPR ITR (large systems performance reference internal throughput rate) numbers. You can get these numbers from your IBM marketing representative or systems engineer.

- ***Operating system and TCP/IP version of the host***

  Different system levels were used for our MVS and VM benchmark measurements.

- ***Number of bytes being sent and received***

  Our transactions sent 100 bytes and returned 800 bytes.

- ***Network connection hardware and software***

We used:

- ESCON channel at a speed of up to 18 MBps
- Parallel channel with a speed of 4.5MBps
- Token ring with a speed of 16Mbps

V2R2 VM Telnet Server Benchmarks

Table 47. Benchmarks for the VM Telnet Server

Hardware and Software	***CPU Load Per Transaction***	
ICC S/W	**Total**	**(No VTAM)**
ICP	0.00425	
Offload	0.00348	

Notes:

1. Client host: 3090-200J
2. Server host: 3090-200J
3. Operating system: VM/ESA* V1.2.0

V2R2.1 MVS Telnet Server Benchmarks

Table 48. Benchmarks for V2R2.1 Telnet on MVS

Hardware and Software					***CPU Per Transaction***		
Channel	**H/W**	**Speed**	**S/W**	**LAN**	**TCP/IP**	**VTAM**	**Total**
ITPECHO Applications							
Parallel	3172-3	50MHz	ICP	Token Ring	0.0077	0.0001	0.0078
ISPF Applications							
Parallel	3172-3	25MHz	ICP	Token Ring	0.0088	0.0001	0.0089
Parallel	3172-3	25MHz	Offload	Token Ring	0.0070	0.0001	0.0071

Notes:

1. Client host: 3090-200J
2. Server host: 3090-200J
3. Operating System: MVS/ESA* V4.2.2, VTAM v3.4

Note: The differences in the benchmarks were caused by the use of ITPECHO versus ISPF applications, not 25MHz versus 50MHz.

V3R1 MVS Telnet Server Benchmarks

Table 49. Benchmarks for V3R1 Telnet on MVS

Hardware and Software					CPU Per Transaction		
Channel	**H/W**	**Speed**	**S/W**	**LAN**	**TCP/IP**	**VTAM**	**Total**
No VTAM Internal Trace with ITPECHO Applications							
Parallel	3172-3	50MHz	ICP	Token Ring	0.0069	0.0001	0.0070
Parallel	3172-3	66MHz	Offload	Token Ring	0.0054	0.0001	0.0055
VTAM Internal Trace with ITPECHO Applications							
Parallel	3172-3	50MHz	ICP	Token Ring	0.0076	0.0001	0.0077
Parallel	3172-3	50MHz	Offload	Token Ring	0.0065	0.0001	0.0066
Parallel	3172-3	50MHz	ICP	FDDI	0.0080	0.0001	0.0081
Parallel	3172-3	50MHz	Offload	FDDI	0.0067	0.0001	0.0068
ESCON	3172-3	50MHz	ICP	Token Ring	0.0079	0.0001	0.0080
ESCON	3172-3	50MHz	ICP	FDDI	0.0080	0.0001	0.0081
ESCON	3172-3	50MHz	Offload	FDDI	0.0067	0.0001	0.0068

Notes:

1. Client host: 3090-200J
2. Server host: 3090-200J
3. Operating system: MVS/ESA* V4.2.2, VTAM v3.4

Note:

Using 50MHz versus 66MHz did not make a significant difference in the benchmarks.

V3R2 MVS Telnet Server Benchmarks (ESCON only)

Table 50. Benchmarks for V3R2 Telnet on MVS

Hardware and Software					CPU Per Transaction		
Channel	**H/W**	**Speed**	**S/W**	**LAN**	**TCP/IP**	**VTAM**	**Total**
VTAM Internal Trace with ITPECHO Applications							
ESCON	3172-3	66MHz	ICP	Token Ring	0.0025	0.0001	0.0026

Notes:

1. Client host: 9021-982 (2CP LPAR)
2. Server host: 9021-982 (2CP LPAR)
3. Operating system: MVS/ESA* V5.2.2, VTAM v4.3.0

Using MVS' benchmark for TCP/IP V3R2 as part of our example, we wanted to estimate the CPU capacity needed with 1 transactions per minute for each of 10000 users. Using our benchmarks, we could use 0.0026 for CPU seconds per transaction in the formula:

```
1 transaction per user * 10000 users * 0.0026 CPU seconds per transaction
------------------------------------------------------------------------
                          60 elapsed seconds

= 0.433 CPU seconds per elapsed second
```

The formula to get the CPU utilization is:

```
CPU seconds per elapsed second
------------------------------ * 100% = CPU utilization
     number of processors
```

We computed the CPU utilization in this example as:

```
0.433
----- * 100% = 43.3%
  1
```

As a result, we would expect our 10000 users to require 43.3% of one processor, leaving 56.7% for other work. If the current CPU utilization was already at 95%, then we would not have enough CPU capacity to accommodate this work load. (According to some standard capacity planning practices, you should plan for your CPU utilization not to exceed 70%.) In other words, 75% of a single processor would be able to support 17,321 users, each doing one transaction per minute.

Note: 17,321 users were derived by (10000*75)/43.3.

If the CPU seconds per elapsed second is higher than 1.0, then the work load cannot be handled by one processor of this speed.

CPU capacity has in some situations become an issue in recent years due to the growth of TCP/IP workloads and to the implementation of TCP/IP on OS/390 computing systems based on microprocessors. With the current TCP/IP stack implementation (up to and including V3R2), most of the protocol stack related processing is done under a single task within the system address space. That is, the CPU requirements of a protocol stack cannot exceed the CPU capacity of a single CPU engine. Multiple stacks can be spread across multiple CPU engines.

Computing the Storage Capacity Needed for MVS TCP/IP

There are two steps to getting the needed storage capacity:

1. Computing the storage needed by the TCP/IP address space regardless of the application
2. Computing the additional storage required to support the application based on the number of expected session.

Storage Required by TCP/IP for TELNET

The following table shows how storage allocation changes as Telnet sessions are established. This table can be used to determine how much storage per user will be required in addition to that allocated during TCP/IP initialization (the 0 Sessions column).

Chapter 9, “Examples of Capacity Sizing for MVS” on page 163 explains how to estimate storage use. Below the line storage grows at about 111 to 114

bytes per user. Above-the-line storage grows at about 3086 to 3620 bytes per user.

Determine how many sessions are required. Use the next higher number of sessions in the table. If 10,000 sessions are needed, use the column for 12,000 sessions.

Table 51. Storage Allocation for V3R2 TELNET (in bytes)

Sessions	**0**	**4000**	**8000**	**12000**	**16000**
TCP/IP Below	404,000	848,000	1,308,000	1,768,000	2,228,000
TCP/IP Above	176,000,000	180,000,000	184,000,000	189,000,000	197,000,000
TCP LSQA Below	212,000	212,000	212,000	212,000	212,000
TCP LSQA Above	29,900,000	30,600,000	31,300,000	32,800,000	32,800,000
System CSA Below	332,000	332,000	332,000	332,000	332,000
System CSA Above	18,700,000	25,800,000	32,900,000	41,700,000	50,700,000
System SQA Below	588,000	588,000	588,000	588,000	588,000
System SQA Above	22,700,000	22,800,000	22,800,000	22,800,000	22,900,000
Total Below	1,536,000	1,980,000	2,440,000	2,900,000	3,360,000
Total Above	247,300,000	259,200,000	271,000,000	286,300,000	303,400,000
Total	248,836,000	261,180,000	273,440,000	289,216,000	306,760,000
Per User Below		111	113	114	114
Per User Above		2,975	2,963	3,250	3,506
Per User Total		3,086	3,076	3,364	3,620

Note: We specified CSA = (3000, 64,000) and SQA = (8,256). The per user numbers are the increase per session and should be added to your 0 session amount.

We used the following TCP/IP profile to support 16,000 sessions.

```
; TCPPRF.DATA.PROFILES(U26TEL)
;
ACBPOOLSIZE                16200
ADDRESSTRANSLATIONPOOLSIZE  1500
CCBPOOLSIZE                  750
DATABUFFERPOOLSIZE     2500 8192
ENVELOPEPOOLSIZE            8500
IPROUTEPOOLSIZE               50
LARGEENVELOPEPOOLSIZE        100
RCBPOOLSIZE                   50
SCBPOOLSIZE                 3000
SKCBPOOLSIZE                 256
SMALLDATABUFFERPOOLSIZE    32400
TINYDATABUFFERPOOLSIZE      5000
TCBPOOLSIZE                16500
UCBPOOLSIZE                  100
;
assortedparms NOUDPQUEUELIMIT
alwayswto tcpipstatistics
endassortedparms
arpage 28800
INTERNALCLIENTPARMS TIMEMARK 28800
disablesga scaninterval 20
ENDINTERNALCLIENTPARMS
;
INFORM  USER1  ENDINFORM
OBEY    USER1 USER2 USER3 USER4 SNMPQE SNMPD
        USER5 USER6 USER7 USER8 USER9 USER10
;       USER11 USER12 USER13 USER14 USER15
;       USER16 USER17 USER18 USER19
;       ROUTED SMTP2 TCPUSR8 NAMESRV
ENDOBEY
;
; Hardware definitions:
; 3172--3 U26 ICP device
DEVICE LCS1   LCS           EC2
; LINK ETH1 ETHERNET 0 LCS1
  LINK TR1  IBMTR    0 LCS1
  LINK FDDI1 FDDI     0 LCS1



AUTOLOG
ENDAUTOLOG

PORT
 ; Values from RFC 1010, "Assigned numbers"
   23 TCP INTCLIEN               ; Telnet server


HOME
; Local host's Internet addresses
  9.67.113.21    TR1
; 9.67.114.12    ETH1
  9.67.115.19    FDDI1

GATEWAY
; Network  First hop   Driver  Packet size  Subnet mask  Subnet value

; Direct routes
; 9               =         ETH1     1500   0.255.255.128   0.67.114.0
9              =         TR1      2000   0.255.255.128   0.67.113.0
9              =       FDDI1      4096   0.255.255.128   0.67.115.0
```

```
DEFAULTNET 9.67.113.1     TR1    2000          0
;

;  Define the VTAM parameters required for the TELNET server
BEGINVTAM
    ; Define logon mode tables to be the defaults shipped with the latest
    ; level of VTAM
; 3277     M23270I
  3278-3-E NSX32703 ; 32 line screen - default of NSX32702 is 24 line screen
  3279-3-E NSX32703 ; 32 line screen - default of NSX32702 is 24 line screen
  3278-4-E NSX32704 ; 48 line screen - default of NSX32702 is 24 line screen
  3279-4-E NSX32704 ; 48 line screen - default of NSX32702 is 24 line screen
  3278-5-E NSX32705 ; 132 column screen - default of NSX32702 is 80 columns
  3279-5-E NSX32705 ; 132 column screen - default of NSX32702 is 80 columns
    ; Define the LUs to be used for general users
  DEFAULTLUS
     M00001..M16000
  ENDDEFAULTLUS
; DEFAULTAPPL TSO ;     Set the default application for all TELNET sessions
; LINEMODEAPPL TSO ; Send all line mode terminals directly to TSO
  ALLOWAPPL TSO* DISCONNECTABLE ; Allow all users access to TSO applications
              ; TSO is multiple applications all beginning with TSO so use
              ; the * to get them all. If a session is closed, disconnect
              ; the user rather than log off the user.
  RESTRICTAPPL A02NV ; Only three users may use IMS
    USER OPER1 ; Allow OPER1 access to Netview appl.                    h a
;     LU A02NV001
    USER USER1 ; Allow user1 access
;     LU OPER1 ; Assign USER1 LU TCPIMS01
    USER USER2 ; Allow user2 access from the default LU pool
    USER USER3 ; Allow user3 access from three telnet sessions, each with a
                ; different reserved LU.
  ALLOWAPPL * ; Allow all applications that have not been previously
              ; specified to be accessed
ENDVTAM

START LCS1
```

Guidelines for Configuring Buffer Pools and Control Blocks

The TCP/IP address space has a base amount of storage for the program and a variable amount based on the number of buffers specified in the pool statements in the MVS *tcpip*.PROFILE.TCPIP data set.

Table 52 lists the number of buffers required per user and their size.

Table 52. Additional Storage Needed Based on Pool Sizes (V3R2 for MVS/ESA).

Type	Size	3172-ICP	3172-Offload
ACBs*	1508	1 +more	2 +more
Small data buffers	2048	2	1 +more
TCBs	1056	1	1
Tiny data buffers	256	0	1 +more

Note: In the above table size for ACB includes actual ACB size of 100 bytes, plus 1408 bytes for internal control blocks.

The size and number of buffers and control blocks are different for Telnet clients and servers.

Additional buffers and control blocks needed are due to work not related to the number of users and transient work such as establishing the initial session. Extra buffers are also needed because of the permit size. You must allocate enough buffers to satisfy the permit size minimums as well as buffers for your users. This can be tricky because the permit size TCP/IP calculates is based on the amount of buffers you allocate. Use the following formula to allow enough for the permit size:

```
total buffers / (1 - permit percentage)
```

Table 53 lists the permit percentage needed to use the formula for Telnet.

Table 53. Buffer Permit Percentage Needed for Telnet (V3R2 for MVS/ESA).

Buffer	Permit Size Percentage
ACBs	10%
Small data buffers	10%
TCBs	10%
Tiny data buffers	10%

For example, if you need 1000 ACBs, then you should allocate:

```
1000 buffers / (1 - 10%) = 1000/.90 = 1112
```

You will also need some regular data buffers and envelopes for your Telnet users.

- For data buffers:

  ```
  total Telnet users * 12% = number of regular data buffers needed
  ```

  For example, if you will be supporting 1000 Telnet users, then you will need at least 120 data buffers.

- For envelopes:

  If the MTU size is greater than 2048 bytes, you should use large envelopes instead of regular envelopes.

  ```
  total Telnet users * 15% = number of envelopes needed
  ```

  For example, if you will be supporting 1000 Telnet users, then you will need at least 150 envelopes.

Remember that the estimated buffer amounts for Telnet do not include the buffers necessary to run FTP.

Computing the Disk I/O Capacity Needed for MVS and VM TCP/IP

The TCP/IP part of our Telnet work load does not require any disk I/O. Nevertheless, you should understand the non-TCP/IP disk I/O requirements for each transaction in your installation and ensure that your system has enough capacity to handle it.

Chapter 8. Capacity Planning Guidelines for FTP

This chapter covers capacity planning for FTP on MVS or VM. Included are sections on computing the CPU, storage, and disk I/O capacity needed.

| TCP/IP for Version 3 Release 2, available since September,1996, improves
| FTP performance significantly.

Review Your Knowledge about FTP and Your Hardware

The typical FTP GET transaction works this way:

1. The user enters GET for a file on the client machine.
2. The server retrieves the file from the server's file system, translates the file into network virtual terminal (NVT) format, and sends the data to the client. (NVT is similar to ASCII but even ASCII format files must be translated into NVT format for FTP.)
3. The FTP client translates the file from NVT format into its local file format and stores the file in the local file system.

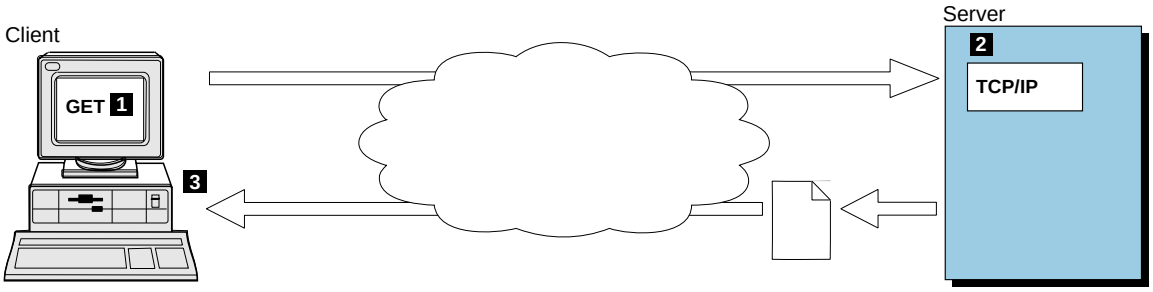


Figure 90. Diagram of the Typical FTP GET Process

The typical FTP PUT transaction works this way:

1. The user enters PUT for a file on the client machine.
2. The FTP client retrieves the file from the local file system, translates it into NVT format, and sends the data to the server.
3. The server translates the file from NVT, formats it into its local format, and stores the file into the server file system.

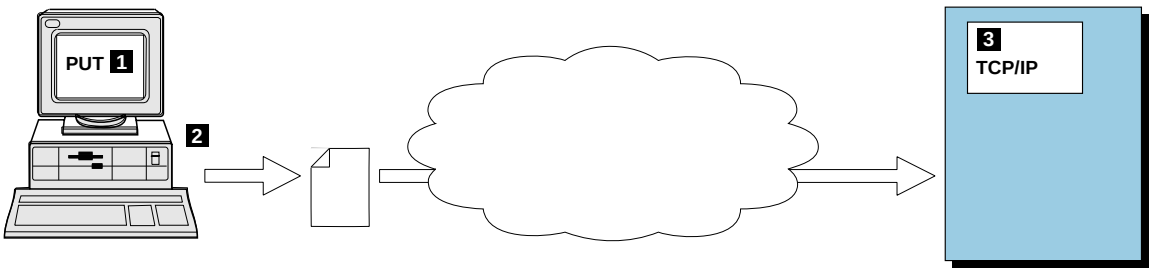


Figure 91. Diagram of the Typical FTP PUT Process

Note that ***some*** data translation takes place on both server and client no matter what the data format is and no matter which direction the data is going.

Knowledge about FTP

In Chapter 2, “Know Your Current Environment” on page 13, you were asked the following questions about FTP:

- Which machines are servers and which are clients?
- How often are your users transferring files?
- What is the average size of a file that is transferred?
- What are the file system characteristics of the client?
- What are the file system characteristics of the server?
- Do the files need to be translated?
- How critical is throughput?
- Where is the usual source/destination?

Knowledge about Your Hardware

You were also asked the following questions about your hardware:

- What machine types are being used?
  - What is the relative power of the CPU?
  - How much memory does it have?
  - How much disk space and how many paths to the disks does it have?
- What hardware and software do you use for your network connection?

Hardware Used in Our FTP Benchmarks

To produce the information in this chapter, we used a server that ran on either a 3090-400J, 3090-300J or ES9021-982 (2 CP LPAR) processor. We were primarily interested in determining the TCP/IP resource requirements for the server.

Computing the CPU Capacity Needed for MVS and VM TCP/IP

The CPU requirements for an FTP work load include both the FTP server and the TCP/IP address space (virtual machine on VM) requirements. The following formula was used to estimate the CPU requirements for each address space:

```
Maximum KB per Second * CPU Seconds per KB

= CPU seconds per elapsed second
```

To use this formula, you need to have an idea of the CPU seconds per KB.

We have measured CPU seconds per KB for some network combinations, as shown in Table 55 on page 150, Table 56 on page 150, Table 57 on page 151, and Table 58 on page 151. You can use our measurement data, but you need to adjust it based on the following variables which impact the CPU seconds per KB:

- ***Speed and architecture of the client and host***

  Our server measurements were done using the following machine types:

  - 3090-400J four-way processor
  - 3090-300J three-way processor
  - ES9021-982 (2 CP LPAR) machine

To compare your processor to ours, use the LSPR ITR (large systems performance reference internal throughput rate) numbers. You can obtain LSPR information from your IBM marketing representative or systems engineer.

The clients we used in our measurements were on RISC/System 6000 models 530 and 540.

- ***Operating system and TCP/IP version of the host***

  We used the following types of MVS, VM and TCP/IP versions.

  For MVS:

  – MVS/ESA 4.2.2 and TCP/IP for MVS Version 2.2.1

  – MVS/ESA 5.2.2 and TCP/IP for MVS Version 3.1

  – MVS/ESA 5.2.2 and TCP/IP for MVS Version 3.2

  For VM:

  – VM/ESA 1.2.0 and TCP/IP for VM Version 2.2

- ***Size of the files being transferred***

  Our file sizes varied with the type of measurement. File sizes between 4, 10 and 20 Mbytes were used for these measurements.

- ***Disk subsystem***

  We used 3380 and 3390 DASD with 3990-3 caching control units.

- ***PUT/GET ratio***

  We assumed an equal amount of PUTs and GETs used in the measurement.

- ***File system characteristics***

  We used data sets with a block size of 23 424 and fixed record of 64 bytes.

- ***Packet size***

  We used 1500 for Ethernet, 2000 for Token Ring, and 4352 for FDDI.

- ***TCP send and receive buffer sizes***

  We used buffers of 8 192, 16 384, 28 672, 32 768 and 65 536 on the MVS host. The TCP send and receive buffers used on RISC/System 6000 were 32 768 and 65 536.

  Note: Buffer sizes are entered as the numbers without spaces when specified in the TCP/IP Profile.

- ***Network connection hardware and software***

  We used in various combinations:

  – Parallel channel with a speed of 4.5MBps
  – ESCON channel with a speed of up to 18MBps
  – Token Ring with a speed of 16Mbps
  – Ethernet with a speed of 10Mbps
  – FDDI with a speed of 100Mbps

These are the groups of benchmarks provided in this chapter:

1. TCP/IP for VM version 2 release 2 FTP
   - With binary files
   - With ASCII files
2. TCP/IP for MVS version 2 release 2.1 FTP
   - With binary files
   - With ASCII files
3. TCP/IP for MVS version 3 release 1 FTP
   - With binary files
   - With ASCII files
   - With MVS as client, AIX as server
   - With data compression
   - With checkpoint/restart
   - With an increase in window size from 32KB to 64KB
4. TCP/IP for MVS version 3 release 2 FTP
   - With binary files
   - With ASCII files
   - With MVS as client, AIX as server

   Note: The C FTP server is significantly enhanced with TCP/IP V3R2. The Pascal FTP Server is no longer supported under TCP/IP V3R2 for MVS. The Pascal FTP Client is still supported under V3R2.

V2R2 VM FTP Benchmarks

Table 54. Benchmarks for FTP on VM on 400J

BINARY DATA (No File Translation)									
Hardware and Software					*CPU Load per 10000KB*			Maximum Aggregate	*Single Session*
Channel	**ICC H/W**	**Speed**	**ICC S/W**	**LAN**	**TCP/IP**	**FTP**	**Total**	**Throughput (KBps)**	**PUT/GET**
Parallel	3172-3	25MHz	ICP	Token Ring	2.404	0.743	3.147	1555	966/893
Parallel	3172-3	25MHz	Offload	Token Ring	1.241	0.830	2.072	556	556/422
Parallel	3172-3	25MHz	ICP	Ethernet	2.794	0.770	3.564	1165	968/798
Parallel	3172-1	25MHz	ICP	Ethernet	2.773	0.771	3.545	813	813/588
Parallel	3172-3	25MHz	Offload	Ethernet	1.212	0.814	2.026	588	559/442
Parallel	3172-3	50MHz	ICP	Ethernet	2.788	0.768	3.555	1291	955/983
Parallel	3172-3	50MHz	Offload	Ethernet	1.119	0.751	1.870	858	577/554
Parallel	3172-3	25MHz	ICP	FDDI	1.921	0.844	2.766	1405	1027/1022
Parallel	3172-3	25MHz	Offload	FDDI	1.174	0.764	1.938	536	433/410
Parallel	3172-3	50MHz	Offload	FDDI	1.257	0.850	2.107	875	577/537
Parallel	3088 CTC				1.548	0.708	2.256	919	901/786
Parallel	RISC	Mod 540	AIX 3.2		2.024	0.729	2.753	1606	1028/1035
Parallel	RISC	Mod 530h	AIX 3.2		2.074	0.751	2.825	1643	1010/1001
ESCON	RISC	Mod 540	AIX 3.2		2.239	0.834	3.074	2132	994/1008
Parallel	RISC	Mod 540	AIX 3.2	FDDI	2.146	0.791	2.936	1559	1025/1039
ESCON	RISC	Mod 540	AIX 3.2	FDDI	1.990	0.791	2.781	2376	975/1089
Parallel	RISC	Mod 540	AIX 3.2	Ethernet	3.796	0.778	4.573	652	441/505
ESCON	RISC	Mod 540	AIX 3.2	Ethernet	3.760	0.780	4.539	851	502/838
ASCII DATA									
Parallel	3172-1	25MHz	ICP	Ethernet	3.217	2.950	6.167		966/893
Parallel	3172-3	25MHz	ICP	Ethernet	3.171	3.025	6.196		410/433

V2R2.1 MVS FTP Benchmarks with No File Translation (Binary Data)

Our measurements were made with 4MB files on Token Ring and Ethernet, 10MB for FDDI.

Table 55. Benchmarks for V2R2.1 FTP on MVS on 400J for Binary Data

	Hardware and Software				CPU Load per 10000 KB			Maximum Aggregate	Single Session
Channel	**H/W**	**Speed**	**S/W**	**LAN**	**TCP/IP**	**FTP**	**Total**	**Throughput (KBps)**	**PUT/GET**
MVS as Server, AIX as Client									
ESCON	3172-3	50MHz	ICP	FDDI	2.567	1.318	3.885	2731	934/1385
ESCON	3172-3	50MHz	Offload	FDDI	1.427	1.343	2.770	1138	642/747
ESCON	3172-3	50MHz	ICP	Token Ring	3.693	1.497	5.190	1365	931/1263
ESCON	3172-3	50MHz	Offload	Token Ring	1.510	1.508	3.018	878	815/614
Parallel	3172-3	50MHz	ICP	FDDI	2.870	1.578	4.448	2156	924/1342
Parallel	3172-3	50MHz	ICP	Token Ring	3.734	1.500	5.234	1489	935/1204
Parallel	3172-3	50MHz	ICP	Ethernet	3.947	1.498	5.445	1092	940/989
Parallel	3172-1	25MHz	ICP	Ethernet	3.805	1.410	5.215	768	751/563
ESCON	RISC	Mod 530	AIX 3.2	None	3.027	1.447	4.474	2048	930/1272
ESCON	RISC	Mod 530	AIX 3.2	FDDI	2.569	1.324	3.893	2194	933/1307
MVS as Client, AIX as Server									
Parallel	3172-3	50MHz	ICP	Token Ring	3.614	2.145	5.759	1271	1271/925

V2R2.1 MVS FTP Benchmarks with ASCII File Translation

Our measurements were made with 4MB files on Token Ring and Ethernet,

Table 56. Benchmarks for V2R2.1 FTP on MVS on 400J for ASCII Data

	Hardware and Software				CPU Load per 10000 KB			Single Session
Channel	**ICC H/W**	**Speed**	**ICC S/W**	**LAN**	**TCP/IP**	**FTP**	**Total**	**PUT/GET**
MVS as Server, AIX as Client								
ESCON	3172-3	50MHz	ICP	FDDI	4.066	2.653	6.720	570/516
ESCON	3172-3	50MHz	Offload	FDDI	1.513	2.175	3.688	578/512
ESCON	3172-3	50MHz	ICP	Token Ring	4.557	2.336	6.893	529/470
ESCON	3172-3	50MHz	Offload	Token Ring	1.597	2.347	3.944	513/462
Parallel	3172-3	50MHz	ICP	FDDI	4.129	2.650	6.779	571/512
Parallel	3172-3	50MHz	ICP	Token Ring	4.652	2.329	6.981	530/460
Parallel	3172-3	50MHz	ICP	Ethernet	4.327	2.339	6.666	534/464
Parallel	3172-1	25MHz	ICP	Ethernet	4.281	2.338	6.519	518/443
ESCON	RISC	Mod 530	AIX 3.2	None	5.369	2.902	8.271	342/375
ESCON	RISC	Mod 530	AIX 3.2	FDDI	3.818	2.271	6.089	556/511
MVS as Client, AIX as Server								
Parallel	3172-3	50MHz	ICP	Token Ring	4.661	4.164	8.824	457/448

Note: For the ASCII measurements, throughput was limited by the constrained CPU on the AIX client. These measurements are included to show the CPU requirements for the server.

V3R1 MVS FTP Benchmarks with No File Translation (Binary Data)

Our measurements were made with 10MB files on Token Ring and Ethernet and 20MB files on FDDI.

Table 57. Benchmarks for V3R1 FTP on MVS on 300J for Binary Data

Hardware and Software						*CPU Load per 10000 KB*		Maximum Aggregate	*Single Session*
Channel	**H/W**	**Speed**	**S/W**	**LAN**	**TCP/IP**	**FTP**	**Total**	**Throughput (KBps)**	**PUT/GET**
MVS as Server, AIX as Client									
ESCON	3172-3	50MHz	ICP	FDDI	2.562	1.160	3.722	2560	928/1326
ESCON	3172-3	50MHz	Offload	FDDI	1.392	1.163	2.555	1138	861/728
ESCON	3172-3	50MHz	ICP	Token Ring	3.616	1.211	4.827	1412	929/1318
ESCON	3172-3	50MHz	Offload	Token Ring	1.429	1.225	2.654	878	844/649
Parallel	3172-3	50MHz	ICP	FDDI	2.791	1.369	4.160	2214	923/1499
Parallel	3172-3	50MHz	Offload	FDDI	1.402	1.162	2.564	1138	822/759
Parallel	3172-3	50MHz	ICP	Token Ring	3.599	1.204	4.803	1412	925/1230
Parallel	3172-3	50MHz	Offload	Token Ring	1.434	1.225	2.659	931	744/667
MVS as Client, AIX as Server									
Parallel	3172-3	50MHz	ICP	Token Ring	3.580	1.930	5.509	1243	1243/900

V3R1 MVS FTP Benchmarks with ASCII File Translation

Our measurements were made with 4MB files.

Table 58. Benchmarks for V3R1 FTP on MVS on 300J for ASCII Data

Hardware and Software					*CPU Load per 10000 KB*			Single Session
Channel	**H/W**	**Speed**	**S/W**	**LAN**	**TCP/IP**	**FTP**	**Total**	**PUT/GET**
MVS as Server, AIX as Client								
Parallel	3172-3	50MHz	ICP	Token Ring	5.260	2.278	7.538	397/354
MVS as Client, AIX as Server								
Parallel	3172-3	50MHz	ICP	Token Ring	4.597	3.835	8.433	447/438

V3R1 MVS FTP Benchmarks with Data Compression

Besides ASCII translation, there are certain features in the FTP in TCP/IP for MVS V3R1 that may require additional overhead. If your users will be taking advantage of any of these features, you will need to plan for additional capacity. We have included some measured data for 2 features: data compression and checkpoint/restart. Overhead and savings for data compression will depend on how tightly compressed the data set you are sending becomes. Here are some guidelines:

- Duplicated, adjacent characters are compressible
- Duplicated records are not compressible
- Duplicated blanks are slightly more compressible than other duplicated characters

We tested a dense file which had no duplicated characters, a medium file which could be compressed by about 50% and a nearly blank file which could be compressed by about 85%.

All of the benchmarks in Table 59 were done with:

- EBCDIC data
- TCP/IP for MVS V3R1
- 3080-300J MVS/ESA as server
- 3080-300J MVS/ESA as client
- Parallel channel
- 3172-3
- 50 MHz
- ICP
- Token Ring

Table 59. Benchmarks for V3R1 FTP Data Compression

		CPU Load per 10000 KB			
File	**Transfer Mode**	**TCP/IP**	**FTP**	**Total**	**Single User PUT/GET**
Dense	Block	3.128	1.377	4.506	922/921
Dense	Compressed	3.209	3.345	6.555	926/910
Medium	Block	3.078	1.360	4.438	927/919
Medium	Compressed	1.861	6.271	8.131	843/824
Sparse	Block	3.084	1.370	4.454	925/929
Sparse	Compressed	0.845	5.405	6.250	777/761

V3R1 MVS FTP Benchmarks with Checkpoint/Restart

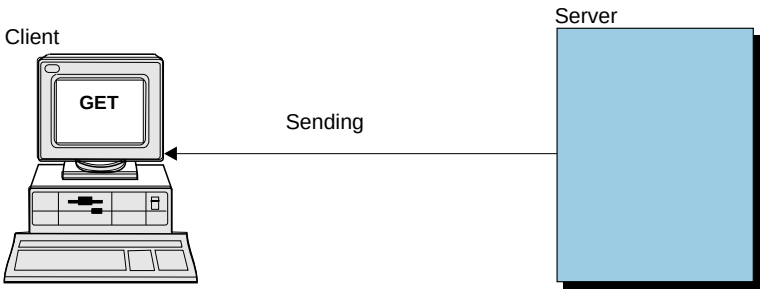
When you use checkpoint/restart with FTP, the overhead will depend upon the checkpoint interval. The larger the interval you choose, the lower the overhead will be. Since checkpointing is done by the sending side of a data transfer:

- the ***server*** is the sending side for a GET
- the ***client*** is the sending side for a PUT

We measured an MVS server with an AIX client doing a GET and an AIX server with an MVS client doing a PUT. MVS overhead for varying checkpoint interval is shown in Table 60 and Table 61.

All of the benchmarks in Table 60 and Table 61 were done with:

- TCP/IP for MVS V3R1
- 3080-300J with MVS/ESA
- RISC/System 6000 model 530 with AIX 3.2
- Parallel channel
- 3172-3
- 50 MHz
- ICP
- Token Ring



valign=bottom scale='.8'.

Table 60. Benchmarks for V3R1 FTP Checkpoint/Restart with MVS Server and AIX Client

	CPU Load * 10000 per KB			***Single User***
Chkpt Int	**TCP/IP**	**FTP**	**Total**	**GET Tput**
0	4.975	0.629	5.604	392
1	5.748	0.751	6.499	326
10	5.668	0.750	6.418	325
100	5.733	0.751	6.484	319
1000	5.286	0.685	5.971	366

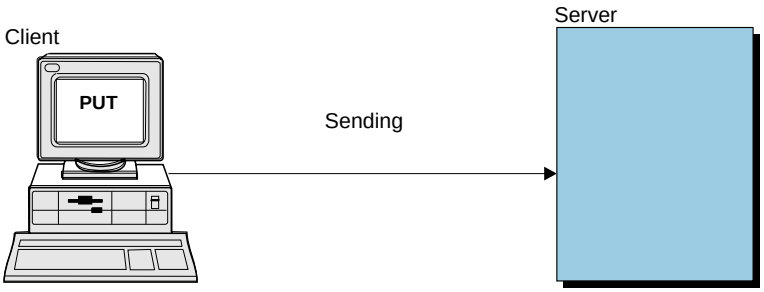


Table 61. Benchmarks for V3R1 FTP Checkpoint/Restart with MVS Client and AIX Server

	*CPU Load * 10000 per KB*			*Single User*
Chkpt Int	**TCP/IP**	**User**	**Total**	**PUT Tput**
0	4.743	0.116	4.859	460
1	5.999	0.106	6.105	451
10	6.019	0.106	6.125	450
100	6.059	0.106	6.165	448
1000	5.505	0.112	5.617	455

In addition, you should be aware that there may be additional overhead associated with checkpointing on the receiving side of a file transfer. The overhead will depend on how the receiver processes the checkpointed data. If each checkpoint marker in the data stream causes a temporary file to be closed and re-opened (as is often the case), then a small checkpoint interval may cause excessive overhead on the receiver and drastically reduce throughput.

Table 62 shows benchmarks for the same configuration but with the window size changing between 32KB and 64KB.

All of the following benchmarks were done with:

- Binary data
- TCP/IP for MVS V3R1
- 3080-300J MVS/ESA as server
- RISC/System 6000 model 530 with AIX 3.2 as client
- Parallel channel
- 3172-3
- 50 MHz
- ICP
- FDDI

Table 62. Benchmarks for V3R1 FTP on MVS for Binary Data Comparing Window Sizes

	CPU Load per 10000 KB			**Maximum Aggregate**	**Single Session**
Window Size	**TCP/IP**	**FTP**	**Total**	**Throughput (KBps)**	**PUT/GET**
32KB	2.791	1.369	4.160	2214	923/1499
64KB	2.478	1.103	3.581	2339	940/2339

V3R2 MVS FTP Benchmarks with No File Translation (Binary Data)

Our measurements were made with 20MB files using an FDDI adapter.

Table 63. Benchmarks for V3R2 FTP on MVS/ESA 5.2.2 on ES9021-982 (2CP LPAR).

Hardware and Software						CPU Load per 10000 KB		Maximum Aggregate	Single Session
Channel	H/W	Speed	S/W	LAN	TCP/IP	FTP	Total	Throughput (KBps)	PUT/GET
MVS as Server, AIX as Client									
ESCON	3172-3	66MHz	ICP	FDDI	0.58	0.33	0.91	3661	1965 (PUT)
ESCON	3172-3	66MHz	ICP	FDDI	0.56	0.17	0.73	4150	2632 (GET)
MVS as Client, AIX as Server									
ESCON	3172-3	66MHz	ICP	FDDI	0.68	0.50	1.18	-	3543 (PUT)
ESCON	3172-3	66MHz	ICP	FDDI	0.77	0.68	1.45	-	1946 (GET)

V3R2 MVS FTP Benchmarks with ASCII File Translation

Our measurements were made with 20MB files.

Table 64. Benchmarks for V3R2 FTP on MVS/ESA on ES9021-982 (2CP LPAR)

Hardware and Software					CPU Load per 10000 KB			Single Session
Channel	H/W	Speed	S/W	LAN	TCP/IP	FTP	Total	PUT/GET
MVS as Server, AIX as Client								
ESCON	3172-3	66MHz	ICP	FDDI	0.56	0.65	1.21	1465 (PUT)
ESCON	3172-3	66MHz	ICP	FDDI	0.59	0.35	0.94	1931 (GET)
MVS as Client, AIX as Server								
ESCON	3172-3	66MHz	ICP	FDDI	0.87	0.65	1.52	666 (PUT)
ESCON	3172-3	66MHz	ICP	FDDI	1.40	1.72	3.12	653 (GET)

V3R2 MVS FTP Window size Benchmarks

Our measurements were made with 20MB files on FDDI.

All of the following benchmarks were done with:

- Binary data
- TCP/IP for MVS/ESA 5.2.2 V3R2
- ES9021-982 (2CP LPAR) MVS/ESA as server
- RISC/System 6000 model 530 with AIX 3.2 as client
- ESCON channel
- 3172-3
- 66 MHz
- ICP
- FDDI

Table 65. Benchmarks for V3R2 FTP on MVS for Binary Data Comparing Window Sizes

	CPU Load per 10000 KB			Maximum Aggregate	Single Session
Window Size	TCP/IP	FTP	Total	Throughput (KBps)	PUT/GET
32KB	0.58	0.33	0.91	3590	1965 (PUT)
32KB	0.56	0.18	0.74	4150	2632 (GET)
64KB	0.58	0.33	0.91	-	2024 (PUT)
64KB	0.56	0.17	0.73	-	3186 (GET)

Example of Computing CPU Capacity on MVS

For this example, we used the FTP benchmark data for the following configuration:

- Binary data (MVS is used as FTP server)
- TCP/IP V3R2 for MVS/ESA 5.2.2
- ES9021-982 2 CP LPAR
- RISC/System 6000 Model 530 with AIX 3.2 Client
- ESCON Channel
- 3172-3 P66 Mhz
- ICP 3.3
- FDDI as LAN media
- Workstation and MVS TCP/IP window size–32768
- Packet size–4352

Table 66. MVS Host CPU and Throughput data (TCP/IP V3R2 for MVS/ESA)

Transfer Type	Throughput in KBytes/sec		Host CPU seconds for 10000KB		
	SS	Max Tput	TCP/IP	FTP	Total
Binary PUT	1965	3590	0.58	0.33	0.91
Binary GET	2632	4150	0.56	0.17	0.73
ASCII PUT	1465	3535	0.56	0.65	1.21
ASCII GET	1931	4185	0.58	0.35	0.93

In this example, we used binary PUT data for computing CPU capacity. From the benchmark, we put 0.000058 for TCP/IP CPU seconds per KB in the formula:

```
Maximum KB per Second * CPU Seconds per KB
   3590 KB per Second * 0.000058 CPU seconds per KB
= 0.20822  CPU seconds per elapsed second
```

Then we put 0.000033 for FTP server CPU seconds per KB in the formula:

```
3590 KB per Second * 0.000033 CPU seconds per KB
= 0.11847  CPU seconds per elapsed second
```

The formula to get the CPU utilization is:

```
CPU seconds per elapsed  second
------------------------------- * 100%  = CPU utilization
      number of processors
```

We computed the TCP/IP address space CPU utilization in this example as:

```
0.20822
------- * 100%  = 20.82%
   1
```

We computed the FTP server address space CPU utilization in this example as:

```
0.11847
------- * 100%  = 11.85%
   1
```

The CPU computation is done on the basis of a single session FTP transfer.

To calculate the CPU affect for multiple session file transfer, additional CPU may be required per FTP session. We recommend that you add 15 to 20% percent contingency on the TCP/IP and FTP utilization. In other words, 20.82% utilization for TCP/IP address space would be 23.94%, when a 15% contingency is added. When a 15% contingency is added, an 11.85% for FTP address space would be 13.63%,

Because FTP runs in a separate address space from the TCP/IP address space, if you are using a multiple CP processor you will see the benefits of load balancig across processors. As a result, instead of 32.67% utilization of a single processor, the CPU load could be split with the 20.82% on one and 11.85% on another.

When you include a 15% contingency, instead of 37.57% utilization of a single processor, the CPU load could be split with 23.94% on one and 13.63% on another.

If the CPU seconds per elapsed second is higher than 1.0, then the work load cannot be handled by one processor of this speed.

Note: The estimated CPU utilization is required only if you sustain the maximum throughput rate on a continuous basis. Most of the time, your users will not be using FTP enough to average at the estimated rate. That rate will only be reached periodically. Keep in mind that system overhead is not included in the above computation.

Computing the Storage Capacity Needed for MVS TCP/IP V3R2

The following is the storage capacity required for the C-FTP server using TCP/IP V3R2.

C-FTP (Binary Get) Storage Usage (TCP/IP V3R2 for MVS/ESA)

The following table shows the MVS virtual storage usage when doing 1,2,3 and 6 FTP Binary Gets. Storage usage is shown for the TCP/IP address space, C-FTP address space and System CSA & SQA.

Number of Sessions	1	2	3	6
Aggr TPUT (KB/Sec)	2632	3688	4097	4150

Number of Sessions	1	2	3	6
Region Size Below	8172K	8172K	8172K	8172K
Region Size Above	1894M	1894M	1894M	1894M
TCP/IP AS Free Below	7544K	7544K	7544K	7544K
TCP/IP AS Free Above	1841M	1841M	1841M	1841M
C-FTP AS Free Below	4964K	4960K	4144K	900K
C-FTP AS Free Above	1878M	1878M	1878M	1876M
TCP/IP AS Below	404K	404K	404K	404K
TCP/IP AS Above	35.3M	35.3M	35.3M	35.4M
TCP/IP AS LSQA Below	224K	224K	224K	224K
TCP/IP AS LSQA Above	17.6M	17.6M	17.6M	17.6M
C-FTP AS Below	2996K	2996K	3804K	7016K
C-FTP AS Above	5176K	5176K	5176K	5260K
C-FTP AS LSQA Below	212K	216K	224K	256K
C-FTP AS LSQA Above	10.8M	11.0M	11.3M	12.9M
System CSA Below	332K	332K	332K	332K
System CSA Above	12.7M	12.7M	12.7M	12.7M
System SQA Below	600K	600K	600K	600K
System SQA Above	22.7M	22.7M	22.7M	22.7M
Total Below	4.77M	4.77M	5.59M	8.83M
Total Above	104.3M	104.5M	104.8M	106.6M
Total	109.1M	109.3M	110.4M	115.4M

Table 67. Storage usage by TCP/IP AS, C-FTP AS & System

1. Above numbers used 3 RS6Ks as clients, 9021-982 (2 CP LPAR) running MVS 5.2.2 and MVS TCP/IP V3R2 as a server.
2. For C-FTP, BUFNO=35, EXTRATASKS=10 were used.
3. TCP/IP and C-FTP region size requested was 0K.
4. Above and below storage taken from RMF VSTOR.
5. Use is constant for TCP/IP, System CSA and System SQA.
6. Use is proportional to users for C-FTP Below, C-FTP Above, C-FTP LSQA Below and C-FTP LSQA Above.

   C-FTP Below : Increase of 808 to 1072 KB/user

   C-FTP Above : Increase of 28 KB/user

   C-FTP LSQA Below: Increase of 4 to 11 KB/user

   C-FTP LSQA Above: Increase of .2 to .5 MB/user

C-FTP (Binary PUT) Storage Usage (TCP/IP V3R2 for MVS/ESA).

The following table shows the MVS virtual storage usage when doing 1,2,3 and 6 FTP Binary Puts. Storage usage is shown for the TCP/IP address space, C-FTP address space and System CSA & SQA.

Number of Sessions	1	2	3	6
Aggr TPUT (KB/Sec)	1965	3100	3661	3590
Region Size Below	8172K	8172K	8172K	8172K
Region Size Above	1894M	1894M	1894M	1894M
TCP/IP AS Free Below	7544K	7544K	7544K	7544K
TCP/IP AS Free Above	1841M	1841M	1841M	1841M
C-FTP AS Free Below	5796K	4972K	4144K	4948K
C-FTP AS Free Above	1841M	1841M	1841M	1841M
TCP/IP AS Below	404K	404K	404K	404K
TCP/IP AS Above	35.3M	35.3M	35.3M	35.3M
TCP/IP AS LSQA Below	224K	224K	224K	224K
TCP/IP AS LSQA Above	17.6M	17.6M	17.6M	17.6M
C-FTP AS Below	2164K	2984K	3804K	2996K
C-FTP AS Above	5120K	5148K	5176K	5260K
C-FTP AS LSQA Below	212K	216K	224K	228K
C-FTP AS LSQA Above	9.9 M	10.2M	10.7M	11.9M
System CSA Below	332K	332K	332K	332K
System CSA Above	12.7M	12.7M	12.7M	12.7M
System SQA Below	600K	600K	600K	600K
System SQA Above	22.7M	22.7M	22.7M	22.7M
Total Below	3.94M	4.76M	5.59M	4.78M
Total Above	103.3M	103.7M	104.2M	105.5M
Total	107.3M	108.5M	109.8M	110.3M

Table 68. Storage usage by TCP/IP AS, C-FTP AS & System

1. Above numbers used 3 RS6Ks as clients, 9021-982 (2 CP LPAR) running MVS 5.2.2 and MVS TCP/IP V3R2 as a server.
2. TCP/IP and C-FTP Region size requested was 0K.
3. For C-FTP, BUFNO=35, EXTRATASKS=10 were used.
4. Above and below storage taken from RMF VSTOR.
5. Use is constant for TCP/IP, System CSA and System SQA.
6. Use is proportional to users for C-FTP Below, C-FTP Above, C-FTP LSQA Below and C-FTP LSQA Above.

   C-FTP Below : Increase of 820 KB/user

   C-FTP Above : Increase of 28 KB/user

   C-FTP LSQA Below: Increase of 2 to 8 KB/user

   C-FTP LSQA Above: Increase of .3 to .5 MB/user

Guidelines for Configuring Buffer Pools and Control Blocks

The TCP/IP address space has a base amount of storage for the program and a variable amount based on the number of buffers specified in the pool statements in the MVS PROFILE.TCPIP data set.

This following table lists the number of buffers required per user and their size:

Table 69. Buffers for an FTP Server (C-FTP and Pascal)

Buffer	Size	3172-Offload	RISC channel	C-FTP Server 3172-ICP
ACBs	96	7	8	8
Data buffers	8-32KB	7	9	10
Envelopes	2048	0	20 if MTU ≤ 2048	1 if MTU ≤ 2048
Large envelopes	8-32KB	0	8 if MTU > 2048	8 if MTU > 2048
SCBs	228	2	5	2
SKCBs	781	0	0	3
Small data buffers	2048	1	0	0
TCBs	1056	4	1	5
Tiny data buffers	256	1	0	0

The size and number of buffers are different for FTP clients and servers. You will also need for each FTP server:

- 1 CCB
- 1 data buffer
- 2 SCBs
- 1 TCB
- 1 UCB (C FTP only)

Additional buffers needed are due to work not related to the number of users and transient work such as establishing the initial session. Extra buffers are also needed because of the permit size. You must allocate enough buffers to satisfy the permit size minimums as well as enough buffers for your users. This can be tricky because the permit size TCP/IP calculates is based on the amount of buffers you allocate. We suggest you use the following formula to allow enough for the permit size:

```
total buffers / (1 - permit percentage)
```

Table 70 lists the permit percentage needed to use the formula for FTP.

Table 70. Buffer Permit Percentage Needed for FTP for TCP/IP V3R2 for MVS/ESA

Buffer	Permit Size Percentage
ACBs	10%
Data buffers	10%
Envelopes	10%
Large envelopes	10%
SCBs	10%
Small data buffers	10%
Tiny data buffers	10%
TCBs	10%

For example, if you need 1000 ACBs, then you should allocate:

```
1000 buffers / (1 - 10%) = 1000/.90 = 1112
```

Remember that the estimated buffer amounts for FTP do not include the buffers necessary to run Telnet.

Computing the Disk I/O Capacity Needed for MVS and VM TCP/IP

When determining the disk I/O capacity requirements for your system, you should understand the file system characteristics. For our measurements, we used a block size of 23 424 and a logical record length of 64 bytes. The record format was fixed, not variable. Smaller block sizes increase the number of disk operations required during an FTP file transfer.

To compute the needed disk I/O capacity, we used the following formula:

```
 KB per second
--------------- = disk I/O operations per second
KB per disk I/O
```

In our example, we wanted to be able to handle 494.5KB per second. With these file system characteristics, we transferred approximately 22KB per disk I/O:

```
494.5KB per second
------------------ = 23 disk I/O operations per second
22KB per disk I/O
```

On VM, the amount of KB per disk I/O varies with the data buffer size.

As long as the files being transferred reside (or will reside) on a disk that gives the level of service that is necessary, then disk I/O will not be a bottleneck.

You must also place your file on a disk system capable of supporting your transfer rate. For high speed LANs, such as FDDI or HPPI, this is especially important. If your device cannot support the higher rate, you cannot achieve the capacity of the LAN media.

Here are some transfer rate maximums:

Table 71. Disk System Maximum Transfer Rates

Device	Type	Maximum
3380	3MB/sec channel	3072KBps
3390	4.5MB/sec channel	4301KBps
3390	SMS with 4 stripes	13312KBps
3390	SMS with 8 stripes	24576KBps

Chapter 9. Examples of Capacity Sizing for MVS

This chapter provides examples of how to size the capacity on MVS for Telnet and FTP.

Examples of Estimating CPU and Storage Usage for Telnet 3270

This section gives an example of estimating both CPU and storage usage for Telnet 3270 on MVS. We used the formulas and benchmarks listed in Chapter 7, “Capacity Planning for Telnet” on page 135 to do our calculations.

Note: Note that the estimates in this examples are in addition to the CPU, storage, and disk requirements for the application being accessed through Telnet. We have used TCP/IP V3R2 MVS telnet in the example.

Step 1. Review Your Knowledge about Telnet and Your Hardware

To illustrate Telnet capacity sizing, we have considered 10000 Telnet users. All 10000 Telnet users are handled by a Telnet server on a single TCP/IP image on an MVS system.

Each of the 10000 users do 1 Telnet SEND of 100 bytes and RECEIVE 800 bytes per minute on the average. The 10000 users with two 3172 Offload machines to support 1500 users each (total 3000 users) and 2 ICP machines to support 3500 users each (total 7000 users).

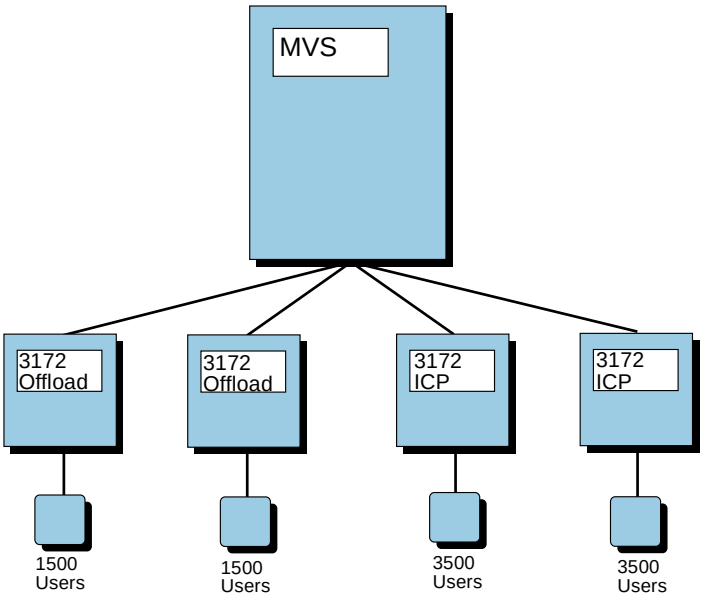


Figure 92. Diagram of Telnet 3270 Estimating Example

Step 2. Computing the CPU Capacity Needed

To estimate the CPU capacity needed, we have used the CPU cost of doing a Telnet transaction based on our Telnet test of 16000 users. Our 16000 Telnet users test was done in an ICP (non-Offload) environment using a 3172-3 controller. For Offload users, we have assumed 12% lower CPU cost per transaction compared to ICP environment. For ICP users, we have selected the appropriate line from Table 50 on page 138.

```
ESCON  3172-3  66MHz ICP       TR  0.0026 CPU seconds/transaction
ESCON  3172-3  66MHz Offload   TR  0.0023 CPU seconds/transaction

The CPU utilization estimate for this example is:

7000 transactions per minute * 0.0026  = 18.2  ICP users
3000 transactions per minute * 0.0023  =  6.9  offload users
                                        -----
                                Total  = 25.1  CPU seconds

(Rounded to 25 for simplicity)
Total  = 25  CPU seconds/minute for TCP/IP + VTAM

 25/60 (to convert minutes to seconds) = 0.42 CPU seconds/elapsed second
```

The ES9021-982 (2CP LPAR) uses two central processors, but the Telnet server uses only one task; therefore, the number of processors is one.

```
25 CPU Seconds/minute   100%
-------------------- * --------- = 41.7% of a processor of ES9021-982
60 Seconds/minute      Processor
```

Note: If the percentage is greater than 100%, then you will need to consider multiple instances of TCP/IP and distribute the 3172-3s between them. One processor or 60 CPU seconds per minute are available for each TCP/IP address space. You will need multiple instances within the multiprocessor environment to handle a CPU load greater than 60 CPU seconds per minute.

Step 3. Computing the Storage Capacity needed

This is a two-step process. In our test, we used a TCP/IP profile with poolsize parameters to support 16000 Telnet users. The TCP/IP profile for 16000 users is shown in Chapter 7, “Capacity Planning for Telnet” on page 135.

In the first step, we determine the change in the poolsize parameters to support the 10000 Telnet sessions in this example and the subsequent change in storage.

In the second step, we add the additional storage required as the 10000 sessions are established.

Step 3A. Computing the Storage change due to Profile change

In Table 51 on page 140 the column for 0 sessions shows the total storage of 248,836,000 bytes is required with the poolsize for 16000 Telnet users.

Determine the poolsize values to support the 10000 sessions. Table 72 shows the derivation of the poolsize parameters for 10000 sessions.

Table 72. Poolsizes to support 10000 users.

Type	For ICP	For Offload	+ More	Permit %	Permit Size	Total for 10000
ACBs*	7000*1	3000*2	500	10%	1500	15000
Small data buffer	7000*2	3000*1	250	10%	1920	19200
TCBs	7000*1	3000*1	0	10%	1100	11100
Tiny data buffer	0	3000*1	250	10%	370	3700
Databuffers	7000*12%	3000*12%	0	10%	300	1300
Envelopes	7000*15%	3000*15%	0	10%	200	1700

Clarification on Calculation

For example, here is the way we have calculated the ACB amount:

```
7000 * 1 = 7000 ACBs for ICP users
3000 * 2 = 6000 ACBs for Offload users
           500 + more
          -----
         13500 Total number of ACBs for ICP and Offload users

13500/(1-10%) = 13500/.90 = 15000 Total ACBs for ICP and Offload
```

You will also need to calculate the data buffers and envelopes for 10000 telnet users.

- For data buffers:
  - 10000 Telnet users
  - (10000*12)/0.80 =1500 data buffers required for 10000 Telnet users.
- For envelopes (since MTU<2048)
  - 10000 Telnet users
  - (10000*15%)/0.90 = 1700 envelopes required for 10000 Telnet users.

Now, determine the poolsize difference between 16000 and 10000 Telnet users.

Table 73. Poolsize differences between 10000 and 16000 users.

Type	Total for 10000 users (a)	Total for 16000 users (b)	Poolsize difference (a-b)
ACBs*	15000	16200	-1200
Small data buffer	19200	32400	-13200
TCBs*	11100	16500	-5400
Tiny data buffer	3700	5000	-1300
Data buffers	1300	2500	-1200
Envelopes	1700	8500	-6800

Now, determine the storage difference between 10000 and 16000 users.

Figure 93. Storage Difference between 16000 and 10000 Telnet users.

```
 1200 ACBs              *  1508 bytes   =     1 809 600
13200 Smalldatabuffers  *  2048 bytes   =    27 033 600
 5400 TCBs              *  1056 bytes   =     5 702 400
 1300 Tinydatabuffers   *   256 bytes   =       332 800
 1200 Data buffers       *  8192 bytes   =     9 830 400
 6800 Envelopes         *  2048 bytes   =    13 926 400
                                          -------------
                                           58 635 200
```

Step 3B. Computing the Per User Storage for 10000 users

To derive the Total Storage requirements, we need to add the storage requirements per user (below and above) for Telnet logon and Steady state environments. For 10000 users, we need to use Table 51 on page 140 and used the storage requirement entry from column for 12000 users for the storage requirement per user. For example,

```
10000 Telnet users = 10000 * 3364 = 33 640 000
```

Note: ACB size includes actual ACB size of 100 bytes, plus 1408 bytes for Internal control blocks used for TCP/IP V3R2 for MVS.

Step 4. Total Estimated Storage for 10000 Telnet Users

```
Total Base Storage (Below and Above)             =  248 836 000
  from Table 51 on page 140
Difference between 16000 and 10000 Telnet users   =  58 635 200
  Total Base Storage for 10000 users(Below+above) = 190 200 800
```

Total Storage (below and Above) with 191 200 800 bytes represents the storage requirements for initializing TCP/IP with the 10000 Telnet users (this includes storage for all control blocks, bufferpools, and other internal control block buffers. This Total storage include buffers (default bufferpool and bufferpool specified using guideline for Telnet user and storage required for other control blocks).

To derive the Total Storage requirements, we need to add the storage requirements per user (below and above) for Telnet logon and Steady state environments.

From Table 51 on page 140 ,we need to take 3364 bytes as the storage requirement per user.

```
For 10000 Telnet users = 10 000 * 3364 = 33 640 000
```

Therefore, Total Storage requirements for 10000 users:

```
= (190 200 800 + 33 640 000) Bytes
= 223 840 800 Bytes
= 213.47 MBytes   (i.e 223 840 800 % (1024*1024) )

total storage required for 10 000 users = 213.47 MBytes
```

Examples of Estimating CPU and Storage Usage for FTP

This section gives an example of estimating CPU, Storage and disk usage for FTP on MVS. We used the formulas and benchmarks listed in Chapter 8, “Capacity Planning Guidelines for FTP” on page 145 to do our calculations. We have used TCP/IP V3R2 MVS FTP (server) performance data in Table xx.

Step 1. Review Your Knowledge about FTP and Your Hardware

The example environment included:

- The server is an IBM ES9121-440 processor. The 9121 is a two-way processor, where each processor has approximately 60% of the processing speed of each processor on ES9021-982.
  The IBM ES9021-982 was used for the benchmarks in our tables.
- There are multiple clients on workstations at least as powerful as the IBM RISC System/6000 model 530, which was the client in the benchmarks.
- The network connection is a 66 MHz 3172-3 running ICP, with an ESCON channel attachment to our MVS host. The LAN used is 100 Mbps FDDI Ring.

We wanted to know how much storage would be required based on the assumption that the maximum number of simultaneous file transfers would be less than 10.

We also wanted to find out what percentage of our available CPU capacity would be required to achieve an aggregate throughput rate of 2500 KBps for binary transfers of large (greater than 10MB) files.

Step 2. Computing the CPU Capacity Needed

To estimate the CPU capacity needed, we selected the binary PUT data for computing CPU capacity.

```
ESCON  3172-3 66MHz ICP FDDI 0.58+0.33 = 0.91 3590 1965  binary PUT
```

Note: The above information is from Table 66 on page 156.

Since the maximum throughput listed in the benchmark was 3590KBps, it would be possible to achieve our goal throughput rate of 2500 KBps if we were using a ES9021-982 2CP LPAR. However, since the maximum throughput was 1965 for a single-user PUT from a workstation, no single user will see 2500 KBps throughput on a PUT. Nevertheless, the goal may still be achievable with multiple simultaneous transfers.

Using the formula, we substituted 0.000058 for CPU seconds for the TCP/IP address space and 2500 KBps for throughput.

```
2500 KB per second  * 0.000058 CPU seconds per KB
          = 0.145 CPU seconds per elapsed second
```

We substituted 0.000033 for CPU seconds for the FTP server and 2500 KBps for throughput.

```
2500 KB per second  * 0.000033 CPU seconds per KB
          = 0.0825 CPU seconds per elapsed second
```

These calculations gave us the capacity estimate for the ES9021-982 2 CP LPAR. Then we took those numbers and compute the requirements for the 9121-440, based on the knowledge that the 2 processors of the 9121-440 are 60% of any one of the two processors in the ES9021-982 used in the benchmarks.

Note: Consult your IBM marketing representative or systems engineer to get the correct relationship in processing power between the different IBM processors.

Therefore:

```
1 CPU second on the 9121-440  = 0.60 CPU seconds on the ES9021-982
(it takes more time to do the same amount of work on the slower
processor).
```

We applied this ratio to the values we previously calculated:

```
0.145  ES9021-982 CPU seconds      1 9121-440 CPU second
-----------------------------  *  ---------------------------
      elapsed second                0.60 ES9021-982 CPU seconds


 = 0.2417 9121-440 CPU seconds per elapsed second
       for TCP/IP address space


0.0825   ES9021-982 CPU seconds     1 9121-440 CPU second
-----------------------------  *  ---------------------------
      elapsed second                0.60 ES9021-982 CPU seconds


 = 0.1375 9121-440 CPU seconds per elapsed second
       for FTP address space
```

Based on these calculations, 24.17% utilization of one processor is needed for the TCP/IP address space and 13.75% utilization of on processor for the FTP address space. As a result, approximately 37.9% of one processor or 18.9% of the total processing complex (2 processors) must be available during file transfers in order to achieve 2500 KBps.

Note: All we have established is the CPU requirement for achieving our target throughput rate. If there are other bottlenecks in our system, we may not achieve the target.

Step 3. Computing the Storage Capacity and Disk I/O

For more information on computing the storage capacity and disk I/O, see Chapter 7, “Capacity Planning for Telnet” on page 135.

Chapter 10. Building Your TCP/IP Profile for MVS

In this chapter, we show you how and why to define:

- Buffer pools
- Multiple FTP servers
- Multiple instances of TCP/IP

Defining Buffer Pools

This section is meant to help you to estimate how many buffers and control blocks you should specify on the pool size statements. The rules are meant as guidelines, not absolutes, and are not all-inclusive.

To estimate the optimum amount to specify on the pool size statements in your profile data set or file, you should monitor your buffer usage over time with the NETSTAT POOLSIZE command. If the low-water mark never approaches the permit size, it is probably safe to decrease the number of buffers you have allocated in that pool. If the low-water mark does approach the permit size, you should increase the number of buffers in the pool.

ACBs

You need many ACBs to run TCP/IP. You should monitor the ACB pool on a regular basis using the NETSTAT POOLSIZE command.

It is so detrimental to run out of ACBs that the TCP/IP for MVS will go out and get more than you have specified if it runs low. Although it is possible to turn off the dynamic allocation of ACBs using the NOACBCUSHION parameter on the ASSORTEDPARMS statement in your configuration file, we do not recommend turning it off.

Include the following in your estimate for ACBs:

- Add at least 1 ACB for each Telnet user your server will be supporting.
- Add a second ACB for each Telnet user who will connect to your server through a 3172 with Offload.
- Add up to 8 ACBs for each concurrently active FTP client that your FTP servers will be supporting. See Chapter 8, “Capacity Planning Guidelines for FTP” on page 145 for specific requirements for each network connection type.

Total the items that pertain to your installation and add at least 15% to your estimate. Then divide this new estimate by 0.90 to account for the permit size percentage.

CCBs

Include the following in your estimate for CCBs:

- Add 1 CCB for each user on the OBEY statement in your configuration file.
- Add 1 CCB for each user on the INFORM statement in your configuration file.
- Add 1 CCB for each INTCLIEN (Telnet) port.

- Add 1 CCB for each pair of FTP server ports.
- Add 1 CCB for each instance of TCP/IP.
- Add 1 CCB for each job executing TCP/IP commands such as NETSTAT.
- Add 1 CCB for each CICS socket connection.

Total the items that pertain to your installation and then divide your estimate by 0.93 to account for the permit size percentage.

Data Buffers: Regular, Small, and Tiny

Some data buffers are required no matter what application you are using.

- If they are available, Telnet will use small data buffers instead of regular data buffers. However, Telnet processing still requires some regular data buffers.
- Offload requires small and tiny data buffers.
- Open Edition (OE) socket applications require tiny data buffers.

Data Buffers

Include the following in your estimate for regular data buffers:

- Do not use the default size of 8192 unless you have a severe storage constraint or you want to slow down FTP. You should increase the size to 16 384 or more.

  Try decreasing the number of data buffers if you want to save storage.
- Add up to 9 data buffers for each concurrently active FTP client your FTP servers will be supporting. See Chapter 8, “Capacity Planning Guidelines for FTP” on page 145 for specific requirements for each network connection type.
- Add data buffers for the number of active Telnet users who connect to your server through a 3172 with ICP multiplied by 0.12.
- Add data buffers for the number of active Telnet users who connect to your server through a 3172 with Offload multiplied by 0.02.

Total the items that pertain to your installation and then divide your estimate by 0.90 to account for the permit size percentage.

Small Data Buffers

Include the following in your estimate for small data buffers:

- Add 1 small data buffer for each active Telnet user.
- Add 1 small data buffer for each active Telnet user who will connect to your server using 3172 with ICP.
- Add 1 small data buffer for each concurrently active FTP client who will connect to your server through a 3172 with Offload.
- Use small data buffers for each UDP socket program connected to your server through a 3172 with Offload. (The amount you need depends on the application and network delays. Sending more data and sending data faster requires more small data buffers.)

Total the items that pertain to your installation and add at least 15% to your estimate. Then divide the new estimate by 0.90 to account for the permit size percentage.

Tiny Data Buffers

Include the following in your estimate for tiny data buffers:

- Add 1 tiny data buffer for each active Telnet user who will connect to your server using 3172 with Offload.
- Add 1 tiny data buffer for each concurrently active FTP client who will connect to your server through a 3172 with Offload.

Total the items that pertain to your installation and add at least 15% to your estimate if you are using 3172 with Offload. Then divide the new estimate by 0.90 to account for the permit size percentage.

Envelopes: Regular and Large

TCP/IP applications use regular envelopes when the MTU size for the network connection is less than or equal to 2048. Large envelopes are used by TCP/IP applications when the MTU size for the network connection is greater than 2048.

Examples of network connections that allow the MTU size to be greater then 2048 are:

- CTC
- FDDI
- HPPI
- NSC Hyperchannel

A token ring also can allow the MTU size to be greater than 2048, but is not commonly configured this way.

Offload does not use envelopes. However, TCP/IP will still use some envelopes even if Offload is the only network connection configured.

Regular Envelopes

Include the following in your estimate for regular envelopes:

- Add up to 25 envelopes for each concurrently active FTP client your FTP servers will be supporting, except for clients connected through the 3172 with Offload. See Chapter 8, “Capacity Planning Guidelines for FTP” on page 145 for specific requirements for each network connection type.

  The number of envelopes FTP requires depends on the MTU size. If the MTU size is close to 2048, fewer envelopes will be required than if the MTU size is 576 or 1500 bytes.
- Add envelopes for the number of active Telnet users who connect to your server through a 3172 with ICP multiplied by 0.15.
- Use envelopes for each UDP socket program connected to your server, except when they connect through a 3172 with Offload. (The amount you need will depend on the application and network delays. Sending more data and sending data faster requires more envelopes.)

Total the items that pertain to your installation and then divide your estimate by 0.90 to account for the permit size percentage.

Large Envelopes

Include the following in your estimate for large envelopes:

- Increase the size from the default of 8192 if you plan to set the MTU size to be larger than 8192 bytes for any of your network connections.
- Add up to 8 envelopes for each concurrently active FTP client your FTP servers will be supporting, except for clients connected through a 3172 with Offload. See Chapter 8, “Capacity Planning Guidelines for FTP” on page 145 for specific requirements for each network connection type.

Total the items that pertain to your installation and then divide your estimate by 0.90 to account for the permit size percentage.

RCBs

RCBs are required by programs using “raw” sockets. To issue a raw socket call, the user must be in the OBEY list in the configuration file. The TRACERTE command on MVS is an example of a program that uses raw sockets.

We recommend that you use the default amount of 50 unless monitoring by using the NETSTAT POOLSIZE command shows the low-water value is approaching the permit size.

SCBs

Include the following in your estimate for SCBs:

- Add 2 SCBs for each concurrently active FTP client your FTP servers will be supporting.
- Add 4 SCBs for each FTP server.

We recommend that you use the default amount of 256 and monitor using the NETSTAT POOLSIZE command to make sure the low-water value does not approach the permit size.

SKCBs

Include the following in your estimate for SKCBs:

- Add 2 SKCBs for each concurrently active FTP client connected to a C FTP server

We recommend that you use the default amount of 256 and monitor using the NETSTAT POOLSIZE command to make sure the low-water value does not approach the permit size.

TCBs

Include the following in your estimate for TCBs:

- Add 4 TCBs for each concurrently active FTP client.
- Add 3 more TCBs for each FTP client connected via a RISC parallel channel adapter.
- Add 1 TCB for each Telnet user your server will be supporting.

Total the items that pertain to your installation and then divide your estimate by 0.90 to account for the permit size percentage.

UCBs

Include the following in your estimate for UCBs:

- Add 1 UCB for each open UDP port.

We recommend that you use the default amount of 100 and monitor using the NETSTAT POOLSIZE command to make sure the low-water value does not approach the permit size.

Example of Analyzing Buffers Used by FTP on MVS

This example of how to analyze the buffers you are using based on tests similar to those used in “Performance Example” on page 109.

We used the NETSTAT POOLSIZE command to show the data buffers and control blocks used based on the default amounts and sizes, as shown in Figure 94.

```
netstat pool size
MVS TCP/IP Netstat V2R2.1
TCPIP Free pool status:
Object       # alloc     # free      Lo-water    Permit size
======       =======     ======      ========    ===========
ACB            1000         993         988         100
CCB             150         133         133          10
Dat buf         160         154         146          32
Sm dat buf        0           0           0           0
Tiny dat buf      0           0           0           0
Env             750         750         729          75
Lrg env          50          49          41          10
RCB              50          50          50           3
SCB             256         245         245          17
SKCB            256         256         256          17
TCB             256         248         247          17
UCB             100         100         100           6
```

Figure 94. NETSTAT POOLSIZE Command in Monitor Buffer Usage

We discovered the maximum buffers used with the following formula:

```
Allocated - Low-Water = Maximum Used
```

We applied this formula to each of the buffers and control blocks, as shown in Table 74.

Table 74. Example of Analysis of Buffers and Control Blocks Necessary

Buffer Type	Allocated	Low Water	Maximum Used	Maximum Before Transfer
ACBs	1000	988	12	4
CCBs	150	133	17	17
Data buffers	160	146	14	6
Small data buffers	0	0	0	0
Tiny data buffers	0	0	0	0
Envelopes	750	729	21	6
Large envelopes	50	41	9	1
RCBs	50	50	0	0
SCBs	256	245	11	9
SKCBs	256	256	0	0
TCBs	256	247	9	5
UCBs	100	100	0	0

We made the following observations:

- No small or tiny data buffers were allocated, so it makes sense that the maximum used would be 0.
- Since FTP was the only application running on MVS:
  - No programs use RawIPOpen, socket(), accept(), or takesocket() calls, so the maximum for RCBs and SKCBs is also 0.
  - No UDP ports were ever open (FTP is a TCP application), so the maximum for UCBs is 0.
- CCBs were required for the following:
  - One for each user on the OBEY statement in the PROFILE.TCPIP data set. We had 12 users on this statement.
  - One for each user on the INFORM statement in the PROFILE.TCPIP data set. We had 1 user on this statement.
  - One for each server. We had 4 FTP servers defined.

  So 12 + 1 + 4 = 17, which is the maximum number of CCBs that were used during our testing.

By comparing to the results of the NETSTAT POOLSIZE command before any FTP data transfers were initiated, we can see that the FTP data transfer required additional ACBs, data buffers, envelopes, SCBs, and TCBs. Therefore the additional buffers needed for this case are:

```
8 ACBs           8 *   96Bytes =     768
8 Data Buffers   8 *  32KBytes = 262 144
15 Envelopes    15 *   2KBytes =  30 720
2 SCBs           2 *  228Bytes =     456
4 TCBs           4 * 1056Bytes =   4 224
                                 --------
                                 298 312 bytes
```

Additional large envelopes were not required until both systems had set the packet size to 4352. The reason for this is envelopes are used to process packets 2048 bytes or smaller, and large envelopes are used to process packets greater than 2048 bytes. When the packet size was greater than 2048 bytes, 8 additional large envelopes were required (8 * 8192 = 65 536 bytes).

Defining Multiple FTP Servers

The internal structure of the C FTP server address space changed between TCP/IP V3R1 for MVS and TCP/IP V3R2 for MVS. In TCP/IP V3R2 for MVS each client that connects to the server is handled for the full duration of the FTP session by a separate subtask within the C FTP server address space. This enables a higher degree of parallel processing on multi-processor CPUs, which results in improved throughput in especially highly loaded C FTP server environments.

When you configure the C FTP server by updating the server FTP.DATA configuration data set, you specify a parameter that is called EXTRATASKS. In TCP/IP V3R2 for MVS, EXTRATASKS may have a value in the range from 0 to 254. The parameter specifies how many tasks should be deployed to handle concurrent FTP sessions. If a higher number of clients connect than what you specified in the EXTRATASKS parameter, these FTP sessions will still be established and serviced, but the sessions that are established after the limit has been reached will all be serviced under one and the same main task within the FTP server - with some performance impact to these sessions.

If you specify an EXTRATASKS parameter of 200, the FTP server may not start 200 subtasks. During initialization the FTP server does some intelligent analysis of the available virtual storage below the 16MB line, and it then calculates how many subtasks can be started safely without risking abends due to lack of virtual storage. The FTP server will print out a message on its SYSPRINT file telling you how many subtasks it decided to actually start, which very well may be lower than the 200 you specified. The actual number will depend on various factors, such as:

- How large is your private region below the 16MB line?
- Which C run-time library are you using?

If you know that you have to be able to support for example, 200 concurrent FTP sessions, you may start two FTP servers, each with an EXTRATASKS parameter of 100.

TCP/IP for MVS allows you to reserve the same port number(s) to more than one address space name in the PORT section of PROFILE.TCPIP. If you want to start two FTP servers, for example, T18NFTP1 and T18NFTP2, you can code the following PORT reservations:

```
PORT
    .. ... ........
    20 TCP T18NFTP1  NOAUTOLOG ; FTP Server 1
    21 TCP T18NFTP1            ; FTP Server 1
    20 TCP T18NFTP2  NOAUTOLOG ; FTP Server 2
    21 TCP T18NFTP2            ; FTP Server 2
    .. ... ........
AUTOLOG
    ........
    T18NFTP1    ; FTP Server 1
    T18NFTP2    ; FTP Server 2
    ........
```

Based on the above definitions, the TCP/IP system address space will start two FTP server address spaces during initialization. FTP clients connect to the FTP server control session port number 21. Because port number 21 is reserved for both servers, the TCP layer in the TCP/IP system address space wil round-robin connection requests to port number 21 between the two server address spaces. The first connection will go to T18NFTP1, the second to T18NFTP2, the third to T18NFTP1, and so on. No attempt is done to do load-balancing in terms of analyzing which of the servers currently has the lowest number of connections; it is purely a connection-based round-robin that is performed.

Defining Multiple Instances of TCP/IP

When the total CPU utilization for the TCP/IP address spaces is 100%, you might want to consider adding another copy of TCP/IP. The load can be balanced between the 2 instances of TCP/IP by assigned them each their own 3172s.

Note: If you set up multiple copies of TCP/IP on a single host, they should be from the same release of TCP/IP.

How to Configure for Multiple Copies of TCP/IP on MVS

To run another TCP/IP program on the same MVS system, you need to have more than one copy of certain key data sets and procedures. The following list describes the steps you might take to run a second TCP/IP:

1. Create another started task (TCPIPROC) for the TCPIP address space.

   Add or modify the //PROFILE statement to point to the second PROFILE.TCPIP data set.

2. Create another PROFILE.TCPIP data set.

   - If you want to use a different high level qualifier for dynamically allocated data sets, modify the DATASETPREFIX parameter.
   - Modify the parameters as if this data set were on another MVS system. For example, you might:
     - Give the HOME statement different IPADDRS values
     - Specify different DEVICE and LINK statements
     - Specify different LUs

3. Create another TCPIP.DATA data set.
   - If you want to use a different high level qualifier for dynamically allocated data sets, modify the DATASETPREFIX parameter.
   - Modify HOSTNAME to identify the second TCPIP address space.
   - Modify TCPIPJOBNAME to point to a second TCPIP started procedure
4. If you are running FTP, create another *hlq*.FTP.DATA data set.
5. Create another TSO logon procedure.
   - Add or modify the //SYSTCPD statement to point to the second TCPIP.DATA data set.
   - If you are running FTP, add or modify the //SYSFTPD statement to point to the second *hlq*.FTP.DATA data set.
6. Create another set of started tasks for each of the servers in the second instance of TCP/IP.
   - Add or modify the //SYSTCPD statements in these procedures to point to the second TCPIP.DATA data set.
   - If you are running FTP, add or modify the //SYSFTPD statements in these procedures to point to the second *hlq*.FTP.DATA data set.

   Refer to the individual chapters in the *TCP/IP for MVS: Customization and Administration Guide* book for more information about configuring the servers.

Part 3. Appendix

Appendix A. Performance Tuning Tips for MVS

You can improve TCP/IP performance significantly by proper tuning. This appendix provides information on tuning techniques that have proven to be successful with TCP/IP for MVS, V2R2.1, V3R1, V3R2, and OS/390. The appendix is divided into two sections:

- ***General performance tuning recommendations.*** This section provides guidance on the use of some of the most important tuning parameters, and suggests values that have been found to optimize performance in production systems.
- ***Performance enhancements***. TCP/IP V3R2 is the most current version of TCP/IP for MVS. TCP/IP V3R2 provides significant reduction in CPU cycles for applications such as FTP, Telnet, Sockets, CICS/IMS and ADSM. This appendix provides a summary of the tuning tips.

General Performance Tuning Recommendations

TCP/IP performance is influenced by a number of parameters that can be tailored for the specific operating environment. In general, these tuning parameters can be grouped into the following categories:

- MVS tuning
- TCP/IP tuning
- Communication tuning
  - Mainframe end
  - Workstation end
- TCP/IP application tuning

Every TCP/IP environment is different. Optimum performance can only be achieved when the system is tuned to match its specific environment. The purpose of this section is to highlight the most important tuning parameters and to recommend parameter values that have been found to maximize performance in existing customer installations.

MVS Tuning

Region size and dispatching priority are important to MVS tuning.

Region size

Because of the potentially large number of users (each of which requires buffers and control blocks), TCP/IP and its applications require relatively large amounts of virtual storage. The default region size of 7.5M is only sufficient for users with relatively small networks. As the number of users increases, the amount of storage used for buffers and control blocks increases as well.

- TCP/IP region size:
  - If region size is not explicitly set in the MVS TCP/IP start-up procedure, the default of 7.5M will be used. Because TCP/IP performance is related to the amount of available virtual storage, it is important that you set the region

size explicitly. If possible, set REGION=0K or 0M; this will provide maximum available memory for TCP/IP.

  - TCP/IP V3R2 and V3R1 with dynamic

- FTP server region size:
  - If possible, set REGION=0K or 0M in the MVS TCP/IP start-up procedure.
  - If virtual storage is limited, make the region size as large as possible — in excess of 7.5M.

If no REGION parameter is specified, the system uses an installation default specified at JES initialization.

If your installation does not change the IBM-supplied default limits in the IEALIMIT or IEFUSI exit routine modules, then specifying various values for the region size have the following results:

- A value equal to 0K or 0M gives the job step all the storage available below and above 16 megabytes.
- A value greater than 0K or 0M and less than or equal to 16384K or 16M establishes the size of the private area below 16 megabytes. If the region size specified is not available below 16 megabytes, the job step abnormally terminates. The extended region size is the default value of 32 megabytes.
- A value greater than 16384K or 16M and less than or equal to 32768K or 32M gives the job step all the storage available below 16 megabytes.The extended region size is the default value of 32 megabytes.
- A value greater than 32768K or 32M and less than or equal to 2096128K or 2047M gives the job step all the storage available below 16 megabytes.The extended region size is the specified value. If the region size specified is not available above 16 megabytes, the job step abnormally terminates.

MVS dispatching priority

Set the dispatching priority for the TCP/IP task as high as possible. Ideally, the dispatching priority for VTAM, TCP/IP and VMCF should be the same — but slightly higher than — that of TCP/IP applications like FTP or ADSM.

The use of a higher dispatching priority for TCP/IP has, in many cases, actually ***improved*** throughput and ***reduced*** CPU utilization. Remember that TCP/IP and VMCF provides services for TCP/IP address spaces; consequently, its performance (or lack thereof) impacts the performance of the TCP/IP applications.

TCP/IP Tuning

You can tune the MVS TCP/IP address space by changing parameters in the *hlq*.PROFILE.TCPIP configuration statements.

Packet size

It is important to remember that when a connection is established between two TCP/IP hosts, the server and client exchange information that specifies the maximum packet size each can receive.

When MVS TCP/IP receives an MSS option from another system, it sets the maximum segment size (MSS) for the connection to the minimum of:

the value specified by the other system.

a value which is 40 bytes (header information) less than the MTU of the route to the other system.

For example, if one system has an MTU=2048 and the other has MTU=1500, the connection will use MSS=1460 byte segments (1500 minus 40 equals 1460). Since small packet sizes generally result in poorer performance, it is important to ensure that the largest packet size possible be specified.

Performance problems can arise when packet sizes are allowed to assume default values. Consider the following example of a GATEWAY statement with several different network interfaces.

```
GATEWAY
*net_number first_hop link_name packet_size subnet_mask  subnet_value
 193.9.200          =        TR1    2000         0
 193.0.2            =        ENET2  1500         0
 128.84        193.9.200.2   FIDI   4352         0.0.255.0   0.0.1.0
 128.84        193.9.200.100 LINK1  DEFAULTSIZE  0.0.255.0   0.0.55.0
 DEFAULTNET    193.0.2.3     ENET2  1500         0
```

In the preceding example, the following is true:

- Where LINK1 specifies 'DEFAULTSIZE', the MTU will be 576 bytes.
- Where ENET2 is the DEFAULTNET, the MTU will be 1500 bytes.

Although you may prefer to use the default value for routes that traverse the public Internet, you probably can improve throughput significantly for routes within your enterprise's network by using the maximum packet size. *Do not accept the default value unless you want the packet size to be 576 bytes.*

The packet size should be the largest value that the route to the workstation can handle without fragmentation. The corresponding parameter on the workstation should be set to the largest value it can receive. The actual packet size used in the communication will be the smaller of the packet sizes exchanged by the client and server when the connection is established. ***To avoid confusion, you should always state the packet size explicitly in the MVS TCP/IP GATEWAY max_packet_size parameter.***

When RouteD is Used

The BSDROUTINGPARMS statement defines the characteristics of each link defined at this host over which RouteD sends routing information. The TRUE/FALSE parameter of the BSDROUTINGPARMS determines whether or not the default packet size of 576 is used.

- When sending to networks that are not locally attached, if TRUE is specified, the MTU parameter can be used to set MTU size.
- When FALSE is specified, the mtu parameter is ignored; the MTU size defaults to 576.

See *TCP/IP for MVS: Customization and Administration Guide* for more information.

Data Buffer Pool Size

The values specified in the DATABUFFERPOOLSIZE statement in (*hlq*.PROFILE.TCPIP) play a significant role in TCP/IP performance, especially FTP performance. The default size for V3R1 and V3R2 for MVS is 16 384 bytes.[1] If you have a large number of Telnet users (thousands) and if storage is a concern, then you may want to use a smaller (8 192) size — but the larger the better.

Buffer allocation guidelines

The MVS TCP/IP address space is divided into a base amount of storage for program logic and a variable amount of storage for buffers and control blocks. MVS TCP/IP allocates virtual storage for control blocks and buffers at start-up time. Note that additional buffers and control blocks may be required by certain TCP/IP applications, such as FTP and Telnet.

You should use the NETSTAT POOLSIZE command to monitor the low water marks for buffer pools. Then, using the guidelines below, you can specify the number of buffers and control blocks needed. TCP/IP V3R1 and V3R2 for MVS expands control blocks and buffer pools dynamically as demand increases.[2]

The following table contains guidelines for each type of control block and data buffer required when the TCP/IP address space is started.

Buffer allocations for MVS TCP/IP start-up

[1] For TCP/IP V3R1 with PTF UN76873, the default is changed to 16 384 from 8192.

[2] If you have the Dynamic Control Block and Buffer Pool PTF (UN76873), needs to be installed for the dynamic expansion of the control blocks and buffer pools.

Table 75. Buffer Allocation for TCP/IP start-up (MVS V2R2.1, V3R1 & V3R2).

Buffer Pool	Number of Active Buffers before Clients Connect
ACB	2 for each instance of TCPIP 1 for each START of 3172 ICP 7 for each START of 3172 offload 2 for each START of RISC System/6000 parallel/ESCON channel 1 for each INTCLIEN (Telnet) port 2 for AUTOLOG if offload 2 if default LU's are in BEGINVTAM More for miscellaneous events
CCB	1 for each instance of TCPIP 1 for each pair of FTP server ports 1 for each INTCLIEN (Telnet) port 1 for each user on OBEY statement 1 for each user on INFORM statement 1 for each job executing TCPIP commands like NETSTAT
Data	2 for each instance of TCPIP 1 for each FTP server to AUTOLOG
Small	1 for each INTCLIEN (Telnet) port
Tiny	10 for each START of 3172 offload
Envelope	4 for each START of 3172 ICP 2 for each START of 3172 offload 2 for each START of RISC System/6000 parallel channel
Large Envelope	1 for each instance of TCPIP
SCB	2 for each pair of FTP server ports 1 for each INTCLIEN (Telnet) port 2 for each INTCLIEN port if offload 2 for AUTOLOG if offload
TCB	1 for each INTCLIEN (Telnet) port 1 for each FTP server to AUTOLOG

- Small and tiny data buffer pools are required for Offload devices.
- Tiny data buffer pools are also used in the OE environment.
- For non-offload devices, if no small data buffers are defined, regular data buffers will be used.

Communication Tuning (Mainframe end)

Mainframe performance is influenced by a number of parameters. This section describes block size parameters for the IBM 3172, and tips for using slow lines such as X.25.

IBM 3172 Block size parameters (ICP)

The primary measurements of network performance are throughput and response time. Increasing the block size increases throughput and decreases host CPU utilization. However, it can also worsen response time because short data blocks are packed into frames that are not always transferred to the host immediately. This increase in frame delay degrades the performance of delay-sensitive traffic for interactive applications like Telnet. On the other hand, throughput-sensitive applications

such as FTP benefit from the improved throughput that results when frames are blocked.

The IBM 3172 combines LAN frames received from the workstation into blocks of up to 20 KB before they are sent to the host. The Inter Connect Program (ICP) implements two programmable parameters that determine when a block of data will be sent to the host:

1. ***Block Delay Timer*** When this timer times out, the 3172 sends the current block, even if the block is not full.
2. ***Response Length*** When frames smaller than this value are received from the workstation, the 3172 sends the block to the host immediately.

The following parameters have been found to work well with systems that support both interactive and bulk data traffic at the same time.

- Set the DELAY TIMER value to 10 ms (default value is 20). This will provide reasonable response times for the interactive user without impacting FTP throughput.
- Set the RESPONSE LENGTH to 500 bytes (default is 100). This parameter causes data streams of less than 500 bytes to be sent without delay, improving response time for both FTP acks and relatively short Telnet input data streams.

You can use the 3172 Operator Control Facility to change these parameters.

X.25 and other relatively slow lines

Specify DELAYACK in the PORT or GATEWAY statements to minimize non-data transmissions. (See *TCP/IP for MVS: Customization and Administration Guide*, SC31-7134 for more information.)

RISC System/6000 Scalable Processor (SP2) tuning

- **Packet size**

  The current implementation of CLAW supports a maximum packet size of 4096 bytes. Set MTU size on the SP2 to 4096.

- **Recommended SP2 tuning parameters**

  sb_max=1310720

  thewall=6000

  lowclust=200

  mb_cl_hiwat=1200

  tcp_sendspace=32768

  tcp_recvspace=32768

  udp_sendspace=32768

  udp_recvspace=32768

  ADSM PTF IP20421 defines additional parameters:

  TCPWindowsize=640

  txnbyte=25600

  tcpnodelay=Y

tcp_mssdflt=4096 (default is 512)

recvnum=64

xmitnum=64

- **Consider multiple channels to TCP/IP** For systems with extremely high-volume, high-performance characteristics, consider running multiple transmissions in parallel through multiple SP/2 nodes via multiple ESCON channels to TCP/IP.

Communication Tuning (Workstation end)

Workstation performance is influenced by a number of parameters. This section describes some of the more important parameters and uses the RISC System/6000[3] AIX system as an example of tuning techniques that will improve performance for a variety of non-IBM workstations.

Parameters should agree!

Assume that an AIX workstation specifies its Ethernet packet size to be 1500 bytes. The corresponding mainframe *max_packet_size* parameter should also specify a packet size of 1500 bytes. If this is not done, the mainframe packet size will default to 576 bytes. When communications are established, the frame size selected will be the smaller of the two values (576 or 1500). In this case, the value specified in the workstation (1500) would be ignored, and the maximum packet size used would be 576, potentially impacting performance.

Packet size

A properly chosen packet size and window size can significantly reduce mainframe load and help improve throughput. On a RISC System/6000, you can use the

```
no -a
```

command to view existing settings, and the

```
netstat -in
```

command to display the current packet size.

On a RISC System/6000, you must be a root user to change existing settings.

- For **Ethernet-connected systems**, the recommended packet size is 1500 bytes. To change it, issue the following commands:

  ```
  ifconfig ether down
  ifconfig ether mtu 1500
  ifconfig ether up
  ```

- For **Token Ring-connected systems**, the recommended packet size is 2000 bytes. To change it, use the following commands:

  ```
  ifconfig tr1 down
  ifconfig tr1 mtu 2000
  ifconfig tr1 up
  ```

[3] If you are using AIX 3.2.5 and RISC System/6000 workstations as your TCP/IP client or server machine, consider installing PTF U435114 to improve FTP throughput for ASCII transfer and reduce host cycles.

- For **FDDI systems**, the recommended packet size is either 4000 or 4352 bytes. To change it, issue the following commands:

```
ifconfig fddi1 down
ifconfig fddi1 mtu 4000
ifconfig fddi1 up
```

For all network types, the workstation-specified MTU size should agree with the packet size specified in the MVS *hlq*.PROFILE.TCPIP GATEWAY statement.

Window size

Use the following commands to change the RISC System/6000 window size to the recommended value of 32 768 or 65 536:

```
no -o tcp_sendspace=32768 -o tcp_recvspace=32768
no -o udp_sendspace=32768 -o udp_recvspace=32768
```

Other important workstation parameters

Use smit to set the following:

sb_max=1310720

thewall=6000

lowclust=200

mb_cl_hiwat=1200

tcp_sendspace=32768

tcp_recvspace=32768

udp_sendspace=32768

udp_recvspace=32768

ASDM PTF IP20421 defines additional parameters:

TCPWindowsize=640

txnbyte=25600

tcpnodelay=Y

tcp_mssdflt=4096 (default is 512)

recvnum=64

xmitnum=64

Application Tuning

FTP Tuning

Only the C-FTP server is supported with TCP/IP V3R2. The Pascal FTP server is no longer supported with TCP/IP V3R2.

A key factor in FTP performance is the efficiency with which disk I/O is performed. The MVS FTP.DATA data set defines the number of disk I/O buffers, as well as record lengths and data set block sizes — each of which impacts data transfer performance.

Block size:

- For IBM 3380, the recommended block size is 23 440
- For IBM 3390 or IBM 9334, the recommended block size is one half track or 29K

Logical record length:

The recommended logical record length is 80 bytes.

Number of disk I/O buffers (Pascal FTP)

The number of disk I/O buffers is defined by the *NCP* parameter.[4] A data set block size equal to one half of the DASD track size provides the best disk I/O performance; if you follow this recommendation, 3 disk I/O buffers (default value) is sufficient. The NCP buffer pool occupies virtual storage below the 16MB line; thus, you may wish to use a smaller data set block size with a larger number of buffers.

The following table illustrates the relationship between virtual storage requirements, data set block size, and number of buffers. For example, with an NCP value of 3, each user requires 70 320 bytes of virtual storage. In general, use a data set block size as large as possible. The smaller the block size, the more buffers you should use.

Table 76. FTP Server NCP Virtual Storage requirements

NCP	Data Set Block Size	Virtual Storage (per user)
3	23440	70320
4	23440	93760
5	23440	117200
7	9980	69860
10	3120	31200

Number of buffers (C-FTP server)

The BUFNO parameter in FTP.DATA specifies the number of buffers used by the C-FTP server task. The size of the buffer is equal to the record length of the dataset for storing data on MVS. The recommended value for BUFNO is 35, and the default value is 5; the allowed value is in the range of 1–255. BUFNO helps in improving FTP throughput when remote FTP clients send data to the C-FTP server on MVS.

Checkpoint Interval

The CHKPTINT variable allows you to specify the frequency with which the system takes checkpoints in a file transfer request from the sending site. When you request checkpoint services in FTP, the overhead is a function of the checkpoint interval: the shorter the interval, the more frequent the checkpoints and the greater the overhead. If possible, set CHKPTINT to 0 (default). With this setting, checkpoints do not occur and CPU load due to checkpointing is eliminated.

[4] Point of confusion: in this context, NCP stands for Number of Channel Programs -- not Network Control Program.

Buffer Allocation Guidelines for FTP Server

The following table contains guidelines for allocating space for buffers and control blocks for the FTP server. These requirements are in addition to those required at MVS TCP/IP start-up time. Note that the FTP Server resides in a different address space from that of TCP/IP.

Table 77. Buffer Allocation Guidelines for FTP (MVS TCP/IP V2R2.1&V3R1)

Buffers per active FTP client	3172-3 ICP	3172-3 Offload	RISC parallel channel
ACB	5 +	7	10
Databuffer	8	7	8
Small	0	1	0
Tiny	0	1	0
Envelope	7, 25 (see below) MTU < 2048	0	19 MTU < 2048
Large envelope	8 MTU > 2048	0	8 MTU > 2048
SCB	2	2	2
TCB	4	4	7
CCB RCB SKCB UCB	0	0	0

- Envelopes are used when packet size is 2048 bytes or less.
- Large envelopes are used when packet size is more than 2048 bytes.
- More envelopes are needed when packet size is 576 bytes (default) than when it is 2000.
- Small and Tiny data buffer pools are required for Offload devices.
- Tiny buffers are also used in the OE environment.
- For non-offload devices, if no small data buffers are defined, regular data buffers will be used.

Recommended FTP.DATA Parameters

Use the following set of recommended settings for the FTP.DATA data set, instead of the defaults.

```
;*********************************************************************
;       Name of File:                          FTP.DATA
;
;       This file, FTP.DATA is used to specify default file and disk
;       parameters used by the FTP client.
;
;       Syntax rules for the FTP.DATA configuration file:
;
;       (a) All characters to the right of, and including the ';' will
;             be treated as a comment.
;       (b) Blanks and <end of line> are used to delimit tokens.
;*********************************************************************
;
;
;
Primary         50      ; Primary allocation is 50 tracks
Secondary       20      ; Secondary allocation is 20 tracks
Directory       15      ; PDS allocated with 15 directory blocks
Lrecl           80      ; Logical record length of 80
BlockSize       23440   ; Block size of 23440 for 3380 dasd
AutoRecall      true    ; migrated HSM files recalled automatically
AutoMount       true    ; non-mounted volumes mounted automatically
DirectoryMode   false   ; use all qualifiers (Data setmode)
Volume          APCSPL  ; Volume serial number for allocation
SpaceType       TRACK   ; data sets allocated in tracks
RECFM           FB      ; Fixed blocked record format
CHKPTInt        0       ; Checkpoint interval is 0
;DcbDSN         mod.dcb ; Data set name used as model for allocation
;UnitName       SYSDA   ; Unit name used for qualification
Filetype        SEQ     ; File type sequential (default)
;RETPD          30      ; New data set expiration date of 30 days
MGMTCLASS       TCPMGT  ; SMS management class for new data sets
RDW             false   ; Do not retain rdw's as data
NCP             3       ; 3 I/O buffers for single session
;                         Used only by Pascal FTP server and Pascal FTP
;                         clients
BUFNO           35      ; Issued by the C-FTP server
EXTRATASKS      20      ; Used to define multiple tasks for C-FTP server
WRAPRECORD      true    ; Do not truncate long record; wrap instead
```

Note: FTP parameters can be defined using the SITE|QUOTE command.

For more information on the FTP commands and FTP data set, see *TCP/IP for MVS: Customization and Administration Guide*.

Buffer Allocation Guidelines for Telnet Server

The following guidelines are suggested to help you allocate virtual storage for the buffer pools and control blocks used by Telnet. These storage requirements are in addition to those required at MVS TCP/IP start-up time. Note that the Telnet Server and its control blocks and buffers reside in the same address space as TCP/IP.

Table 78. Buffer Allocation Guidelines for Telnet (MVS TCP/IP V2R2.1 & V3.1)

Per Telnet user	3172-3 ICP	3172-3 Offload
ACB	1+	2+
Small	2	1+
Tiny	0	1+
TCB	1	1
Data[5] Envelope[5] Large envelope[5] CCB RCB SCB SKCB UCB		

- Small and tiny data buffer pools are required for Offload devices.
- Tiny buffer pools are also used in the OE environment.
- A Small data buffer pool is recommended for Telnet users.
- For non-offload devices, if no small data buffers are defined, regular data buffers will be used.

ADSTAR Distributed Storage Management (ADSM) Tuning

The ADSM server is widely used for backup and recovery of bulk data.

Because ADSM users send large amounts of data over the TCP/IP connection, ADSM performs best when TCP/IP is tuned for bulk data transfer (as for FTP). See *ADSM Tuning Guide,*, for more information.

For a RISC System/6000 SP2, the following network options should be set at the client:

```
/etc/no -o sb_max=130720
/etc/no -o rfc1323=1
/etc/no -o thewall=6000
/etc/no -o lowclust=200
/etc/no -o mb_cl_hiwat=1200
/etc/no -o tcp_sendspace=32768
/etc/no -o tcp_recvspace=32768
/etc/no -o udp_sendspace=32768
/etc/no -o udp_recvspace=32768
```

The following ADSM option is set in the server options file:

```
TXNGroupmax   40
```

Set the following ADSM options in the stanza for the correct server in the /usr/lpp/bin/dsm.sys file on the ADSM client.

[5] Some data buffers, envelopes and large envelopes are required, but not on a per-user basis.

```
TCPBuffsize       32
TCPWindowsize    640
TXNBytelimit   25600
TCPNodelay         y
```

Performance Enhancements

The following PTFs are available on TCP/IP V3R1 for performance enhancements. These PTFs have already been integrated into TCP/IP V3R2 for MVS. The PTF set includes:

- PN64344/UN69518 Telnet Scan Interval Option
- PN67848/UN77980 Storage Management
- PN68920/UN77994 Telnet LU Lookup
- PN69060/UN75573 VTAM Interface
- PN69881/UN75804 ACB Release
- PN70195/UN76873 Dynamic TCP Control Blocks
- PN70330/UN77979 IRB Scheduling
- PN70420/UN77871 ESA Datamoves
- PN68607/UN75139 Control Block Allocate
- PN72737/UN79157 SCB Allocation
- PNG7770/UN90824 C-FTP Performance Improvements

Dynamic Control Block and Buffer Pool Memory Allocation

An MVS TCP/IP system with a large number of users requires a large number of control blocks and buffers. If sufficient minimums were not specified, the system could run out of resources and either slow down or shut down entirely.

With the enhancements, minimums must no longer be pre-defined because initial pool sizes are based on defaults, that are dynamically expanded — as needed — until storage limits are reached. Even when storage limits have been reached, the system attempts to satisfy requests for small blocks of storage from available larger blocks. In this way, TCP/IP V3R1 and V3R2 for MVS attempts to keep running until all resources are exhausted.

With the new function, the following buffer pools are automatically expanded as needed:

- ACBPOOLSIZE
- ADDRESSTRANSLATIONPOOLSIZE
- CCBPOOLSIZE
- DATABUFFERPOOLSIZE
- ENVELOPEPOOLSIZE
- IPROUTEPOOLSIZE
- LARGEENVELOPEPOOLSIZE
- RCBPOOLSIZE
- SCBPOOLSIZE
- SKCBPOOLSIZE
- SMALLDATABUFFERPOOLSIZE
- TCBPOOLSIZE

- TINYDATABUFFERPOOLSIZE
- UCBPOOLSIZE

(See *TCP/IP for MVS: Customization and Administration Guide* for more detail).

Smoothing Telnet Responsiveness

Inactive clients continue to use control blocks and buffers in server memory; consequently, they may prevent other clients from connecting (or re-connecting after a connection failure) to TCP/IP.

There are at least two reasons why a given client might appear to be inactive to the Telnet server:

- The user may have simply gone to lunch, leaving his workstation turned on and connected.
- The communication link may have failed and broken the connection.

The problem is that the Telnet server cannot easily determine whether the connection is still desired, or should be "cleaned up" and made available to another user.

| To address this problem, TCP/IP V3R1 and V3R2 provide a periodic scan to determine which (if any) client connections have become inactive and to clean up those which have. You can control this process through the use of two parameters in the *hlq*.PROFILE.TCPIP *INTERNALCLIENTPARMS* statement.

- *INACTIVE* allows you to specify how long a terminal can remain unused (no communication with the server) before it will be deemed inactive and disconnected by the server.
- *TIMEMARK* allows you to specify how often the server will send an "are you there" probe to clients that appear to be inactive. Clients who receive three consecutive probes without intervening activity are considered to be inactive.

The Telnet server includes a scan function that periodically examines every current TCP/IP connection to determine whether any action is required. At the time of the scan,

- If the client has been active since the last *TIMEMARK* period began, the connection is deemed to be active, and no further action is taken for that client.
- If the client has exceeded its *INACTIVE* period, the "clean-up" function is scheduled. This function closes control blocks, frees up buffers, and makes the connection available to another user.
- If the client has not been heard from for the *TIMEMARK* interval, an "Are you there?" probe is scheduled to be sent to the client to determine whether or not it is still there. If the client fails to respond, the connection is assumed to be broken, and clean-up is scheduled.
- If the client has been sent three consecutive probes (indicating no activity for three times the *TIMEMARK* interval) the client is determined to be inactive, and the "clean-up" function is scheduled.

Once all of the connections have been scanned, scheduled *TIMEMARK* probes and "clean-ups" are performed. Note that the TIMEMARK probes and clean-up activity are not actually performed until scan processing is complete.

Prior to the availability of this function, the scan occurred once every two minutes — an interval that was hard-coded into TCP/IP. Although this technique made the cleanup function efficient, it caused a flurry of *TIMEMARK* and "clean-up" processing activity every two minutes. In large networks the processing that had been scheduled during the scan could impact the response time of the remaining active clients.

To smooth out this periodic after-scan processing, a new parameter was introduced — *SCANINTERVAL*. Using this parameter, you can change the hard-coded two-minute interval to a value that is consistent with your *INACTIVE* and *TIMEMARK* specifications.

To understand the interaction between these three parameters, let us consider an example. Assume, for the purpose of the example:

- *INACTIVE 3600* one hour
- *TIMEMARK 1200* 20 minutes
- *SCANINTERVAL 30* 30 seconds

With these values, every 30 seconds the server would scan the entire list of TCP/IP connections.

- If scan processing found a client that had not had activity for one hour, it would schedule that client for disconnection so that the connection could be used by another client.
- If scan processing found a client that had no activity in the last 20 minutes, a *TIMEMARK* probe would be sent to that client.
- If scan processing found a client that had been sent three probes without activity, that client would be scheduled for disconnection.

Since the scan process takes place (in this example) every 30 seconds, it can be assumed that at the conclusion of the scan there will be much less work (*TIMEMARK* and "clean-up" processing) to do than would have been the case with the 2 minute hard-coded scan interval. A longer *SCANINTERVAL* —say 60 seconds — would have resulted in less frequent scan processing, but longer periods of probe and clean-up activity because more clients would have reached the end of their *TIMEMARK* and *INACTIVE* periods.

These values are very dependent upon a number of other variables: number of connections, client think time, processor speed, and so on. Experimentation is the only way to obtain the optimum values. One user, with a 10 000 terminal network has found that the following parameters in TCPIP.PROFILE.TCPIP provide optimum performance.

```
INTERNALCLIENTPARMS
INACTIVE 3600
SCANINTERVAL  30
TIMEMARK      1200
DISABLESGA
ENDINTERNALCLIENTPARMS
```

It is also recommended that you disable GO AHEAD transmissions for both client and server when using Telnet. This can be done with the DISABLESGA statement, which reduces the overhead for a full duplex terminal using a full duplex connection. With this change, Telnet users should notice better response times at

logon and lower CPU utilization. (See *TCP/IP for MVS: Customization and Administration Guide* for further information.)

IP Over Channel to 374x/NCP And Native IP Over Escon Channel to 3746-9x0

The current NCP release is V7R5. The NCP resides in the 3745. The NCP IP router component can now connect to TCP/IP for MVS using one of three different technologies:

1. A SNALINK using the NCST component of the NCP. This was the only technique available for NCPs prior to V7R3.
2. A direct channel-attachment using a parallel channel and the CDLC channel protocol. The parallel channel attaches to a parallel channel adapter in the 3745.
3. A direct channel-attachment using an ESCON channel and the CDLC channel protocol. This configuration requires a 3746-900 to be installed together with the 3745. Only the 3746 is able to attach to an ESCON channel.

The 3746 Nways Controller (model 900 and 950) implements both an APPN network node and an IP router function. The 3746-9x0 IP router function is completely independent of the 3745 NCP IP router function. The 3746-9x0 IP router function connects to TCP/IP for MVS over an ESCON channel using the CDLC channel protocol.

As 3746-9X0 is now supporting native IP over escon channel, the throughput is increased as the data goes directly from the escon adapter to MVS TCP/IP in the host. This enhancement allows IP datagrams to be sent and received across the channel without having been encapsulated into SNA frames as otherwise done with SNALINK. The result is reduced mainframe CPU consumption and improved throughput because interaction with SNALINK address space is no longer required.

Performance Tuning is simplified because:

- Controller Configuration Management (CCM) Tool is configured such that default parameters are optimized for performance
- MTU value or IP packet size needs to be set in MVS TCP/IP profile (GATEWAY or BSDROUTING parameters). Recommended MTU size or packet size is 4096, make sure you explicitly state the packet size instead of default size.
- Make sure you use 4096 as the block size on the DEVICE statement in TCPIP profile dataset

For additional information, see the *TCP/IP for MVS: Planning and Migration Guide*.

Other Tuning Considerations

- For normal environments, trace activities should be disabled. They create significant system overhead, and should only be activated when needed.
- When tuning, change only one parameter at a time and measure results before making any other changes.

- To monitor traffic output that is captured in TCP.OUTPUT when you bring down TCP/IP, add the following in the TCPIP.PROFILE data set:

  ASSORTEDPARMS
  TCPIPSTATISTICS
  ASSORTEDPARMS

  If you have SDSF, TCP/IP statistics will show under TCP.jobname.

- MVS TCP/IP buffers
  - Set up MVS TCP/IP buffers (MVS TCP/IP Profile) using guidelines shown for FTP, Telnet and other applications.
  - Monitor MVS TCP/IP buffer use periodically using NETSTAT POOLSIZE on MVS.
  - Allocate more TCP/IP buffers if running low.
  - Remember, the buffer charts are guidelines; allocation may vary with different application mix.

Appendix B. Sample Files and Reports for Examples

This appendix contains sample files and reports used in examples.

Sample Files

This section contains sample files used in examples in this book.

Input Files for the VM Monitor Example

The following files are part of the VM Monitor example shown in “Example of the Use of VM Monitor and VMPRF” on page 62. They are input to the SSHXPN MASTER file.

- SSHXPN SETTINGS
- SSHXPN REPORTS
- SSHXPN INCLUSER
- SSHXPN UCLASS

The first input file is SSHXPN SETTINGS, as shown in Figure 95.

```
NODUMP
SYSTEM "smmddyyx"
SYSTEMID VMQ
REPORTS
  BYTIME 2 SECONDS
  INTERIM 10 SECONDS
  STIME 00:00:00
  ETIME 23:59:59
  MAXDASD 2000
  MAXMINIDISKS 1000
  MAXUSERS 500
  PAGESIZE 48
SUMMARY
  BYTIME 10 SECONDS
  INTERIM 10 SECONDS
  STIME 00:00:00
  ETIME 23:59:59
  MAXDASD 2000
  MAXMINIDISKS 1000
  MAXUSERS 700
  PAGESIZE 48
```

Figure 95. Sample Parameter Settings File for the VM Monitor Example

The second input file to SSHXPN MASTER is SHANIS REPORTS, as shown in Figure 96.

```
REPORT_TABLE_OF_CONTENTS
SYSTEM_SUMMARY_BY_TIME
UCLASS_RESOURCE_UTIL
UCLASS_RESPONSE
UCLASS_STATES
UCLASS_VMCOMM_ACTIVITY
USER_RESOURCE_UTIL
USER_RESOURCE_UTIL_BY_USERID
USER_RESPONSE
USER_STATES
USER_TO_USER_VMCOMM
USER_VMCOMM_ACTIVITY
SUMMARY_USER
DASD_BY_ACTIVITY
```

Figure 96. Sample Reports File for the VM Monitor Example

The third input file to SSHXPN MASTER is SHANIS INCLUSER, as shown in Figure 97.

```
TCPIP3
FTPSERV3
FTPD3
FTPD31
FTPD32
TCPIP
FTPSERV
FTPD
FTPD1
FTPD2
VMNFS
VMNFS3
RTMESA
MONWRITE
```

Figure 97. Sample Included Users File for the VM Monitor Example

The INCLUSER file makes sure that certain users are always included in the reports. Below is a sample for the INCLUSER file, using the default user IDs.

Please note that most installations will not use **all** of the default user IDs, or some might not be that important, therefore this list will be shorter than the UCLASS list. Your list should only include the users that you want to see on every report. Add any user IDs that are unique to your installation. For example, we have included FTPD2, FTPD3, and FTPD4 here, as shown in Figure 98.

```
TCPIP
FTPSERVE
FTPD2
FTPD3
FTPD4
SMTP
NAMESRV
PORTMAP
VMNFS
SNMPQE
SNMPQD
ROUTED
LPSERVE
```

Figure 98. Sample Included Users File Showing All Default User IDs

The fourth input file to SSHXPN MASTER is SHANIS UCLASS, as shown in Figure 99.

```
         CMS User  Default Class
FTP*     FTP Serv
TCPMON   Monitors
TCPIP*   TCPIP Sv
VMNFS*   FileServ
PORT*    Portmapr
RTMESA*  Monitors
MON*     Monitors
M0*      OurUsers

TCPIP    AllTCPIP
TCPDNS   AllTCPIP
FTPSERV  AllTCPIP
VMNFS    AllTCPIP
```

Figure 99. Sample User Classification File for the VM Monitor Example

The default user IDs for TCP/IP for VM are in the UCLASS file are shown in Figure 100. You should remember that if you are not using the default names, or if you are running multiple FTP servers or REXEC slave virtual machines, you need to add the default user IDs to your list. Otherwise the statistics for those virtual machines will not be included in the AllTCPIP grouping.

You might want to use `FTP*  AllTCPIP` and `RSLAV*  AllTCPIP` as entries to account for the cases where there are multiples. VMPRF documentation recommends entering these specifications in decreasing order of generality, so the FTP* and RSLAV* should be inserted after the CMS User statement and before the TCPMAINT statement.

```
         CMS User  Default Class
TCPMAINT AllTCPIP
TCPIP    AllTCPIP
FTPSERVE AllTCPIP
SMTP     AllTCPIP
NAMESRV  AllTCPIP
SNALNKA  AllTCPIP
REXECD   AllTCPIP
X25IPI   AllTCPIP
PORTMAP  AllTCPIP
VMNFS    AllTCPIP
SNMPQE   AllTCPIP
SNMPQD   AllTCPIP
VMKERB   AllTCPIP
ADM_SERV AllTCPIP
NCS      AllTCPIP
NCSLLBD  AllTCPIP
NCSGLBD  AllTCPIP
ROUTED   AllTCPIP
LPSERVE  AllTCPIP
NDBSERVE AllTCPIP
```

Figure 100. Sample User Classification File Showing Default Users

Reports Used in Examples

This section contains the reports used in the examples in this book. There are two types of reports used:

- VMPRF reports based on data from VM Monitor
- RMF interval reports

VMPRF Reports Based on Data from VM Monitor

The following VMPRF reports are used in the VM Monitor example. See “Example of the Use of VM Monitor and VMPRF” on page 62 to see the context.

- PRF002 SYSTEM_SUMMARY_BY_TIME
- PRF014 UCLASS_RESOURCE_UTIL
- PRF036 UCLASS_RESPONSE
- PRF019 UCLASS_STATES
- PRF035 UCLASS_VMCOMM_ACTIVITY
- PRF008 USER_RESOURCE_UTIL
- PRF054 USER_RESOURCE_UTIL_BY_USERID
- PRF033 USER_RESPONSE
- PRF018 USER_STATES
- PRF030 USER_TO_USER_VMCOMM
- PRF032 USER_VMCOMM_ACTIVITY
- PRF012 DASD_BY_ACTIVITY

Please notice that in the interest of space, some of the reports were reduced from full length.

```
1PRF002  Run 12/02/1993 09:44:34         SYSTEM_SUMMARY_BY_TIME
4
                                         System Performance Summary by Time
 From 05/09/1993 22:55:25                                                                                VMQ
 To   05/09/1993 23:10:01                                                                                CPU 3090
SN
76530
 For    876 Secs 00:14:35                NMY0993E                                                        ES/VM
20.01
SLU
0000
```

```
_
```

```
            <--------CPU---------> <Vec> <--Users--> <---I/O---> <Stg> <-Paging-> <Spl> <----------UP+MP Transactions---------->
                 <--Ratio-->                                                  <--Rate-->       <--Response Time--> <----Throughput---->
0                                                                      DASD        PGIN   Rd
 From  To    Pct         Cap-   On-   Pct  Log-              Activ Rate  Resp         and  and              Non-  Quick          Non-  Quick
 Time  Time  Busy    T/V ture  line  Busy   ged              Time Elist PGOUT   Wr Rate  Triv   Triv   Disp   Triv   Triv   Disp
 23:06 23:06   0.6  1.93 .3767  4.0     0    14     3   27    3.4     0     0    0    0 0.562      0  0.359   0.17
0
0.17
 23:06 23:06   0.4  2.59 .0903  4.0     0    14     1   27    3.6     0     0    0    0      0      0  0.345      0
0
0.17
 23:06 23:06   0.6  1.96 .3919  4.0     0    14     3   27    3.5     0     0    0    0 0.473      0  0.397   0.17
0
0.17
 23:06 23:06   0.6  1.94 .3756  4.0     0    14     3   27    3.5     0     0    0    0 0.477      0  0.367   0.17
0
0.17
 23:06 23:06   0.4  2.60 .0871  4.0     0    14     2   26    3.4     0     0    0    0      0      0  0.353      0
0
0.17
 23:06 23:06   0.7  1.91 .3860  4.0     0    14     4   27    3.5     0     0    0    0 0.476  5.856  0.355   0.17
0.17
0.17
 23:06 23:06   0.4  2.04 .1093  4.0     0    14     2   26    3.4     0     0    0    0      0      0  0.358      0
0
0.17
 23:06 23:06   0.6  1.92 .3914  4.0     0    14     3   27    3.5     0     0    0    0 0.475      0  0.361   0.17
0
0.17
 23:06 23:06   0.7  1.98 .3737  4.0     0    14     3   27    3.5     0     0    0    0 0.487      0  0.364   0.17
0
0.17
 23:06 23:07   0.4  2.45 .0953  4.0     0    14     2   27    3.4     0     0    0    0      0      0  0.350      0
0
0.17
 23:07 23:07   0.7  1.97 .3729  4.0     0    14     3   27    3.6     0     0    0    0 0.478  0.208  0.351   0.17
0.17
0.17
 23:07 23:07   3.6  1.57 .8278  4.0     0    14     5   86    7.7     0     0    0    0 0.139      0  0.384   0.50
0
0.17
 23:07 23:07   9.6  1.47 .8962  4.0     0    14     5  129    8.0     0     0    0    0 0.476      0  0.370   0.17
0
0.17
 23:07 23:07   1.9  1.56 .7258  4.0     0    14     5   42    4.7     0     0    0    0 0.361      0  0.378   0.33
0
0.17
 23:07 23:07   0.4  2.51 .0887  4.0     0    14     2   26    3.4     0     0    0    0      0      0  0.381      0
0
0.17
 23:07 23:07   0.7  1.96 .3768  4.0     0    14     3   27    3.4     0     0    0    0 0.564      0  0.362   0.17
0
0.17
 23:07 23:07   0.4  2.44 .0926  4.0     0    14     2   26    3.6     0     0    0    0      0      0  0.343      0
0
0.17
 23:07 23:07   0.6  1.93 .3923  4.0     0    14     3   27    3.5     0     0    0    0 0.472      0  0.369   0.17
0
```

```
0.17
 23:07 23:07   0.4  2.45 .0890  4.0    0   14    3   26    3.4    0    0   0   0 0.474     0  0.372   0.17 0 0.17
 23:07 23:08   0.6  1.93 .3880  4.0    0   14    3   27    3.7    0    0   0   0     0     0  0.352     0 0 0.17
 23:08 23:08   0.7  1.98 .3748  4.0    0   14    3   27    3.5    0    0   0   0 0.362     0  0.357   0.33 0 0.17
 23:08 23:08   0.4  2.56 .0900  4.0    0   14    2   26    3.4    0    0   0   0     0     0  0.365     0 0 0.17
 23:08 23:08   0.6  1.95 .3919  4.0    0   14    3   27    3.5    0    0   0   0 0.476     0  0.346   0.17 0 0.17
 23:08 23:08   0.4  2.53 .0841  4.0    0   14    2   27    3.5    0    0   0   0     0     0  0.361     0 0 0.17
 23:08 23:08   0.6  1.95 .3891  4.0    0   14    3   27    3.5    0    0   0   0 0.475     0  0.375   0.17 0 0.17
 23:08 23:08   0.7  1.96 .3769  4.0    0   14    3   27    3.5    0    0   0   0 0.474     0  0.340   0.17 0 0.17
 23:08 23:08   0.4  2.33 .0974  4.0    0   14    2   26    3.6    0    0   0   0     0     0  0.370     0 0 0.17
 23:08 23:08   0.6  1.93 .3937  4.0    0   14    3   27    3.5    0    0   0   0 0.477     0  0.385   0.17 0 0.17
 23:08 23:08   0.4  2.50 .0841  4.0    0   14    2   26    3.5    0    0   0   0     0     0  0.365     0 0 0.17
 23:08 23:09   0.6  1.97 .3918  4.0    0   14    3   28    3.5    0    0   0   0 0.477     0  0.362   0.17 0 0.17 023:09 23:09   0.6
1.94 .3763  4.0    0   14    3   27    3.4    0    0   0   0 0.474     0  0.376   0.17 0 0.17
 23:09 23:09   2.9  1.67 .8018  4.0    0   14    6   75    8.4    0    0   0   0 0.197  8.278  0.356   0.33 0.33 0.17
 23:09 23:09  10.0  1.52 .8863  4.0    0   14    7  194    9.5    0    0   0   0 0.239     0  0.354   0.33 0 0.17
 23:09 23:09   9.6  1.48 .8894  4.0    0   14    6  148    9.8    0    0   0   0     0     0  0.764     0 0 0.17
 23:09 23:09   6.9  1.49 .8793  4.0    0   14    7   99   13.6    0    0   0   0 0.363     0  0.521   0.33 0 0.17
 23:09 23:09   0.6  1.94 .3730  4.0    0   14    3   27    3.5    0    0   0   0 0.476     0  0.887   0.17 0 0.17
 23:09 23:09   0.4  2.54 .0975  4.0    0   14    2   26    3.4    0    0   0   0     0     0  0.368     0 0 0.17
 23:09 23:09   0.6  1.94 .3919  4.0    0   14    3   27    3.8    0    0   0   0 0.477     0  0.349   0.17 0 0.17
 23:09 23:09   0.4  2.50 .0849  4.0    0   14    1   27    3.4    0    0   0   0     0     0  0.381     0 0 0.17
 23:09 23:10   0.6  1.94 .3896  4.0    0   14    3   27    3.5    0    0   0   0 0.475     0  0.363   0.17      0 0.17
```

Figure 101. SYSTEM_SUMMARY_BY_TIME Report (partial)

```
1PRF014  Run 12/02/1993 09:44:34         UCLASS_RESOURCE_UTIL
6
                                         Resource Utilization by User Class
 From 05/09/1993 22:55:25                                                                                VMQ
 To   05/09/1993 23:10:01                                                                                CPU 3090
SN
76530
 For    876 Secs 00:14:35                NMY0993E                                                        ES/VM
20.01
SLU
0000
```

```
_
                <---------CPU---------> <Vec> <-User Time> <-DASD-> <----Storage----> <-----------Paging----------> <---Spool--->
                     <-Seconds->             <--Minutes->          <-----Pages-----> <--Slots--> <Rate wh/ Logged>
0          Log-                                          Rate                                          XSTORE     DASD          Rate
 User       ged                        T/V                While   Est                                   PGIN+    Read+         While
 Class    Users   Pct Total  Virt Ratio  Secs Logged Activ   Logged   WSS Resid Lockd XSTOR  DASD    PGOUT    Write  Pages Logged

 TCPIP Sv     1   0.6    22    14   1.5     0     15    14        0  2098  2147    42     0     0        0        0      0      0
 FTP Serv     4   0.1     4     2   2.1     0     58     6     2.37   383   392     0     0     0        0        0      0      0
 OurUsers     1   0.1     4     2   1.9     0     15     9     1.18   250   252     0     0     0        0        0     29   0.07
 Monitors     1   0.0     1     1   2.5     0     15    15     2.39   133   133     0     0     0        0        0      0      0
 CMS User     3   0.0     0     0   3.4     0     44     1        0   682   685     0     0     0        0        0      2   0.00
 FileServ     1     0     0     0     0     0     15     0        0  1645  1645     0     0     0        0        0      0      0
 Portmapr     1     0     0     0     0     0     15     0        0   372   372     0     0     0        0        0      0      0
 GCS          2     0     0     0     0     0     29     0        0    66    67     1     0     0        0        0      0      0

 Sum/Mean    14   0.9    32    19   1.7     0    204    45     5.95   586   593     3     0     0        0        0     31   0.07
```

Figure 102. UCLASS_RESOURCE_UTIL Report

```
1PRF036  Run 12/02/1993 09:44:34         UCLASS_RESPONSE
7
                                         Transaction Response Time and Throughput by User Class
 From 05/09/1993 22:55:25                                                                        VMQ
 To   05/09/1993 23:10:01                                                                        CPU 3090
SN
76530
 For    876 Secs 00:14:36                NMY0993E                                                ES/VM
20.01
SLU
0000
```

```
                      <--Response Time--> <-Transactions/Hour> <---Transactions--->
0          Log-
 User       ged  Think          Non-                 Non-                 Non-
 Class    Users   Time   Triv   Triv   Mean   Triv   Triv  Total   Triv   Triv  Total

 Monitors     1    5.6      0  0.385  0.385      0    596    596      0    145    145
 OurUsers     1    9.7  0.481      0  0.481    349      0    349     85      0     85
 CMS User     3   43.0  0.324      0  0.324     78      0     78     19      0     19
 FTP Serv     4  176.5  0.004  8.546  5.029     29     41     70      7     10     17
 Portmapr     1      0      0      0      0      0      0      0      0      0      0
 FileServ     1      0      0      0      0      0      0      0      0      0      0
 TCPIP Sv     1      0      0      0      0      0      0      0      0      0      0
 GCS          2      0      0      0      0      0      0      0      0      0      0

 Sum/Mean    14   19.5  0.423  0.912  0.710    448    637   1085    109    155    264
```

Figure 103. UCLASS_RESPONSE Report

```
1PRF019  Run 12/02/1993 09:44:34         UCLASS_STATES
8
                                         User State Samples by User Class
 From 05/09/1993 22:55:25                                                                        VMQ
 To   05/09/1993 23:10:01                                                                        CPU 3090
SN
76530
 For    876 Secs 00:14:35                NMY0993E                                                ES/VM
20.01
SLU
0000
```

```
               <-User Time>        <Pct of Time> <----------------------Percent of True Non-Dormant Time---------------------->
                                                 <---------------------------Waiting on---------------------------->
                                                                                                  <----SVM and---->
0          Log-                                  True                                                                        I/O
 User       ged    CPU Activ  State   True   Non-  Run-       Load-                 Inst  Test  Cons  Test Elig-  Dor-    Ac
 Class    Users   Secs  Mins Sample Dormnt Dormnt  ning   CPU   ing  Page   I/O   Sim  Idle  Func  Idle  ible  mant  tive Other

 TCPIP Sv     1     22    14    876   13.7   86.3   2.8     0     0     0     0   0.5     0     0     0     0     0  96.7     0
 FTP Serv     4      4     6   3504   92.2    7.8   1.5   1.1     0     0     0  19.3   6.6     0  55.1     0  16.4     0     0
 OurUsers     1      4     9    876      0  100.0   0.2   0.3     0     0   1.3     0     0  98.2     0     0     0     0     0
 Monitors     1      1    15    876   99.5    0.5     0     0     0     0     0  75.0  25.0     0     0     0     0     0     0
 CMS User     3      0     1   2628   99.8    0.2     0     0     0     0     0     0  20.0     0     0     0     0  20.0  60.0
 FileServ     1      0     0    876  100.0      0     0     0     0     0     0     0     0     0     0     0     0     0     0
 Portmapr     1      0     0    876      0  100.0     0     0     0     0     0     0     0     0     0     0 100.0     0     0
 GCS          2      0     0   1752  100.0      0     0     0     0     0     0     0     0     0     0     0     0     0     0

 Sum/Mean    14     32    45  12264   77.2   22.8   1.0   0.2     0     0   0.4   2.1   0.7  30.8   5.4     0  33.0  26.2   0.1
```

Figure 104. UCLASS_STATES Report

```
1PRF035  Run 12/02/1993 09:44:34         UCLASS_VMCOMM_ACTIVITY
9
                                         Virtual Machine Communications Activity by User Class
 From 05/09/1993 22:55:25                                                                    VMQ
 To   05/09/1993 23:10:01                                                                    CPU 3090
SN
76530
 For    876 Secs 00:14:35                NMY0993E                                           ES/VM
20.01
SLU
0000
```

```
_
                    <----------------------Message Rate--------------------> <----Mean Msgs in Queue---> <-----Per Message----->
                    IUCV+VMCF<-------IUCV------> <--------VMCF-------->       <--------IUCV--------> <VMCF>
0          Log-                                                                                                  DASD
 User       ged  Total                                                  DIAG                                 CPU    SSCH     Page
 Class    Users   Msgs  Total   Send   Recv   Fail   Send   Recv   Fail    98   Send   Recv  Reply    All    Secs   +RSCH   Rd+Wr

 TCPIP Sv     1   3744  4.274      0      0      0  1.467  2.807      0 14.897      0  1.000      0  0.521  0.0057       0       0
 FTP Serv     4   3400  3.882      0      0      0  2.513  1.369      0     0      0      0      0  0.002  0.0013    0.61       0
 Monitors     1    585  0.668  0.334  0.333      0      0      0      0     0      0      0      0      0  0.0021    3.58       0
 OurUsers     1    344  0.393      0      0      0  0.295  0.098      0     0      0      0      0      0  0.0120    3.00       0
 FileServ     1      0      0      0      0      0      0      0      0     0      0      0      0      0       0       0       0
 Portmapr     1      0      0      0      0      0      0      0      0     0      0      0      0      0       0       0       0
 CMS User     3      0      0      0      0      0      0      0      0     0      0      0      0      0       0       0       0
 GCS          2      0      0      0      0      0      0      0      0     0      0      0      0      0       0       0       0

 Sum/Mean    14   8073  9.216  0.334  0.333      0  4.274  4.274      0 14.897      0  0.071      0  0.038  0.0039    0.64       0
```

Figure 105. UCLASS_VMCOMM_ACTIVITY Report

```
1PRF008  Run 12/02/1993 09:44:34         USER_RESOURCE_UTIL
10
                                         Resource Utilization by User
 From 05/09/1993 22:55:25                                                                    VMQ
 To   05/09/1993 23:10:01                                                                    CPU 3090
SN
76530
 For    876 Secs 00:14:35                NMY0993E                                           ES/VM
20.01
SLU
0000
```

```
_
          <---------CPU---------> <Vec> <-User Time-> <-DASD-> <----Storage----> <-----------Paging----------> <---Spool--->
               <-Seconds->             <--Minutes-->          <-----Pages-----> <--Slots--> <Rate wh/ Logged>
0                                                        Rate                                  XSTORE     DASD          Rate   Data
                                 T/V                    While   Est                             PGIN+     Read+         While Spaces
 Userid    Pct  Total  Virt Ratio  Secs Logged Active   Logged   WSS Resid Lockd XSTOR  DASD    PGOUT     Write  Pages Logged  Owned

 TCPIP3    0.6    22    14   1.5     0     15     14        0  2098  2147    42     0     0        0         0      0      0      0
 FTPD32    0.1     2     1   2.1     0     15      2     1.02   384   402     0     0     0        0         0      0      0      0
 FTPD31    0.0     1     1   2.1     0     15      2     0.68   383   395     0     0     0        0         0      0      0      0
 MONWRITE  0.0     1     1   2.5     0     15     15     2.39   133   133     0     0     0        0         0      0      0      0
 FTPSERV3  0.0     1     0   2.1     0     15      1     0.34   382   387     0     0     0        0         0      0      0      0
 FTPD3     0.0     1     0   2.1     0     15      1     0.34   382   384     0     0     0        0         0      0      0      0
 VMNFS3      0     0     0     0     0     15      0        0  1645  1645     0     0     0        0         0      0      0      0

 M0200     0.1     4     2   1.9     0     15      9     1.18   250   252     0     0     0        0         0     29   0.07      0
 MAINT     0.0     0     0   3.4     0     15      1        0   712   712     0     0     0        0         0      2   0.00      0
 OPERATOR  0.0     0     0     0     0     15      0        0   667   667     0     0     0        0         0      0      0      0
 PORTMAP3    0     0     0     0     0     15      0        0   372   372     0     0     0        0         0      0      0      0
 GCSXA2      0     0     0     0     0     15      0        0    66    67     1     0     0        0         0      0      0      0
 GCSXA       0     0     0     0     0     15      0        0    66    67     1     0     0        0         0      0      0      0
 OPERSYMP    0     0     0     0     0     15      0        0   667   677     0     0     0        0         0      0      0      0
0The table above contains the top    14 of    14 items.
 Sum/Mean  0.9    32    19   1.7     0    204     45     0.42   586   593     3     0     0        0         0     31   0.01      0
```

Figure 106. USER_RESOURCE_UTIL Report

```
1PRF054  Run 12/02/1993 09:44:34         USER_RESOURCE_UTIL_BY_USERID
11
                                         Resource Utilization by User
 From 05/09/1993 22:55:25                                                                    VMQ
 To   05/09/1993 23:10:01                                                                    CPU 3090
SN
76530
 For    876 Secs 00:14:35                NMY0993E                                           ES/VM
20.01
SLU
0000
```

```
_
```

```
          <---------CPU---------> <Vec> <-User Time-> <-DASD-> <----Storage----> <-----------Paging----------> <---Spool--->
                <-Seconds->             <--Minutes-->          <-----Pages-----> <--Slots--> <Rate wh/ Logged>
0                                                        Rate                                    XSTORE     DASD           Rate   Data
                                    T/V                 While   Est                              PGIN+     Read+          While Spaces
 Userid    Pct  Total   Virt Ratio  Secs Logged Active   Logged   WSS Resid Lockd XSTOR  DASD    PGOUT     Write  Pages Logged  Owned

 FTPD3     0.0      1      0   2.1     0     15      1     0.34   382   384     0     0     0        0         0      0      0      0
 FTPD31    0.0      1      1   2.1     0     15      2     0.68   383   395     0     0     0        0         0      0      0      0
 FTPD32    0.1      2      1   2.1     0     15      2     1.02   384   402     0     0     0        0         0      0      0      0
 FTPSERV3  0.0      1      0   2.1     0     15      1     0.34   382   387     0     0     0        0         0      0      0      0
 GCSXA       0      0      0     0     0     15      0        0    66    67     1     0     0        0         0      0      0      0
 GCSXA2      0      0      0     0     0     15      0        0    66    67     1     0     0        0         0      0      0      0
 MAINT     0.0      0      0   3.4     0     15      1        0   712   712     0     0     0        0         0      2   0.00      0
 MONWRITE  0.0      1      1   2.5     0     15     15     2.39   133   133     0     0     0        0         0      0      0      0
 M0200     0.1      4      2   1.9     0     15      9     1.18   250   252     0     0     0        0         0     29   0.07      0
 OPERATOR  0.0      0      0     0     0     15      0        0   667   667     0     0     0        0         0      0      0      0
 OPERSYMP    0      0      0     0     0     15      0        0   667   677     0     0     0        0         0      0      0      0
 PORTMAP3    0      0      0     0     0     15      0        0   372   372     0     0     0        0         0      0      0      0
 TCPIP3    0.6     22     14   1.5     0     15     14        0  2098  2147    42     0     0        0         0      0      0      0
 VMNFS3      0      0      0     0     0     15      0        0  1645  1645     0     0     0        0         0      0      0      0

 Sum/Mean  0.9     32     19   1.7     0    204     45     0.42   586   593     3     0     0        0         0     31   0.01      0
```

Figure 107. USER_RESOURCE_UTIL_BY_USERID Report

```
1PRF033  Run 12/02/1993 09:44:34         USER_RESPONSE
12
                                         Transaction Response Time and Throughput by User
 From 05/09/1993 22:55:25                                                                    VMQ
 To   05/09/1993 23:10:01                                                                    CPU 3090
SN
76530
 For    876 Secs 00:14:36                NMY0993E                                           ES/VM
20.01
SLU
0000
```

```
_
```

```
                 <--Response Time--> <-Transactions/Hour> <---Transactions--->
0
           Think          Non-                Non-                Non-
 Userid     Time   Triv   Triv   Mean   Triv   Triv  Total   Triv   Triv  Total

 MONWRITE    5.6      0  0.385  0.385      0    600    600      0    145    145
 FTPD32    149.7  0.008  7.814  4.887     14     23     37      3      5      8
 FTPD31    242.0  0.001  9.285  5.572     13     19     32      2      3      5
 FTPSERV3  119.9  0.001 11.047  5.524      8      8     17      1      1      2
 FTPD3     119.9  0.001  7.492  3.746      8      8     17      1      1      2
 VMNFS3        0      0      0      0      0      0      0      0      0      0
 TCPIP3        0      0      0      0      0      0      0      0      0      0

 M0200       9.7  0.481      0  0.481    353      0    353     85      0     85
 MAINT      43.0  0.324      0  0.324     81      0     81     19      0     19
 OPERATOR      0      0      0      0      0      0      0      0      0      0
 PORTMAP3      0      0      0      0      0      0      0      0      0      0
 GCSXA2        0      0      0      0      0      0      0      0      0      0
 GCSXA         0      0      0      0      0      0      0      0      0      0
 OPERSYMP      0      0      0      0      0      0      0      0      0      0
0The table above contains the top    14 of    14 items.
 Sum/Mean   19.5  0.423  0.912  0.710     82    117    199    109    155  D 264
```

Figure 108. USER_RESPONSE

1PRF018 Run 12/02/1993 09:44:34 USER_STATES
13
User State Samples by User
From 05/09/1993 22:55:25 VMQ
To 05/09/1993 23:10:01 CPU 3090
SN
76530
For 876 Secs 00:14:35 NMY0993E ES/VM
20.01
SLU
0000

_

0	<--User Time->			<Pct of Time>		<--------------------Percent of True Non-Dormant Time------------------------>												
							<--------------------------Waiting on--------------------------------->											
														<----SVM and---->				
Userid	CPU Secs	Active Mins	State Samples	True Dormnt	True Non-Dormnt	Run-ning	CPU	Load-ing	Page	I/O	Inst Sim	Test Idle	Cons Func	Test Idle	Elig-ible	Dor-mant	I/O Active	Other
TCPIP3	22	14	876	13.7	86.3	2.8	0	0	0	0	0.5	0	0	0	0	0	96.7	0
FTPD32	2	2	876	89.3	10.7	2.1	1.1	0	0	0	22.3	8.5	0	51.1	0	14.9	0	0
FTPD31	1	2	876	91.2	8.8	1.3	1.3	0	0	0	24.7	6.5	0	54.5	0	11.7	0	0
MONWRITE	1	15	876	99.5	0.5	0	0	0	0	0	75.0	25.0	0	0	0	0	0	0
FTPSERV3	1	1	876	93.8	6.2	1.9	0	0	0	0	16.7	7.4	0	59.3	0	14.8	0	0
FTPD3	1	1	876	94.4	5.6	0	2.0	0	0	0	8.2	2.0	0	59.2	0	28.6	0	0
VMNFS3	0	0	876	100.0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
M0200	4	9	876	0	100.0	0.2	0.3	0	0	1.3	0	0	98.2	0	0	0	0	0
MAINT	0	1	876	99.5	0.5	0	0	0	0	0	0	0	0	0	0	0	25.0	75.0
OPERATOR	0	0	876	99.9	0.1	0	0	0	0	0	0	100.0	0	0	0	0	0	0
PORTMAP3	0	0	876	0	100.0	0	0	0	0	0	0	0	0	0	0	100.0	0	0
GCSXA2	0	0	876	100.0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
GCSXA	0	0	876	100.0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
OPERSYMP	0	0	876	100.0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0The table above contains the top 14 of 14 items.																		
Sum/Mean	32	45	12264	77.2	22.8	1.0	0.2	0	0	0.4	2.1	0.7	30.8	5.4	0	33.0	26.2	0.1

Figure 109. USER_STATES

```
1PRF030  Run 12/02/1993 09:44:34         USER_TO_USER_VMCOMM
14
                                         Virtual Machine Communications - User to User Samples
 From 05/09/1993 22:55:25                                                                                VMQ
 To   05/09/1993 23:10:01                                                                                CPU 3090
SN
76530
 For    876 Secs 00:14:35                NMY0993E                                                        ES/VM
20.01
SLU
0000
```

```
_
                                          <------------------Message Rate-----------------> <----Mean Msgs in Queue--->
                                          IUCV+VMCF<-------IUCV------> <-------VMCF-------> <-------IUCV-------> <VMCF>
0                 S Logged  Number
          Msg     V   Time      of  Total
 Userid   Target  M  (min) Samples   Msgs  Total   Send   Recv   Fail   Send   Recv   Fail   Send   Recv  Reply    All

 TCPIP3   FTPD32   N      1      12   1484 20.610      0      0      0  7.139 13.472      0      0  1.000      0  2.667
 FTPD32   TCPIP3   N      2      23   1460 10.580      0      0      0  6.841  3.739      0      0      0      0  0.043
 FTPD31   TCPIP3   N      2      18    974  9.018      0      0      0  5.833  3.185      0      0      0      0      0
 TCPIP3   FTPD31   N      1       5    718 23.933      0      0      0  7.600 16.333      0      0  1.000      0  2.200
 TCPIP3   FTPSERV3 N      0       4    649 27.045      0      0      0 11.501 15.543      0      0  1.000      0  2.750
 MONWRITE *MONITOR N     15     146    585  0.668  0.334  0.333      0      0      0      0      0      0      0      0
 TCPIP3   FTPD3    N      0       3    569 31.612      0      0      0 10.445 21.167      0      0  1.000      0  7.333
 FTPSERV3 TCPIP3   N      1      10    484  8.067      0      0      0  5.233  2.833      0      0      0      0      0
 FTPD3    TCPIP3   N      1      11    482  7.303      0      0      0  4.742  2.561      0      0      0      0      0
 TCPIP3   M0200    N      7      74    324  0.730      0      0      0  0.178  0.552      0      0  1.000      0      0

 M0200    TCPIP3   N      9      86    344  0.667      0      0      0  0.500  0.167      0      0      0      0      0
0The table above contains the top    11 of    11 items.
 Sum/Mean               39     392   8073  3.432  0.125  0.124      0  1.592  1.592      0      0  0.250      0  0.196
```

Figure 110. USER_TO_USER_VMCOMM

```
1PRF032  Run 12/02/1993 09:44:34         USER_VMCOMM_ACTIVITY
15
                                         Virtual Machine Communications Activity by User
 From 05/09/1993 22:55:25                                                                VMQ
 To   05/09/1993 23:10:01                                                                CPU 3090
SN
76530
 For    876 Secs 00:14:35                NMY0993E                                        ES/VM
20.01
SLU
0000
```

```
_
```

```
                <----------------------Message Rate--------------------> <----Mean Msgs in Queue---> <-----Per Message----->
                IUCV+VMCF<-------IUCV------> <-------VMCF-------->       <-------IUCV-------> <VMCF>
0                                                                                                          DASD
           Total                                                  DIAG                                CPU    SSCH    Page
 Userid     Msgs  Total   Send   Recv   Fail   Send   Recv   Fail   98   Send   Recv  Reply   All    Secs   +RSCH   Rd+Wr

 TCPIP3     3744  4.274      0      0      0  1.467  2.807      0 14.897      0  1.000      0  0.521  0.0057       0       0
 FTPD32     1460  1.667      0      0      0  1.078  0.589      0      0      0      0      0  0.007  0.0013    0.61       0
 FTPD31      974  1.112      0      0      0  0.719  0.393      0      0      0      0      0      0  0.0013    0.61       0
 MONWRITE    585  0.668  0.334  0.333      0      0      0      0      0      0      0      0      0  0.0021    3.58       0
 FTPSERV3    484  0.553      0      0      0  0.358  0.194      0      0      0      0      0      0  0.0013    0.61       0
 FTPD3       482  0.550      0      0      0  0.357  0.193      0      0      0      0      0      0  0.0013    0.61       0
 VMNFS3        0      0      0      0      0      0      0      0      0      0      0      0      0       0       0       0

 M0200       344  0.393      0      0      0  0.295  0.098      0      0      0      0      0      0  0.0120    3.00       0
 MAINT         0      0      0      0      0      0      0      0      0      0      0      0      0       0       0       0
 OPERATOR      0      0      0      0      0      0      0      0      0      0      0      0      0       0       0       0
 PORTMAP3      0      0      0      0      0      0      0      0      0      0      0      0      0       0       0       0
 GCSXA2        0      0      0      0      0      0      0      0      0      0      0      0      0       0       0       0
 GCSXA         0      0      0      0      0      0      0      0      0      0      0      0      0       0       0       0
 OPERSYMP      0      0      0      0      0      0      0      0      0      0      0      0      0       0       0       0
0The table above contains the top    14 of    14 items.
 Sum/Mean   8073  0.658  0.024  0.024      0  0.305  0.305      0  1.064      0  0.071      0  0.038  0.0039    0.64       0
```

Figure 111. USER_VMCOMM_ACTIVITY

```
1PRF012  Run 12/02/1993 09:44:35         DASD_BY_ACTIVITY
16
                                         DASD Activity Ordered by Activity
 From 05/09/1993 22:55:25                                                                                VMQ
 To   05/09/1993 23:10:01                                                                                CPU 3090
SN
76530
 For    876 Secs 00:14:36                NMY0993E                                                        ES/VM
20.01
SLU
0000
```

```
_
```

<-----Device------>					<---------SSCH+RSCH-------->				<-----------Time------------>						<-SSCHs in> <--Queue-->		
0 Num- ber	Volume Serial	Type	Control Unit	Owner	Mini- disk Links	On- line Secs	Count	Rate	Plus Avoided	Plus Avoid Rate	Pct Busy	Pend	Disc	Conn	Serv	Resp	Mean	Max	Err
0364	VMPRF9	3380-K	3990-3		41	876	2260	2.6	2737	3.1	7.0	0.1	19.8	7.1	27.1	27.9	0.0	0.2	0
0372	VMPRF2	3380-K	3990-3		27	876	974	1.1	2100	2.4	1.5	0.0	6.2	7.2	13.4	13.4	0	0	0
0371	VMPRF1	3380-K	3990-3		5	876	1270	1.4	1282	1.5	2.0	0.0	5.5	8.0	13.5	13.5	0	0	0
0374	ES2RES	3380-K	3990-3		106	876	315	0.4	315	0.4	0.1	0.0	0.8	0.8	1.6	1.6	0	0	0
023F	TCPIP2	3380-K	3880-03		0	876	307	0.4	311	0.4	0.8	0.0	10.1	11.8	22.0	22.0	0	0	0
0361	WRKLB3	3380-K	3990-3		0	876	298	0.3	298	0.3	0.0	0.0	0.1	0.7	0.8	0.8	0	0	3
0360	WRKLB1	3380-K	3990-3		0	876	296	0.3	296	0.3	0.0	0.0	0.1	0.7	0.8	0.8	0	0	2
02C0	DSW777	3380-D	3880-23		0	876	292	0.3	292	0.3	0.1	0.0	0.1	3.0	3.2	3.2	0	0	0
02C1	F43AAA	3380-D	3880-23		0	876	292	0.3	292	0.3	0.1	0.0	0.1	3.1	3.2	3.2	0	0	0
02C2	A45AAA	3380-D	3880-23		0	876	292	0.3	292	0.3	0.1	0.0	0.1	3.1	3.2	3.2	0	0	0
02C3	CPDLB2	3380-D	3880-23		0	876	292	0.3	292	0.3	0.1	0.0	0.1	3.1	3.2	3.2	0	0	0
02C4	DSW888	3380-D	3880-23		0	876	292	0.3	292	0.3	0.1	0.0	0.1	3.0	3.2	3.2	0	0	0
02C5	DSW999	3380-D	3880-23		0	876	292	0.3	292	0.3	0.1	0.0	0.1	3.1	3.2	3.2	0	0	0
02C6	DSWAAA	3380-D	3880-23		0	876	292	0.3	292	0.3	0.1	0.0	0.1	3.1	3.2	3.2	0	0	0
02C7	CPDLB2	3380-D	3880-23		0	876	292	0.3	292	0.3	0.1	0.0	0.1	3.1	3.2	3.2	0	0	0
02C8	DSD2C8	3380-D	3880-23		0	876	292	0.3	292	0.3	0.1	0.0	0.1	3.1	3.2	3.2	0	0	0
02C9	E41AD6	3380-D	3880-23		0	876	292	0.3	292	0.3	0.1	0.0	0.1	3.1	3.2	3.2	0	0	0
02CA	CPDLB2	3380-D	3880-23		0	876	292	0.3	292	0.3	0.1	0.0	0.1	3.1	3.2	3.2	0	0	0
02CB	XASPOL	3380-D	3880-23		0	876	292	0.3	292	0.3	0.1	0.0	0.1	3.1	3.2	3.2	0	0	0
02CC	DSD2CC	3380-D	3880-23		0	876	292	0.3	292	0.3	0.1	0.0	0.1	3.1	3.2	3.2	0	0	0
02CD	TCPIP1	3380-D	3880-23		0	876	292	0.3	292	0.3	0.1	0.0	0.1	3.1	3.2	3.2	0	0	0
02CE	CPDLB2	3380-D	3880-23		0	876	292	0.3	292	0.3	0.1	0.0	0.1	3.1	3.2	3.2	0	0	0
02CF	CPDLB2	3380-D	3880-23		0	876	292	0.3	292	0.3	0.1	0.0	0.1	3.1	3.2	3.2	0	0	0
0362	WRKLB4	3380-K	3990-3		0	876	292	0.3	292	0.3	0.0	0.0	0.1	0.7	0.8	0.8	0	0	0
0363	TCPIP4	3380-K	3990-3		0	876	292	0.3	292	0.3	0.0	0.0	0.1	0.7	0.8	0.8	0	0	0
0365	WRKLB6	3380-K	3990-3		0	876	292	0.3	292	0.3	0.0	0.0	0.1	0.7	0.8	0.8	0	0	0
0366	WRKLB7	3380-K	3990-3		0	876	292	0.3	292	0.3	0.0	0.0	0.1	0.7	0.8	0.8	0	0	0
0367	WRKLB8	3380-K	3990-3		0	876	292	0.3	292	0.3	0.0	0.0	0.1	0.7	0.8	0.8	0	0	0
0368	VMSSYS	3380-K	3990-3		1	876	292	0.3	292	0.3	0.0	0.0	0.1	0.7	0.8	0.8	0	0	0
0369	G202F1	3380-K	3990-3		0	876	292	0.3	292	0.3	0.0	0.0	0.1	0.7	0.8	0.8	0	0	0
036A	G2036A	3380-K	3990-3		0	876	292	0.3	292	0.3	0.0	0.1	0.1	0.7	0.8	0.8	0	0	0
036B	TCPIP1	3380-K	3990-3		0	876	292	0.3	292	0.3	0.0	0.0	0.1	0.7	0.8	0.8	0	0	0
036C	PIDCPY	3380-K	3990-3		0	876	292	0.3	292	0.3	0.0	0.0	0.1	0.7	0.8	0.8	0	0	0
036D	BLD111	3380-K	3990-3		0	876	292	0.3	292	0.3	0.0	0.0	0.1	0.7	0.8	0.8	0	0	0
0036E	A45AAA	3380-K	3990-3		0	876	292	0.3	292	0.3	0.0	0.0	0.1	0.7	0.8	0.8	0	0	0
036F	WRKLB2	3380-K	3990-3		0	876	292	0.3	292	0.3	0.0	0.0	0.1	0.7	0.8	0.8	0	0	0
0370	CPDLB2	3380-K	3990-3		0	876	292	0.3	292	0.3	0.0	0.0	0.1	0.7	0.8	0.8	0	0	0
0373	ESAV11	3380-K	3990-3		0	876	292	0.3	292	0.3	0.0	0.0	0.1	0.7	0.8	0.8	0	0	0
0375	G20CCC	3380-K	3990-3		0	876	292	0.3	292	0.3	0.0	0.0	0.1	0.7	0.8	0.8	0	0	0
0376	WRKLB4	3380-K	3990-3		0	876	292	0.3	292	0.3	0.0	0.0	0.1	0.7	0.8	0.8	0	0	0
0The table above contains the top				40 of	146 items.														
Sum/Mean					1	127884	25284	28.9	26903	3o.7	0.1	0.0	2.5	2.5	5.0	5.1	0.0	0.2	5

Figure 112. DASD_BY_ACTIVITY

RMF Interval Reports

Figure 113 and Figure 114 are RMF reports are used in the testing example. See “Performance Example” on page 109 to see the context of how they are used.

```
1                                                  RMF MONITOR II HARDCOPY LOG
-                                                       SESSION NAME BB
-            MVS/ESA                  SYSTEM ID MVSQ             DATE 02/20/94
             SP4.2.2                  RPT VERSION 4.2.2          TIME 22.53.01
0
     +++++++ TOTAL MODE ++++++++++++++++++++++++++ ARD      ++++++++++++++++++++
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++ 22.53.01 ++++++++++++++++++++
...
 ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++ 22.53.42 ++++++++++++++++++++
 22:53:42 DEV    FF PRIV LSQA LSQA X SRM  TCB    CPU   EXCP SWAP LPA CSA NVI V&H
 JOBNAME  CONN  BEL   FF  CSF  ESF M ABS  TIME   TIME  RATE RATE  RT  RT  RT  RT
 *MASTER* 26.79   0   28   70    0   0.0  11.28  18.89 0.00 0.00 0.0 0.0 0.0 0.0
 PCAUTH   0.000   0    2   21    0 X 0.0   0.03   0.03 0.00 0.00 0.0 0.0 0.0 0.0
 RASP     0.000 --- ---- ---- ---- X 0.0   0.02   0.04 0.00 0.00 0.0 0.0 0.0 0.0
 TRACE    0.000   0    3 1041    0 X 0.0   0.01   0.01 0.00 0.00 0.0 0.0 0.0 0.0
 XCFAS    0.082   0   21  213    0 X 0.0   1.49   1.64 0.00 0.00 0.0 0.0 0.0 0.0
 GRS      0.000   0   21   27    0 X 0.0   0.03   0.04 0.00 0.00 0.0 0.0 0.0 0.0
 SMXC     0.000   0    2   16    0   0.0   0.00   0.00 0.00 0.00 0.0 0.0 0.0 0.0
 SYSBMAS  0.000   0    8   79    0   0.0   0.01   0.01 0.00 0.00 0.0 0.0 0.0 0.0
 DUMPSRV  1.046   0    2   36    0   0.0   0.23   0.24 0.00 0.00 0.0 0.0 0.0 0.0
 CONSOLE  3.243   0    2   28    0 X 0.0   4.22   4.48 0.00 0.00 0.0 0.0 0.0 0.0
 ALLOCAS  0.000   0    2   58    0 X 0.0   0.04   0.04 0.00 0.00 0.0 0.0 0.0 0.0
 SMF      12.65   0    2   35    0 X 0.0   0.56   1.08 0.00 0.00 0.0 0.0 0.0 0.0
 LLA      4.373   7    9   44    1 X 0.0   1.40   1.52 0.00 0.00 0.0 0.0 0.0 0.0
 JES2     122.9  21   29   66    1   0.0  24.30  27.02 3.80 0.00 0.0 0.0 0.0 0.0
 NETTCP   5.573   1   15   51    1   0.0  13.29  14.41 0.00 0.00 0.0 0.0 0.0 0.0
 IOSAS    0.631   0    3   25    0   0.0   1.11   1.12 0.00 0.00 0.0 0.0 0.0 0.0
 TNF      0.169   0    2   16    0 X 0.0   0.01   0.01 0.00 0.00 0.0 0.0 0.0 0.0
 VMCF     0.310   0    2   16    0 X 0.0   0.02   0.03 0.00 0.00 0.0 0.0 0.0 0.0
 CATALOG  10.94   0    2  162    0 X 0.0   4.29   4.40 0.00 0.00 0.0 0.0 0.0 0.0
 VLF      0.231   0   10   31    1 X 0.0   0.23   0.26 0.00 0.00 0.0 0.0 0.0 0.0
 TCP22R   32.22   0   13   43    2   0.0  63.99  68.42 0.00 0.00 0.0 0.0 0.0 0.0
 FTPSRV6  0.648   0    2   30    2   0.0   0.20   0.21 0.00 0.00 0.0 0.0 0.0 0.0
 FTPSRV7  0.720   0    2   30    2   0.0   0.19   0.20 0.00 0.00 0.0 0.0 0.0 0.0
 FTPSRV8  0.598   0    2   30    2   0.0   0.19   0.20 0.00 0.00 0.0 0.0 0.0 0.0
 FTPSRV9  50.46   0    2   31    2   0.0  17.46  18.25 3.40 0.00 0.0 0.0 0.0 0.0
 RMFL     0.390   0    2   35    1   0.0  68.95  76.64 0.00 0.00 0.0 0.0 0.0 0.0
 ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++ 22.53.47 ++++++++++++++++++++
...
```

Figure 113. RMF Interval Report Time Stamped 22:53:42

```
...
 ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++ 22.56.23 ++++++++++++++++++++
 22:56:23 DEV    FF PRIV LSQA LSQA X SRM  TCB    CPU   EXCP SWAP LPA CSA NVI V&H
 JOBNAME  CONN  BEL   FF  CSF  ESF M ABS  TIME   TIME  RATE RATE  RT  RT  RT  RT
 *MASTER* 26.94   0   28   70    0   0.0  11.29  19.14 0.20 0.00 0.0 0.0 0.0 0.0
 PCAUTH   0.000   0    2   21    0 X 0.0   0.03   0.03 0.00 0.00 0.0 0.0 0.0 0.0
 RASP     0.000 --- ---- ---- ---- X 0.0   0.02   0.04 0.00 0.00 0.0 0.0 0.0 0.0
 TRACE    0.000   0    3 1041    0 X 0.0   0.01   0.01 0.00 0.00 0.0 0.0 0.0 0.0
 XCFAS    0.082   0   21  213    0 X 0.0   1.54   1.69 0.00 0.00 0.0 0.0 0.0 0.0
 GRS      0.000   0   21   27    0 X 0.0   0.03   0.04 0.00 0.00 0.0 0.0 0.0 0.0
 SMXC     0.000   0    2   16    0   0.0   0.00   0.00 0.00 0.00 0.0 0.0 0.0 0.0
 SYSBMAS  0.000   0    8   79    0   0.0   0.01   0.01 0.00 0.00 0.0 0.0 0.0 0.0
 DUMPSRV  1.046   0    2   36    0   0.0   0.23   0.24 0.00 0.00 0.0 0.0 0.0 0.0
 CONSOLE  3.316   0    2   28    0 X 0.0   4.24   4.50 0.20 0.00 0.0 0.0 0.0 0.0
 ALLOCAS  0.000   0    2   58    0 X 0.0   0.04   0.04 0.00 0.00 0.0 0.0 0.0 0.0
 SMF      13.38   0    2   35    0 X 0.0   0.56   1.14 0.00 0.00 0.0 0.0 0.0 0.0
 LLA      4.373   7    9   44    1 X 0.0   1.40   1.52 0.00 0.00 0.0 0.0 0.0 0.0
 JES2     126.2  21   29   66    1   0.0  24.87  27.66 3.40 0.00 0.0 0.0 0.0 0.0
 NETTCP   5.573   1   15   51    1   0.0  13.35  14.49 0.00 0.00 0.0 0.0 0.0 0.0
 IOSAS    0.631   0    3   25    0   0.0   1.11   1.12 0.00 0.00 0.0 0.0 0.0 0.0
 TNF      0.169   0    2   16    0 X 0.0   0.01   0.01 0.00 0.00 0.0 0.0 0.0 0.0
 VMCF     0.310   0    2   16    0 X 0.0   0.02   0.03 0.00 0.00 0.0 0.0 0.0 0.0
 CATALOG  10.97   0    2  162    0 X 0.0   4.35   4.46 0.00 0.00 0.0 0.0 0.0 0.0
 VLF      0.231   0   10   31    1 X 0.0   0.24   0.26 0.00 0.00 0.0 0.0 0.0 0.0
 TCP22R   42.51   0   13   43    2   0.0  87.14  93.06 0.00 0.00 0.0 0.0 0.0 0.0
 FTPSRV6  0.648   0    2   30    2   0.0   0.20   0.21 0.00 0.00 0.0 0.0 0.0 0.0
 FTPSRV7  0.720   0    2   30    2   0.0   0.19   0.20 0.00 0.00 0.0 0.0 0.0 0.0
 FTPSRV8  0.598   0    2   30    2   0.0   0.19   0.20 0.00 0.00 0.0 0.0 0.0 0.0
 FTPSRV9  70.04   0    2   31    2   0.0  27.29  28.35 11.6 0.00 0.0 0.0 0.0 0.0
 RMFL     0.390   0    2   35    1   0.0  75.52  83.89 0.00 0.00 0.0 0.0 0.0 0.0
 ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++ 22.56.28 ++++++++++++++++++++
...
```

Figure 114. RMF Interval Report Time Stamped 22:56:23

Appendix C. IBM's TCP/IP V3R2 Performance Improvements

V3R2 Performance Improvements

- IBM TCP/IP, Version 3 Release 2, for MVS exploits MVS/ESA architecture.
- IBM TCP/IP, Version 3 Release 2, for MVS includes enhanced interfaces which provide significant performance improvement for IBM supplied TCP/IP applications such as FTP Server, socket applications (CICS, IMS) and user written C-Socket applications to the HPNS interface. IBM TCP/IP V3R2 provides significant reduction in CPU utilization and increased throughput. Other TCPIP enhancements will also reduce CPU cycles for all TCP/IP applications.
  - Pathlength Reductions
  - Reduced CPU Consumption
  - Reduced Context Switches
  - Improved Throughput
  - Improved Serialization
  - Limited Parameter List Validations
  - Limited MVS Redispatches to Complete a Socket Request
  - Improved Table Lookups
  - Code Runs Authorized, Key 6
  - Applications are Not Non-Swappable
  - Checksum Improvements
  - Fast I/O path for FTP server
  - All Data Movements Occur Under Application Task
  - More TCP/IP processing Runs at Application Priority
  - *Telnet Improvements*
  - Fully Exploits MVS ESA Architecture
  - Improved sockets usage by C-FTP server
  - Improved sockets library
- All the performance enhancements, available to TCP/IP V3R1 for MVS users through installation of the performance PTFs, are integrated in TCP/IP V3R2.
- IBM Applications Enabled for MVS TCP/IP V3R2 (High Performance Native Sockets)
  - C File Transfer Protocol (FTP) Server
  - LPR Client
  - MISCSERV
  - Kerberos
  - NCPROUTE
  - RouteD Server

- PortMapper (Sun RPC) Server
- NCS
- NDB Client/Server
- SNMP Agent and Monitor
- REXEC Server
- X Window Client
- RPC
- ADSM Client & Server

IBM TCPIP V3R2 Performance Summary

Table 79. TPUT and CPU comparisons between TCP/IP V3R1, V3R1+ and V3R2 for MVS

	V3R1 vs V3R2		V3R1+ vs V3R2	
Application	**TPUT Improve-ment**	**S/390 CPU Reduction (TCPIP/ CFTP/ USERx)**	**TPUT Improve-ment**	**S/390 CPU Reduction (TCPIP/ CFTP/ USERx)**
C-FTP (Bin Put)	up to 30%	48%	equal	35%
C-FTP (BIN Get)	equal	53%	equal	32%
C-FTP (Asc Put)	up to 30%	40%	equal	25%
C-FTP (Asc Get)	equal	61%	equal	44%
C-Sockets TCP (MVS Send)	up to 265%	55 to 70%	up to 181%	38 to 61%
C-Sockets TCP (MVS Recv)	up to 130%	53 to 67%	up to 100%	40 to 63%
C-Sockets UDP (MVS Send)	up to 18%	50 to 66%	equal	43 to 56%
C-Sockets UDP (MVS Recv)	up to 128%	59 to 71%	up to 128%	50 to 65%
Telnet (TN3270)	equal	38%	equal	22%

System Setup/Parameters Used for Benchmarks

Performance Setup:

C-FTP Server filesize = 20 MB, Seq DS, Recfm: FB, Lrecl: 64, Blksize: 23424,
Lan: 16 Mb Token Ring, MTU: 2000.
PUT: WS --> MVS
Get: WS (/dev/null) <-- MVS

Telnet TN3270 Transaction workload: 100 Bytes IN and 800 bytes OUT from MVS Host

C-Sockets TCP sockets, Message Sizes: 128, 1024, 4096, 8192 bytes.
Lan: FDDI, MTU: 4352.

Client RS/6000 WS

Server 9021-982 (2 CP LPAR), MVS 5.2.2.

IBM TCP/IP Versions Used for Benchmarks

- V3R1 - Measurements do not include Performance PTFs
- V3R1+ - Measurements include the following Performance PTFs
  The set includes following PTFs plus prerequisites:
  - PN64344/UN69518 Telnet Scan Interval Option
  - PN67848/UN77980 Storage Management
  - PN68920/UN77994 Telnet LU Lookup
  - PN69060/UN75573 VTAM Interface
  - PN69881/UN75804 ACB Release
  - PN70195/UN76873 Dynamic TCP Control Blocks
  - PN70330/UN77979 IRB Scheduling
  - PN70420/UN77871 ESA Datamoves
  - PN68607/UN75139 Control Block Allocate
  - PN72737/UN79157 SCB Allocation
  - PN67770/UN90824 C-FTP performance Improvements
- V3R2 - V3R2 GA level code used for measurements

FTP Performance Improvements for V3R2

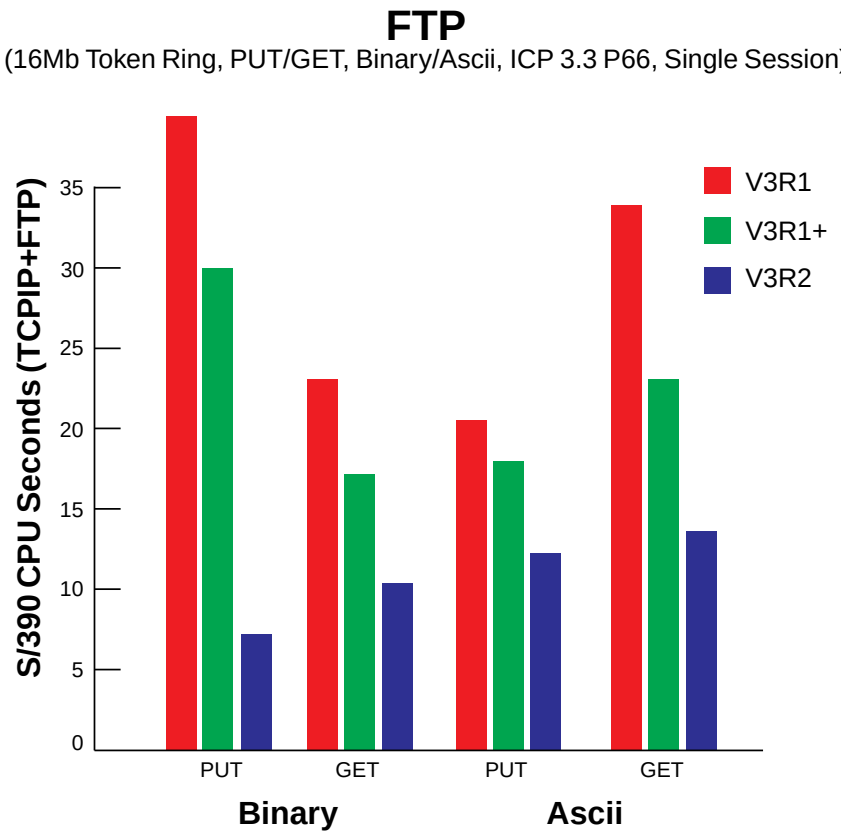


Figure 115. FTP performance comparisons of V3R1, V3R1+ and V3R2.

Host Processor: ES9021-982 (2CP LPAR)
3172-3: Pentium 66Mhz (ICP V3.3., ESCON,TR)
S/390 CPU is the sum of TCP/IP and FTP address spaces
Workstation: RISC/6000 Model 530h
PUT: RISC/6000 Hard Disk ----> Host (3380) DASD
GET: Host (3380) Cache DASD ---> RISC/6000 /dev/null
Packet Size: 2000 bytes
Host Data Buffer Size: 32 Kbytes (ICP)
W/S Data Buffer Size(s): 32 Kbytes
File size used: 20 Mbytes
Transfer type: Binary/ASCII, Five Transfers
IBM TCP/IP Versions: V3R1, V3R1+ and V3R2 for MVS

Single session FTP was used for the Host CPU comparisons. TCP/IP V3R2 reduces S/390 CPU cycles (47.8- 52.9%) for binary and (39.9-61%) for ASCII transfer compared to V3R1. Throughput is equivalent for both V3R1 and V3R2.

IBM TCP/IP V3R2 reduces S/390 CPU cycles (31.7- 34.8%) for binary and (25.5-44.2%) for ASCII transfer compared to V3R1+. Throughput is equivalent for both V3R1+ and V3R2.

Telnet Performance Improvements for V3R2

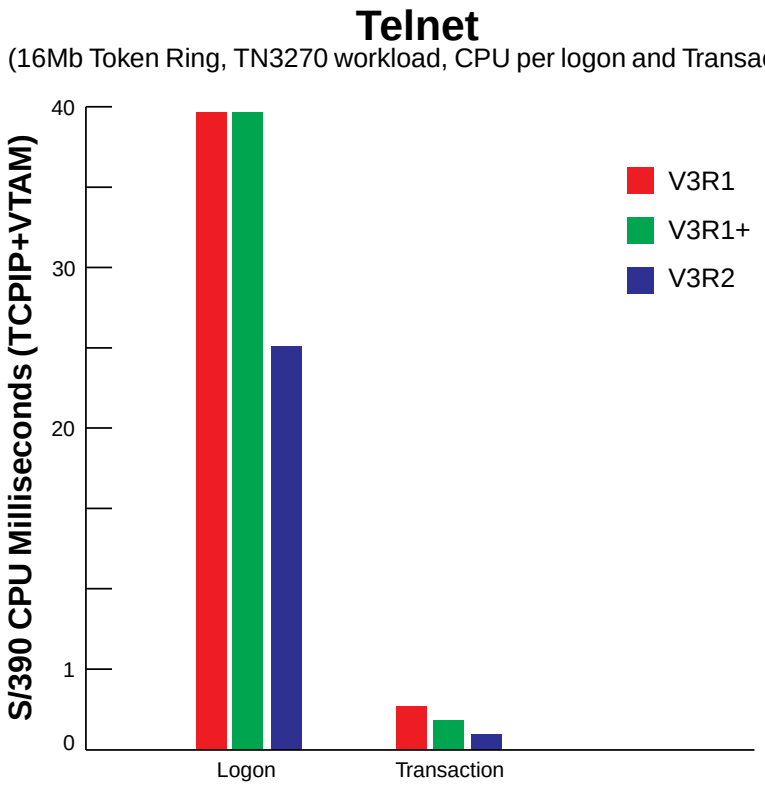


Figure 116. Telnet performance comparisons of V3R1, V3R1+ and V3R2.

Host: ES9021-982 (2 CP LPAR)
IBM TCP/IP Versions: V3R1, V3R1+ and V3R2 for MVS
3172-3: Pentium 66Mhz (ICP 3.3, ESCON,16Mb TR)
S/390 CPU is the sum of TCPIP and VTAM address spaces
Transaction Workload: 100 bytes sent, 800 bytes received from Host

When compared with TCP/IP V3R1, TCP/IP V3R2 reduces S/390 CPU costs per telnet transaction by 38%.

When compared with TCP/IP V3R1+, TCP/IP V3R2 reduces S/390 CPU costs per telnet logon by 36% and by 22% for Steady state transaction.

V3R2 Performance (TCP C-Sockets, MVS Send)

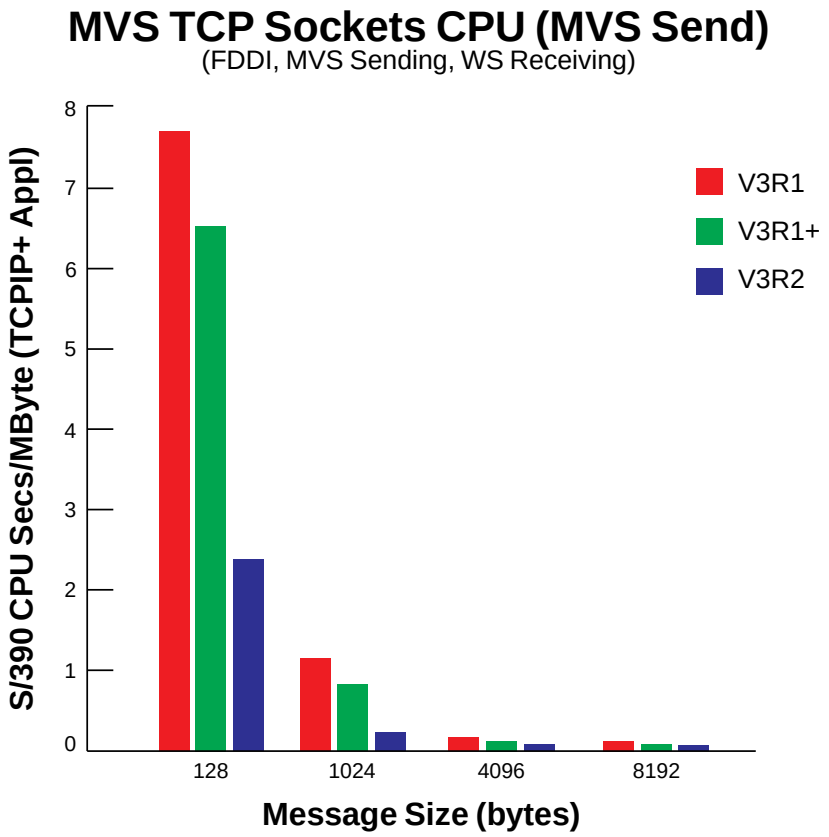


Figure 117. Performance comparisons of CPU cycles for TCP sockets when MVS is sending.

IBM TCP/IP Versions: V3R1, V3R1+ and V3R2 for MVS
Host Processor: ES9021-982 (2 CP LPAR)
W/S (Receiver): RISC/6000 M530

Message Size(s): 128, 1024, 4096 and 8192 bytes
Packet Size: 4352 bytes
3172-3: ICP, Pentium 66MHz, ESCON channel
S/390 CPU is the sum of TCPIP and User address spaces

When compared with TCP/IP V3R1, TCP/IP V3R2 reduces S/390 CPU cycles for TCP sockets by 55-70% for MVS Send; throughput is improved by 265%.

When compared with TCP/IP V3R1+, TCP/IP V3R2 reduces S/390 CPU cycles for TCP sockets by 38-61% for MVS Send; throughput is improved by 181%.

V3R2 Performance (TCP C-Sockets, MVS Rcv)

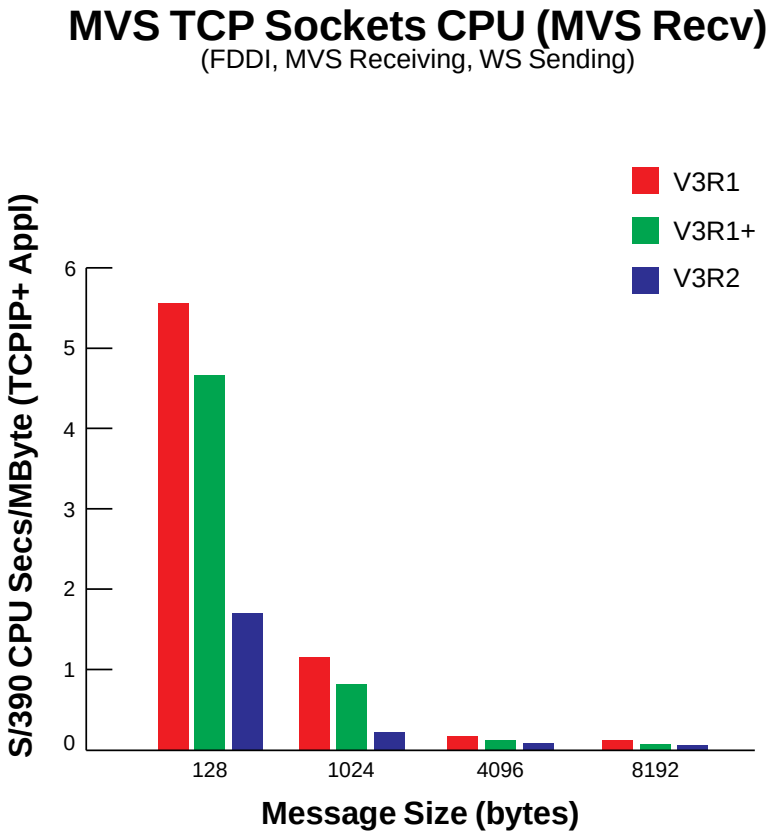


Figure 118. Performance comparisons of CPU cycles for TCP sockets when MVS is receiving.

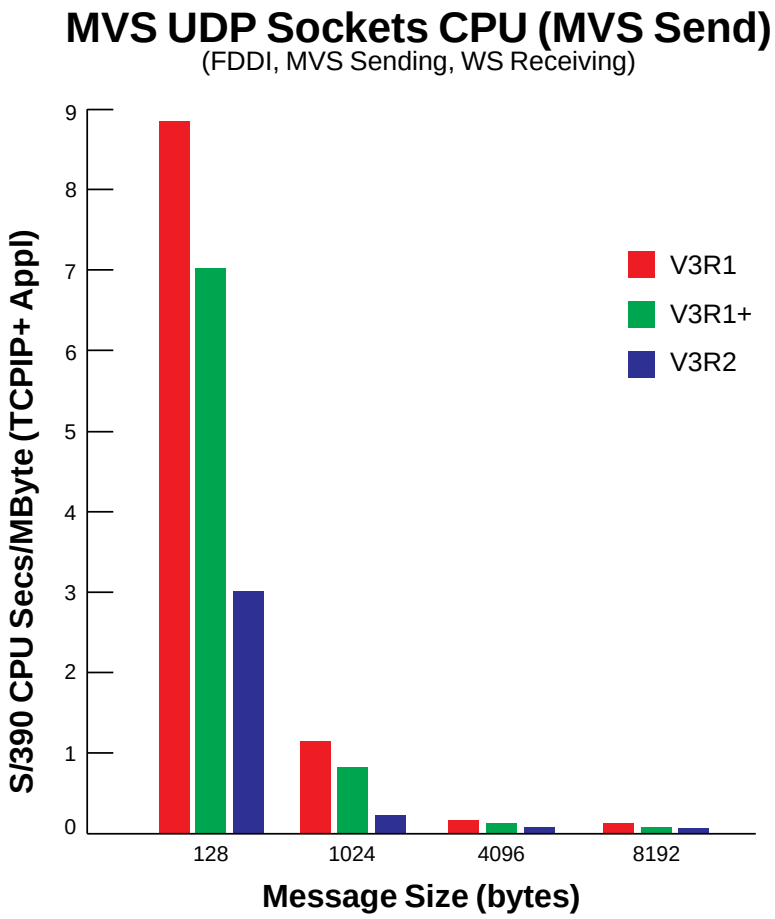
IBM TCP/IP Versions: V3R1, V3R1+ and V3R2 for MVS
Host Processor: ES9021-982 (2 CP LPAR)
W/S (Sender): RISC/6000 M530

Message Size(s): 128, 1024, 4096 and 8192 bytes
Packet Size: 4352 bytes
3172-3: ICP, Pentium 66MHz, ESCON channel
S/390 CPU is the sum of TCPIP and User address spaces

When compared with TCP/IP V3R1, TCP/IP V3R2 reduces S/390 CPU cycles for TCP sockets by 53-67% when MVS is the Receiver; throughput is improved by 130%.

When compared with TCP/IP V3R1+, TCP/IP V3R2 reduces S/390 CPU cycles for TCP sockets by (40-63%) when MVS is the Receiver; throughput is improved by 100%.

V3R2 Performance (UDP C-Sockets, MVS Send)



| *Figure 119. Performance comparisons of CPU cycles for UDP sockets when MVS is sending.*

| IBM TCP/IP Versions: V3R1, V3R1+ and V3R2 for MVS
| Host Processor: ES9021-982 (2 CP LPAR)
| W/S (Receiver): RISC/6000 M530

| Message Size(s): 128, 1024, 4096 and 8192 bytes
| Packet Size: 4352 bytes
| 3172-3: ICP, Pentium 66MHz, ESCON channel
| S/390 CPU is the sum of TCPIP and User address spaces

| To avoid UDP packet loss, delays were added between Sending Packets

| When compared with TCP/IP V3R1, TCP/IP V3R2 reduces S/390 CPU cycles for
| UDP sockets by 50-66% for MVS Send; throughput is improved up to 18%.

| When compared with TCP/IP V3R1+, TCP/IP V3R2 reduces S/390 CPU cycles for
| UDP sockets by 43-56% for MVS Send; throughput is equivalent for both V3R2 and
| V3R1+.

V3R2 Performance (UDP C-Sockets, MVS Rcv)

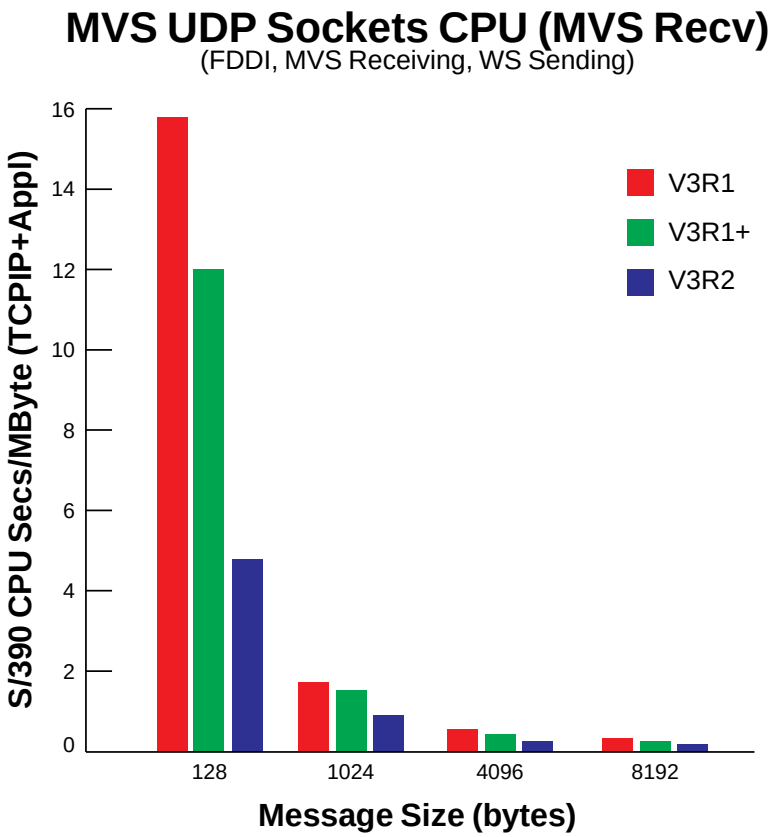


Figure 120. Performance comparisons of CPU cycles for UDP sockets when MVS is receiving.
..

IBM TCP/IP Versions: V3R1, V3R1+ and V3R2 for MVS
Host Processor: ES9021-982 (2 CP LPAR)
W/S (Sender): RISC/6000 M530

Message Size(s): 128, 1024, 4096 and 8192 Bytes
Packet Size: 4352 bytes
3172-3: ICP, Pentium 66MHz, ESCON channel
S/390 CPU is the sum of TCPIP and User address spaces
To avoid UDP packet loss, delays were added between Sending Packets

When compared with TCP/IP V3R1, TCP/IP V3R2 reduces S/390 CPU cycles for UDP sockets by 59-71% when MVS is the Receiver; throughput is improved by 128%.

When compared with TCP/IP V3R1+, TCP/IP V3R2 reduces S/390 CPU cycles for UDP sockets by 50-65% when MVS is the Receiver; throughput is improved by 128%.

Summary of Performance Improvements for V3R2

Performance Improvements (V3R1 to V3R2)

- FTP Server
  - TCP/IP V3R2 reduces S/390 CPU cycles by 48-53% for binary transfer and improves throughput for binary PUT compared to V3R1.
  - TCP/IP V3R2 reduces S/390 CPU cycles by 40-61% for ASCII transfer and improves throughput for ASCII PUT compared to V3R1.
- Telnet TN3270
  - TCP/IP V3R2 reduces S/390 CPU cycles by 38% for steady state telnet Transaction compared to V3R1.
- C-Sockets
  - TCP/IP V3R2 reduces S/390 CPU cycles by (53-70%) for TCP C-Socket Send/Receive and improves throughput up to 265%.
  - TCP/IP V3R2 reduces S/390 CPU cycles by (50-71%) for UDP C-Socket Send/Receive and improves throughput up to 128%.

Performance Improvements (V3R1+ to V3R2)

- FTP Server
  - TCP/IP V3R2 reduces S/390 CPU cycles by 32-35% for binary transfer and maintains equivalent throughput.
  - TCP/IP V3R2 reduces S/390 CPU cycles by 25-44% for ASCII transfer and maintains equivalent throughput.
- Telnet TN3270
  - TCP/IP V3R2 reduces S/390 CPU cycles by 36% for Telnet Logon and by 22% for telnet Transaction.
- C-Sockets
  - TCP/IP V3R2 reduces S/390 CPU cycles by 38-63% for TCP C-Socket Send/Receive and improves throughput up to 181%.
  - TCP/IP V3R2 reduces S/390 CPU cycles by 43-65% for UDP C-Socket Send/Receive and improves throughput up to 128%.

    Performance Note: V3R1+ includes Performance PTFs

Appendix D. Notices

References in this publication to IBM products, programs, or services do not imply that IBM intends to make them available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Subject to IBM's valid intellectual property or other legally protectable rights, any functionally equivalent product, program, or service may be used instead of the IBM product, program, or service. The evaluation and verification of operation in conjunction with other products, except those expressly designated by IBM, are the responsibility of the user.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
500 Columbus Avenue
Thornwood, NY 10594
USA

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

Site Counsel
IBM Corporation
P.O. Box 12195
3039 Cornwallis Road
Research Triangle Park, NC 27709-2195
USA

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement.

This document is not intended for production use and is furnished as is without any warranty of any kind, and all warranties are hereby disclaimed including the warranties of merchantability and fitness for a particular purpose.

Trademarks

The following terms are trademarks of the IBM Corporation in the United States or other countries or both:

	LANStreamer
ACF/VTAM	Library Reader
AD/Cycle	MVS/ESA
AIX	MVS/SP
AIX/ESA	MVS/XA
BookManager	NetView
C/370	OpenEdition
CICS	OS/2
DB2	OS/390
DFSMS	PS/2
DFSMS/MVS	RACF
ESCON	RISC System/6000
ES/9000	RS/6000
EtherStreamer	SAA
Extended Services	System/370
GDDM	System/390
Hardware Configuration Definition	VTAM
IBM	3090

The following terms are trademarks of other companies:

UNIX is a registered trademark in the United States and other countries licensed exclusively through X/Open Company Limited.

Microsoft, Windows, and the Windows 95 logo are trademarks or registered trademarks of Microsoft Corporation.

Other company, product, and service names, which may be denoted by a double asterisk (**), may be trademarks or service marks of others.

Part 4. Glossary, Bibliography, and Index

Glossary

This glossary includes terms and definitions from:

- The *American National Standards Dictionary for Information Systems*, ANSI X3.172-1990, copyright 1990 by the American National Standards Institute (ANSI). Copies may be purchased from the American National Standards Institute, 11 West 42nd Street, New York, New York 10036. Definitions are identified by the symbol (A) after the definition.
- The ANSI/EIA Standard—440-A, *Fiber Optic Terminology*. Copies may be purchased from the Electronic Industries Association, 2001 Pennsylvania Avenue, N.W., Washington, DC 20006. Definitions are identified by the symbol (E) after the definition.
- The *Information Technology Vocabulary*, developed by Subcommittee 1, Joint Technical Committee 1, of the International Organization for Standardization and the International Electrotechnical Commission (ISO/IEC JTC1/SC1). Definitions of published parts of this vocabulary are identified by the symbol (I) after the definition; definitions taken from draft international standards, committee drafts, and working papers being developed by ISO/IEC JTC1/SC1 are identified by the symbol (T) after the definition, indicating that final agreement has not yet been reached among the participating National Bodies of SC1.
- The *IBM Dictionary of Computing*, New York: McGraw-Hill, 1994.
- Internet Request for Comments: 1208, *Glossary of Networking Terms*.
- Internet Request for Comments: 1392, *Internet Users' Glossary*.
- The *Object-Oriented Interface Design: IBM Common User Access Guidelines*, Carmel, Indiana: Que, 1992.

The following cross-references are used in this glossary:

Contrast with: This refers to a term that has an opposed or substantively different meaning.

Synonym for: This indicates that the term has the same meaning as a preferred term, which is defined in its proper place in the glossary.

Synonymous with: This is a backward reference from a defined term to all other terms that have the same meaning.

See: This refers the reader to multiple-word terms that have the same last word.

See also: This refers the reader to terms that have a related, but not synonymous, meaning.

Deprecated term for: This indicates that the term should not be used. It refers to a preferred term, which is defined in its proper place in the glossary.

A

abend. Abnormal end of task; the termination of a task before its completion because of an error condition that cannot be resolved by recovery facilities while the task is executing.

abnormal end of task. See *abend*.

abstract syntax. A data specification that includes all distinctions that are needed in data transmissions, but that omits (abstracts) other details such as those that depend on specific computer architectures. See also *abstract syntax notation 1 (ASN.1)* and *basic encoding rules (BER)*.

abstract syntax notation 1 (ASN.1). The Open Systems Interconnection (OSI) method for abstract syntax specified in the following standards:

- ITU-T Recommendation X.208 (1988) | ISO/IEC 8824: 1990
- ITU-T Recommendation X.680 (1994) | ISO/IEC 8824-1: 1994

See also *basic encoding rules (BER)*.

accelerator. In the AIXwindows Toolkit, a keyboard alternative to a mouse button action; for example, holding the <Shift> and <M> keys on the keyboard can be made to post a menu in the same way that a mouse button action does. Accelerators typically provide increased input speed and greater convenience.

action bar. Deprecated term for *menu bar*.

active gateway. A gateway that is treated like a network interface in that it is expected to exchange routing information. If it does not do so for a period of time, the route associated with the gateway is deleted.

active open. In the TCP/UDP socket interface, the state of a connection that is providing a service. Contrast with *passive open*.

adapter. A part that electrically or physically connects a device to a computer or to another device.

address. In data communication, the unique code assigned to each device, workstation, or user connected to a network.

address class. In Internet communications, the categorization by the part of an IP address that distinguishes the network address from the host address. Class A addresses allocate 7 bits to the network ID and 24 bits to the host ID. Class B addresses allocate 14 bits to the network ID and 16 bits to the host ID. Class C addresses allocate 21 bits to the network ID and 8 bits to the host ID. Class D addresses contain 1110 in the first 4 bits and identify the address as a multicast. The remaining 28 bits in the class D address specify a particular multicast group.

address mask. For internet subnetworking, a 32-bit mask used to identify the subnetwork address bits in the host portion of an IP address. Synonymous with *subnet mask* and *subnetwork mask*.

address resolution. A method for mapping network-layer addresses to media-specific addresses.

Address Resolution Protocol (ARP). (1) In the Internet suite of protocols, the protocol that dynamically maps an IP address to an address used by a supporting metropolitan or local area network such as Ethernet or token-ring. (2) See also *Reverse Address Resolution Protocol (RARP)*.

addressing. In data communication, the way in which a station selects the station to which it is to send data.

Administrative Domain. A collection of hosts and routers, and the interconnecting networks, managed by a single administrative authority.

advanced program-to-program communication (APPC). (1) The general facility characterizing the LU 6.2 architecture and its various implementations in products. (2) Sometimes used to refer to the LU 6.2 architecture and its product implementations as a whole, or to an LU 6.2 product feature in particular, such as an APPC application programming interface.

Advanced Research Projects Agency (ARPA). The United States Department of Defense agency responsible for creating ARPANET. This agency is now called the Defense Advanced Research Projects Agency (DARPA).

agent. (1) In systems management, a user that, for a particular interaction, has assumed an agent role. (2) An entity that represents one or more managed objects by (a) emitting notifications regarding the objects and (b) handling requests from managers for management operations to modify or query the objects. (3) A system that assumes an agent role.

AIX. Advanced Interactive Executive.

AIX operating system. IBM's implementation of the UNIX operating system. The RISC System/6000 system, among others, runs the AIX operating system.

all-stations address. In communications, synonym for *broadcast address*.

American National Standards Institute (ANSI). An organization consisting of producers, consumers, and general interest groups, that establishes the procedures by which accredited organizations create and maintain voluntary industry standards in the United States. (A)

AND operation. Synonym for *conjunction*.

ANSI. American National Standards Institute.

API. Application programming interface.

APPC. Advanced program-to-program communication.

application. A collection of software components used to perform specific types of user-oriented work on a computer.

application layer. In the Open Systems Interconnection (OSI) reference model, the layer that provides means for application processes residing in open systems to exchange information and that contains the application-oriented protocols by which these processes communicate. (T) See *Open Systems Interconnection (OSI) reference model*.

application program. (1) A program written for or by a user that applies to the user's work, such as a program that does inventory control or payroll. (2) A program used to connect and communicate with stations in a network, enabling users to perform application-oriented activities.

application program interface. See *application programming interface (API)*.

application programming interface (API). (1) The set of programming language constructs or statements that can be coded in an application program to obtain the specific functions and services provided by an underlying operating system or service program. (2) In VTAM, the language structure used in control blocks so that application programs can reference them and be identified to VTAM.

argument. A parameter passed between a calling program and a called program.

ARP. Address Resolution Protocol.

ARPA. Advanced Research Projects Agency.

ARPANET. A network established by the United States Department of Defense Advanced Research Projects Agency (now the Defense Advanced Research Projects Agency).

ASCII (American National Standard Code for Information Interchange). The standard code, using a coded character set consisting of 7-bit coded characters (8 bits including parity check), that is used for information interchange among data processing systems, data communication systems, and associated equipment. The ASCII set consists of control characters and graphic characters. (A)

ASN.1. Abstract syntax notation 1.

ASYNC. Asynchronous.

asynchronous (ASYNC). (1) Pertaining to two or more processes that do not depend upon the occurrence of specific events such as common timing signals. (T) (2) Without regular time relationship; unexpected or unpredictable with respect to the execution of program instructions.

asynchronous communication. A method of communication supported by the operating system that allows an exchange of data with a remote device, using either a start-stop line or an X.25 line. Asynchronous communication includes the file transfer support and the interactive terminal facility support.

Athena widgets. The X Window System widget set developed by the Massachusetts Institute of Technology (MIT) for Project Athena.

attachment unit interface (AUI). In a local area network, the interface between the medium attachment unit and the data terminal equipment within a data station. (I) (A)

attention key. A function key on terminals that, when pressed, causes an I/O interruption in the processing unit.

attribute. (1) A characteristic that identifies and describes a managed object. The characteristic can be determined, and possibly changed, through operations on the managed object. (2) Information within a managed object that is visible at the object boundary. An attribute has a type, which indicates the range of information given by the attribute, and a value, which is within that range. (3) Variable data that is logically a part of an object and that represents a property of the object. For example, a serial number is an attribute of an equipment object.

AUI. Attachment unit interface.

authentication. (1) In computer security, verification of the identity of a user or the user's eligibility to access an object. (2) In computer security, verification that a message has not been altered or corrupted. (3) In computer security, a process used to verify the user of an information system or protected resources.

authorization. In computer security, the right granted to a user to communicate with or make use of a computer system. (T)

AUTOEXEC.BAT file. In the DOS operating system, a batch file that resides in the root directory of the boot drive. AUTOEXEC.BAT contains commands that DOS executes every time the PC is booted.

automated operator. In IMS/VS, an application program that can issue a subset of IMS/VS operator commands and receive status information on the execution of the commands.

B

backbone. (1) A set of nodes and their interconnecting links providing the primary data path across a network. (2) In a local area network multiple-bridge ring configuration, a high-speed link to which the rings are connected by means of bridges or routers. A backbone may be configured as a bus or as a ring. (3) In a wide area network, a high-speed link to which nodes or data switching exchanges (DSEs) are connected.

background task. A task that is running even though the user is not currently interacting with it. Contrast with *foreground task*.

backout. In IMS/VS, the process of removing all the database updates performed by an application program that has terminated abnormally.

baseband. (1) A frequency band occupied by a signal, or by a number of multiplexed signals. (T) (2) A frequency band that uses the complete bandwidth of a transmission.

basic encoding rules (BER). The rules specified in ISO 8825 for encoding data units described in abstract syntax notation 1 (ASN.1). The rules specify the encoding technique, not the abstract syntax.

Basic Input/Output System (BIOS). Code that controls basic hardware operations, such as interactions with diskette drives, hard disk drives, and the keyboard.

batch. Pertaining to activity involving little or no user action. Contrast with *interactive*.

batch message processing (BMP) program. In IMS/VS, a batch processing program that has access to online databases and message queues.

Bayonet Neill-Concelman (BNC). A standardized connector used with coaxial cable. Ethernet is an example of a network that uses these connectors.

Because It's Time Network (BITNET). A low-cost, low-speed network of hosts interconnected by nonswitched SDLC and BSC lines that was started at the City University of New York. The network is primarily composed of universities, nonprofit organizations, and research centers. BITNET has merged with the Computer Science Network (CSNET) to form the Consortium for Research and Education Network (CREN).

BER. Basic encoding rules.

Berkeley Software Distribution (BSD). Pertaining to any of the series of UNIX specifications or implementations distributed by the University of California at Berkeley. The mnemonic “BSD” is usually followed by a number to specify the particular version of UNIX that was distributed (for example, BSD 4.3). Many vendors use BSD specifications as standards for their UNIX products.

BGP. Border Gateway Protocol.

big endian. A format for storage or transmission of binary data in which the most significant bit (or byte) is placed first. Contrast with *little endian*.

binary synchronous communication (BSC). (1) A form of telecommunication line control that uses a standard set of transmission control characters and control character sequences, for binary synchronous transmission of binary-coded data between stations. (2) Contrast with *Synchronous Data Link Control (SDLC)*.

BIOD. Block input/output daemon.

BIOS. (1) Basic Input/Output System. (2) See also *NetBIOS*.

bit map. (1) A representation of an image by an array of bits. (2) A pixmap with a depth of one bit plane.

BITNET. Because It's Time Network.

block. A string of data elements recorded or transmitted as a unit. The elements may be characters, words, or physical records. (T)

block input/output daemon (BIOD). In the Network File System (NFS), a daemon that performs parallel read/write requests on behalf of an NFS client.

blocking mode. (1) A way of requesting a service over an interface so that if the request cannot be completed immediately, the requesting process is suspended until the request is completed. (2) Contrast with *nonblocking mode*.

BMP. Batch message processing.

BNC. Bayonet Neill-Concelman.

Boolean. (1) Pertaining to the processes used in the algebra formulated by George Boole. (A) (2) A value of 0 or 1 represented internally in binary notation.

Boolean operation. (1) Any operation in which each of the operands and the result take one of two values. (I) (A) (2) An operation that follows the rules of Boolean algebra. (I) (A)

boot. To prepare a computer system for operation by loading an operating system.

Border Gateway Protocol (BGP). An Internet Protocol (IP) routing protocol used between domains and autonomous systems. Contrast with *Exterior Gateway Protocol (EGP)*.

border router. In Internet communications, a router, positioned at the edge of an autonomous system, that communicates with a router that is positioned at the edge of a different autonomous system.

bridge. (1) A functional unit that interconnects two local area networks that use the same logical link control protocol but may use different medium access control protocols. (T) (2) A functional unit that interconnects multiple LANs (locally or remotely) that use the same logical link control protocol but that can use different medium access control protocols. A bridge forwards a frame to another bridge based on the medium access control (MAC) address. (3) In the connection of local loops, channels, or rings, the equipment and techniques used to match circuits and to facilitate accurate data transmission. (4) Contrast with *gateway* and *router*.

broadband. (1) A frequency band broad enough to be divided into several narrower bands, each of which can be used for different purposes or be made available to different users. Synonymous with *wideband*. (T) (2) A frequency band divisible into several narrower bands so that different kinds of transmission (such as voice, video, and data) can occur at the same time. Synonymous with *wideband*. See also *baseband*. (3) Transmission media and techniques that use a broad frequency range, divided into sub-bands of narrower frequency, so that different kinds of transmission can occur at the same time.

broadcast. (1) Transmission of the same data to all destinations. (T) (2) Simultaneous transmission of data to more than one destination. (3) Contrast with *multicast*.

broadcast address. In communications, a station address (eight 1's) reserved as an address common to all stations on a link. Synonymous with *all-stations address*.

BSC. Binary synchronous communication.

BSD. Berkeley Software Distribution.

buffer. A portion of storage used to hold input or output data temporarily.

bus. (1) A facility for transferring data between several devices located between two end points, only one device being able to transmit at a given moment. (T) (2) A computer configuration in which processors are interconnected in series.

bus network. (1) A local area network in which there is only one path between any two data stations and in which data transmitted by any station is concurrently available to all other stations on the same transmission medium. (2) A network configuration that provides a bidirectional transmission facility to which all nodes are attached. A sending node transmits in both directions to the ends of the bus. All nodes in the path copy the message as it passes.

Note: A bus network may be a linear network, a star network, or a tree network. In the case of a tree or star network, there is a data station at each endpoint node. There is no data station at an intermediate node; however, one or more devices such as repeaters, connectors, amplifiers, and splitters are located there. (T)

button. See *mouse button*, *push button*, *radio button*, and *spin button*.

byte ordering. The arrangement of bytes under specific machine architectures. Two common methods of byte ordering are big endian and little endian.

C

C language. A language used to develop software applications in compact, efficient code that can be run on different types of computers with minimal change.

carrier sense. In a local area network, an ongoing activity of a data station to detect whether another station is transmitting. (T)

carrier sense multiple access with collision detection (CSMA/CD). A protocol that requires carrier sense and in which a transmitting data station that detects another signal while transmitting, stops sending, sends a jam signal, and then waits for a variable time before trying again. (T) (A)

case-sensitive. Pertaining to the ability to distinguish between uppercase and lowercase letters.

CCITT. International Telegraph and Telephone Consultative Committee. This was an organization of the International Telecommunication Union (ITU). On 1 March 1993 the ITU was reorganized, and responsibilities for standardization were placed in a subordinate organization named the Telecommunication Standardization Sector of the International Telecommunication Union (ITU-TS). "CCITT" continues to be used for recommendations that were approved before the reorganization.

channel. (1) A path along which signals can be sent, for example, data channel, output channel. (A) (2) A functional unit, controlled by the processor, that handles the transfer of data between processor storage and local peripheral equipment.

channel-attached. (1) Pertaining to the attachment of devices directly by input/output channels to a host processor. (2) Pertaining to devices attached to a controlling unit by cables, rather than by telecommunication lines. (3) Contrast with *link-attached*. (4) Synonymous with *local*.

checkpoint. (1) A point at which information about the status of a job and the system can be recorded so that the job step can be later restarted. (2) To record such information.

checksum. (1) The sum of a group of data associated with the group and used for checking purposes. (T) (2) In error detection, a function of all bits in a block. If the written and calculated sums do not agree, an error is indicated. (3) On a diskette, data written in a sector for error detection purposes; a calculated checksum that does not match the checksum of data written in the sector indicates a bad sector. The data are either numeric or other character strings regarded as numeric for the purpose of calculating the checksum.

CICS. Customer Information Control System.

class A network. In Internet communications, a network in which the high-order (most significant) bit of the IP address is set to 0 and the host ID occupies the three low-order octets.

class B network. In Internet communications, a network in which the two high-order (most significant and next-to-most significant) bits of the IP address are set to 1 and 0, respectively, and the host ID occupies the two low-order octets.

class C network. In Internet communications, a network in which the two high-order (most significant and next-to-most significant) bits of the IP address are both set to 1 and the next high-order bit is set to 0. The host ID occupies the low-order octet.

CLAW. Common Link Access to Workstation.

click. To press and release a button on a pointing device without moving the pointer off of the object or choice.

client. (1) A functional unit that receives shared services from a server. (T) (2) A user. (3) Synonymous with *requester*.

client/server. In communications, the model of interaction in distributed data processing in which a program at one site sends a request to a program at another site and awaits a response. The requesting program is called a client; the answering program is called a server.

clipboard. An area of storage provided by the system to hold data temporarily.

closed system. A system whose characteristics comply with proprietary standards and that therefore cannot readily be connected to other systems. (T)

CLPA. Create link pack area.

CLUT. Color lookup table.

color lookup table (CLUT). Synonym for *color map*. In multimedia, synonym for *color palette*.

color map. (1) A lookup table in which each index is associated with a red, green, and blue value. Synonymous with *color lookup table (CLUT)* and *color table*. (2) A set of color cells. A pixel value indexes the color map to produce RGB-intensities. A color map consists of a set of entries defining color values that, when associated with a window, is used to display the contents of the window. (3) A lookup table that translates color indexes into RGB triplets.

color palette. A set of colors that can be displayed on the screen at one time. This can be a standard set used for all images or a set that can be customized for each image. Synonymous with *color lookup table (CLUT)*.

color table. Synonym for *color map*.

command. A request from a terminal for the performance of an operation or the execution of a particular program.

command interpreter. In the AIX operating system, a program that sends instructions to the kernel.

command name. The first term in a command, usually followed by operands.

command operator. In DOS or OS/2, a special character used for conditional processing or grouping, or to redirect input or output.

command prompt. A displayed character or string of characters that indicates that a user may enter a command to be processed.

Common Link Access to Workstation (CLAW). A continuously executing program designed to minimize host interrupts while maximizing channel utilization.

communication adapter. (1) A circuit card with associated software that enables a processor, controller, or other device to be connected to a network. (2) A mechanism that enables communication facilities to be attached to host processors.

Communications Manager/2. An IBM licensed program that supports the development and use of OS/2 applications involving two or more connected systems or workstations. Communications Manager/2 provides multiple concurrent connectivities using different protocols for

IBM 3270 and 5250 emulation sessions, printer sessions, and file transfers. It supports a range of application programming interfaces (APIs), which may be called concurrently and are designed for a variety of applications. Communications Manager/2 includes the necessary interfaces for network management.

community. In the Simple Network Management Protocol (SNMP), an administrative relationship between entities.

community name. In the Simple Network Management Protocol (SNMP), a string of octets identifying a community.

compile. (1) To translate all or part of a program expressed in a high-level language into a computer program expressed in an intermediate language, an assembly language, or a machine language. (T) (2) To prepare a machine language program from a computer program written in another programming language by making use of the overall logic structure of the program, or generating more than one computer instruction for each symbolic statement, or both, as well as performing the function of an assembler. (A) (3) To translate a source program into an executable program (an object program). (4) To translate a program written in a high-level programming language into a machine language program.

compiler. A program that translates a source program into an executable program (an object program).

Computer Science Network (CSNET). A large computer network, mostly in the United States but with international connections. CSNET sites include universities, research labs, and some commercial companies. CSNET has merged with the Because It's Time Network (BITNET) to form the Consortium for Research and Education Network (CREN).

computer word. A word suitable for processing by a given computer, usually treated as a unit. (T) Synonymous with *fullword*.

concurrent server. A server that can handle many connections at the same time. It can accept new connection requests while still processing the transactions started by previous requests. Synonymous with *multiconnection server*. Contrast with *iterative server*.

CONFIG.SYS file. In the OS/2 operating system, a file used by the base operating system that describes the devices, system parameters, and resource options of a workstation. See also *configuration file*.

configuration file. A file that specifies the characteristics of a system device or network.

conjunction. The Boolean operation whose result has the Boolean value 1 if and only if each operand has the Boolean value 1. (I) (A) Synonymous with *AND operation*.

connection. (1) In data communication, an association established between functional units for conveying information. (I) (A) (2) In Open Systems Interconnection architecture, an association established by a given layer between two or more entities of the next higher layer for the purpose of data transfer. (T) (3) In TCP/IP, the path between two protocol applications that provides reliable data stream delivery service. In the Internet, a connection extends from a TCP application on one system to a TCP application on another system.

Consortium for Research and Education Network (CREN). A large computer network formed from the merging of the Because It's Time Network (BITNET) and the Computer Science Network (CSNET).

control program. (1) A computer program designed to schedule and to supervise the execution of programs of a computer system. (I) (A) (2) The part of the AIX Base Operating System that determines the order in which basic functions should be performed.

create link pack area (CLPA). An option used during IPL to initialize the link pack pageable area.

CREN. Consortium for Research and Education Network.

CSMA/CD. Carrier sense multiple access with collision detection.

CSNET. Computer Science Network.

cursor. (1) A movable, visible mark used to indicate a position of interest on a display surface. (A) (2) A visible indication of the position where user interaction with the keyboard will appear. The keyboard cursors are the selection cursor and the text cursor.

Customer Information Control System (CICS). An IBM licensed program that enables transactions entered at remote terminals to be processed concurrently by user-written application programs. It includes facilities for building, using, and maintaining databases.

D

daemon. A program that runs unattended to perform a standard service. Some daemons are triggered automatically to perform their task; others operate periodically.

DARPA. Defense Advanced Research Projects Agency.

DASD. Direct access storage device.

data circuit-terminating equipment (DCE). In a data station, the equipment that provides the signal conversion and coding between the data terminal equipment (DTE) and the line. (I)

Notes:

1. The DCE may be separate equipment or an integral part of the DTE or of the intermediate equipment.
2. A DCE may perform other functions that are usually performed at the network end of the line.

data link layer. In the Open Systems Interconnection reference model, the layer that provides services to transfer data between entities in the network layer over a communication link. The data link layer detects and possibly corrects errors that may occur in the physical layer. (T)

data link switching (DLSw). A method of transporting network protocols that use IEEE 802.2 logical link control (LLC) type 2. SNA and NetBIOS are examples of protocols that use LLC type 2. See also *encapsulation* and *spoofing*.

data set. (1) Synonym for *file*. (2) Deprecated term for *modem*.

data stream. (1) All information (data and control commands) sent over a data link usually in a single read or write operation. (2) A continuous stream of data elements being transmitted, or intended for transmission, in character or binary-digit form, using a defined format.

data terminal equipment (DTE). That part of a data station that serves as a data source, data sink, or both. (I) (A)

database. (1) A collection of data with a given structure for accepting, storing, and providing, on demand, data for multiple users. (T) (2) A collection of interrelated data organized according to a database schema to serve one or more applications. (T) (3) A collection of data fundamental to a system. (A) (4) A collection of data fundamental to an enterprise. (A)

database administrator (DBA). A person who is responsible for a database system, particularly for defining the rules by which data are accessed and stored. The database administrator is usually responsible also for database integrity, security, performance, and recovery.

database record. In IMS/VS, a collection of segments that contains one occurrence of the root segment type and all of its dependents arranged in a hierarchical sequence. It may be smaller than, equal to, or larger than the access method logical record.

DATABASE 2 (DB2). An IBM relational database management system.

datagram. (1) In packet switching, a self-contained packet, independent of other packets, that carries information sufficient for routing from the originating data terminal equipment (DTE) to the destination DTE without relying on earlier exchanges between the DTEs and the network. (I) (2) In TCP/IP, the basic unit of information passed across the Internet environment. A datagram contains a source and destination address along with the data. An Internet Protocol (IP) datagram consists of an IP header followed by the transport layer data. (3) See also *packet* and *segment*.

DBA. Database administrator.

DBCS. Double-byte character set.

DB2. DATABASE 2.

DCE. (1) Data circuit-terminating equipment. (2) Distributed Computing Environment.

DDN. Defense Data Network.

decryption. In computer security, transforming encoded text or ciphertext into plaintext.

default. Pertaining to an attribute, condition, value, or option that is assumed when none is explicitly specified. (I)

Defense Advanced Research Projects Agency (DARPA). The United States Department of Defense agency responsible for creating ARPANET. This agency was formerly called the Advanced Research Projects Agency (ARPA).

Defense Data Network (DDN). MILNET and several other United States Department of Defense networks.

destination node. The node to which a request or data is sent.

destination service access point (DSAP). In SNA and TCP/IP, a logical address that allows a system to route data from a remote device to the appropriate communications support. Contrast with *source service access point (SSAP)*.

dialog box. In Advanced CUA architecture, a movable window, fixed in size, containing controls that a user uses to provide information required by an application so that it can continue to process a user request.

Note: In CUA architecture, this is a programmer term. The user term is *pop-up window*.

direct access storage. A storage device that provides direct access to data. (I) (A)

direct access storage device (DASD). A device in which access time is effectively independent of the location of the data.

directory. A table of identifiers and references to the corresponding items of data. (I) (A)

directory service (DS). An application service element that translates the symbolic names used by application processes into the complete network addresses used in an OSI environment. (T)

disk drive. (1) In an IBM personal computer, a diskette drive or a hard disk drive. (2) The mechanism used to seek, read, and write information on a disk.

disk operating system. An operating system for computer systems that use disks and diskettes for auxiliary storage of programs and data.

diskette. (1) A small magnetic disk enclosed in a jacket. (T) (2) A thin, flexible magnetic disk and a semi-rigid protective jacket, in which the disk is permanently enclosed.

display. (1) A visual presentation of data. (I) (A) (2) To present data visually. (I) (A)

display station. An input/output device containing a display screen and an attached keyboard that allows a user to send information to or receive information from the system. See also *terminal* and *workstation*.

Distributed Computing Environment (DCE). The Open Software Foundation (OSF) specification (or a product derived from this specification) that assists in networking. DCE provides such functions as authentication, directory service (DS), and remote procedure call (RPC).

distributed processing. Processing that takes place across two or more linked systems.

Distributed Protocol Interface (DPI). An interface between a Simple Network Management Protocol (SNMP) agent and its subagents that is defined in Request for Comments (RFC) 1592.

DLL. Dynamic link library.

DLSw. Data link switching.

DNS. Domain Name System.

document type definition (DTD). The rules, determined by an application, that apply SGML to the markup language of documents of a particular type. SGML provides the syntax for the markup language, and the DTD provides the vocabulary for the markup language.

domain. (1) That part of a computer network in which the data processing resources are under common control. (T) (2) See *Administrative Domain* and *domain name*.

domain name. In the Internet suite of protocols, a name of a host system. A domain name consists of a sequence of subnames separated by a delimiter character. For example, if the fully qualified domain name (FQDN) of a host system is `ralvm7.vnet.ibm.com`, each of the following is a domain name:

- `ralvm7.vnet.ibm.com`
- `vnet.ibm.com`
- `ibm.com`

domain name server. In the Internet suite of protocols, a server program that supplies name-to-address translation by mapping domain names to IP addresses. Synonymous with *name server*.

Domain Name System (DNS). In the Internet suite of protocols, the distributed database system used to map domain names to IP addresses.

DOS. Disk Operating System. See *IBM Disk Operating System*.

dotted decimal notation. The syntactical representation for a 32-bit integer that consists of four 8-bit numbers written in base 10 with periods (dots) separating them. It is used to represent IP addresses.

double-byte character set (DBCS). A set of characters in which each character is represented by 2 bytes. Languages such as Japanese, Chinese, and Korean, which contain more symbols than can be represented by 256 code points, require double-byte character sets.

Because each character requires 2 bytes, the typing, display, and printing of DBCS characters requires hardware and programs that support DBCS. Contrast with *single-byte character set (SBCS)*.

double-precision. A specification that causes a floating-point value to be stored internally in the long format.

doubleword. A contiguous sequence of bits or characters that comprises two computer words and is capable of being addressed as a unit. (A)

DPI. Distributed Protocol Interface.

DPI API. In the Simple Network Management Protocol (SNMP), a program interface for a subagent that provides an extension to the function provided by the SNMP agent.

drag. To use a pointing device to move an object. For example, a user can drag a window border to make the window larger.

drag and drop. To directly manipulate an object by moving it and placing it somewhere else using a pointing device.

DS. Directory service.

DSAP. Destination service access point.

DTD. Document type definition.

DTE. Data terminal equipment. (A)

dynamic. (1) In programming languages, pertaining to properties that can only be established during the execution of a program; for example, the length of a variable-length data object is dynamic. (I) (2) Pertaining to an operation that occurs at the time it is needed rather than at a predetermined or fixed time. (3) Contrast with *static*.

dynamic link library (DLL). A file containing executable code and data bound to a program at load time or run time, rather than during linking. The code and data in a dynamic link library can be shared by several applications simultaneously.

dynamic resource allocation. An allocation technique in which the resources assigned for execution of computer programs are determined by criteria applied at the moment of need. (I) (A)

E

EBCDIC. Extended binary-coded decimal interchange code. A coded character set of 256 8-bit characters.

EGP. Exterior Gateway Protocol.

EIA. Electronic Industries Association.

Electronic Industries Association (EIA). An organization of electronics manufacturers that advances the technological growth of the industry, represents the views of its members, and develops industry standards.

encapsulation. (1) In communications, a technique used by layered protocols by which a layer adds control information to the protocol data unit (PDU) from the layer it supports. In this respect, the layer encapsulates the data from the supported layer. In the Internet suite of protocols, for example, a packet would contain control information from the physical layer, followed by control information from the network layer, followed by the application protocol data. (2) See also *data link switching*.

encryption. In computer security, the process of transforming data into an unintelligible form in such a way that the original data either cannot be obtained or can be obtained only by using a decryption process.

Enhanced X-Windows Toolkit. (1) In the AIX operating system, a collection of basic functions for developing a variety of application environments. Toolkit functions manage Toolkit initialization, widgets, memory, events, geometry, input focus, selections, resources, translation of events, graphics contexts, pixmaps, and errors. (2) See also *AIXwindows Toolkit* and *X Window System*.

Enterprise Systems Connection (ESCON). A set of IBM products and services that provide a dynamically connected environment within an enterprise.

entry field. An area into which a user types or places text. Its boundaries are usually indicated.

envelope. (1) That part of a message containing information used in the submission, sending, or delivery of the message. (T) (2) A storage area used to hold packets, datagrams, or fragments during TCP/IP processing.

environment variable. A variable that specifies (a) how an operating system or another program will run or (b) the devices that the operating system will recognize.

ephemeral port number. In some TCP/IP implementations, a temporary port number assigned to a process for the duration of a call. Ephemeral port numbers are typically assigned to client processes that must provide servers with a client port number so that the server can respond to the correct process.

ESCON. Enterprise Systems Connection.

ESCON channel. A channel having an Enterprise Systems Connection channel-to-control-unit I/O interface that uses optical cables as a transmission medium. Contrast with *parallel channel*.

Ethernet. A 10-Mbps baseband local area network that allows multiple stations to access the transmission medium at will without prior coordination, avoids contention by using carrier sense and deference, and resolves contention by using collision detection and delayed retransmission. Ethernet uses carrier sense multiple access with collision detection (CSMA/CD).

extended character. A character other than a 7-bit ASCII character. An extended character can be a 1-byte code point with the eighth bit set (ordinal 128 through 255).

extended recovery facility (XRF). A facility that minimizes the effect of failures in MVS, VTAM, the host processor, or high availability applications during sessions between high availability applications and designated terminals. This facility provides an alternate subsystem to take over sessions from the failing subsystem.

exterior gateway. In Internet communications, a gateway on one autonomous system that communicates with another autonomous system. Contrast with *interior gateway*.

Exterior Gateway Protocol (EGP). In the Internet suite of protocols, a protocol, used between domains and autonomous systems, that enables network reachability information to be advertised and exchanged. IP network addresses in one autonomous system are advertised to another autonomous system by means of EGP-participating routers. Contrast with *Border Gateway Protocol (BGP)*.

eXternal Data Representation (XDR). A standard, developed by Sun Microsystems, Incorporated, for representing data in machine-independent format.

F

FAT. File allocation table.

FDDI. Fiber Distributed Data Interface.

Fiber Distributed Data Interface (FDDI). An American National Standards Institute (ANSI) standard for a 100-megabit-per-second LAN using optical fiber cables.

fiber optic network. A network based on the technology and standards that define data transmission using cables of glass or plastic fibers carrying light. The advantages of a fiber optic network are higher transmission speeds, greater carrying capacity, lower error rates, and lighter, more compact cables that are less susceptible to electromagnetic interference.

fiber optics. The branch of optical technology concerned with the transmission of radiant power through fibers made of transparent materials such as glass, fused silica, and plastic. (E)

file. A named set of records stored or processed as a unit. (T) Synonymous with *data set*.

file allocation table (FAT). In IBM personal computers, a table used to allocate space on a disk for a file and to locate and chain together parts of the file that may be scattered on different sectors so that the file can be used in a random or sequential manner.

file name substitution. In the AIX operating system, the process in which the shell substitutes an alphabetically sorted list of file names in the place of a pattern. The shell recognizes a pattern (as opposed to a file name) by the occurrence of a word (character string) with either of the following characteristics:

- The word contains any of these characters: *, ?, [, or {.
- The word begins with this character: ~.

Synonymous with *globbing*.

File Transfer Protocol (FTP). In the Internet suite of protocols, an application layer protocol that uses TCP and Telnet services to transfer bulk-data files between machines or hosts.

file transfer, access, and management (FTAM). An application service element that enables user application processes to manage and access a file system, which may be distributed. (T)

filter. A device or program that separates data, signals, or material in accordance with specified criteria. (A)

fixed disk. Synonym for *hard disk*.

font. A family of characters of a given size and style; for example, 9-point Helvetica. (T)

foreground task. The task with which the user is interacting. Contrast with *background task*.

foreign host. Synonym for *remote host*.

foreign network. In an internet, any network interconnected to the local network by one or more intermediate gateways or routers.

FQDN. Fully qualified domain name.

frame. (1) In Open Systems Interconnection architecture, a data structure pertaining to a particular area of knowledge and consisting of slots that can accept the values of specific

attributes and from which inferences can be drawn by appropriate procedural attachments. (T) (2) The unit of transmission in some local area networks, including the IBM Token-Ring Network. It includes delimiters, control characters, information, and checking characters. (3) In SDLC, the vehicle for every command, every response, and all information that is transmitted using SDLC procedures.

FTAM. File transfer, access, and management.

FTP. File Transfer Protocol.

fullword. Synonym for *computer word*.

fully qualified domain name (FQDN). In the Internet suite of protocols, the name of a host system that includes all of the subnames of the domain name. An example of a fully qualified domain name is `ralvm7.vnet.ibm.com`. See also *host name*.

fully qualified name. In the Internet suite of protocols, see *fully qualified domain name (FQDN)*.

G

gadget. In the AIXwindows Toolkit, a windowless graphical object that looks like its equivalent like-named widget but does not support the translations, actions, or pop-up widget children supplied by that widget.

gateway. (1) A functional unit that interconnects two computer networks with different network architectures. A gateway connects networks or systems of different architectures. A bridge interconnects networks or systems with the same or similar architectures. (T) (2) In TCP/IP, synonym for *router*.

GC. Graphics context.

GContext. Graphics context.

GDDM. Graphical Data Display Manager.

GDDM interface for X Window System (GDDMXD). A graphical interface that formats and displays characters, graphics, and images on workstation display devices that support the X Window System.

GDDMXD. Graphical Data Display Manager interface for X Window System.

GDF. Graphics data file.

GID. Group ID.

global character. Synonym for *pattern-matching character*.

globbing. Deprecated term for *file name substitution*.

glyph. An image, usually of a character, in a font.

Graphical Data Display Manager (GDDM). In the NetView Performance Monitor (NPM), an IBM licensed program used in conjunction with the Presentation Graphics Feature (PGF) to generate online graphs in the NPM Graphic Subsystem.

graphics context (GC, Gcontext). In the Enhanced X-Windows Toolkit, the storage area for various kinds of graphics output, such as foreground pixels, background pixels, line widths, and clipping regions. A graphics context can be used only with drawables that have the same root and the same depth as the graphics context.

graphics data file (GDF). A picture definition in a coded format used internally by the Graphical Data Display Manager (GDDM) that optionally provides the user with a lower level program interface than the GDDM application program interface (API).

group ID (GID). In the AIX operating system, a number that corresponds to a specific group name. The group ID can often be substituted in commands that take a group name as a value.

H

halfword. A contiguous sequence of bits or characters that constitutes half a computer word and can be addressed as a unit. (A)

handle. (1) In the Advanced DOS and OS/2 operating systems, a binary value created by the system that identifies a drive, directory, and file so that the file can be found and opened. (2) In the AIX operating system, a data structure that is a temporary local identifier for an object. Allocating a handle creates it. Binding a handle makes it identify an object at a specific location.

hard disk. A rigid magnetic disk such as the internal disks used in the system units of personal computers and in external hard disk drives. Synonymous with *fixed disk*.

HASP. Houston Automatic Spooling Program.

HDLC. High-level data link control.

header file. Synonym for *include file*.

hexadecimal. (1) Pertaining to a selection, choice, or condition that has 16 possible different values or states. (I) (2) Pertaining to a fixed-radix numeration system, with radix of 16. (I) (3) Pertaining to a system of numbers to the base 16; hexadecimal digits range from 0 through 9 and A through F, where A represents 10 and F represents 15.

high-level data link control (HDLC). In data communication, the use of a specified series of bits to control data links in accordance with the International Standards for HDLC: ISO 3309 Frame Structure and ISO 4335 Elements of Procedures.

high-performance file system (HPFS). In the OS/2 operating system, an installable file system that uses high-speed buffer storage, known as a cache, to provide fast access to large disk volumes. The file system also supports the coexistence of multiple, active file systems on a single personal computer, with the capability of multiple and different storage devices. File names used with the HPFS can have as many as 254 characters.

hiragana. One of the two common Japanese phonetic alphabets (the other is katakana). In hiragana, each character is represented by 1 byte. See also *kanji*.

hop count. (1) A metric or measure of distance between two points. (2) In Internet communications, the number of routers that a datagram passes through on its way to its destination. (3) In SNA, a measure of the number of links to be traversed in a path to a destination.

host. In the Internet suite of protocols, an end system. The end system can be any workstation; it does not have to be a mainframe.

host address. See *IP address*.

host ID. In the Internet suite of protocols, that part of the IP address that defines the host system on the network. The length of the host ID depends on the type of network or network class (A, B, or C).

host name. In the Internet suite of protocols, the name given to a machine. Sometimes, "host name" is used to mean *fully qualified domain name (FQDN)*; other times, it is used to mean the most specific subname of a fully qualified domain name. For example, if `ralvm7.vnet.ibm.com` is the fully qualified domain name, either of the following may be considered the host name:

- `ralvm7.vnet.ibm.com`
- `ralvm7`

hot key. The key combination used to change from one session to another on the workstation.

Houston Automatic Spooling Program (HASP). A computer program that provides supplementary job management, data management, and task management functions, such as control of job flow, ordering of tasks, and spooling.

HPFS. High-performance file system.

HTML. HyperText Markup Language. A markup language that is specified by an SGML document type definition (DTD) and that is understood by all World Wide Web servers.

HyperText Markup Language (HTML). See *HTML*.

I

IAB. Internet Architecture Board.

IBM Disk Operating System (DOS). A disk operating system based on MS-DOS that operates with all IBM personal computers.

IBM Operating System/2 (OS/2). An IBM licensed program that can be used as the operating system for personal computers. The OS/2 licensed program can perform multiple tasks at the same time.

IBM OS/2 Presentation Manager. The front-end data manager and user interface for the IBM OS/2 operating system; an example of a graphical user interface.

ICMP. Internet Control Message Protocol.

IEEE. Institute of Electrical and Electronics Engineers.

IESG. Internet Engineering Steering Group.

IETF. Internet Engineering Task Force.

IGP. Interior Gateway Protocol.

immediate access storage. A storage device whose access time is negligible in comparison with other operating times. (A)

IMS. Information Management System. Synonym for *IMS/VS*.

IMS/VS. Information Management System/Virtual Storage. Synonymous with *IMS*.

include file. A text file that contains declarations used by a group of functions, programs, or users. Synonymous with *header file*.

Information Management System/Virtual Storage (IMS/VS). A database/data communication (DB/DC) system that can manage complex databases and networks.

initial program load (IPL). (1) The initialization procedure that causes an operating system to commence operation. (2) The process by which a configuration image is loaded into storage at the beginning of a work day or after a system malfunction. (3) The process of loading system programs and preparing a system to run jobs.

installation. (1) In system development, preparing and placing a functional unit in position for use. (T) (2) A particular computing system, including the work it does and the people who manage it, operate it, apply it to problems, service it, and use the results it produces.

Institute of Electrical and Electronics Engineers (IEEE). A professional society accredited by the American National Standards Institute (ANSI) to issue standards for the electronics industry.

integrated services digital network (ISDN). A digital end-to-end telecommunication network that supports multiple services including, but not limited to, voice and data.

Note: ISDNs are used in public and private network architectures.

interactive. Pertaining to a program or system that alternately accepts input and then responds. An interactive system is conversational, that is, a continuous dialog exists between user and system. Contrast with *batch*.

interior gateway. In Internet communications, a gateway that communicates only with its own autonomous system. Contrast with *exterior gateway*.

Interior Gateway Protocol (IGP). In the Internet suite of protocols, a protocol used to propagate network reachability and routing information within an autonomous system. Examples of IGPs are Routing Information Protocol (RIP) and Open Shortest Path First (OSPF).

International Organization for Standardization (ISO). An organization of national standards bodies from various countries established to promote development of standards to facilitate international exchange of goods and services, and develop cooperation in intellectual, scientific, technological, and economic activity.

International Telecommunication Union (ITU). The specialized telecommunication agency of the United Nations, established to provide standardized communication procedures and practices, including frequency allocation and radio regulations worldwide.

internet. A collection of networks interconnected by a set of routers that allow them to function as a single, large network. See also *Internet*.

Internet. The internet administered by the Internet Architecture Board (IAB), consisting of large national backbone networks and many regional and campus networks all over the world. The Internet uses the Internet suite of protocols.

Internet address. See *IP address*.

Internet Architecture Board (IAB). The technical body that oversees the development of the Internet suite of protocols that are known as TCP/IP.

Internet Control Message Protocol (ICMP). The protocol used to handle errors and control messages in the Internet Protocol (IP) layer. Reports of problems and incorrect datagram destinations are returned to the original datagram source. ICMP is part of the Internet Protocol.

Internet drafts. Proposals, techniques, and mechanisms that document the Internet Engineering Task Force (IETF) work in progress and that define protocols and their characteristics in an internet. After the drafts are approved, they become Request for Comments (RFC) standards.

Internet Engineering Steering Group (IESG). The executive committee of the Internet Engineering Task Force (IETF).

Internet Engineering Task Force (IETF). The task force of the Internet Architecture Board (IAB) that is responsible for solving the short-term engineering needs of the Internet.

internet layer. In Internet communications, the Internet Protocol (IP) layer that provides transparency over the physical network (transmission media) and over the internet topology. The internet layer is equivalent to the network layer in Open Systems Interconnection (OSI) architecture.

Internet Protocol (IP). A connectionless protocol that routes data through a network or interconnected networks. IP acts as an intermediary between the higher protocol layers and the physical network. However, this protocol does not provide error recovery and flow control and does not guarantee the reliability of the physical network.

Internet router. A device that enables an internet host to act as a gateway for routing data between separate networks that use a specific adapter.

Internet suite of protocols. A set of protocols developed for use on the Internet and published as Requests for Comments (RFCs).

internetwork. Any wide area network connecting more than one network.

Internetwork Packet Exchange (IPX). The network protocol used to connect Novell's servers, or any workstation or router that implements IPX, with other workstations. Although similar to the Internet Protocol (IP), IPX uses different packet formats and terminology.

interoperability. The capability to communicate, execute programs, or transfer data among various functional units in a way that requires the user to have little or no knowledge of the unique characteristics of those units. (T)

interrupt. (1) A suspension of a process, such as execution of a computer program caused by an external event, and performed in such a way that the process can be resumed. (A) (2) To stop a process in such a way that it can be resumed.

interrupt number. The identification that is used to send a signal from an installed hardware feature to the CPU requesting attention. Different hardware features use different interrupt numbers.

intersystem communication (ISC). An extension of IMS/VS Multiple Systems Coupling that permits the connection of IMS/VS to another IMS/VS subsystem, to CICS/OS/VS, or to a user-written subsystem, provided both subsystems use intersystem communication.

intrapartition. In CICS, pertaining to the same CICS address space.

intrinsics. In the Enhanced X-Windows Toolkit, a set of management mechanisms that provides for constructing and interfacing between composite widgets, their children, and other clients. Also, intrinsics provide the ability to organize a collection of widgets into an application.

IP. Internet Protocol.

IP address. The 32-bit address defined by the Internet Protocol, standard 5, Request for Comments (RFC) 791. It is usually represented in dotted decimal notation.

IP datagram. In the Internet suite of protocols, the fundamental unit of information transmitted through an internet. It contains source and destination addresses, user data, and control information such as the length of the datagram, the header checksum, and flags indicating whether the datagram can be or has been fragmented.

IPL. Initial program load.

IPX. Internetwork Packet Exchange.

ISC. Intersystem communication.

ISDN. Integrated services digital network.

ISO. International Organization for Standardization.

iterative server. A server that can handle only one connection at a time. It can accept a new connection request only when it has completed processing the transaction started by a previous request. Contrast with *concurrent server*.

ITU. International Telecommunication Union.

ITU-T. See *ITU-TS*.

ITU-TS. International Telecommunication Union - Telecommunication Standardization Sector. The part of the International Telecommunication Union (ITU) that is responsible for developing recommendations for telecommunications.

J

Japanese Industrial Standards Committee (JISC). An organization that issues standards for coding character sets.

JCL. Job control language.

JES. Job Entry Subsystem.

JES reader. The part of the Job Entry Subsystem (JES) that receives job input and records it in the job queue and spool data set.

JES writer. The part of the Job Entry Subsystem (JES) that receives job output and writes it to end-use devices.

JES2. An MVS subsystem that receives jobs into the system, converts them to internal format, selects them for execution, processes their output, and purges them from the system. In an installation with more than one processor, each JES2 processor independently controls its job input, scheduling, and output processing.

JES3. An MVS subsystem that receives jobs into the system, converts them to internal format, selects them for execution, processes their output, and purges them from the system. In complexes that have several loosely coupled processing units, the JES3 program manages processors so that the global processor exercises centralized control over the local processors and distributes jobs to them via a common job queue.

JISC. Japanese Industrial Standards Committee.

job control language (JCL). A control language used to identify a job to an operating system and to describe the job's requirements.

Job Entry Subsystem (JES). An IBM licensed program that receives jobs into the system and processes all output data produced by the jobs.

JUNET. The Japanese Academic and Research Network that connects various UNIX operating systems.

K

kanji. A Japanese ideographic alphabet. In kanji, each character is represented by 2 bytes. See also *hiragana* and *katakana*.

katakana. One of the two common Japanese phonetic alphabets (the other is hiragana). In katakana, each character is represented by 1 byte. Katakana is primarily used to write foreign words phonetically. See also *kanji*.

Kb. Kilobit.

KB. Kilobyte.

Kbps. Kilobits per second.

Kerberos. The security system of the Massachusetts Institute of Technology's (MIT's) Project Athena. It uses symmetric key cryptography to provide security services to users in a network.

kernel. The part of an operating system that performs basic functions such as allocating hardware resources.

kill. In the UNIX operating system, to stop a process from running.

kilobit (Kb). (1) For processor storage, real and virtual storage, and channel volume, 2^{10} or 1024 bits. (2) For disk storage capacity and communications volume, 1000 bits.

kilobyte (KB). (1) For processor storage, real and virtual storage, and channel volume, 2^{10} or 1024 bytes. (2) For disk storage capacity and communications volume, 1000 bytes.

L

LaMail. The client that communicates with the OS/2 Presentation Manager to manage mail on the network.

LAN. Local area network.

LED. Light-emitting diode.

light-emitting diode (LED). A semiconductor chip that gives off visible or infrared light when activated.

Line Printer Daemon (LPD). The printer server that allows other hosts to access its printer.

Line Printer Protocol. In the Internet suite of protocols, a protocol used for printing files on printers attached to remote hosts.

Line Printer Requester (LPR). A client that allows the local host to submit a file for printing on a remote printer server.

link-attached. (1) Pertaining to devices that are connected to a controlling unit by a data link. (2) Contrast with *channel-attached*. (3) Synonymous with *remote*.

Listener. In the TCP/IP socket interface for CICS, an IBM-supplied application program that performs the functions of a concurrent server.

little endian. A format for storage or transmission of binary data in which the least significant bit (or byte) is placed first. Contrast with *big endian*.

LLB. Local Location Broker.

local. (1) Pertaining to a device accessed directly without use of a telecommunication line. (2) Contrast with *remote*. (3) Synonym for *channel-attached*.

local area network (LAN). (1) A computer network located on a user's premises within a limited geographical area. Communication within a local area network is not subject to external regulations; however, communication across the LAN boundary may be subject to some form of regulation. (T) (2) A network in which a set of devices are connected to one another for communication and that can be connected to a larger network. (3) See also *Ethernet* and *token ring*. (4) Contrast with *metropolitan area network (MAN)* and *wide area network (WAN)*.

local host. (1) In TCP/IP, the host on the network at which a particular operator is working. (2) In an internet, the host to which a user's terminal is connected without using the internet.

Local Location Broker (LLB). In the AIX Network Computing System (NCS) Location Broker, a server that maintains information about objects on the local host and provides the Location Broker forwarding facility.

local network. In an internet, the portion of a network that is physically connected to the local host without intermediate gateways or routers. Contrast with *foreign network.*

logical terminal. In IMS/VS, a destination with a name related to one or more physical terminals. Abbreviated as *LTERM*.

logical unit (LU). A type of network accessible unit that enables users to gain access to network resources and communicate with each other.

logon. The procedure by which a user begins a terminal session.

loopback interface. (1) An interface that bypasses unnecessary communications functions when the information is addressed to an entity within the same system. (2) See also *loopback test*.

loopback test. A test in which signals from a tester are looped at a modem or other network element back to the tester for measurements that determine or verify the quality of the communications path.

LPD. Line Printer Daemon.

LPR. Line Printer Requester.

LTERM. In IMS/VS, logical terminal.

LU. Logical unit.

LU-LU session. A logical connection between two logical units (LUs) in an SNA network that typically provides communication between two users.

LU type. The classification of an LU in terms of the specific subset of SNA protocols and options it supports for a given session, namely:

- The mandatory and optional values allowed in the session activation request
- The usage of data stream controls, function management headers (FMHs), request unit parameters, and sense data values
- Presentation services protocols such as those associated with FMH usage

LU types 0, 1, 2, 3, 4, 6.1, 6.2, and 7 are defined.

LU 6.2. A type of logical unit that supports general communication between programs in a distributed processing environment. LU 6.2 is characterized by (a) a peer relationship between session partners, (b) efficient utilization of a session for multiple transactions, (c)

comprehensive end-to-end error processing, and (d) a generic application programming interface (API) consisting of structured verbs that are mapped into a product implementation.

LU 6.2 session. A session that is initiated by VTAM on behalf of a logical unit (LU) 6.2 application program, or a session initiated by a remote LU in which the application program specifies that VTAM is to control the session by using the APPCCMD macroinstruction.

M

MAC. Medium access control.

mail gateway. A machine that connects two or more electronic mail systems (often, mail systems on different networks) and transfers messages between them.

MAN. Metropolitan area network.

Management Information Base (MIB). (1) A collection of objects that can be accessed by means of a network management protocol. (2) A definition for management information that specifies the information available from a host or gateway and the operations allowed. (3) In OSI, the conceptual repository of management information within an open system.

mapping. The process of converting data that is transmitted in one format by the sender into the data format that can be accepted by the receiver.

markup. The identification of the components of a document to enable each component to be appropriately formatted, displayed, or used.

markup language. A notation for identifying the components of a document to enable each component to be appropriately formatted, displayed, or used.

marshall. To copy data into a remote procedure call (RPC) packet. Stubs perform marshalling. Contrast with *unmarshall*.

mask. (1) A pattern of characters used to control retention or elimination of portions of another pattern of characters. (I) (A) (2) To use a pattern of characters to control retention or elimination of portions of another pattern of characters. (I) (A)

Master Terminal Operator (MTO). The console operator for CICS.

maximum transfer unit (MTU). The maximum number of bytes that an Internet Protocol (IP) datagram can contain.

maximum transmission unit (MTU). In LANs, the largest possible unit of data that can be sent on a given physical medium in a single frame. For example, the MTU for Ethernet is 1500 bytes.

Mb. Megabit.

MB. Megabyte.

Mbps. Megabits per second.

medium access control (MAC). In LANs, the sublayer of the data link control layer that supports medium-dependent functions and uses the services of the physical layer to provide services to the logical link control (LLC) sublayer. The MAC sublayer includes the method of determining when a device has access to the transmission medium.

megabit (Mb). (1) For processor storage, real and virtual storage, and channel volume, 2^{20} or 1 048 576 bits. (2) For disk storage capacity and communications volume, 1 000 000 bits.

megabyte (MB). (1) For processor storage, real and virtual storage, and channel volume, 2^{20} or 1 048 576 bytes. (2) For disk storage capacity and communications volume, 1 000 000 bytes.

menu. (1) A list of options displayed to the user by a data processing system, from which the user can select an action to be initiated. (T) (2) In text processing, a list of choices displayed to the user by a text processor from which the user can select an action to be initiated. (T) (3) A list of choices that can be applied to an object. A menu can contain choices that are not available for selection in certain contexts. Those choices are indicated by reduced contrast.

menu bar. In the AIX operating system, a rectangular area at the top of the client area of a window that contains the titles of the standard pull-down menus for that application.

message. An assembly of characters and sometimes control codes that is transferred as an entity from an originator to one or more recipients. A message consists of two parts: envelope and content. (T)

message format service (MFS). In IMS/VS, an editing facility that allows application programs to deal with simple logical messages instead of device-dependent data, thus simplifying the application development process.

message handling service (MHS). An application service element that provides generalized facility for exchanging electronic messages between systems. (T)

message processing program (MPP). In IMS/VS, an application program that is driven by transactions and has access to online IMS/VS databases and message queues.

message queue. In IMS/VS, the data set on which messages are queued before being processed by an application program or sent to a terminal.

metropolitan area network (MAN). A network formed by the interconnection of two or more networks which may operate at higher speed than those networks, may cross administrative boundaries, and may use multiple access methods. (T) Contrast with *local area network (LAN)* and *wide area network (WAN)*.

MFS. Message format service.

MHS. Message handling service.

MIB. Management Information Base.

MIB module. In the Simple Network Management Protocol (SNMP), a collection of objects relating to a common management area. See also *Management Information Base (MIB)* and *MIB variable*.

MIB object. Synonym for *MIB variable*.

MIB tree. In the Simple Network Management Protocol (SNMP), the structure of the Management Information Base (MIB).

MIB variable. In the Simple Network Management Protocol (SNMP), a specific instance of data defined in a MIB module. Synonymous with *MIB object*.

MIB view. In the Simple Network Management Protocol (SNMP), the collection of managed objects, known to the agent, that is visible to a particular community.

MIB walking. In the Simple Network Management Protocol (SNMP), a technique of looking for Management Information Base (MIB) tree information when it is presented in a hierarchical format.

microcode. (1) One or more microinstructions. (2) A code, representing the instructions of an instruction set, that is implemented in a part of storage that is not program-addressable. (3) To design, write, and test one or more microinstructions.

MILNET. The military network that was originally part of ARPANET. It was partitioned from ARPANET in 1984. MILNET provides a reliable network service for military installations.

modem (modulator/demodulator). (1) A functional unit that modulates and demodulates signals. One of the functions of a modem is to enable digital data to be transmitted over analog transmission facilities. (T) (A) (2) A device that converts digital data from a computer to an analog signal that can be transmitted on a telecommunication line, and converts the analog signal received to data for the computer.

Motif. See *OSF/Motif*.

mount. (1) To place a data medium in a position to operate. (T) (2) To make recording media accessible.

mouse. A commonly used pointing device, containing one or more buttons, with which a user can interact with a product or the operating environment.

mouse button. A mechanism on a mouse pointing device used to select objects or choices, initiate actions, or directly manipulate objects; a user presses a mouse button to interact with a computer system. The button makes a “clicking” sound when pressed and released.

MPP. Message processing program.

MSC. Multiple Systems Coupling.

MTO. Master Terminal Operator.

MTU. (1) Maximum transfer unit. (2) Maximum transmission unit.

multicast. (1) Transmission of the same data to a selected group of destinations. (T) (2) A special form of broadcast in which copies of a packet are delivered to only a subset of all possible destinations. (3) Contrast with *broadcast*.

multiconnection server. Synonym for *concurrent server*.

multihomed host. In the Internet Protocol (IP), a host that is connected to more than one network.

Multiple Systems Coupling (MSC). An IMS/VS feature that permits geographically dispersed IMS/VS systems to communicate with each other.

Multiple Virtual Storage (MVS). See *MVS*.

multitasking. A mode of operation that provides for concurrent performance, or interleaved execution of two or more tasks. (I) (A)

MVS. Multiple Virtual Storage. Implies MVS/370 and the MVS/ESA product.

N

name server. In the Internet suite of protocols, synonym for *domain name server*.

National Science Foundation (NSF). A United States government agency that is a sponsor of the National Science Foundation Network (NFSNET).

National Science Foundation Network (NSFNET). A collection of local and regional net-

works in the United States that are connected by a high-speed backbone. NSFNET provides scientists access to a number of supercomputers across the country.

NCK. Network Computing Kernel.

NCP. Network Control Program.

NCS. Network Computing System.

NDB. Network Database System.

NDIS. Network driver interface specification.

NetBIOS. Network Basic Input/Output System. A standard interface to networks, IBM personal computers (PCs), and compatible PCs, that is used on LANs to provide message, print-server, and file-server functions. Application programs that use NetBIOS do not need to handle the details of LAN data link control (DLC) protocols.

NetView program. An IBM licensed program used to monitor and manage a network and to diagnose network problems.

network. (1) An arrangement of nodes and connecting branches. (T) (2) A configuration of data processing devices and software connected for information interchange. (3) A group of nodes and the links interconnecting them.

network adapter. A physical device, and its associated software, that enables a processor or controller to be connected to a network.

network administrator. A person who manages the use and maintenance of a network.

Network Computing Kernel (NCK). In the Network Computing System (NCS), the combination of the remote procedure call (RPC) runtime library and the Location Broker, which provide the function necessary required to run distributed applications.

Network Computing System (NCS). A set of software tools, developed by Apollo Computer Inc., that conform to the Network Computing Architecture. These tools include the remote procedure call runtime library, the Location Broker, and the NIDL compiler.

network control program. A program, generated by the user from a library of IBM-supplied modules, that controls the operation of a communication controller.

Network Control Program (NCP). An IBM licensed program that provides communication controller support for single-domain, multiple-domain, and interconnected network capability.

Network Database System (NDB). The part of TCP/IP that uses the remote procedure call (RPC) protocol to allow interoperability among a variety of workstation users and a mainframe relational database system. It provides access to a mainframe relational database from workstations and mainframes and allows workstation users to issue SQL statements interactively, or to invoke NDB services from within a C application program. NDB services can then be used to pass SQL statements to a DB2 or SQL/DS system and to handle responses from the DB2 or SQL/DS system.

network driver interface specification (NDIS). An application programming interface (API) definition that allows DOS or OS/2 systems to support one or more network adapters and protocol stacks. NDIS is a 16-bit, Ring O (for the OS/2 operating system) API that defines a specific way for writing drivers for layers 1 and 2 of the OSI model. NDIS also handles the configuration and binding of these network drivers to multiple protocol stacks.

network element. In the Simple Network Management Protocol (SNMP), a gateway, router, or host that contains management agents responsible for performing the network management functions requested by the network management stations.

Network File System (NFS). A protocol developed by Sun Microsystems, Incorporated, that allows any host in a network to mount another host's file directories. Once mounted, the file directory appears to reside on the local host.

network identifier. In TCP/IP, that part of the IP address that defines a network. The length of the network ID depends on the type of network class (A, B, or C).

Network Information Center (NIC). In Internet communications, local, regional, and national groups throughout the world who provide assistance, documentation, training, and other services to users.

Network Interface Definition Language (NIDL). A declarative language for the definition of interfaces that has two forms, a Pascal-like syntax and a C-like syntax. NIDL is a component of the Network Computing Architecture.

network job entry. In object distribution, an entry in the network job table that specifies the system action required for incoming network jobs sent by a particular user or group of users. Each entry is identified by the user ID of the originating user or group.

network layer. In Open Systems Interconnection (OSI) architecture, the layer that is responsible for routing, switching, and link-layer access across the OSI environment.

network management station. In the Simple Network Management Protocol (SNMP), a station that executes management application programs that monitor and control network elements.

Network Print Facility (NPF). In TCP/IP for MVS, a feature that routes VTAM, JES2, or JES3 printer output to printers in a TCP/IP network.

NFS. Network File System.

NFS client. A program or system that mounts remote file directories from another host called a Network File System (NFS) server.

NFS server. A program or system that allows authorized remote hosts called Network File System (NFS) clients to mount and access its local file directories.

NIC. Network Information Center.

NIDL. Network Interface Definition Language.

NJE. Network job entry.

node. (1) In a network, a point at which one or more functional units connect channels or data circuits. (I) (2) Any device, attached to a network, that transmits and receives data. (3) An endpoint of a link or a junction common to two or more links in a network. Nodes can be processors, communication controllers, cluster controllers, or terminals. Nodes can vary in routing and other functional capabilities.

nonblocking mode. (1) A way of requesting a service over an interface so that if the request cannot be completed immediately, the requesting process is able to continue and is not suspended. (2) Contrast with *blocking mode*.

NPF. Network Print Facility.

NPSI. X.25 NCP Packet Switching Interface.

NSF. National Science Foundation.

NSFNET. National Science Foundation Network.

O

octal. Pertaining to a selection, choice, or condition that has eight possible different values or states. (I) (A)

octet. A byte that consists of 8 bits. (T)

OEM. Original equipment manufacturer.

OfficeVision Series. IBM's family of office application programs that can be used for office tasks such as creating, sending, receiving, and filing electronic mail.

Offload host. Any device that is handling the TCP/IP processing for the host where TCP/IP with the Offload feature is installed.

Offload system. The Offload host plus the host where TCP/IP with the Offload feature is installed.

Open Shortest Path First (OSPF). In the Internet suite of protocols, a function that provides intradomain information transfer. An alternative to the Routing Information Protocol (RIP), OSPF allows the lowest-cost routing and handles routing in large regional or corporate networks.

Open Software Foundation (OSF). A nonprofit research and development organization whose goals are (a) to develop specifications and software for use in an open software environment and (b) to make the specifications and software available to information technology vendors under fair and equitable licensing terms. For example, OSF developed the Distributed Computing Environment (DCE).

open system. A system whose characteristics comply with standards made available throughout the industry and that therefore can be connected to other systems complying with the same standards. (T)

Open Systems Interconnection (OSI). (1) The interconnection of open systems in accordance with standards of the International Organization for Standardization (ISO) for the exchange of information. (T) (A) (2) The use of standardized procedures to enable the interconnection of data processing systems.

Note: OSI architecture establishes a framework for coordinating the development of current and future standards for the interconnection of computer systems. Network functions are divided into seven layers. Each layer represents a group of related data processing and communication functions that can be carried out in a standard way to support different applications.

Open Systems Interconnection (OSI) architecture. Network architecture that adheres to that particular set of ISO standards that relates to Open Systems Interconnection. (T)

Open Systems Interconnection (OSI) reference model. A model that describes the general principles of the Open Systems Interconnection, as well as the purpose and the hierarchical arrangement of its seven layers. (T)

original equipment manufacturer (OEM). A manufacturer of equipment that may be marketed by another manufacturer.

orphan. In the UNIX operating system, a child process that runs even though its parent process has been killed.

OS/2 operating system. IBM Operating System/2.

OSF. Open Software Foundation.

OSF/Motif. A graphical interface that contains a toolkit, a presentation description language, a window manager, and a style guideline.

OSI. Open Systems Interconnection.

OSPF. Open Shortest Path First.

P

packet. In data communication, a sequence of binary digits, including data and control signals, that is transmitted and switched as a composite whole. The data, control signals, and, possibly, error control information are arranged in a specific format. (I)

packet internet groper (PING). In Internet communications, a program used in TCP/IP networks to test the ability to reach destinations by sending the destinations an Internet Control Message Protocol (ICMP) echo request and waiting for a reply.

packet switching. The process of routing and transferring data by means of addressed packets so that a channel is occupied only during transmission of a packet. On completion of the transmission, the channel is made available for transfer of other packets. (I)

packet switching data network (PSDN). A network that uses packet switching as a means of transmitting data.

parallel. (1) Pertaining to a process in which all events occur within the same interval of time, each handled by a separate but similar functional unit; for example, the parallel transmission of the bits of a computer word along the lines of an internal bus. (T) (2) Pertaining to concurrent or simultaneous operation of two or more devices or to concurrent performance of two or more activities in a single device. (A) (3) Pertaining to concurrent or simultaneous occurrence of two or more related activities in multiple devices or channels. (A) (4) Pertaining to the simultaneity of two or more processes. (A) (5) Pertaining to the simultaneous processing of the individual parts of a whole, such as the bits of a character and the characters of a word, using separate facilities for the various parts. (A) (6) Contrast with *serial*.

parallel channel. A channel having a System/360 and System/370 channel-to-control-unit I/O interface that uses bus-and-tag cables as a transmission medium. Contrast with *ESCON channel*.

parameter. (1) A variable that is given a constant value for a specified application and that may denote the application. (I) (A) (2) In Basic CUA architecture, a variable used in conjunction with a command to affect its result. (3) An item in a menu for which the user specifies a value or for which the system provides a value when the menu is interpreted. (4) Data passed to a program or procedure by a user or another program, namely as an operand in a language statement, as an item in a menu, or as a shared data structure.

parse. To analyze the operands entered with a command and create a parameter list for the command processor from the information.

partition. A fixed-size division of storage.

partitioned data set (PDS). A data set in direct access storage that is divided into partitions, called members, each of which can contain a program, part of a program, or data.

passive open. In the TCP/UDP socket interface, the state of a connection that is prepared to provide a service (such as accept datagrams) on demand. Contrast with *active open*.

pattern-matching character. A special character such as an asterisk (*) or a question mark (?) that can be used to represent one or more characters. Any character or set of

characters can replace a pattern-matching character. Synonymous with *global character* and *wildcard character*.

PC. Personal computer.

PC network. A low-cost, broadband network that allows attached IBM personal computers to communicate and share resources.

PCNFSD. Personal-computer network-file-system daemon.

PDN. Public data network.

PDS. Partitioned data set.

PDU. Protocol data unit.

peer. In network architecture, any functional unit that is in the same layer as another entity. (T)

pel. Picture element.

peripheral PU. In SNA, a physical unit in a peripheral node. Contrast with *subarea PU*.

personal computer (PC). (1) A microcomputer primarily intended for stand-alone use by an individual. (T) (2) A desktop, floor-standing, or portable microcomputer that usually consists of a system unit, a display monitor, a keyboard, one or more diskette drives, internal fixed-disk storage, and an optional printer. PCs are designed primarily for stand-alone operation but may be connected to mainframes or networks.

personal-computer NFS daemon (PCNFSD). A daemon that manages user authentication and print spooling.

Personal System Communications Adapter (PSCA). An adapter card that connects a microchannel-based personal computer (or processor) to a System/370 or System/390 parallel channel.

physical layer. In the Open Systems Interconnection reference model, the layer that provides the mechanical, electrical, functional, and procedural means to establish, maintain, and release physical connections over the transmission medium. (T)

physical unit (PU). (1) The component that manages and monitors the resources (such as attached links and adjacent link stations) associated with a node, as requested by an SSCP via an SSCP-PU session. An SSCP activates a session with the physical unit in order to indirectly manage, through the PU, resources of the node such as attached links. This term applies to type 2.0, type 4, and type 5 nodes only. (2) See also *peripheral PU* and *subarea PU*.

picture element (pel, pixel). (1) In computer graphics, the smallest element of a display surface that can be independently assigned color and intensity. (T) (2) The area of the finest detail that can be reproduced effectively on the recording medium. (3) An element of a raster pattern about which a toned area on a photoconductor can appear.

PING. Packet internet groper.

ping command. The command that sends an Internet Control Message Protocol (ICMP) echo-request packet to a gateway, router, or host with the expectation of receiving a reply.

piping. A feature that allows the output of a program as it is displayed on the screen to be used as input to another program without reentering the data on the keyboard.

pixel. Picture element.

pixel map. A three-dimensional array of bits. A pixel map can be thought of as a two-dimensional array of pixels, with each pixel being a value from zero to 2 to the power N -1, where N is the depth of the pixel map. Synonymous with *pixmap*.

pixmap. (1) In the AIXwindows Toolkit and the Enhanced X-Windows Toolkit, a data type to which icons, originally created as bitmaps, are converted. After this conversion, the appropriate AIXwindows subroutines can generate pixmaps through references to a defaults file, by name, and through an argument list, by pixmap. (2) Synonym for *pixel map*.

PM. Presentation Manager.

polling. (1) On a multipoint connection or a point-to-point connection, the process whereby data stations are invited, one at a time, to transmit. (I) (2) Interrogation of devices for such purposes as to avoid contention, to determine operational status, or to determine readiness to send or receive data. (A)

POP. Post Office Protocol.

pop-up window. In Advanced CUA architecture, a movable window, fixed in size, in which a user provides information required by an application so that it can continue to process a user request.

port. (1) An access point for data entry or exit. (2) A connector on a device to which cables for other devices such as display stations and printers are attached. (3) The representation of a physical connection to the link hardware. A port is sometimes referred to as an adapter; however, there can be more than one port on an adapter. There may be one or more ports controlled by a single DLC process. (4) In the Internet suite of protocols, a 16-bit number used to communicate between TCP or the User Datagram Protocol (UDP) and a higher-level protocol or application. Some protocols, such as File Transfer Protocol (FTP) and Simple Mail Transfer Protocol (SMTP), use the same well-known port number in all TCP/IP implementations. (5) An abstraction used by transport protocols to distinguish among multiple destinations within a host machine. (6) Synonymous with *socket*.

port number. In Internet communications, the identification of an application entity to the transport service.

portmapper. A program that maps client programs to the port numbers of server programs. Portmapper is used with remote procedure call (RPC) programs.

Post Office Protocol (POP). A protocol used for exchanging network mail and accessing mailboxes.

PostScript. A standard specified by Adobe Systems, Incorporated, that defines how text and graphics are presented on printers and display devices.

presentation layer. In the Open Systems Interconnection reference model, the layer that provides for the selection of a common syntax for representing information and for transformation of application data into or from this common syntax. (T)

Presentation Manager (PM). See *IBM OS/2 Presentation Manager*.

process. (1) A course of the events defined by its purpose or by its effect, achieved under given conditions. (2) Any operation or combination of operations on data. (3) A function being performed or waiting to be performed.

Professional Office Systems (PROFS). See *OfficeVision Series*.

PROFS. Professional Office Systems.

protocol. (1) A set of semantic and syntactic rules that determine the behavior of functional units in achieving communication. (I) (2) In Open Systems Interconnection architecture, a

set of semantic and syntactic rules that determine the behavior of entities in the same layer in performing communication functions. (T)

protocol data unit (PDU). A unit of data specified in a protocol of a given layer and consisting of protocol control information of this layer, and possibly user data of this layer. (T)

protocol suite. A set of protocols that cooperate to handle the transmission tasks for a communication system.

PSCA. Personal System Communications Adapter.

PSDN. Packet switching data network.

PU. Physical unit.

public data network (PDN). A communications common carrier network providing data communications services over switched or nonswitched lines. See *public network*.

public network. A network established and operated by a telecommunication administration or by a Recognized Private Operating Agency (RPOA) for the specific purpose of providing circuit-switched, packet-switched, and leased-circuit services to the public. Contrast with *user-application network*.

pull-down menu. See *menu*.

push button. A button, labeled with text, graphics, or both, that represents an action that will be initiated when a user selects it.

Q

queue. (1) A line or list of items waiting to be processed; for example, work to be performed or messages to be displayed. (2) To arrange in or form a queue.

R

RACF. Resource Access Control Facility.

radio button. A circle with text beside it. Radio buttons are combined to show a user a fixed set of choices from which the user can select one. The circle becomes partially filled when a choice is selected.

radix. The positive integer by which the weight of the digit place is multiplied to obtain the weight of the digit place with the next higher weight; for example, in the decimal numeration system the radix of each digit place is 10, in a biquinary code the radix of each fives position is 2. (I) (A)

RAM. Random access memory. (A)

random access memory (RAM). (1) A storage device in which data can be written and read. (2) Deprecated term for *direct access storage device (DASD)*. (T) (3) A storage device into which data is entered and from which data is retrieved in a nonsequential manner. (4) See also *direct access storage*.

RARP. Reverse Address Resolution Protocol.

read-only memory (ROM). Memory in which stored data cannot be modified by the user except under special conditions.

Recommendation X.21. See *X.21*.

Recommendation X.25. See *X.25*.

recursion. The performance of an operation in several steps, with each step using the output of the preceding step.

reduced instruction-set computer (RISC). A computer that uses a small, simplified set of frequently used instructions for rapid execution.

reentrant. The attribute of a program or routine that allows the same copy of the program or routine to be used concurrently by two or more tasks.

remote. (1) Pertaining to a system, program, or device that is accessed through a telecommunication line. (2) Synonym for *link-attached*. (3) Contrast with *local*.

Remote Execution Protocol (REXEC). A protocol that allows the execution of a command or program on any host in the network. The local host receives the results of the command execution.

remote host. Any host on a network except the host at which a particular operator is working. Synonymous with *foreign host*.

remote procedure call (RPC). A facility that a client uses to request the execution of a procedure call from a server. This facility includes a library of procedures and an external data representation.

Request for Comments (RFC). In Internet communications, the document series that describes a part of the Internet suite of protocols and related experiments. All Internet standards are documented as RFCs.

request unit (RU). A message unit that contains control information, end-user data, or both.

request/response unit (RU). A generic term for a request unit or a response unit. See *request unit (RU)* and *response unit (RU)*.

requester. Synonym for *client*.

resolver. In TCP/IP, a program or subroutine that obtains information from a domain name server or a local table for use by an application program.

Resource Access Control Facility (RACF). An IBM licensed program that provides for access control by identifying and verifying the users of the system, by authorizing access to protected resources, by logging the detected unauthorized attempts to enter the system, and by logging the detected accesses to protected resources.

resource records. In the Internet suite of protocols, individual records of data used by the Domain Name System (DNS). Examples of resource records include a host's Internet Protocol (IP) addresses, preferred mail addresses, and aliases.

response unit (RU). A message unit that acknowledges a request unit. It may contain prefix information received in a request unit. If positive, the response unit may contain additional information (such as session parameters in response to BIND SESSION). If negative, the response unit contains sense data defining the exception condition.

Restructured Extended Executor (REXX). A general-purpose, procedural language for end-user personal programming, designed for ease by both casual general users and computer professionals. It is also useful for application macros. REXX includes the capability of issuing commands to the underlying operating system from these macros and procedures. Features include powerful character-string manipulation, automatic data typing, manipulation of objects familiar to people, such as words, numbers, and names, and built-in interactive debugging.

return code. A value returned to a program to indicate the results of an operation requested by that program.

Reverse Address Resolution Protocol (RARP). (1) In the Internet suite of protocols, the protocol that maps a hardware (MAC) address to an IP address. RARP can be used to determine a port's IP address. (2) See also *Address Resolution Protocol (ARP)*.

REXEC. Remote Execution Protocol.

REXX. Restructured Extended Executor.

RFC. Request for Comments.

RGB. (1) Color coding in which the brightness of the additive primary colors of light, red, green, and blue, are specified as three distinct values of white light. (2) Pertaining to a color display that accepts signals representing red, green, and blue.

RIP. Routing Information Protocol.

RISC. Reduced instruction-set computer.

ROM. Read-only memory. (A)

route. The path that network traffic uses to get from source to destination.

router. (1) A computer that determines the path of network traffic flow. The path selection is made from several paths based on information obtained from specific protocols, algorithms that attempt to identify the shortest or best path, and other criteria such as metrics or protocol-specific destination addresses. (2) An attaching device that connects two LAN segments, which use similar or different architectures, at the reference model network layer. (3) In OSI terminology, a function that determines a path by which an entity can be reached. (4) In TCP/IP, synonymous with *gateway*. (5) Contrast with *bridge*.

Routing Information Protocol (RIP). In the Internet suite of protocols, an interior gateway protocol used to exchange intradomain routing information and to determine optimum routes between internet hosts. RIP determines optimum routes on the basis of route metrics, not link transmission speed.

routing table. A collection of routes used to direct datagram forwarding or to establish a connection. The information is passed among routers to identify network topology and destination feasibility.

RPC. Remote procedure call.

RPCGEN. A tool that generates C code to implement a Sun Microsystems, Incorporated, remote procedure call (RPC) protocol.

RU. Request/response unit.

S

SBCS. Single-byte character set.

scan code. A code that the keyboard generates when a key is pressed. Every key on a keyboard has a unique scan code associated with it.

SDLC. Synchronous Data Link Control.

secondary window. In Advanced CUA architecture, a type of window that can be moved and sized and is always associated with a primary window. Help is an example of a secondary window.

segment. (1) A section of cable between components or devices. A segment may consist of a single patch cable, several patch cables that are connected, or a combination of building cable and patch cables that are connected. (2) In Internet communications, the unit of transfer between TCP functions in different machines. Each segment contains control and data fields; the current byte-stream position and actual data bytes are identified along with a checksum to validate received data. (3) In IMS/VS, the unit of access to a database; for the database system, the smallest amount of data that can be transferred by one DL/I operation. For input terminal operations using the DC feature, a segment is defined by the particular terminal type and is obtained by the application program with one call.

select. To explicitly identify one or more objects to which a subsequent choice will apply.

semantics. The relationships between symbols and their meanings. (A)

Sendmail. In the UNIX operating system, the mail server that uses the Simple Mail Transfer Protocol (SMTP) to route mail from one host to another on the network.

serial. (1) Pertaining to a process in which all events occur one after the other; for example, serial transmission of the bits of a character according to V24 CCITT protocol. (T) (2) Pertaining to the sequential or consecutive occurrence of two or more related activities in a single device or channel. (A) (3) Pertaining to the sequential processing of the individual parts of a whole, such as the bits of a character or the characters of a word, using the same facilities for successive parts. (A) (4) Contrast with *parallel*.

serial line. A transmission medium commonly used for point-to-point link connections. Often, a serial line consists of an RS-232 connection into a modem over a telephone line.

serial line Internet Protocol (SLIP). A protocol used to run Internet Protocol (IP) over serial lines, such as telephone circuits or RS-232 cables, interconnecting two systems.

server. A functional unit that provides shared services to workstations over a network; for example, a file server, a print server, a mail server. (T)

SGML. Standard Generalized Markup Language. A syntax for markup languages that formalizes markup and frees it of system and processing dependencies.

shell. (1) A software interface between a user and the operating system of a computer. Shell programs interpret commands and user interactions on devices such as keyboards, pointing devices, and touch-sensitive screens and communicate them to the operating system. Shells simplify user interactions by eliminating the user's concern with operating system requirements. A computer may have several layers of shells for various levels of user interaction. (2) In the AIX operating system, a command interpreter that acts as an interface between the user and the operating system. A shell can contain another shell nested inside it; the outer shell is the parent shell, and the inner shell is the child.

Simple Mail Transfer Protocol (SMTP). In the Internet suite of protocols, an application protocol for transferring mail among users in the Internet environment. SMTP specifies the mail exchange sequences and message format. It assumes that the Transmission Control Protocol (TCP) is the underlying protocol.

Simple Network Management Protocol (SNMP). In the Internet suite of protocols, a network management protocol that is used to monitor routers and attached networks. SNMP is an application layer protocol. Information on devices managed is defined and stored in the application's Management Information Base (MIB).

single-byte character set (SBCS). A character set in which each character is represented by a one-byte code. Contrast with *double-byte character set (DBCS)*.

SLIP. Serial line Internet Protocol.

SMI. Structure of Management Information.

SMTP. Simple Mail Transfer Protocol.

SNA. Systems Network Architecture.

SNA Network Link (SNALINK). In TCP/IP for MVS and VM, a function that allows the use of an SNA subarea routing network to transfer data using TCP/IP protocols. SNALINK provides the interface between TCP/IP and the SNA network. SNALINK must be defined as an application program to VTAM, which causes LU 0 sessions to be established between the SNALINK logical unit and other logical units in the SNA network.

SNALINK. SNA Network Link.

SNMP. Simple Network Management Protocol.

SOA record. Start-of-authority record.

socket. (1) An endpoint for communication between processes or application programs. (2) Synonym for *port*.

socket address. The address of an application program that uses the socket interface on the network. In Internet format, it consists of the IP address of the socket's host and the port number of the socket. The application program is usually not aware of the structure of the address.

socket interface. A Berkeley Software Distribution (BSD) application programming interface (API) that allows users to easily write their own communication application programs.

source service access point (SSAP). In SNA and TCP/IP, a logical address that allows a system to send data to a remote device from the appropriate communications support. Contrast with *destination service access point (DSAP)*.

spin button. A component used to display, in sequence, a ring of related but mutually exclusive choices. A user can accept the value displayed in the entry field or can type a valid choice into the entry field.

spoofing. For data links, a technique in which a protocol initiated from an end station is acknowledged and processed by an intermediate node on behalf of the final destination. In IBM 6611 data link switching, for example, SNA frames are encapsulated into TCP/IP packets for transport across a non-SNA wide area network, unpacked by another IBM 6611, and passed to the final destination. A benefit of spoofing is the prevention of end-to-end session timeouts.

spool. Simultaneous peripheral operation online. See also *spooling*.

spooling. (1) The use of auxiliary storage as buffer storage to reduce processing delays when transferring data between peripheral equipment and the processors of a computer. The term is derived from the expression Simultaneous Peripheral Operation On Line. (T) (A) (2) Reading and writing input and output streams on an intermediate device in a format convenient for later processing or output. (3) Performing a peripheral operation such as printing while the computer is busy with other work.

SQL. Structured Query Language.

SQL/DS. Structured Query Language/Data System.

SSAP. Source service access point.

SSCP. System services control point.

Standard Generalized Markup Language (SGML). See *SGML*.

standard output. The primary destination of data coming from a command. Standard output goes to the display unless redirection or piping is used, in which case standard output can go to a file or to another command. In the AIX operating system, abbreviated as *STDOUT*.

start-of-authority (SOA) record. In the Domain Name System (DNS), the resource record that defines a zone.

static. (1) In programming languages, pertaining to properties that can be established before execution of a program; for example, the length of a fixed length variable is static. (I) (2) Pertaining to an operation that occurs at a predetermined or fixed time. (3) Contrast with *dynamic*.

STDOUT. In the AIX operating system, standard output.

stream. (1) To send data from one device to another. (2) See *data stream*.

Structure of Management Information (SMI). (1) In the Simple Network Management Protocol (SNMP), the rules used to define the objects that can be accessed by means of a network management protocol. (2) In OSI, the set of standards relating to management information. The set includes the *Management Information Model* and the *Guidelines for the Definition of Managed Objects*.

Structured Query Language (SQL). An English-like programming language used to define, to manipulate, to control data in, and to perform queries on, relational databases.

Structured Query Language/Data System (SQL/DS). An IBM relational database management system.

stub. (1) A program module that transfers remote procedure calls (RPCs) and responses between a client and a server. Stubs perform marshalling, unmarshalling, and data format conversion. Both clients and servers have stubs. The Network Interface Definition Language (NIDL) compiler generates client and server stub code from an interface definition. (2) Hooking functions used as extensions to the protocol to generate protocol requests for the Enhanced X-Windows Toolkit. (3) A small module, link-edited into application code, that locates and transfers control to a larger body of related code.

subagent. In the Simple Network Management Protocol (SNMP), something that provides an extension to the utility provided by the SNMP agent.

subarea PU. In SNA, a physical unit in a subarea node. Contrast with *peripheral PU*.

subdirectory. A directory contained within another directory in a file system hierarchy.

subnet. (1) In TCP/IP, a part of a network that is identified by a portion of the IP address. (2) Synonym for *subnetwork*.

subnet address. In Internet communications, an extension to the basic IP addressing scheme where a portion of the host address is interpreted as the local network address.

subnet mask. Synonym for *address mask*.

subnetwork. (1) Any group of nodes that have a set of common characteristics, such as the same network ID. (2) In the AIX operating system, one of a group of multiple logical network divisions of another network, such as can be created by the Transmission Control Protocol/Internet Protocol (TCP/IP) interface program. (3) Synonymous with *subnet*.

subnetwork mask. Synonym for *address mask*.

subsystem. A secondary or subordinate system, usually capable of operating independently of, or asynchronously with, a controlling system. (T)

synchronous. (1) Pertaining to two or more processes that depend upon the occurrence of specific events such as common timing signals. (T) (2) Occurring with a regular or predictable time relationship.

Synchronous Data Link Control (SDLC). (1) A discipline conforming to subsets of the Advanced Data Communication Control Procedures (ADCCP) of the American National Standards Institute (ANSI) and High-level Data Link Control (HDLC) of the International Organization for Standardization, for managing synchronous, code-transparent, serial-by-bit information transfer over a link connection. Transmission exchanges may be duplex or half-duplex over switched or nonswitched links. The configuration of the link connection may be point-to-point, multipoint, or loop. (I) (2) Contrast with *binary synchronous communication (BSC)*.

system services control point (SSCP). A component within a subarea network for managing the configuration, coordinating network operator and problem determination requests, and providing directory services and other session services for users of the network. Multiple SSCPs, cooperating as peers with one another, can divide the network into domains of control, with each SSCP having a hierarchical control relationship to the physical units and logical units within its own domain.

Systems Network Architecture (SNA). The description of the logical structure, formats, protocols, and operational sequences for transmitting information units through, and controlling the configuration and operation of, networks. The layered structure of SNA allows the ultimate origins and destinations of information, that is, the users, to be independent of and unaffected by the specific SNA network services and facilities used for information exchange.

T

talk. In the Internet suite of protocols, a protocol that allows two users on remote computers to communicate in a real-time fashion.

task. In a multiprogramming or multiprocessing environment, one or more sequences of instructions treated by a control program as an element of work to be accomplished by a computer. (I) (A)

task control. In CICS, a program that synchronizes CICS task activity. Under task control, the highest priority task that is ready for processing is started next.

task manager. In the OS/2 operating system, the function that controls the starting and stopping of programs, including shutting down the system.

Task-Related User Exit (TRUE). A CICS module used for invoking resource managers that are outside of CICS. A TRUE is one of the components of CICS sockets.

task switch. A change in a task in control of the processor. State of the task changes from ready to active, and current task is placed in a state other than active.

TCP. Transmission Control Protocol.

TCP/IP. Transmission Control Protocol/Internet Protocol.

Telnet. In the Internet suite of protocols, a protocol that provides remote terminal connection service. It allows users of one host to log on to a remote host and interact as directly attached terminal users of that host.

terminal. A device, usually equipped with a keyboard and a display device, that is capable of sending and receiving information.

terminal emulator. A program that allows a device such as a microcomputer or personal computer to enter and receive data from a computer system as if it were a particular type of attached terminal.

terminate-and-stay-resident (TSR) program. A program that installs part of itself as an extension of DOS when it is executed.

TFTP. Trivial File Transfer Protocol.

ticket-granting server. In Kerberos, a function that grants Kerberos tickets to authenticated users as permission to access an end service.

Time Sharing Option (TSO). An operating system option; for the System/370 system, the option provides interactive time sharing from remote terminals.

time stamp. (1) To apply the current system time. (2) The value on an object that is an indication of the system time at some critical point in the history of the object. (3) In query, the identification of the day and time when a query report was created that query automatically provides on each report.

tn3270. An informally defined protocol for transmitting 3270 data streams over Telnet.

token. (1) In a local area network, the symbol of authority passed successively from one data station to another to indicate the station temporarily in control of the transmission medium. Each data station has an opportunity to acquire and use the token to control the medium. A token is a particular message or bit pattern that signifies permission to transmit. (T) (2) In LANs, a sequence of bits passed from one device to another along the transmission medium. When the token has data appended to it, it becomes a frame.

token-bus network. A bus network in which a token passing procedure is used. (T)

token passing. In a token-ring network, the process by which a node captures a token; inserts a message, addresses, and control information; changes the bit pattern of the token to the bit pattern of a frame; transmits the frame; removes the frame from the ring when it has made a complete circuit; generates another token; and transmits the token on the ring where it can be captured by the next node that is ready to transmit.

token ring. (1) According to IEEE 802.5, network technology that controls media access by passing a token (special packet or frame) between media-attached stations. (2) A FDDI or IEEE 802.5 network with a ring topology that passes tokens from one attaching ring station (node) to another. (3) See also *local area network (LAN)*.

token-ring network. (1) A ring network that allows unidirectional data transmission between data stations, by a token passing procedure, such that the transmitted data return to the transmitting station. (T) (2) A network that uses a ring topology, in which tokens are passed in a circuit from node to node. A node that is ready to send can capture the token and insert data for transmission.

transaction. In IMS/VS, a specific set of input data that triggers execution of a specific process or job. A transaction is a message destined for an application program.

transaction code. The first 1 to 8 characters of the first segment of a message sent to IMS/VS. The transaction code identifies the application program for which the message is intended.

Transmission Control Protocol (TCP). A communications protocol used in the Internet and in any network that follows the U.S. Department of Defense standards for internetwork protocol. TCP provides a reliable host-to-host protocol between hosts in packet-switched communications networks and in interconnected systems of such networks. It uses the Internet Protocol (IP) as the underlying protocol.

Transmission Control Protocol/Internet Protocol (TCP/IP). A set of communications protocols that support peer-to-peer connectivity functions for both local and wide area networks.

transport layer. In the Open Systems Interconnection reference model, the layer that provides a reliable end-to-end data transfer service. There may be relay open systems in the path. (T) See also *Open Systems Interconnection reference model*.

trap. In the Simple Network Management Protocol (SNMP), a message sent by a managed node (agent function) to a management station to report an exception condition.

Trivial File Transfer Protocol (TFTP). In the Internet suite of protocols, a protocol for file transfer that requires minimal overhead and minimal capability. TFTP uses the connectionless datagram delivery services of the User Datagram Protocol (UDP), which allows hosts that have no disk storage to implement TFTP in read-only memory (ROM) and use it to boot themselves.

TRUE. Task-Related User Exit.

TSO. Time Sharing Option.

TSR program. Terminate-and-stay-resident program.

U

UDP. User Datagram Protocol.

UID. User number.

uniform resource locator (URL). For HTML documents and for the World Wide Web, a sequence of characters that represent information resources. This sequence of characters includes (a) the abbreviated name of the protocol used to access the information resource and (b) the information used by the protocol to locate the information resource. For example, in the context of the Internet, these are abbreviated names of some protocols used to access various information resources: `http`, `ftp`, `gopher`, `telnet`, and `news`; and this is the URL for the IBM home page: `http://www.ibm.com/`.

UNIX operating system. An operating system developed by Bell Laboratories that features multiprogramming in a multiuser environment. The UNIX operating system was originally developed for use on minicomputers but has been adapted for mainframes and microcomputers. The AIX operating system is IBM's implementation of the UNIX operating system.

unmarshall. To copy data from a remote procedure call (RPC) packet. Stubs perform unmarshalling. Contrast with *marshall*.

URL. Uniform resource locator.

user. Any person or any thing that may issue or receive commands and messages to or from the information processing system. (T)

user-application network. A configuration of data processing products, such as processors, controllers, and terminals, established and operated by users for the purpose of data processing or information exchange, which may use services offered by communication common carriers or telecommunication Administrations. (T) Contrast with *public network*.

User Datagram Protocol (UDP). In the Internet suite of protocols, a protocol that provides unreliable, connectionless datagram service. It enables an application program on one machine or process to send a datagram to an application program on another machine or process. UDP uses the Internet Protocol (IP) to deliver datagrams.

user exit. (1) A point in an IBM-supplied program at which a user exit routine may be given control. (2) A programming service provided by an IBM software product that may be requested during the execution of an application program for the service of transferring control back to the application program upon the later occurrence of a user-specified event.

user number (UID). In the AIX operating system, a number that uniquely identifies a user to the system. It is the internal number associated with a user ID.

user profile. In computer security, a description of a user that includes such information as user ID, user name, password, access authority, and other attributes obtained at logon.

V

V.35. In data communication, a specification of the CCITT that defines the list of definitions for interchange circuits between data terminal equipment (DTE) and data circuit-terminating equipment (DCE) at various data rates.

virtual address. The address of a location in virtual storage. A virtual address must be translated into a real address in order to process the data in processor storage.

Virtual Telecommunications Access Method (VTAM). An IBM licensed program that controls communication and the flow of data in an SNA network. It provides single-domain, multiple-domain, and interconnected network capability.

volume table of contents (VTOC). (1) A table on a direct access volume that describes each data set on the volume. (2) An area on a disk or diskette that describes the location, size, and other characteristics of each file and library on the disk or diskette.

VTAM. Virtual Telecommunications Access Method.

VTOC. Volume table of contents.

W

WAN. Wide area network.

well-known port. In Internet communications, one of a set of preassigned protocol port numbers that address specific functions used by transport level protocols (for example, TCP and UDP).

wide area network (WAN). (1) A network that provides communication services to a geographic area larger than that served by a local area network or a metropolitan area network, and that may use or provide public communication facilities. (T) (2) A data communication network designed to serve an area of hundreds or thousands of miles; for example, public and private packet-switching networks, and national telephone networks. (3) Contrast with *local area network (LAN)* and *metropolitan area network (MAN)*.

wideband. Synonym for *broadband*. (T)

widget. (1) In the AIX operating system, a graphic device that can receive input from the keyboard or mouse and can communicate with an application or with another widget by means of a callback. Every widget is a member of only one class and always has a window associated with it. (2) The fundamental data type of the Enhanced X-Windows Toolkit. (3) An object that provides a user-interface abstraction; for example, a Scrollbar widget. It is the combination of an Enhanced X-Windows window (or subwindow) and its associated semantics. A widget implements procedures through its widget class structure.

wildcard character. Synonym for *pattern-matching character*.

window. A portion of a display surface in which display images pertaining to a particular application can be presented. Different applications can be displayed simultaneously in different windows. (A)

WinSock application programming interface (API). A socket-style transport interface developed for the Windows family of operating systems.

working directory. The directory to which an application program will write data if no path is specified.

workstation. (1) A functional unit at which a user works. A workstation often has some processing capability. (T) (2) One or more programmable or nonprogrammable devices that allow a user to do work. (3) A terminal or microcomputer, usually one that is connected to a mainframe or to a network, at which a user can perform applications.

workstation network. A low-cost broadband network that allows attached personal computers to communicate and share resources.

World Wide Web (WWW). A network of servers that contain programs and files. Many of the files contain hypertext links to other documents available through the network.

WWW. World Wide Web.

X

X Client. An application program that uses the X protocol to communicate windowing and graphics requests to an X Server.

X protocol. The rules for writing programs that interact with the X Window System.

X Server. A program that interprets the X protocol and controls one or more screens, a pointing device, a keyboard, and various resources associated with the X Window System, such as graphics contexts, pixmaps, and color tables.

X Window System. A software system, developed by the Massachusetts Institute of Technology, that allows the user of a display to concurrently use multiple application programs through different windows of the display. The application programs may execute on different computers.

X.21. An International Telegraph and Telephone Consultative Committee (CCITT) recommendation for a general-purpose interface between data terminal equipment and data circuit-terminating equipment for synchronous operations on a public data network.

X.25. (1) An International Telegraph and Telephone Consultative Committee (CCITT) recommendation for the interface between data terminal equipment and packet-switched data networks. (2) See also *packet switching*.

X.25 interface. An interface consisting of a data terminal equipment (DTE) and a data circuit-terminating equipment (DCE) in communication over a link using the procedures described in the CCITT Recommendation X.25.

X.25 NCP Packet Switching Interface (NPSI). An IBM licensed program that allows SNA users to communicate over packet switching data networks that have interfaces complying with CCITT Recommendation X.25. It allows SNA programs to communicate with SNA or non-SNA equipment over such networks.

XDR. Abbreviation for “eXternal Data Representation.”

XRF. Extended recovery facility.

Z

zombie. In the UNIX operating system, a child process that has finished and is inactive, but has not yet been killed by its parent process.

zone. In AppleTalk networks, a subset of nodes within an internet.

Bibliography

This bibliography lists the publications for IBM TCP/IP products.

For publications on IBM TCP/IP V3R2 for MVS, see "TCP/IP for MVS Publications."

IBM TCP/IP Publications

The following sections describe the books associated with IBM TCP/IP products.

TCP/IP for MVS Publications

- *TCP/IP Version 3 for OpenEdition MVS: Applications Feature Guide*, SC31-8069

  This book explains how to plan for, install, customize, and use the OpenEdition MVS Applications Feature. The Feature consists of applications and interfaces for direct access to the OpenEdition MVS environment. For example, users of the Feature can use MVS, UNIX, or AIX commands to transfer files, log in to the OpenEdition environment without going through TSO, and run commands remotely. This book also explains how to improve performance and diagnose problems when using the Feature.

- *TCP/IP for MVS: Application Programming Interface Reference*, SC31-7187

  This book describes the syntax and semantics of program source code necessary to write your own application programming interface (API) into TCP/IP. You can use this interface as the communication base for writing your own client or server application. You can also use this book to adapt your existing applications to communicate with each other using sockets over TCP/IP.

- *TCP/IP for MVS: CICS TCP/IP Socket Interface Guide and Reference*, SC31-7131

  This book is for people who want to set up, write application programs for, and diagnose problems with the socket interface for CICS using TCP/IP for MVS.

- *TCP/IP for MVS: Customization and Administration Guide*, SC31-7134

  This book is for people who want to configure, customize, administer, and maintain TCP/IP for MVS. Familiarity with MVS operating system, TCP/IP protocols, and IBM Time Sharing Option (TSO) is recommended.

- *TCP/IP for MVS: Diagnosis Guide*, LY43-0105

  This book explains how to diagnose TCP/IP problems and how to determine whether a specific problem is in the IBM TCP/IP for MVS product code. It explains how to gather information for and describe problems to the IBM Software Support Center.

- *TCP/IP for MVS: IMS TCP/IP Application Development Guide and Reference*, SC31-7186

  This book is for programmers who want to write application programs that use the IMS TCP/IP application development services provided by IBM TCP/IP for MVS.

- *TCP/IP for MVS: Messages and Codes*, SC31-7132

  This book explains the informational and error messages issued by IBM TCP/IP for MVS. It can help users, operators, or system programmers to diagnose and fix problems identified by TCP/IP for MVS error messages.

- *TCP/IP for MVS: Network Print Facility*, SC31-8074

  This book is for system programmers and network administrators who need to prepare their network to route VTAM, JES2, or JES3 printer output to remote printers using TCP/IP for MVS.

- *TCP/IP for MVS: Offloading TCP/IP Processing*, SC31-7133

This book is for people who want to install and configure the Offload feature on IBM 3172 Model 3 Interconnect Controllers. This book is also for people who want to use and customize the Offload feature of TCP/IP for MVS.

- *TCP/IP for MVS: Planning and Migration Guide*, SC31-7189

  This book is intended to help you plan for TCP/IP for MVS whether you are migrating from a previous version or installing TCP/IP for MVS for the first time. This book also identifies the suggested and required modifications needed to enable you to use the enhanced functions provided with TCP/IP for MVS.

- *TCP/IP: Performance Tuning Guide*, SC31-7188

  This book describes how to improve the performance of your network operations.

- *TCP/IP for MVS: Programmer's Reference*, SC31-7135

  This book describes the syntax and semantics of a set of high-level application functions that you can use to program your own applications in a TCP/IP environment. These functions provide support for application facilities, such as user authentication, distributed databases, distributed processing, network management, and device sharing.

  This book is for people who want to use the supplied interfaces while writing application programs that access TCP/IP for MVS. Familiarity with the MVS operating system, TCP/IP protocols, and IBM Time Sharing Option (TSO) is recommended.

- *TCP/IP for MVS: User's Guide*, SC31-7136

  This book is for people who want to use TCP/IP for MVS for data communication. Familiarity with MVS operating system and IBM Time Sharing Option (TSO) is recommended.

TCP/IP for VM Publications

The following list describes books in the IBM TCP/IP for VM library.

- *IBM TCP/IP Version 2 Release 2 for VM: Messages and Codes*, SC31-6151.

  This book is for system programmers who want to diagnose and fix problems identified by TCP/IP for VM error messages.

- *IBM TCP/IP Version 2 Release 2 for VM: Offload of TCP/IP Processing*, SC31-6176.

  This book is for people who want to install and configure the Offload feature on IBM 3172 Model 3 Interconnect Controllers. This book is also for people who want to use and customize the Offload feature for TCP/IP for VM.

- *IBM TCP/IP Version 2 Release 2 for VM: Planning and Customization*, SC31-6082.

  This book is for system programmers who want to plan and customize the TCP/IP for VM environment.

- *IBM TCP/IP Version 2 Release 2 for VM: Programmer's Reference*, SC31-6084.

  This book is for application and system programmers who want to write application programs that use TCP/IP for VM. Application programmers should know the VM operating system.

- *IBM TCP/IP Version 2 Release 2 for VM: User's Guide*, SC31-6081.

  This book is for people who want to use TCP/IP for VM for data communication. Familiarity with VM operating system, IBM Command Processor (CP), and IBM Conversational Monitor System (CMS) is recommended.

TCP/IP for OS/2 Publication

- *IBM TCP/IP Version 3.0 for OS/2: Programmer's Reference*, SC31-6077.

  This book provides application and system programmers with the information required to write application programs that use TCP/IP for OS/2. Programmers should know the OS/2 operating system.

TCP/IP for DOS Publications

The following list describes books in the IBM TCP/IP for DOS library.

- *IBM TCP/IP Version 2.1.1 for DOS: Command Reference*, SX75-0083.

  This book is for people who use a workstation with TCP/IP for DOS, such as end users and system programmers. The people who use this book should be familiar with DOS and the workstation, understand DOS operating system concepts, and be familiar with the *IBM TCP/IP Version 2.1.1 for DOS: User's Guide*.

- *IBM TCP/IP Version 2.1.1 for DOS: Installation and Administration*, SC31-7047.

  This book provides system programmers, network administrators, and workstation users responsible for installing TCP/IP for DOS with the information required to plan and implement the installation of TCP/IP for DOS. The topics include hardware and software requirements, pre-installation system performance considerations, instructions for installing TCP/IP for DOS, instructions for customizing the TCP/IP for DOS environment, and installation examples.

- *IBM TCP/IP Version 2.1.1 for DOS: Programmer's Reference*, SC31-7046.

  This book is for application and system programmers to aid them in writing application programs that use TCP/IP for DOS on a workstation. Application programmers should know the DOS operating system and multitasking operating system concepts. Application programmers should be knowledgeable in the C programming language.

- *IBM TCP/IP Version 2.1.1 for DOS: User's Guide*, SC31-7045.

  This book is for people who use a workstation with TCP/IP for DOS, such as end users and system programmers. The people who use this book should be familiar with DOS and the workstation, and also understand DOS operating system concepts.

TCP/IP for AIX (RS/6000, PS/2, RT, 370) Publications

The following list shows books in the TCP/IP for AIX library.

- *AIX Operating System TCP/IP User's Guide*, SC23-2309.
- *AIX PS/2 TCP/IP User's Guide*, SC23-2047.
- *TCP/IP for IBM X-Windows on DOS 2.1*, SC23-2349.

TCP/IP for AS/400 Publications

The following list shows books in the TCP/IP for AS/400 library.

- *IBM AS/400 Communications: TCP/IP Guide*, SC41-9875.
- *IBM AS/400 Communications: User's Guide*, SC21-9601.

Other IBM TCP/IP Publications

The following list shows other available IBM TCP/IP books.

- *IBM Local Area Network Technical Reference*, SC30-3383.
- *IBM TCP/IP for VM and MVS: Diagnosis Guide*, LY43-0013.
- *TCP/IP and National Language Support*, GG24-3840.
- *TCP/IP Introduction*, GC31-6080.

- *TCP/IP Tutorial and Technical Overview*, GG24-3376.

IBM Operating System Publications

The following lists show books about various IBM operating systems.

AIX Publications

- *AIX Communications Concepts and Procedures for IBM RISC System/6000*, GC23-2203.
- *AIX Communications Programming Concepts*, SC23-2206.
- *IBM AIX Operating System Technical Reference, Volume 1*, SC23-2300.
- *IBM AIX Operating System Technical Reference, Volume 2*, SC23-2301.

AS/400 Publications

- *IBM AS/400 CL Reference Manual Volume 1*, SC21-9775.
- *IBM AS/400 CL Reference Manual Volume 2*, SC21-9776.
- *IBM AS/400 CL Reference Manual Volume 3*, SC21-9777.
- *IBM AS/400 CL Reference Manual Volume 4*, SC21-9778.
- *IBM AS/400 CL Reference Manual Volume 5*, SC21-9779.
- *IBM AS/400 Communications: APPN Network User's Guide*, SC21-8188.
- *IBM AS/400 Communications: Programmer's Guide*, SC21-9590.
- *IBM AS/400 Communications: User's Guide*, SC21-9601.
- *IBM AS/400 Device Configuration Guide*, SC21-8106.
- *IBM AS/400 Programming: Command Reference Summary*, SC21-8076.
- *IBM AS/400 Programming: Data Management Guide*, SC21-9658.
- *IBM AS/400 System Operations: Database Coordinator' Guide*, SC21-8086.
- *IBM AS/400 System Operations: Operator's Guide*, SC21-8082.

DOS Publications

- *DOS Getting Started Version 5.00*, SA40-0637.
- *DOS 5.02 Technical Reference*, S16G-4559.
- *DOS/Windows Client Getting Started*, SC09-3000.
- *PC DOS 6.1 Command Reference*, S71G-3634.

MVS Publications

For a complete description of the library for MVS/ESA Version 4, see *MVS/ESA Library Guide for MVS/ESA System Product Version 4*, GC28-1601. For a complete description of the library for MVS/ESA Version 5, see *OS/390 Information Roadmap*, GC28-1727. Selected books from these libraries are listed in the following sections. See also "JES Publications" on page 282.

MVS/ESA Version 4

- *MVS/ESA Application Development Guide: Assembler Language Programs*, GC28-1644.
- *MVS/ESA Application Development Guide: Authorized Assembler Language Programs*, GC28-1645.
- *MVS/ESA Application Development Reference: Services for Authorized Assembler Language Programs, Volume 1*, GC28-1647.

- *MVS/ESA Application Development Reference: Services for Authorized Assembler Language Programs, Volume 2*, GC28-1648.
- *MVS/ESA Diagnosis: Component Reference*, LY28-1814.
- *MVS/ESA Diagnosis: Procedures*, LY28-1667.
- *MVS/ESA Diagnosis: System Reference*, LY28-1820.
- *MVS/ESA Diagnosis: Using Dumps and Traces*, LY28-1813.
- *MVS/ESA Initialization and Tuning Guide*, GC28-1634.
- *MVS/ESA JCL Reference*, GC28-1654.
- *MVS/ESA Service Aids*, GC28-1669.
- *MVS/ESA System Messages, Volume 4 (IEC–IFD)*, GC28-1659.
- *MVS/ESA TSO Programming* GC28-1671.

MVS/ESA Version 5

- *OS/390 MVS Assembler Services Guide*, GC28-1762.
- *OS/390 MVS Auth Assembler Services Guide*, GC28-1763.
- *OS/390 MVS Auth Assembler Services Reference ALE-DYN*, GC28-1764.
- *OS/390 MVS Auth Assembler Services Reference ENF-ITT*, GC28-1765.
- *OS/390 MVS Diagnosis: Procedures*, LY28-1082.
- *OS/390 MVS Diagnosis: Reference*, LY28-1084.
- *OS/390 MVS Diagnosis: Tools and Service Aids*, LY28-1085.
- *OS/390 MVS Initialization and Tuning Guide*, SC28-1751.
- *OS/390 MVS IPCS Commands*, GC28-1754.
- *OS/390 MVS IPCS User's Guide*, GC28-1756.
- *OS/390 MVS JCL Reference*, GC28-1757.
- *OS/390 MVS Sysplex Services Guide*, GC28-1771.
- *OS/390 MVS Sysplex Services Reference*, GC28-1772.
- *OS/390 MVS System Commands*, GC28-1781.
- *OS/390 MVS System Messages, Vol 4 (IEC-IFD)*, GC28-1787.
- *OS/390 MVS Using the Functional Subsystem Interface*, SC28-1911.

MVS Utilities

DFSMS/MVS Utilities, SC26-4926.

RMF User's Guide, GC33-6483.

OS/2 Publications

- *IBM OS/2 Warp Server Up and Running!*, S25H-8004
- *IBM Official Guide to Using OS/2 Warp*, ISBN 1-56884-466-2 (Karla Stagray and Linda S. Rogers; Foster City, CA: An IBM Press Book published by IDG Books Worldwide, Inc., 1995)
- *IBM OS/2 Warp Internet Connection: Your Key to Cruising the Internet and the World Wide Web*, ISBN 1-56884-465-4 (Deborah Morrison; Foster City, CA: An IBM Press Book published by IDG Books Worldwide, Inc., 1995)

OS/390 Publications

- *OS/390 Information Roadmap*, GC28-1727

  This book describes the documentation for the specific elements included in OS/390.

- *OS/390 Up and Running!*, GC28-1726

  This book is intended to help you plan for the installation of OS/390. It describes migration, installation, hardware and software requirements, and coexistence considerations.

VM Publications

- *VM/ESA CMS Command Reference Summary*, SX24-5249.
- *VM/ESA CP Planning and Administration for 370*, SC24-5430.
- *VM/ESA CP Programming Services for 370*, SC24-5435.
- *VM/ESA Group Control System Reference for 370*, SC24-5426.
- *VM/ESA: Library Guide and Master Index*, GC23-0367.
- *VM/ESA: Master Index for 370*, GC24-5436.
- *VM/ESA Service Introduction and Reference*, SC24-5444.
- *VM/SP CMS Command Reference*, ST00-1981.
- *VM/SP Group Control System Macro Reference*, SC24-5250.
- *VM/SP Installation Guide*, SC24-5237.
- *VM/SP High Performance Option: Library Guide and Master Index*, GC23-0187.
- *VM/SP System Facilities for Programming*, SC24-5288.
- *VM/XA CP Programming Services*, SC23-0370.
- *VM/XA Diagnosis Reference*, LY27-8054.
- *VM/XA Installation and Service*, SC23-0364.
- *VM/XA SP Group Control System Command and Macro Reference*, SC23-0433.

IBM Software Publications

The following sections describe the books associated with IBM software products.

ACF/VTAM Publications

The following list shows books in the VTAM Version 4 library.

- *VTAM AnyNet Feature: Guide to SNA over TCP/IP*, SC31-6527.
- *VTAM Customization*, LY43-0068.
- *VTAM Diagnosis*, LY43-0069.
- *VTAM Messages and Codes*, SC31-6546.
- *VTAM Installation and Migration Guide*, GC31-6547.
- *VTAM Network Implementation Guide*, SC31-6548.
- *VTAM Operation*, SC31-6549.
- *VTAM Programming*, SC31-6550.
- *VTAM Programming for LU 6.2*, SC31-6551.
- *VTAM Release Guide*, GC31-6555.
- *VTAM Resource Definition Reference*, SC31-6552.

CICS/ESA Publications

The following lists show books in the CICS/ESA Version 3 Release 3 library.

- *CICS/ESA 3.3 Customization Guide*, SC33-0665.
- *CICS/ESA 3.3 Diagnosis Reference*, LY33-6072.
- *CICS/ESA 3.3 CICS-Supplied Transactions*, SC33-0669.

DATABASE 2 Publications

The following lists show books in the DATABASE 2 library.

DATABASE 2 Version 2

- *IBM DATABASE 2 Version 2: Administration Guide*, SC26-4374.
- *IBM DATABASE 2 Version 2: Application Programming and SQL Guide*, SC26-4377.
- *IBM DATABASE 2 Version 2: Messages and Codes*, SC26-4379.
- *IBM DATABASE 2 Version 2: Reference Summary*, SX26-3771.
- *IBM DATABASE 2 Version 2: SQL Reference*, SC26-4380.

DATABASE 2 Version 3

- *IBM DATABASE 2 Version 3: DB2 Administration Guide*, SC26-4888.
- *IBM DATABASE 2 Version 3: DB2 Application Programming and SQL Guide*, SC26-4889.
- *IBM DATABASE 2 Version 3: DB2 Messages and Codes*, SC26-4892.
- *IBM DATABASE 2 Version 3: DB2 Reference Summary*, SX26-3801.
- *IBM DATABASE 2 Version 3: DB2 SQL Reference*, SC26-4890.

GDDM Publications

The following list shows books in the GDDM Version 3 Release 1 library.

- *GDDM Base Application Programming Guide*, SC33-0867.
- *GDDM Base Application Programming Reference*, SC33-0868.
- *GDDM/MVS Installation: Planning, Testing, and Servicing*, SC33-0872.
- *GDDM 3.1 Diagnosis Guide*, SC33-0870.
- *GDDM 3.1 User's Guide*, SC33-0875.

IMS/ESA Publications

- *IMS/ESA V3R1 Application Programming: DL/I Calls*, SC26-4274.
- *IMS/ESA V3R1 System Administration Guide*, SC26-4282.
- *IMS/ESA V3R1 Utilities Reference*, SC26-4284.

ISPF Publication

ISPF Dialog Management Guide and Reference, SC34-4266.

JES Publications

JES2

- *MVS/ESA JES2 Commands*, GC23-0084 (for MVS/ESA Version 4), GC28-1443 (for MVS/ESA Version 5).
- *MVS/ESA JES2 Initialization and Tuning Guide*, SC23-0082 (for MVS/ESA Version 4), SC28-1453 (for MVS/ESA Version 5)
- *MVS/ESA JES2 Initialization and Tuning Reference*, SC23-0083 (for MVS/ESA Version 4), SC28-1454 (for MVS/ESA Version 5)

JES3

- *MVS/ESA JES3 Commands*, GC23-0090 (for MVS/ESA Version 4).
- *MVS/ESA JES3 Initialization and Tuning Guide*, SC23-0088 (for MVS/ESA Version 4), SC28-1455 (for MVS/ESA Version 5)
- *MVS/ESA JES3 Initialization and Tuning Reference*, SC23-0089 (for MVS/ESA Version 4), SC28-1456 (for MVS/ESA Version 5)

MVS/DFP Publications

- *MVS/DFP Version 3 Release 3: Customizing and Operating the Network File System Server*, SC26-4832.
- *MVS/DFP Version 3 Release 3: Macro Instructions for Data Sets*, S26-4747.
- *MVS/DFP Version 3 Release 3: Using Data Sets*, SC26-4749.
- *MVS/DFP Version 3 Release 3: Using the Network File System Server*, SC26-4732.

Network Control Program (NCP) Publications

- *ACF/NCP V7R1 IP Router Planning and Installation Guide*, GG24-3974.
- *NCP and EP Reference*, LY43-0029.
- *NCP, SSP, and EP Generation and Loading Guide*, SC31-6221.
- *NCP, SSP, and EP Resource Definition Guide*, SC31-6223.
- *NCP, SSP, and EP Resource Definition Reference*, SC31-6224.

NetView Publications

The following list shows books in the NetView Version 3 library.

- *NetView for MVS Administration and Security Reference*, SC31-8045.
- *NetView for MVS Application Programming Guide*, SC31-8061.
- *NetView for MVS Automation Implementation*, SC31-8050.
- *NetView for MVS Bridge Implementation*, SC31-6131.
- *NetView for MVS Customization Guide*, SC31-8052.
- *NetView for MVS Customization: Using Assembler*, SC31-8053.
- *NetView for MVS Customization: Using PL/I and C*, SC31-8054.
- *NetView for MVS Customization: Writing Command Lists*, SC31-8055.
- *Managing Your Future: NetView for MVS*, G325-3530.
- *NetView for MVS Installation and Administration Guide*, SC31-8043.
- *NetView for MVS Messages*, SC31-8046.
- *NetView for MVS Command Reference*, SC31-8047.
- *NetView for MVS Problem Determination and Diagnosis*, LY43-0102.

- *NetView for MVS Resource Alerts Reference*, SC31-7097.
- *NetView for MVS RODM and GMFHS Programming Guide*, SC31-8049.
- *NetView for MVS User's Guide*, SC31-8056.
- *NIAF/2 Guide*, SC31-8044.

Networking Systems Cross-Product Library

The following list shows books in the Networking Systems cross-product library.

- *Planning Aids: Pre-Installation Planning Checklist for NetView, NCP, and VTAM*, SX75-0092.
- *Planning for Integrated Networks*, SC31-8062.
- *Planning for NetView, NCP, and VTAM*, SC31-8063.

OpenEdition MVS Publications

The following list shows selected books in the OpenEdition MVS library.

- *OS/390 OpenEdition Introduction*, GC28-1889
- *OS/390 OpenEdition Planning*, SC28-1890

Programming Publications

The following list shows books about various programming applications.

- *IBM C/370 Diagnosis Guide and Reference*, LY09-1804 (feature 8082).
- *IBM C/370 General Information Manual*, GC09-1386.
- *IBM C/370 Installation and Customization Guide Version 2 Release 1.0*, GC09-1387.
- *IBM C/370 Programming Guide*, SC09-1384.
- *IBM C/370 Reference Summary*, SX09-1211.
- *IBM C/370 User's Guide*, SC09-1264.
- *OS/390 C/C++ Run-Time Library Reference*, SC28-1663.
- *IBM TSO Extensions CLISTs*, SC28-1876.
- *IBM TSO Extensions Command Language Reference*, GX23-0015.
- *IBM TSO Extensions Interactive Data Transmission Facility: User's Guide*, SC28-1104.
- *IMS/ESA V3R1 Application Programming: DL/I Calls*, SC26-4274.
- *HiPPI User's Guide and Programmer's Reference*, SA23-0369.
- *Parallel I/O Access Methods Programmer's Guide*, SC26-4648.
- *VS Pascal Application Programming Guide*, SC26-4319.
- *VS Pascal Diagnosis Guide and Reference*, LY27-9525.
- *VS Pascal General Information*, GT00-2664.
- *VS Pascal Installation and Customization for MVS*, SC26-4321.
- *VS Pascal Installation and Customization for VM*, SC26-4342.
- *VS Pascal Language Reference*, SC26-4320.

RACF Publications

The following list shows books in the RACF library.

- *IBM Resource Access Control Facility (RACF): General Information Manual*, GT00-2820.
- *IBM Resource Access Control Facility (RACF): User's Guide*, SC28-1341.
- *External Security Interface (RACROUTE) Macro Reference*, GC28-1366.
- *RACF Publications Order Guide*, GX22-0002.
- *Resource Access Control Facility (RACF) Security Administrator's Guide*, SC28-1340.
- *System Programming Library: RACF*, SC28-1343.

SMP/E Publications

The following list shows books in the SMP/E Release 8 library.

- *SMP/E Diagnosis Guide*, SC23-3130.
- *SMP/E Messages and Codes*, SC28-1107.
- *SMP/E Reference*, SC28-1107.
- *SMP/E Reference Summary*, SX22-0006.
- *SMP/E User's Guide*, SC28-1302.

VSAM Publications

The following list shows books in the VSAM library.

- *MVS/370 VSAM Administration Guide*, GC26-4066.

X.25 NPSI Publications

The following list shows books in the X.25 NPSI library.

- *X.25 Network Control Program Packet Switching Interface Diagnosis, Customization, and Tuning Version 3*, LY30-5610.
- *X.25 Network Control Program Packet Switching Interface Host Programming*, SC30-3502.
- *X.25 Network Control Program Packet Switching Interface Planning and Installation*, SC30-3470.

IBM Hardware Publications

The following sections describe the books associated with IBM hardware products.

System/370 and System/390 Publications

The following list shows the principles of operation manuals for the System/370 and System/390 processors.

- *IBM ESA/370 Principles of Operation*, SA22-7200.
- *IBM ESA/390 Principles of Operation*, SA22-7201.
- *IBM System/370 Extended Architecture Principles of Operation*, SA22-7085.
- *IBM System/370 Principles of Operation*, GA22-7000.
- *S/360, S/370, and S/390 I/O Interface Channel to Channel Control Unit OEMI*, GA22-6974.

3172 Interconnect Controller Publications

The following list shows books in the IBM 3172 Interconnect Controller library.

- *IBM Interconnect Controller Program User's Guide*, SC30-3525.
- *IBM 3172 Interconnect Controller Installation and Service Guide*, GA27-3861.
- *IBM 3172 Interconnect Controller Operator's Guide*, GA27-3860.
- *IBM 3172 Interconnect Controller Planning Guide*, GA27-3867.
- *IBM 3172 Interconnect Controller Status Codes*, GA27-3951.

3270 Information Display System Publication

3270 Information Display System: 3270 Data Stream Programmer's Reference, GA23-0059.

8232 LAN Channel Station Publications

The following list shows books in the IBM 8232 LAN Channel Station library.

- *IBM LAN Channel Support Program: Version 1.0 User's Guide*, SC30-3458.
- *IBM 8232 LAN Channel Station: Installation and Testing*, GA27-3796.
- *IBM 8232 LAN Channel Station: Operating Guide*, GA27-3785.

9370 Publications

The following list shows books in the 9370 library.

- *IBM 9370 Information System: Using the X.25 Communications Subsystem*, SA09-1742.
- *IBM 9370 Information System X.25 Communications Subsystem Description*, SA09-1743.
- *VM/ESA: Connectivity Planning, Administration, and Operation Release 1*, SC24-5448.

Other TCP/IP-Related Publications

The following sections describe other books associated with TCP/IP.

- *The Art of Distributed Application: Programming Techniques for Remote Procedure Calls*, John R. Corbin, Springer-Verlog, 1991.
- *CAE Specification: X/Open Transport Interface (XTI),* X/Open Company Ltd., U. K., 1992, SC31-8005.
- *IEEE Network Magazine*, July 1990.
- *Internetworking with TCP/IP Volume I: Principles, Protocols, and Architecture*, Douglas E. Comer, Prentice Hall, Englewood Cliffs, New Jersey, 1991 SC31-6144.
- *Internetworking with TCP/IP Volume II: Implementation and Internals*, Douglas E. Comer, Prentice Hall, Englewood Cliffs, New Jersey, 1992 SC31-6145.
- *Internetworking with TCP/IP Volume III: Client-Server Programming and Applications*, Douglas E. Comer, Prentice Hall, Englewood Cliffs, New Jersey, 1991 SC31-7194.
- *Interoperability Report*, Volume 3, No. 3, March 1989.
- "MIB II Extends SNMP Interoperability," C. Vanderberg, *Data Communications*, October 1990.
- *"Network Management and the Design of SNMP,"* J.D. Case, J.R. Davin, M.S. Fedor, M.L. Schoffstall.
- *"Network Management of TCP/IP Networks: Present and Future,"* A. Ben-Artzi, A. Chandna, V. Warrier.

- *The Simple Book: An Introduction to Management of TCP/IP-based Internets*, Marshall T Rose, Prentice Hall, Englewood Cliffs, New Jersey, 1993.
- "Special Issue: Network Management and Network Security," *ConneXions-The Interoperability Report*, Volume 4, No. 8, August 1990.
- *TCP/IP Illustrated Volume 1*, W. Richard Stevens, Addison Wesley, 1994. ISBN 0-201-63346-9.
- *UNIX Programmer's Reference Manual*, (4.3 Berkeley Software Distribution, Virtual VAX-11 Version). Department of Electrical Engineering and Computer Science. University of California, Berkeley, 1988.

Hewlett-Packard/Apollo (NCS) Publications

The following list shows NCS books.

- *Managing the NCS Location Broker*, Apollo Computer Inc., 330 Billerica Road, Chelmsford, MA 01824, 1988. Apollo Order No. 011895-A00.
- *Network Computing Architecture*, Lisa Zahn, (Terence H. Dineen, Paul J. Leach, Elizabeth A. Martin, Nathaniel W. Mishkin, Joseph N. Pato, Geoffrey L. Wyant), Apollo Computer Inc., a subsidiary of Hewlett-Packard Company, Chelmsford, Massachusetts, Prentice Hall, Englewood Cliffs, New Jersey 07632, 1990.
- *Network Computing Architecture (NCA) Protocol Specifications*, Apollo Computer Inc., 330 Billerica Road, Chelmsford, MA 01824, (508) 256-6600, 1989. Apollo Order No. 010201-A00
- *Network Computing System (NCS) Reference*, Apollo Computer Inc., 330 Billerica Road, Chelmsford, MA 01824, 1987. Apollo Order No. 010200, Revision 00.
- *Network Computing System Reference Manual*, Mike Kong, (Terence H. Dineen, Paul J. Leach, Elizabeth A. Martin, Nathaniel W. Mishkin, Joseph N. Pato, Geoffrey L. Wyant), Apollo Computer Inc., a subsidiary of Hewlett-Packard Company, Chelmsford, Massachusetts, Prentice Hall, Englewood Cliffs, New Jersey 07632, 1990.

HYPERchannel Publication

The following is a HYPERchannel book.

- *HYPERchannel A220 Processor Adapter* 4290007, Network Systems Corporation.

Kerberos Publications

The following list shows Kerberos books.

- Jennifer G. Steiner, Clifford Neuman, and Jeffrey I. Schiller. *"Kerberos: An Authentication Service for Open Networks."* Massachusetts Institute of Technology, 12 January 1988.
- S.P. Miller et al. *"Kerberos Authentication and Authorization System,"* Project Athena Technical Plan, Section E.2.1. Massachusetts Institute of Technology, 21 December 1987.

OSF/Motif Publications

The following list shows OSF/Motif books.

- *OSF/Motif Application Environment Specifications*, (AES), Open Software Foundation, Prentice Hall, Inc., 1990, ISBN 0-13-640483-9.
- *OSF/Motif Programmer's Guide*, Open Software Foundation, Prentice Hall, Inc., 1990, ISBN 0-13-640509-6.
- *OSF/Motif Programmer's Reference*, Open Software Foundation, Prentice Hall, Inc., 1990, ISBN 0-13-640517-7.

- *OSF/Motif Style Guide*, Open Software Foundation, Prentice Hall, Inc., 1990, ISBN 0-13-640491-X.
- *OSF/Motif User's Guide*, Open Software Foundation, Prentice Hall, Inc., 1990, ISBN 0-13-640525-8.

Sun (RPC) Publications

The following list shows Sun Microsystems books.

- *Networking on the Sun Workstation: Remote Procedure Call Programming Guide*, (800-1324-03), Sun Microsystems, Inc.
- *Network Programming*, (800-1779-10), Sun Microsystems, Inc.

X Window System Publications

The following list shows X Window System books.

- *Introduction to the X Window System*, Oliver Jones, Prentice-Hall, 1988, ISBN 0-13-499997-5.
- *PEXlib Specification and C Language Binding*, Jeff Stevenson, Hewlett-Packard Company, 1992, SR28-5116.
- *The X Window System Series* (6 volumes), O'Reilly & Associates, 1988, 1989, 1990, ISBN 0-937175-40-4, 0-937175-27-7, 0-937175-28-5, 0-937175-35-6, 0-937175-33-1, 0-937175-35-8.
- *X Protocol Reference Manual*, Adrian Nye, ed. O'Reilly & Associates, Inc., 1990, ISBN 0-937175-50-1.
- *X Window System: C Library and Protocol Reference*, Robert Scheifler, James Gettys, and Ron Newman, DEC Press, 1988, ISBN 1-55558-012-2.
- *X Window System: Programming and Applications with Xt*, Douglas A. Young, Prentice-Hall, 1989, ISBN 0-13-972167-3.
- *X Window System: Programming and Applications with Xt, OSF/Motif Edition*, Douglas A. Young, Prentice-Hall, 1990, ISBN 0-13-497074-8.
- *X Window System Technical Reference*, Steven Mikes, Addison-Wesley, 1990, ISBN 0-201-52370-1.
- *X Window System User's Guide*, Valerie Quercia and Tim O'Reilly, O'Reilly & Associates, Inc., 1990, ISBN 0-937175-14-5.

Network Architecture Publications

The following sections list books associated with network architecture.

Open Systems Interconnection (OSI) Publication

The following book is in the OSI library.

- *Open Systems Interconnection*, Z320-9757.

Systems Network Architecture (SNA) Publications

The following list shows books in the SNA library.

- *Systems Network Architecture: Sessions between Logical Units*, GC20-1868.
- *Systems Network Architecture Format and Protocol Reference Manual: Architecture Logic*, SC30-3112.
- *Systems Network Architecture Format and Protocol Reference Manual: Management Services*, SC30-3346.

- *Systems Network Architecture Formats*, GA27-3136.
- *Systems Network Architecture Network Product Formats*, LY43-0081.

Index

Printed in U.S.A.

SC31-7188-02