IBM TCP/IP Version 2 Release 2 for VM
Installation and Interoperability

Document Number GG24-3624-02

December 1992

International Technical Support Center
Raleigh

Take Note!

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

Third Edition (December 1992)

This edition applies to IBM TCP/IP Version 2 Release 2 for VM 5735-FAL for use with the VM Operating System.

Order publications through your IBM representative or the IBM branch office serving your locality. Publications are not stocked at the address given below.

An ITSC Technical Bulletin Evaluation Form for reader's feedback appears facing Chapter 1. If the form has been removed, comments may be addressed to:

IBM Corporation, International Technical Support Center
Dept. 985, Building 657
P. O. Box 12195
Research Triangle Park, NC 27709 USA

When you send information to IBM, you grant IBM a non-exclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

Abstract

This document provides information to ease the installation and use of IBM TCP/IP Version 2 Release 2 for VM. It focuses on how IBM TCP/IP Version 2 Release 2 for VM can be used to interoperate in different environments.

This document is intended for the customers and the systems engineers who will evaluate the product possibilities and install the product.

The book is organized to help you understand the architecture of IBM TCP/IP V2 for VM, the new functions and enhancements brought by V2R1 and V2R2, and the requirements and guidelines for its installation and tailoring. Each TCP/IP server is discussed and sample configuration files are provided to help you in your system customization.

The reader is assumed to have a basic knowledge of the TCP/IP protocol suite and be familiar with the VM and SNA architectures and products.

CO EN OP VM (336 pages)

Contents

Figures

Tables

Special Notices

This publication is intended to help the customer to install, use and customize IBM TCP/IP Version 2 Release 2 for VM. The information in this publication is not intended as the specification of any programming interfaces that are provided by IBM TCP/IP Version 2 Release 2 for VM. See the PUBLICATIONS SECTION of the IBM Programming Announcement for IBM TCP/IP Version 2 Release 2 for VM for more information about what publications are considered to be product documentation.

References in this publication to IBM products, programs or services do not imply that IBM intends to make these 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 IBM′s product, program, or service may be used. Any functionally equivalent program that does not infringe any of IBM′s intellectual property rights may be used instead of the IBM product, program or service.

Information in this book was developed in conjunction with use of the equipment specified, and is limited in application to those specific hardware and software products and levels.

IBM may have patents or pending patent applications covering subject matter 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 the IBM Director of Commercial Relations, IBM Corporation, Purchase, NY 10577.

The information contained in this document has not been submitted to any formal IBM test and is distributed AS IS. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer′s ability to evaluate and integrate them into the customer′s operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environments do so at their own risk.

The following document contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples contain the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

The following terms, which are denoted by an asterisk (*) in this publication, are trademarks of the International Business Machines Corporation in the United States and/or other countries:

AIX
DB2
ES/9000
ES/9370
IBM
NetView
Operating System/2 and OS/2
Presentation Manager
PROFS

Personal System/2 and PS/2
RACF
RISC System/6000
RT, RT PC and RT Personal Computer
S/370
S/390
System/88
VM/ESA
VM/XA
VTAM

The following terms, which are denoted by a double asterisk (**) in this publication, are trademarks of other companies:

APOLLO is a registered trademark of the Hewlett-Packard Company.
Hewlett-Packard is a registered trademark of the Hewlett-Packard Company.
HYPERchannel is a trademark of the Network Systems Corporation.
IEEE is a trademark of the the Institute of Electrical and Electronics Engineers, Inc..
NCS is a trademark of the Hewlett-Packard Company.
Network Computing System is a trademark of the Hewlett-Packard Company.
Network File System is a trademark of SUN Microsystems, Inc.
NFS is a trademark of SUN Microsystems, Inc.
OSF/Motif is a trademark of Open Software Foundation, Inc.
OSF is a trademark of Open Software Foundation, Inc.
Portmapper is a trademark of SUN Microsystems, Inc.
SUN Microsystems is a trademark of SUN Microsystems, Inc.
UNIX was developed and licensed by AT&T. It is a trademark of
UNIX Systems Laboratories, Inc.
X Window System is a trademark of the Massachusetts Institute of Technology.

Preface

The purpose of this document is to provide information that will ease the evaluation, installation and use of IBM TCP/IP Version 2 Release 2 for VM.

This document is intended for persons who will:

- Evaluate the interoperability functions of the product so that they meet their requirements.
- Install the product.
- Use VM/CMS to interoperate with different environments via TCP/IP.

The document shows the installation and use of the IBM TCP/IP Version 2 Release 2 for VM product on the VM hosts of the TCP/IP network at the ITSC Raleigh. All the examples provided in this document were produced in the ITSC environment.

The document is organized as follows:

- Chapter 1, “Introduction”

  This chapter gives a functional overview of IBM TCP/IP Version 2 Release 2 for VM. It also contains technical information on how TCP/IP is implemented in VM, a summary of the interoperability capabilities and a description of the new functions.

- Chapter 2, “Installation and Tailoring”

  This chapter provides information on how to install and tailor IBM TCP/IP Version 2 Release 2 for VM.

- The following chapters provide information on how to use the interoperability possibilities between a IBM TCP/IP Version 2 Release 2 for VM host and another TCP/IP host:
  - Chapter 3, “Using TCP/IP between Two VM Systems”
  - Chapter 4, “Using TCP/IP between VM and MVS”
  - Chapter 5, “Using TCP/IP between VM and OS/2”
  - Chapter 6, “Using TCP/IP between VM and DOS”
  - Chapter 7, “Using TCP/IP between VM and UNIX/AIX”
  - Chapter 8, “Using TCP/IP between VM and OS/400”
- Chapter 9, “Network Management in TCP/IP”

  This chapter lists and explains the available network management functions for IBM TCP/IP Version 2 Release 2 for VM, especially the Simple Network Management Protocol (SNMP) and the NETSTAT command. People interested in SNMP should read *Managing TCP/IP Networks using Netview and the SNMP Interface* (GG24-3696).

- Chapter 10, “Debugging”

  This chapter shows what can be done when problems occur.

- Chapter 11, “National Language Support (NLS)”

  This chapter shows how IBM TCP/IP Version 2 Release 2 for VM implements NLS. It also shows how user translate tables can be defined and when they

are used. NLS is not discussed in full detail. People interested in NLS should read *TCP/IP and National Language Support - GG24-3840*.

- Appendix B, “Name Server Installation Console Log”

  This chapter lists the console log for a step-by-step installation of a primary name server.

- The following appendixes list the configuration files for all systems shown in Figure 39 on page 267.
  - Appendix C, “Configuration Listings for VM System VM14”
  - Appendix D, “Configuration Listings for VM System VM15”
  - Appendix E, “Configuration Listings for VM System RALYESA (VMESA)”
  - Appendix F, “Configuration Listings for MVS System MVS20”
  - Appendix G, “Configuration Listings for MVS System MVS18”
  - Appendix H, “Configuration Listings for OS/400 System RALYAS4B”
  - Appendix I, “Configuration Listings for OS/2 System FRED”
  - Appendix J, “Configuration Listings for RS/6000 System RS60001”
  - Appendix K, “Configuration Listings for PS/2 AIX System PSAIX”
  - Appendix L, “3172 Configuration”

Related Publications

The following publications are considered particularly suitable for a more detailed discussion of the topics covered in this document. For a more exhaustive bibliography, please refer to the *TCP/IP Tutorial and Technical Overview*, GG24-3376.

Prerequisite Publications

- *IBM TCP/IP V2 R2 for VM: Planning and Customization*, SC31-6082-01
- *IBM TCP/IP V2 R2 for VM: Programmer's Reference*, SC31-6084-01
- *IBM TCP/IP V2 R2 for VM: User's Guide*, SC31-6081-01
- *IBM TCP/IP V2 R2 for VM: Messages and Codes*, SC31-6151-01
- *IBM TCP/IP V2 for VM and MVS: Diagnosis Guide*, LY43-0013-00 (available to IBM-licensed customers only)

Additional Publications

- *Internetworking With TCP/IP, Volume I: Principles, Protocols and Architecture*, SC31-6144
- *Internetworking With TCP/IP, Volume II: Design, Implementation and Internals*, SC31-6145
- *PROFS Extended Mail User's Guide and Installation Manual*, SH21-0044
- *AIX Version 3.2 for RISC System/6000* - Block Multiplexer Channel Adapter - User's Guide and Programming Reference*, SC23-2427
- *Virtual Machine/System Product - CMS Shared File System Administration - Release 6*, SC24-5367.

Specific TCP/IP Request for Comments (RFCs)

- *Domain Administrator's Guide*, RFC 1032
- *Domain Administrator's Operations Guide*, RFC 1033
- *Domain Names - Concepts and Facilities*, RFC 1034
- *Domain Names - Implementation and Specification*, RFC 1035
- *Routing Information Protocol*, RFC 1058
- *Management Information Base for Network Management of TCP/IP-based Internets*, RFC 1156
- *Management Information Base for Network Management of TCP/IP-based Internets: MIB II*, RFC 1213
- *A Simple Network Management Protocol (SNMP)*, RFC 1157
- *Line Printer Daemon Protocol (LPD)*, RFC 1179.
- *Proposed Standard for the Transmission of IP Datagrams over FDDI Networks*, RFC 1188.

Acknowledgements

First Edition

The advisors for this project were:

Lesia Cox
Philippe Beaupied
International Technical Support Center, Raleigh

The author of this document is:

Rolf Traber
IBM Switzerland

This publication is the result of a residency conducted at the International Technical Support Center, Raleigh

Thanks to the following people for the invaluable advice and guidance provided in the production of this document:

Gisbert Kruesemann
Ruy Medeiros
International Technical Support Center, Raleigh

TCP/IP Development, Raleigh

Thanks to the following people who helped to install and customize some of the prerequisite products for TCP/IP:

Georg Steinborn
Javier Cuan
International Technical Support Center, Raleigh

Second Edition

The project advisor for this publication was:

Philippe Beaupied
International Technical Support Center, Raleigh

The authors of this publication are:

Julio Cesar Salim Gouy
IBM Brazil

Christof Eppler
IBM Germany

Philippe Beaupied
International Technical Support Center, Raleigh

Thanks to the following people who helped creating the necessary system environment:

Carla Sadtler
International Technical Support Center, Raleigh

Also thanks for their contributions to:

Dr. Wilhelm Buerger
IBM Germany
for the introduction on the Kerberos system

TCP/IP Development
Thomas Watson Research Center, Yorktown Heights

TCP/IP Development
Research Triangle Park, Raleigh

3172 Design and Development
Research Triangle Park, Raleigh

ACF/NCP Design and Development
Research Triangle Park, Raleigh

Third Edition

The project advisor for this publication was:

Philippe Beaupied
International Technical Support Center, Raleigh

The author of this publication is:

Frédéric Débulois
IBM France

Thanks to the following people who helped create the necessary system environment:

Carla Sadtler
International Technical Support Center, Raleigh

Scott Vetter, Michael Schwartz
International Technical Support Center, Poughkeepsie

Chapter 1. Introduction

This chapter explains how TCP/IP is implemented in the VM operating system. It lists all TCP/IP functions and their hardware and software requirements.

Note

References to IBM TCP/IP V2 for VM denote that the corresponding function/feature is included both in IBM TCP/IP Version 2 Release 1 for VM and IBM TCP/IP Version 2 Release 2 for VM.

1.1 Functional Overview

TCP/IP functions can be characterized as belonging to one of five categories:

1. Connectivity and gateway functions, which handle the physical interfaces and the routing of IP datagrams.
2. Server functions are usually implemented as one or more disconnected virtual machines. As the name implies, they deliver a certain service to a client, for example a file transfer.
3. Client functions are implemented as user commands, requesting a certain service from a server anywhere in the network.
4. Network status/management functions are used to detect and solve network problems.
5. Application Programming Interfaces (APIs) allow you to write your own client/server applications.

1.1.1 Connectivity and Gateway Functions

Communication interfaces available are:

X.25: Access TCP/IP hosts via X.25.

SNALINK: Access TCP/IP hosts via SNA backbone network (LU 0 based).

IEEE** 802.5: Access TCP/IP hosts via a token-ring network.

Ethernet: Access TCP/IP hosts via Ethernet V2 and IEEE 802.3.

FDDI: Access TCP/IP hosts via Fiber Distributed Data Interface LAN.

1.1.2 Server Functions

Servers that can be implemented are:

FTP: Serve file transfer requests from any TCP/IP hosts.

SMTP: Deliver mail sent from any TCP/IP or RSCS host.

REXEC: Execution of VM/CMS commands initiated from any TCP/IP host.

LPD: Local print of files sent from any TCP/IP host.

TELNET: Remote logon from any TCP/IP host, either in 3270 mode or line by line mode.

NFS**: Remote access of a CMS minidisk in transparent mode from any TCP/IP host.

NDB: Remote access to an SQL/DS-based relational database system through the TCP/IP network.

DNS: Maps host names to internet addresses. Can be customized in an authoritative way (primary and/or secondary - SQL/DS is required) or as a cache only (SQL/DS is not required).

1.1.3 Client Functions

The following client functions may be used by a VM/CMS user:

FTP: File transfer to any TCP/IP host.

TFTP: File transfer using Trivial File Transfer Protocol to any TCP/IP host.

NOTE: Send note to any TCP/IP host using SMTP.

SENDFILE: Send a 7-bit ASCII text file to any TCP/IP host using SMTP.

TELNET: Remote logon to any TCP/IP host, either in 3270 mode or line by line mode.

LPR: Remote printout of files to any TCP/IP host.

REXEC: Execute user-supplied command on any TCP/IP host.

X Window: GDDM based graphics can be sent to and TCP/IP X server host.

DIG, NSLOOKUP: Query a name server for name to IP address mapping.

1.1.4 Network Status and Management Functions

The following functions may be used for network status and management:

NETSTAT: Show TCP/IP status of local system.

ROUTED: Dynamic update of routing tables using the RIP protocol.

RPCINFO: Display information about registered RPC procedures on any TCP/IP host.

PING: Test connection to any TCP/IP host.

SNMP: Manage TCP/IP network via SNMP VM agent (NetView* VM).

OBEYFILE: Make temporary changes to the system operation and network configuration.

1.1.5 Application Programming Interfaces (APIs)

The following APIs are available to write your own TCP/IP applications:

Sockets: Berkeley Socket library from 4.3BSD.

Kerberos: Kerberos security services and libraries.

NCS: Apollo** NCS** services and libraries.

RPC: Sun** RPC** services and libraries.

SNMP DPI: Distributed Programming Interface for own SNMP applications.

X Windows: X Window System** based on OSF/Motif**.

1.2 Functions and Enhancements Included in IBM TCP/IP Version 2 Release 1 for VM

Compared to IBM TCP/IP V1 R2, IBM TCP/IP Version 2 Release 1 for VM offers some new, important functions, especially in network management (SNMP, RIP) and security (RACF*, Kerberos). New APIs (Kerberos, NCS, SNMP agent DPI) are also provided. The socket library has been enhanced and is now based on the Berkeley Software Distribution Socket library, 4.3BSD.

1.2.1 Simple Mail Transfer Protocol (SMTP) Enhancements

Two new features were added to SMTP:

- SMTP, in conjunction with a name server, now supports MX records, in order to send the mail to an alternate host if the primary destination is not available.

  The alternate host will store the mail and will try, periodically, to send it to the final destination.

  Let's consider the following example (part of the "MASTER DATA" file of the NAMESRV virtual machine):

```
fsc5      IN    A      152.9.250.150
          IN    MX 0   fsc5
          IN    MX 2   psfred
          IN    MX 4   fsc6
```

  Any TCP/IP host, that uses the name server, which sends mail to *fsc5* will try, in turn, the following hosts:

  1. *fsc5* itself first (the number after the keyword MX is a kind of priority). If SMTP is not available in this host then the host will attempt to send the mail to *psfred*.
  2. *psfred*. If SMTP is available on *psfred* then the mail will be stored there. *psfred* will then, periodically, try to send the mail back to *fsc5*.
  3. *fsc6*. This host will receive the mail for *fsc5* only when no SMTP is available on *fsc5* or *psfred*.

- The RSCS to SMTP gateway function of SMTP can now be restricted to specific RSCS users with the secure gateway function. Please refer to Section 2.11.6, "Configuring an SMTP Secure Gateway" on page 104 for more details.

1.2.2 Kerberos Services and API

Kerberos is an API that allows the programmer to force additional security checks before the client can use a server. Both server and client applications must have these APIs implemented.

Kerberos has been described in details in the *TCP/IP Tutorial and Technical Overview - GG24-3376-02* manual.

Something about the name Kerberos: In Greek mythology, Cerberus guarded the door to the underworld. Nobody could pass by without his permission. Like this ancient counterpart, modern Kerberos guards access to data on remote computer systems.

While the need for security is easy to explain and understand, most discussions of security are incredibly complex, and often readers fail to gain the knowledge they seek. This brief discussion of the Kerberos system should give you a useful overview.

Why has Kerberos been developed: With the growth of networks and the advent of open network computing environments, there is a growing concern for network security. The lack of physical control over workstations and network connections in a large and open environment requires measures to positively identify users and workstations to the network services (and vice versa). The Kerberos system was developed at MIT to address precisely this issue of mutual authentication of users and services in an environment that may contain pieces that cannot be trusted, for example network connections and workstations.

Design of Kerberos: How is authentication achieved? The user (we′ll call him Bill) identifies himself to the other party (we′ll call her Mary) by having something unique in common, for example, a password. A password authenticates Bill to the login program if Bill supplies the proper password stored in the login database. Of course, Bill has to make sure that he talks to the right login program. TCP/IP offers the secure attention key for this purpose. If used, it guarantees that Bill is not tricked into giving his password to a password-stealing program that masquerades as the login program.

In the open environment we must also deal with intruders trying to imper- sonate other users, with fake services, and with stolen communication. Kerberos employs a trusted third party that helps to establish the identity of users and services. Authentication is based on the ability to encrypt and decrypt data with a unique key that is common to the third party and the object to be identified. To guard against stolen or altered communications, messages between Bill and Mary are also encrypted. The authentication mechanism is used to distribute a session key to Bill and Mary for this purpose. It also ensures that messages come from the original sender. Finally, to save Bill the trouble of having to identify himself each time he wants another service, the third party is divided into two services, the user-authentication service, and the ticket granting service (TGS).

Invoke a Service with Kerberos

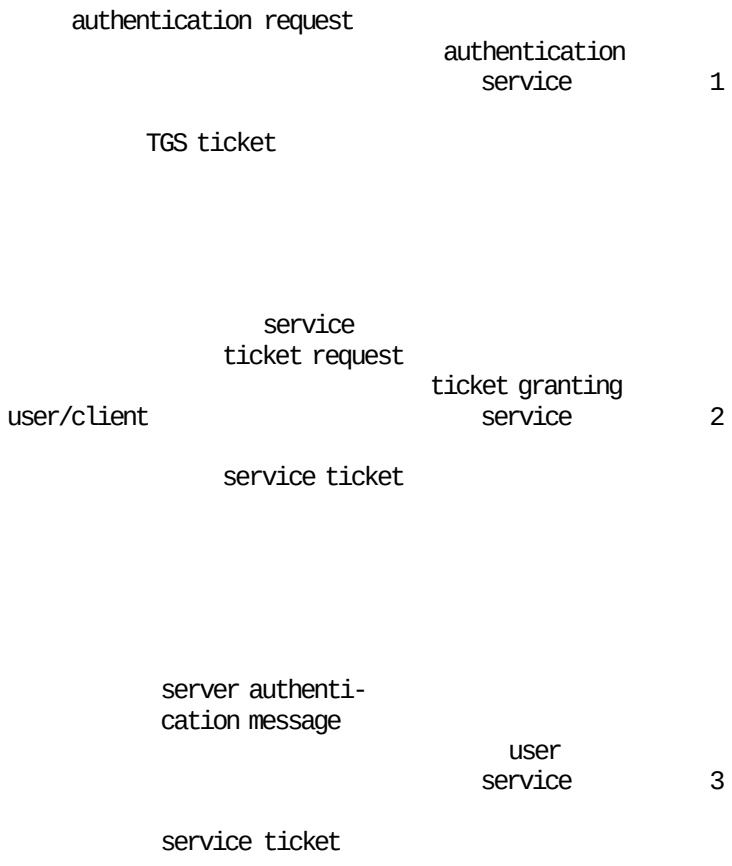


Figure 1. Kerberos System Basic Functions

To invoke the Kerberos services, Bill goes through a three step process:

1 Bill is authenticated by what looks like a normal login. The result of the authentication is a ″ticket″ that allows Bill to access the ticket granting service (TGS). This ticket is good for a limited time to ask the TGS for tickets for other services.

2 Using the TGS ticket Bill gets the ticket for the desired service.

3 The just obtained ticket is sent to the desired service, which then becomes available to Bill. He can, as an option, assure himself that he received the correct service, not a fake one.

How Kerberos works: Kerberos uses a database of identities and encryption keys that uniquely identify clients and services. The database, the authentication service, and the TGS are provided in a trusted environment. To get the initial ticket, Bill sends only his identity to the authentication server. The message that is returned is encrypted with Bill′s password. We can view this message like an envelope that contains the ticket. Bill can only open the envelope if he provides the proper password.

Important

It is important to note, that with this mechanism a password is never sent across the network.

The ticket consists of two parts:

- One that Bill can read
- One that only the designated service can read, because this part is encrypted with the unique key of that service.

Both parts contain the encryption key that is going to be used for the communication between user and service. The identity of Bill, lifetime of the ticket, workstation address and time stamp are only contained in the service readable part.

Ticket Request: When requesting a ticket for the desired service from the TGS, Bill sends the service readable part of the TGS ticket together with the name of the requested service and his own envelope containing an ″authenticator″ to the TGS. The authenticator consists of Bill′s identity, a time stamp and the station address. The TGS decrypts the ticket. As it contains the session key, it can now open the envelope and check that the user (Bill) sending the envelope is the same one that is requesting the service, and that the ticket is still valid. The TGS creates a ticket for accessing the desired service. It has the same structure as the TGS ticket, except that the service readable part is encrypted with the encryption key of the desired service.

Service Request: To access the service, Bill sends the service readable part of the service ticket together with his own envelope containing an authenticator to the service. The service decrypts the service ticket, and opens the envelope. To authenticate itself to Bill, it uses the time stamp that it received in the envelope, adds ″1″ to it and sends the result in an envelope to Bill. This assures Bill that the correct service was invoked; otherwise the service could not have opened the envelope, modified the time stamp and returned it to Bill.

Additional Features: Kerberos provides additional facilities for administering the database of identities and encryption keys, as well as a mechanism to distribute this information to additional Kerberos servers for higher availability in a large network.

Important

It is important to note that, today, all applications shipped do not include Kerberos calls. To implement a kerberized environment you need to manually modify the client and server code to include calls to Kerberos.

1.2.3 Network Computing System (NCS) API

The APOLLO** NCS** is an API that assists the programmer in writing applications that require distributed data as well as distributed computing.

1.2.4 Simple Network Management Protocol (SNMP)

SNMP allows you to manage your TCP/IP network. IBM TCP/IP V2 for VM supports both the SNMP monitor and agent function.

Note: For the SNMP monitor function, NetView VM is a requirement.

1.2.4.1 Overview

SNMP is an Internet standard that defines a set of functions that may be used to monitor and control network elements in a TCP/IP based network.[1] A network element is a piece of equipment that is connected to a network. It includes intelligent devices like computers and dumb devices like repeaters.

An SNMP agent is software in a network element that is to be monitored or managed and which will respond to SNMP queries or sets. The Network Management Station (NMS) is the computer with software that can generate requests to an SNMP agent and accept the responses which are produced. The SNMP agent also sends messages (TRAPs) to a SNMP NMS to indicate a significant change in status.

Objectives: The goals of SNMP are relatively simple. SNMP provides functions for network monitoring and management. As the name implies, it is a simple protocol minimizing the number and complexity of network management functions. Currently it is mostly a read-only system. The architecture is independent of specific hosts, routers or gateways.

Components: Basically there are two components for SNMP, a server and a client. They are called an ***Agent*** and a ***Network Management Station*** (NMS), sometimes referred to as a ***Monitor***.

- ***SNMP Client - NMS or Monitor***

  The monitor executes management applications that monitor and control network elements (Agents). It issues SNMP commands like GET, GETN and SET to the agent and it processes messages (GET responses, traps) from the agent. The commands and messages in SNMP are called Protocol Data Units (PDUs).

- ***SNMP Server - Agent***

  The agent responds to SNMP commands via GET responses. It allows access to the Management Information Base (MIB) where the current status of the TCP/IP system is reflected. A message (trap) is sent by a SNMP agent to a SNMP NMS to indicate a significant change in status. For example, when a link goes up or down or when a authentication failure occurs. These traps are then forwarded to defined monitors.

[1] See RFC 1156 and 1157 for a detailed definition of SNMP.

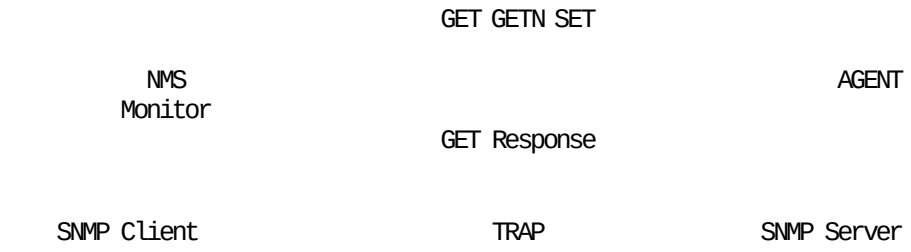


Figure 2. SNMP Components

SNMP Service Primitives

- ***GET Request***

  Allows a network management station to query a network element.

- ***GET Next Request***

  Same as get request but used when information is in a table --> get next entry. For example to retrieve all the entries in the routing or interface table this command must be used.

- ***GET Response***

  Allows a network element to reply to a management query.

- ***SET Request***

  Allows a network management station to set the value of a managed object.

- ***TRAP***

  Allows a network element to report certain events concerning managed objects. For example when a link goes down or an authentication failure occurs a TRAP is sent to the monitor.

  The following traps are defined in RFC 1157:

  – Cold_Start: The agent is restarting. Informations may have been lost.

  – Warm_Start: The agent is restarting. No informations lost.

  – Link_Up: A link is up again.

  – Link_Down: A link is down.

  – Authentication_Failure: Invalid community name provided.

  – EGP Neighbor_Loss: Can't communicate anymore with an EGP neighbor.

  – Enterprise_Specific: An enterprise-specific event has occurred.

Management Information Base (MIB): The items within the network elements which are manageable are called ***Managed Objects*** (packet counts, routing tables, etc.). The universe of network manageable objects is called the MIB. The objects themselves are arranged in groups:

Group	Objects for	£
system	basic system configuration	3
interfaces	network attachment	22
at	address translation	3
ip	internet protocol	33
icmp	internal control message protocol statistics	26
tcp	transmission control protocol	17
udp	user datgram protocol	4
egp	exterior gateway protocol	6

£: Number of objects in that group

Figure 3. MIB I

Group	Objects for	£
system	basic system configuration	7
interfaces	network attachment	23
at	address translation	3
ip	internet protocol	38
icmp	internal control message protocol statistics	26
tcp	transmission control protocol	19
udp	user datgram protocol	7
egp	exterior gateway protocol	18
transmiss.	transmission. Media-specific	0
snmp	snmp applications entities	30

£: Number of objects in that group

Figure 4. MIB II

MIB I and MIB II are different versions of the Management Information Base. All IBM hosts implement MIB II.

1.2.4.2 SNMP Implementation in IBM TCP/IP V2 for VM

The SNMP implementation in IBM TCP/IP V2 for VM contains both components, an agent and a monitor. Both components can be configured and started with or without the other. If you have many VM systems with TCP/IP installed, there is no need to have the monitor component on each system. See also Figure 19 on page 57 for the implementation of SNMP in IBM TCP/IP V2 for VM.

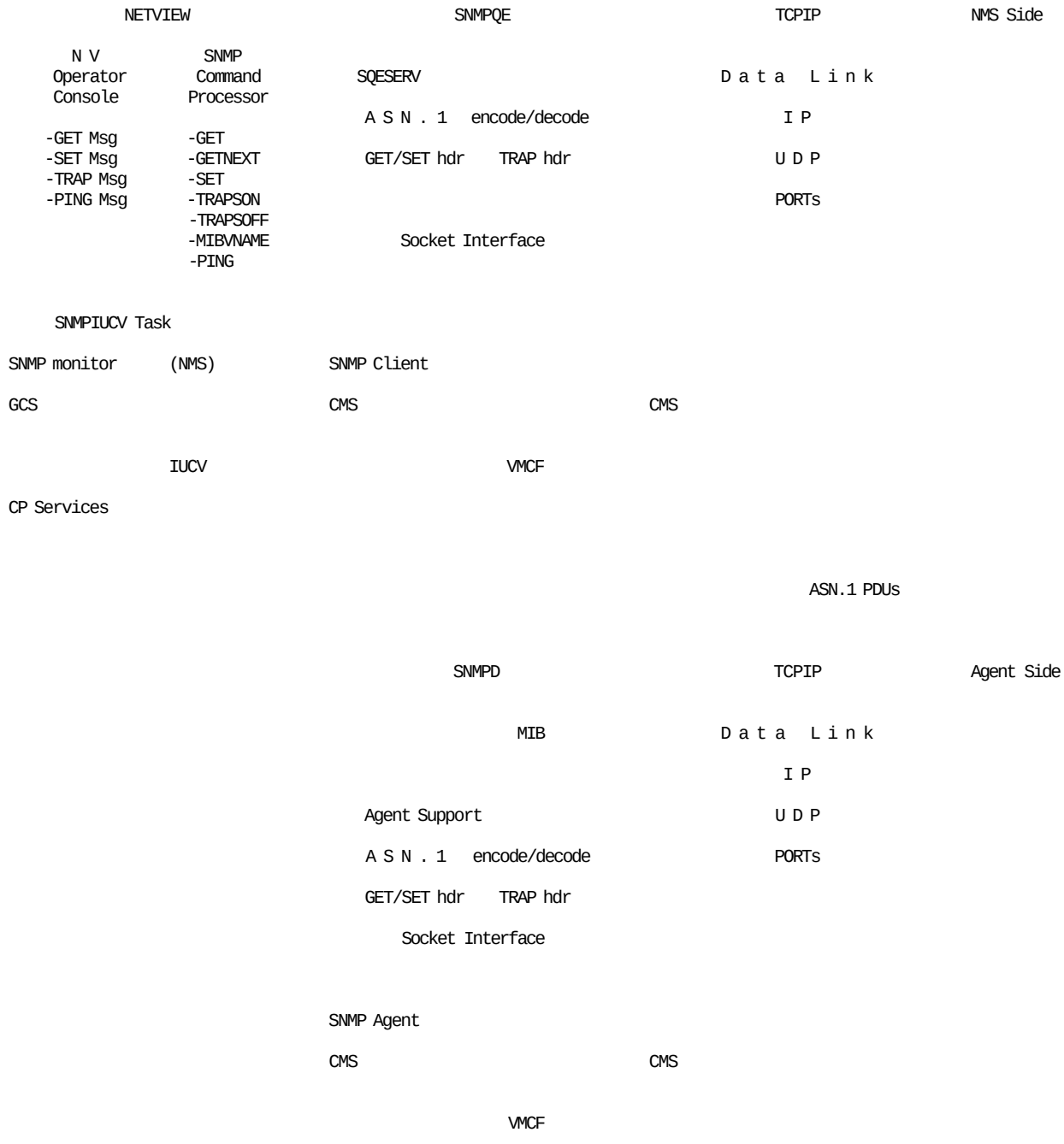


Figure 5. SNMP Implementation in TCP/IP V2 for VM

1.2.5 SNMP Command Processing

The following steps describe how the SNMP command is processed (refer to Figure 5 on page 10):

1. The NetView operator or a NetView CLIST issues an SNMP command.
2. The SNMP command is validated by the SNMP command processor.
3. The command processor passes the request to the SNMPIUCV task.
4. The SNMPIUCV task passes the request to the SNMP query engine.
5. The SNMP query engine validates the request, encodes it using the ASN.1 notation, creates an SNMP request PDU, and sends it to the SNMP agent.
6. The SNMP query engine receives a response from the SNMP agent.
7. The SNMP query engine decodes the response and sends it to the NetView SNMPIUCV task.
8. The SNMPIUCV task sends the response as a multi-line message to the requesting operator or authorized receiver.

A full SNMP implementation in IBM TCP/IP V2 for VM requires four virtual machines:

- ***TCPIP virtual machine***, sends the PDUs between the TCP/IP nodes.
- ***NETVIEW virtual machine***, implements the user command interface and the monitor function.
- ***SNMPQE virtual machine***, processes the ASN.1 PDUs and communicates with TCPIP and NETVIEW virtual machines.
- ***SNMPD virtual machine***, implements the SNMP agent.

Warning

The VM SNMP agent does not support the SET command; that is, a VM agent can′t modify or set a MIB variable.

1.2.5.1 SNMP Agent Distributed Program Interface (DPI)

As said previously SNMP traps are defined in RFC 1157, and the specifications of the SNMP protocol do not define how an ″end user″ can have his SNMP agent monitor specific processes. To achieve this you have to either modify the code of your SNMP agent or use what is called a subagent. A subagent will use an API to communicate with the agent. This API is called the Distributed Programming Interface (DPI). The routines located in the ″DPILIB TXTLIB″ will allow the subagent to:

- Establish a connection with the agent: *query_DPI_port()*

  This returns the port of the agent. The subagent will then use the Socket API to connect to that port.

- Register its variables to the agent: *mkDPIregister()*

  The subagent is ready to receive queries from the agent.

- Get variables from its MIB: *pDPIpacket()*
- Set a MIB variable: *mkDPIset()*
- Answer a query from the agent: *mkDPIresponse()*
- Send a trap to the agent: *mkDPItrap()*

The agent will forward these traps to the monitor(s).

When a monitor queries a MIB variable, the requests are always sent to the agent. If the variable is not owned by the agent but has been registered by a subagent, then the query is forwarded to the subagent. The answer is sent back from the subagent to the agent, which in turn sends it back to the monitor.

Using such APIs (on UNIX machines the DPI is called SMUX), one can add MIB variables without having to modify the code of the agent.

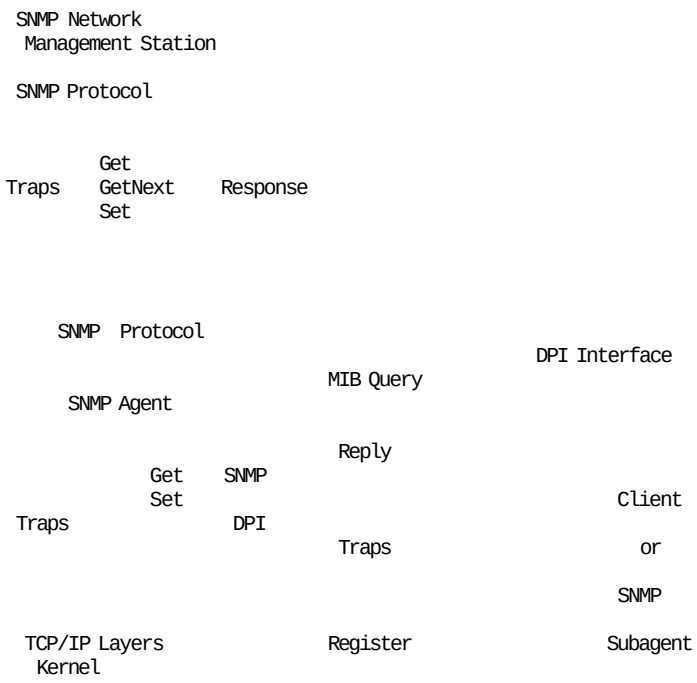


Figure 6. SNMP SubAgent Overview

Section 1.3.7, “SNMP / 3172 Network Management” on page 27 gives an example of the use of the SNMP DPI to implement a subagent within the TCPIP engine, which allows the query of the 3172 enterprise-specific variables.

1.2.6 Routing Information Protocol (RIP)

RIP[2] is a protocol that assists in finding a path in the network through which to route IP datagrams. The gateways using RIP exchange their routing information in order to allow the neighbors to learn of topology changes. The RIP server updates the local routing tables dynamically, resulting in current and accurate routing tables.

[2] See RFC 1058 for a complete definition of RIP.

1.2.6.1 Routing Protocol Overview

In a nationwide network, such as the current Internet, it is very unlikely that a single routing protocol will be used for the whole network. Rather, the network will be organized as a collection of ***autonomous systems***. An autonomous system will, in general, be administered by a single entity, or at least will have some reasonable degree of technical and administrative control. The following definitions are commonly used:

- ***Autonomous System:*** Set of computers which exchange routing information using the same Interior Domain protocol.
- ***Interior Domain (or Gateway) Protocol:*** Routing update protocols used within an autonomous system. Some examples are: Routing Information Protocol (RIP), Open Shortest Path First (OSPF), Hello.
- ***Exterior Domain (or Gateway) Protocol:*** Routing update protocols used to exchange routing information between autonomous systems. Some examples are: Exterior Gateway Protocol (EGP), Border Gateway Protocol (BGP). Such protocols are now usually referred to as ***Inter-AS Routing Protocols***.
- ***Algorithm:*** Mathematical formula used to create the routing table based on the routing updates received from other gateways (or routers). Some examples are: Vector distance, Shortest Path First.

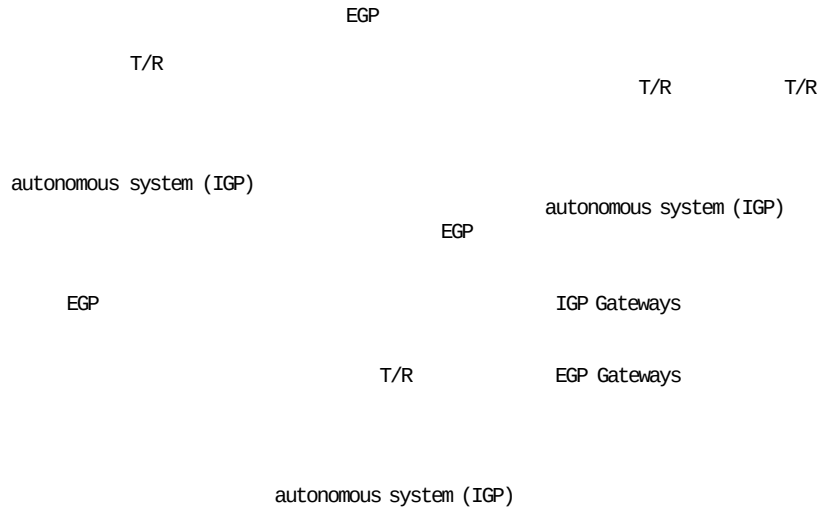


Figure 7. Routing Protocol Positioning

Note: IBM TCP/IP V2 for VM supports only RIP (TCP/IP V2R1 and TCP/IP V2R2).

1.2.6.2 RIP Overview

RIP is a protocol designed to handle routing within small to moderate-sized networks, where line speeds do not differ radically. Therefore this protocol is most useful as an Interior Gateway Protocol.

Objective of a routing protocol

The goal of RIP is to supply the information that is needed to do routing. Routing is the task of finding a path from a sender to a desired destination.

RIP is one of a class of algorithms known as ***vector distance algorithms***. Vector distance algorithms are based on the exchange of only a small amount of information. Each entity (gateway or host) that participates in the routing protocol is assumed to keep information about all of the destinations within the system. Each entry in this routing database includes the next gateway to which datagrams destined for the entity should be sent. In addition, it includes a ***metric*** measuring of the total distance to the entity. Vector distance algorithms get their name from the fact that it is possible to compute optimal routes when the only information exchanged is the list of these distances. Furthermore, information is only exchanged among entities that are adjacent.

RIP Limitations: Since RIP is designed for a specific network environment it has some limitations. These should be considered before implementing RIP in your network.

- It cannot be used with networks where any path goes through more than 15 gateways.
- It cannot share traffic between paths.
- It cannot adapt to changes in network load.
- It cannot be used in a network with variable length subnet masks.
- It cannot discriminate between network speeds. For example, if a route R1 to network N goes through a token-ring LAN but shows a distance D1 greater than the distance D2 associated with the route R2 to network N which goes through a 2400 bps serial line, R2 will be chosen, even if much slower.

RIP Database: We said above that each entity keeps a routing database with one entry for every possible destination in the system. An actual implementation is likely to need to keep the following information about each destination:

Address The IP address of the host or network.

Gateway The first gateway along the route to the destination.

Interface The physical network which must be used to reach the first gateway.

Metric A number indicating the distance to the destination.

Timer The amount of time since the entry was last updated.

Basic Vector Distance Algorithm: The following procedure is carried out by every entity that participates in the routing protocol. This must include all of the gateways in the system. Hosts that are not gateways may participate as well.

Basic Distance Vector Algorithm

- Keep a table with an entry for every possible destination in the system. The entry contains the distance D to the destination, and the first gateway G on the route to the network.
- Periodically, send a routing update to every neighbor. The update is a set of messages that contain all of the information from the routing table. It contains an entry for each destination, with the distance shown to that destination.
- When a routing update arrives from the neighbor G′, add 1 to the distance advertised by G′. Call the resulting distance D′. Compare the resulting distance with the current routing table entries. If the new distance D′ for N is smaller than the existing value D, adopt the new route. That is, change the table entry for N to have metric D′ and gateway G′. If G′ is the gateway from which the existing route came, G′ = G, then use the new metric even if it is larger than the old one.

Note: This is not a statement of the RIP protocol. There are several refinements still to be added.

RIP Usage

- This works well in a token-ring or Ethernet LAN environment, because RouteD will receive its own broadcast packets.
- Other networks, such as point-to-point links, cannot be managed by RouteD unless a RouteD server is running on the host on the other end of the link. If the other host is not running RouteD, the RouteD server will not receive updates over the link, and will assume the link is down.
- RouteD cannot be used to manage HYPERchannel** networks, because HYPERchannel does not support the link-level broadcasting of packets.

Warning

If different subnet masks are used throughout the network, you should not use RIP, because the subnet mask is not passed in the routing information. It is assumed that the router receiving the information is using the same subnet mask as the router sending the information.

1.2.7 Remote Printing (LPD and LPR)

The line printing component includes both client and server programs. The clients (LPR, LPQ, and LPRM) provide functions equivalent to those in the VM/CMS PRINT command. The server provides support for both direct printing and RSCS connection, as well as remote printers on systems which use the LPR/LPD protocol.

From the viewpoint of system VM15 (Figure 8) there are three printers and one punch defined. The printer ***LOCPR*** is directly attached to VM15 and therefore defined as a local printer in the LPD printer configuration file ″LPD CONFIG″. The printer ***LPT1*** is a PC printer attached to an OS/2* running TCP/IP V1.2.1 for OS/2 and also the LPD server. This printer is a remote printer to system VM15 linked via a TCP connection. The printer ***ITSC*** is also defined as a remote printer; it is available through the RSCS network and it is also available for any TCP/IP user.

Note that the system RALYDPD4 has no TCP/IP installed; therefore, there is no TCP connection to this printer. The punch device ***JCLMVS20*** is a virtual device only. JCL can be sent to this device to execute a job on the MVS system.

The data stream is sent via RSCS and NJE to the MVS system and therefore TCP/IP on the MVS system is not required. To summarize the picture, all TCP/IP network users can send their print output to one of the three printers, or they can even send JCL to a remote MVS system.

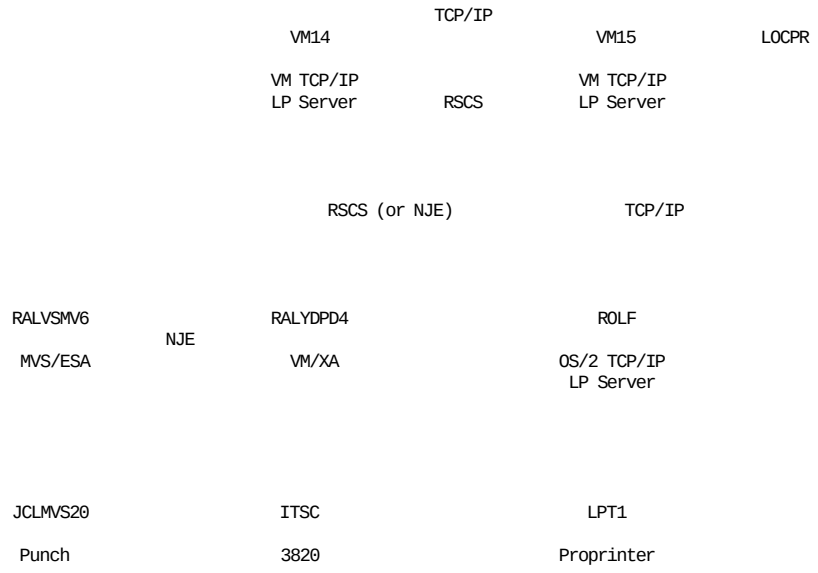


Figure 8. VM LPD Server Configurations

1.3 New Functions and Enhancements in IBM TCP/IP Version 2 Release 2 for VM

IBM TCP/IP Version 2 Release 2 for VM offers some new, important functions, especially in the areas of network management (3172), connectivity (FDDI, 3745-Ethernet) and relational database access (NDB). Improvements for NFS, SMTP and name server have also been added.

1.3.1 Supported Systems

VM/XA*: VM/XA is supported in *exploitation* mode. The TCPIP virtual machine can now run in XA-mode. The virtual machine memory size is tested during startup, and the 31-bit addressing support module (TCPIPXA) is automatically invoked if this size has been set to a value greater than 16MB. This module allows buffers and control blocks to be loaded *above the 16MB line* and **it requires access to the Pascal runtime library**.

VM/ESA*: VM/ESA is supported in *toleration* mode. The implementation of V2R2 does not use any capability specific to VM/ESA (that is, for example, the VM Data Spaces), but it is the first release which has been tested and which is therefore officially supported on the VM/ESA platform.

1.3.2 Fiber Distributed Data Interface (FDDI) LAN Attachment

The *Fiber Distributed Data Interface* (FDDI) provides a high bandwidth (100 megabits per second peak data rate) general interconnection among computers and peripheral equipment. It establishes the connection among many stations (nodes) distributed over distances of several kilometers.

IBM TCP/IP Version 2 Release 2 for VM supports the attachment to an FDDI LAN through the IBM 3172 Model 2 Interconnect Controller running Version 2 Release 1 of the 3172 Interconnect Controller Program. This provides an FDDI gateway function for high-speed access to TCP/IP applications on IBM hosts via parallel channel.

Basic characteristics of the FDDI specification include:

- A fiber optic transmission media
- A ring topology
- A token access protocol
- An architecture with a distributed management approach.

1.3.2.1 FDDI Topology

An FDDI ring consists of two separate counter-rotating rings forming a primary ring and a secondary ring. The secondary ring may be used for concurrent transmission or standby.

Note: The IBM 3172 FDDI Adapter card supports one single MAC entity, and therefore can use the secondary ring for standby only.

The *dual counter-rotating ring* topology objective is to provide greater reliability and fault tolerance of a damaged network. It allows the network to be reconfigured around hardware failures that occur. Any single failure that occurs within a dual ring connection can be isolated from the ring and the integrity of the rest of the ring is still maintained. Multiple hardware failures on the dual ring

will result in dividing the ring into separate rings that still function but are isolated from each other.

FDDI also provides facilities which allow an FDDI network to be configured into a star topology.

Two classes of connections are defined:

- *Dual Attach Stations* (class A):

  Which are connected to both the primary and secondary rings and are capable of reconfiguring the ring to bypass an inoperative segment by redirecting traffic from the ring of one rotation to the other.

- *Single Attach Stations* (class B):

  Which are connected to a single ring and are attached to a *Dual Attach Concentrator*, itself dual-attached to the main rings to maintain the reliablility of the dual ring.

Single Attach Concentrators are used to form a tree-structured hierarchical topology, connected to the trunk dual ring, as shown in Figure 9.

Figure 9. FDDI Topology

A maximum of 1000 physical connections may be present on the FDDI ring, with a maximum distance of 2 KM between adjacent nodes and a total fiber path length not longer than 200 KM.

This technology is intended:

- To connect LAN-attached workstations whose application data flow requires high bandwidth (advanced CAD/CAM, high resolution imaging, system simulation and modeling)
- To consolidate multiple LANs in a campus environment onto a single high-speed backbone
- To implememt LANs that spread over greater distances and allow the attachment of more nodes than the other existing technologies.

1.3.2.2 Protocol Concepts

FDDI borrows very extensively from the techniques used in the IEEE 802.5 token-ring specification.

The two basic entities transmitted in an FDDI ring are called *frames* and *tokens*. Access to begin transmission on the ring is gained by capturing a token. There is only *one* token allowed on a ring, and therefore only one station may transmit new data onto the ring at a time. A station transmits data by receiving a token, transmitting multiple frames until a timer expires, and re-transmitting the token onto the ring.

A station is said to be holding the token when the token is not present on the ring, and that station is transmitting. Since the token immediately follows the last transmitted frame, a downstream station may repeat this process resulting in the FDDI ring simultaneously carrying multiple frames from multiple stations.

An FDDI station transmits information to its downstream neighbor and receives information from its upstream neighbor. A frame is repeated from each station to its downstream neighbor until it returns to the sender, which ″removes″ it from the ring (by not re-transmitting it). When a station receives a frame that is addressed to it or to a group of which it is a member, the station:

1. Copies the frame if buffering allows it
2. Sets the ″addressed recognized″ symbol in the frame
3. Sets the ″frame copied″ symbol in the frame
4. Transmits the frame to its downstream neighbor.

Otherwise, the frame is repeated to its downstream neighbor and it is not copied.

FDDI bandwidth is partitioned into three classes of service:

- Synchronous Transmission
- Asynchronous Transmission
- Immediate Service

Synchronous transmission is intended for applications whose bandwidth and response time limits are predictable, permitting them to be pre-allocated. This is not supported by the IBM 3172 implementation.

Asynchronous transmission is used by applications whose bandwidth requirements are less predictable (for instance bursty traffic). The bandwidth for transmitting asynchronous frames is dynamically allocated from unused and/or unallocated FDDI bandwidth.

Immediate service is used for extraordinary applications such as ring recovery.

1.3.2.3 OSI Compliance

The Fiber Distributed Data Interface (FDDI) is a standard developed by the technical committee X3T9.5 of the American National Standards Institute (ANSI). It is compliant with the lower layers of the OSI protocol stack and provides services specified by the Data Link and Physical layers of the OSI model.

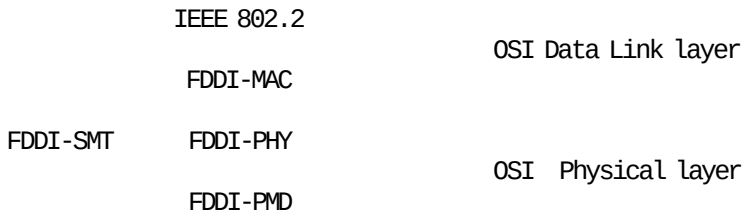


Figure 10. FDDI Standard Protocol Layering and Layer Interactions

It consists of the following sublayers, as illustrated in Figure 10:

- FDDI-PMD (Physical Media Dependent - ANSI 1988 - ISO 9314/3) describes the physical connection to the medium and includes the specification for all of the transmission media hardware (fibers, connectors, optical pulses characteristics).
- FDDI-PHY (Physical Control - ANSI 1986 - ISO 9314/1) handles all of the symbol-based functions. A *symbol* is the basic sequence of bits which represents data and control information. Thus this sublayer describes the encode/decode of the data and control symbols, the serial-to-parallel and parallel-to-serial conversions and the clocking requirements.
- FDDI-MAC (Medium Access Control - ANSI 1986 - DIS 9314/2) handles all of the frame-based functions and controls access to the transmission media. It handles the data framing (including the starting delimiter, frame class and end delimiter), the addressing (generation/recognition of source and destination addresses), the data checking (generation/verification of the CRC frame check sequence), the bandwidth allocation and the negotiation of the timed-token protocol.
- FDDI-SMT (Station Management) describes the local portion of the system management including the control required for proper operation of an FDDI station (initialization, configuration, performance monitoring, fault detection and error recovery). SMT is not yet an approved ANSI standard.

The Data Link layer is subdivided into the LLC and MAC sublayers. FDDI only specifies the MAC sublayer, while the LLC sublayer is the IEEE 802.2, which interfaces directly to the FDDI-MAC sublayer as well as the other IEEE 802.x MAC sublayers of the other network specifications.

1.3.2.4 FDDI and TCP/IP

The encapsulating of IP datagrams and ARP requests and replies in FDDI frames is defined in *RFC 1188 - A Proposed Standard for the Transmission of IP Datagrams over FDDI Networks*, for single MAC stations (which the IBM 3172 Interconnect Controller is). Operation on dual MAC stations will be described in a forthcoming document.

RFC 1188 states that all frames are transmitted in standard IEEE 802.2 LLC Type 1 Unnumbered Information format (Type 1 refers to the *connectionless* or *user datagram* mode of operation which implies no guaranteed delivery of the frames to the destination station), with the DSAP and SSAP fields of the 802.2 header set to the assigned global SAP value for SNAP (decimal 170). The 24-bit Organization Code in the SNAP header is set to zero, and the remaining 16 bits are the EtherType as defined by *RFC 1060 - Assigned Numbers*, that is:

- 2048 for IP
- 2054 for ARP.

Please refer to the *TCP/IP Tutorial and Technical Overview*, GG24-3376 for more details.

The mapping of 32-bit Internet addresses to 48-bit FDDI addresses is done via the ARP dynamic discovery procedure. The broadcast Internet addresses (whose <host address> is set to all ones) are mapped to the broadcast FDDI address (all ones).

IP datagrams are transmitted as a series of 8-bit bytes using the usual TCP/IP transmission order called ″big-endian″ or ″network byte order″.

The FDDI-MAC specification (*ISO 9314/2 - ISO, Fiber Distributed Data Interface - Media Access Control*) defines a maximum frame size of 4500 bytes for all frame fields. After taking the LLC/SNAP header into account, and to allow future extensions to the MAC header and frame status fields, the MTU of FDDI networks is set to 4352 bytes.

1.3.2.5 FDDI Support by the IBM 3172 Model 2 Interconnect Controller

Support of the FDDI LAN attachment by the 3172 requires:

- IBM 3172 FDDI LAN adapter (feature #2250)
- IBM 3172 Interconnect Control Program (ICP) V2R1

The IBM 3172 configured as an FDDI LAN-Host gateway may act as a *Dual Attach Station*. It supports up to one FDDI adapter card and two parallel channel adapters. A token-ring adapter may be installed simultaneously with the FDDI adapter, but can be used only to connect the 3172 ICP Operator Facility station. The use of other LAN attachments in a gateway configuration together with the FDDI LAN adapter is, at the time of publication of this document, a Statement of Direction. Figure 11 on page 22 shows an IBM 3172 used as an FDDI LAN-Host gateway to allow high-speed access to a TCP/IP host running IBM TCP/IP Version 2 Release 2 for VM (Section 2.8, “FDDI LAN Attachment Support” on page 91 describes the new *FDDI LINK* statement to be included in the ″PROFILE TCPIP″ configuration file).

Although the 3172 FDDI LAN-Host gateway supports both VTAM*/SNA and TCP/IP data flows, only one can be selected at a time; that is, both flows are not supported simultaneously:

- When VTAM/SNA is selected, up to two channel adapters are supported.
- When TCP/IP is selected, only one channel adapter is supported.

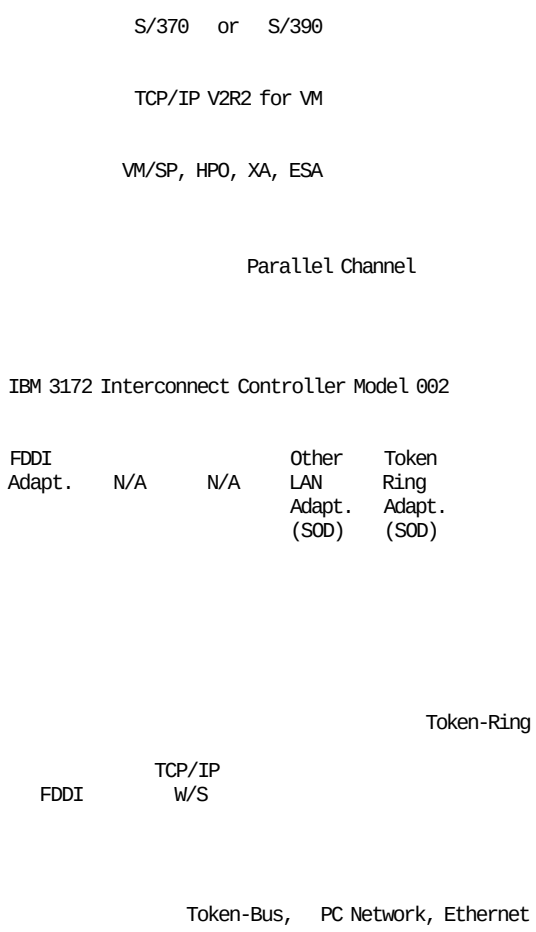


Figure 11. IBM 3172-002 Interconnect Controller FDDI LAN-Host Gateway Configuration

1.3.2.6 References

Suggested readings:

- *TCP/IP Tutorial and Technical Overview - GG24-3376-02*
- *Local Area Network Concepts and Products - GG24-3178-02*
- *FDDI - A tutorial*, published by Connexions, Vol.4, No.10
- *RFC 1188 - A Proposed Standard for the Transmission of IP Datagrams over FDDI Networks*.

1.3.3 3745 Ethernet LAN Adapter and ACF/NCP V6 IP Router Support

1.3.3.1 Overview

ACF/NCP V6 and the 3745 Ethernet LAN Adapter (ELA) have been announced in September 1991 and will be available in September 1992. The 3745 ELA will provide attachment to the 3745 Communication Controller of LANs that conform to the Ethernet V2 or IEEE 802.3 standard. With ACF/NCP V6, IP datagrams sent by a TCP/IP workstation attached to a 3745 ELA, which must go through the 3745 to reach their destination, are enveloped in SNA RUs by ACF/NCP and can be routed:

- Through the SNALINK virtual machine, to any TCP/IP application residing on a host which is channel-attached to the 3745
- Through the SNA backbone, to any other *reachable* TCP/IP host/workstation.

IBM TCP/IP Version 2 Release 2 for VM is the first level of TCP/IP for VM to support the IP routing function included in ACF/NCP V6.

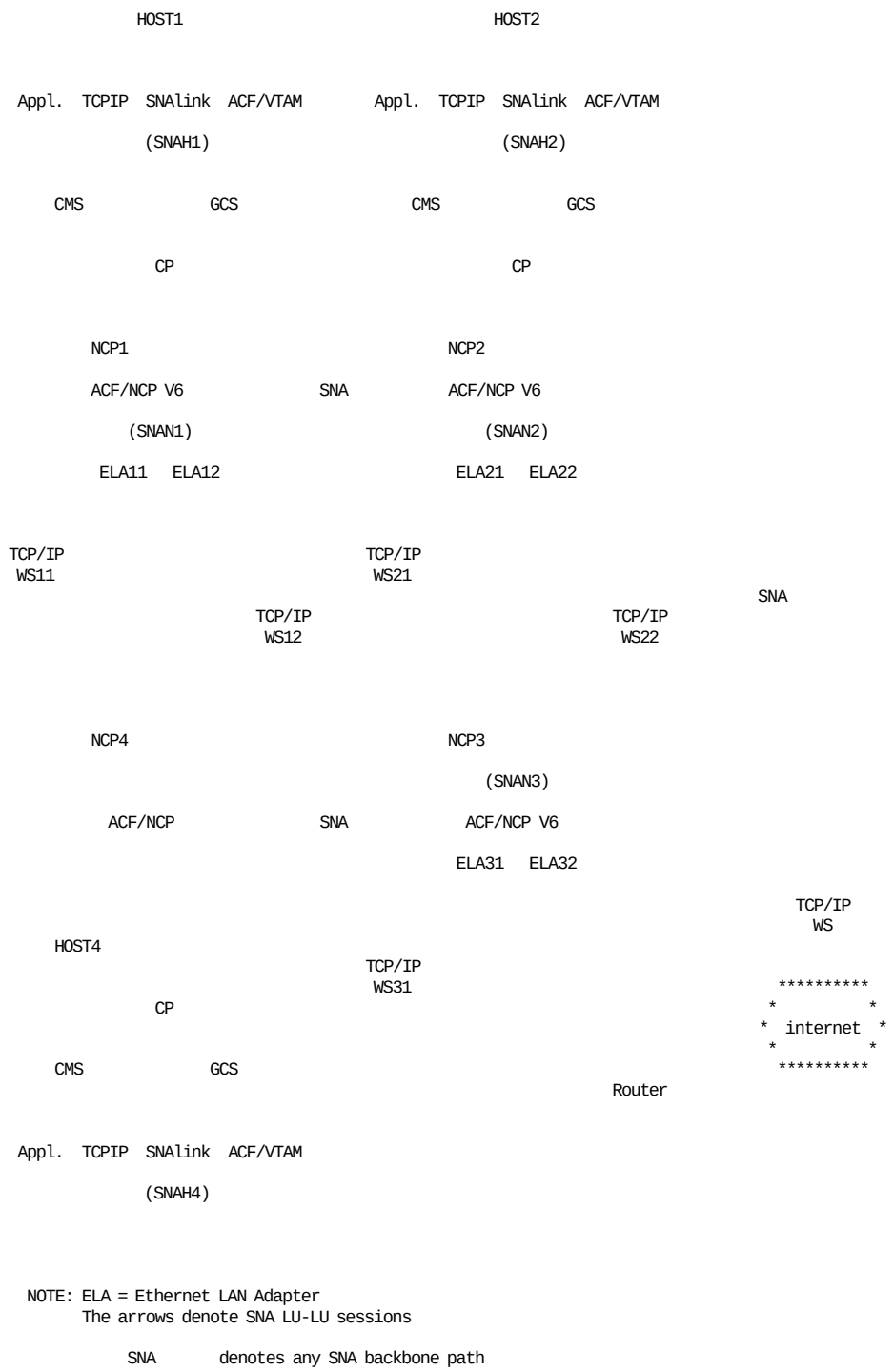


Figure 12. 3745 IP Routing Function Configurations

In the configuration shown in Figure 12 on page 24, any TCP/IP workstation WSnn (WS11, WS12, WS21, etc.) connected to a 3745 ELA, can access:

- Any TCP/IP application running on HOST1, HOST2 or HOST4
- Any other TCP/IP workstation WSpp connected to a 3745 ELA installed on the same or on another 3745
- Any TCP/IP host/workstation that one of the HOSTx or WSpp hosts/workstations, acting as a router, can route IP traffic to.

1.3.3.2 ACF/NCP V6 IP Router

The design of an IP router in ACF/NCP V6 has been especially based on *RFC 1009 - Requirements for Internet Gateways*. In addition to IP and ICMP, it implements the Address Resolution Protocol (ARP) to perform the *direct routing* of an IP datagram destined to a TCP/IP workstation connected to a directly attached Ethernet LAN. The implementation supports routing of both Ethernet V2 and IEEE 802.3 over the same interface.

The ACF/NCP V6 IP router supports the fragmentation process, and an MTU can be defined for each interface.

The interface between SNA and IP in ACF/NCP V6 is referred to as NCP Connectionless SNA Transport (NCST). The basic connection provided by NCST is a single full-duplex type 0 LU session which can link:

- Two NCST endpoints in two NCPs. For example, Figure 12 on page 24 shows the following LU-LU sessions of this type:

  (SNAN1) - (SNAN2)

  (SNAN2) - (SNAN3)

- One NCST endpoint in an NCP to any SNA session partner that provides equivalent services, such as the SNALINK virtual machine implemented in IBM TCP/IP Version 2 Release 2 for VM. For example, Figure 12 on page 24 shows the following LU-LU sessions of this type:

  (SNAN1) - (SNAH1)

  (SNAN2) - (SNAH2)

  (SNAN3) - (SNAH4)

Section 2.9, “3745 ELA and ACF/NCP V6 IP Router Support” on page 92 describes the various definitions which are needed in ACF/VTAM*, TCP/IP for VM and ACF/NCP when configuring the 3745 ELA and NCP IP router support in conjunction with IBM TCP/IP Version 2 Release 2 for VM.

1.3.4 Network File System (NFS)

The following enhancements have been implemented in the NFS** server included in IBM TCP/IP Version 2 Release 2 for VM:

- The VMNFS module uses multiple-block **BLOCKIO* requests (a fast VM control program system service) to write to CMS disks. It detects when CP does not support multiple-block requests, and in that case reverts to single-block requests.
- The write-block logic of the VMNFS module has been improved to eliminate duplicate updates of a block during the processing of a single client request. This has been done to improve the poorer performance of *record=nl* write compared to binary write operations.
- The NFS server now automatically detaches a CMS minidisk when all client mounts of that minidisk have been released.
- VMNFS now supports the CMS special message interface (SMSG) which allows CMS users to query the NFS server activity and to perform CMS minidisk-related operations (see Section 2.13.1, “SMSG Interface to NFS” on page 113 for more details).

1.3.5 Domain Name Server (DNS)

IBM TCP/IP Version 2 Release 2 for VM includes some enhancements in its name server implementation:

- Two new commands for querying name servers from a CMS user ID: **DIG** and **NSLOOKUP** (see Section 2.18.3.3, “The DIG Command” on page 164 and Section 2.18.3.4, “The NSLOOKUP Command” on page 166 for more details).
- An option to remove the queue limit from all open UDP ports (see Section 2.18, “Domain Name Server” on page 142 for more details).

1.3.6 Simple Mail Transfer Protocol (SMTP)

IBM TCP/IP Version 2 Release 2 for VM enhances the performance and reliability of the VM SMTP server with the implementation of the special message interface (SMSG), and provides additional flexibility with mail header customization and some new configuration statements:

- The SMSG interface allows the query of SMTP mail delivery queues and statistics, and provides a set of privileged commands for system administration tasks (see Section 2.11.9, “SMSG Interface to SMTP” on page 106 and Section 3.4.5, “SMSG Interface to SMTP” on page 184 for more details).
- With mail header customization, users can change the rules used to re-write header address transformations (see Section 2.11.10, “SMTP Mail Headers” on page 107 for more details).
- The *ONDISKFULL* statement provides the ability to have the TCPIP owner’s user ID notified when the SMTP A-disk is filled up beyond a given threshold.
- *OUTBOUNDOPENLIMIT* provides the ability to limit the number of simultaneously-delivered pieces of mail in order to save system resources.

See Section 2.11.1, “Configuring SMTP” on page 97 for the new/changed SMTP configuration statements.

1.3.7 SNMP / 3172 Network Management

1.3.7.1 Overview

When participating in a TCP/IP network, the IBM 3172 Interconnect Controller is an extension of the host node, providing the host with LAN connections to the TCP/IP network. The 3172 is not, by itself, addressable as a TCP/IP node, and has no real TCP/IP or SNMP protocol support. In particular, it has no support for the ASN.1 encoding used in the exchange of SNMP requests/responses.

Network Management support of the IBM 3172 Interconnect Controller in a TCP/IP environment is based on the ability to retrieve 3172 network management data (called ***attributes***) using an SNMP *subagent* (also called *proxy-agent*) running on the channel-attached TCP/IP host. This SNMP subagent will handle the mapping between the requested/retrieved 3172 attributes and the ASN.1 notation of the corresponding MIB variables, and will send them to the local SNMP agent for inclusion in this host′s TCP/IP Management Information Base (MIB) and transmission via Simple Network Management Protocol (SNMP) to an SNMP monitor in the TCP/IP network. This function will allow the network operator to access SNMP information about the 3172 LAN and channel interfaces, to correctly represent the host node in the TCP/IP network.

1.3.7.2 3172 Network Management Implementation

TCP/IP network management for the IBM 3172 is provided via an extension to the ″8232 command set″ supported by the 3172. This new command called Network Management (*NETMAN*), is sent by the host over any one of the sub-channels connecting the 3172 to the TCP/IP host, to retrieve 3172 network management data (*attributes*) from the 3172.

The term *attribute* refers to a piece of information about the 3172, which may be descriptive information, such as the box location, or statistical information such as the traffic at a given interface.

A *NETMAN* command allows the retrieval of one or more attributes from a given 3172. Upon receipt of this command, the 3172 will format an appropriate response string containing the required data, and will send it to the requesting subagent over the same sub-channel pair that the original request was received from.

The 3172 supports the following operations via the *NETMAN* command:

- *Get attribute* used to retrieve one or more attributes
- *Set attribute* used to change the value of one or more attributes
- *Send trap* used to send trap information upon a 3172-detected event.

Note: Only the *get attribute* operation is supported by the subagent included in IBM TCP/IP Version 2 Release 2 for VM.

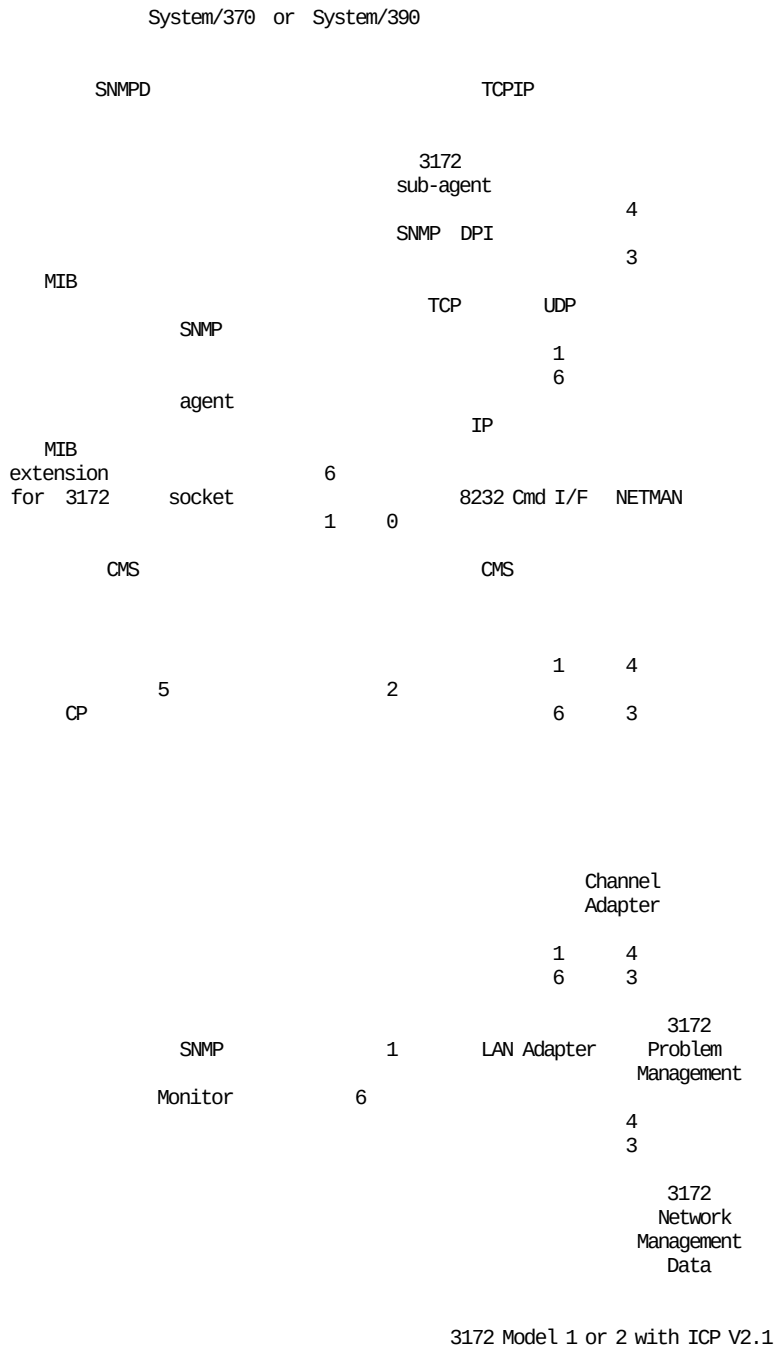


Figure 13. 3172 Network Management Data Flow

Figure 13 on page 28 shows the data flow associated with an *SNMP GET* request/response used to retrieve 3172 Network Management data using the subagent implemented in IBM TCP/IP Version 2 Release 2 for VM:

0 The subagent registers the 3172 enterprise-specific variables to the agent.

1 An SNMP monitor sends an *SNMP GET* request PDU to the agent, to retrieve a 3172 enterprise-specific variable.

2 The agent forwards the request to the subagent through the SNMP DPI interface using a TCP connection.

3 The subagent maps the ASN.1 notation of the requested variable to an attribute index and sends a *NETMAN* command to the 3172, requesting the corresponding attribute.

4 The 3172 processes the request and sends back a *NETMAN* command response, containing the requested attribute, to the subagent.

5 The subagent builds an *SNMP GET* response PDU containing the requested variable and sends it to the agent through the SNMP DPI interface.

6 The agent forwards the *SNMP GET* response PDU, containing the requested variable, to the requesting SNMP monitor.

Note: Appendix F of the *IBM TCP/IP V2 R2 for VM: User's Guide* contains the mapping between the ASN.1 notation and the attribute index for all of the retrievable 3172 network management data.

1.3.7.3 3172 Network Management Variables

3172 Enterprise-Specific Variables

Most of the attributes that the *NETMAN* command provides to the SNMP agent do not map into existing MIB variables. These new items are added as new variables to the IBM Research *private* MIB tree, as shown in Figure 14 on page 30.

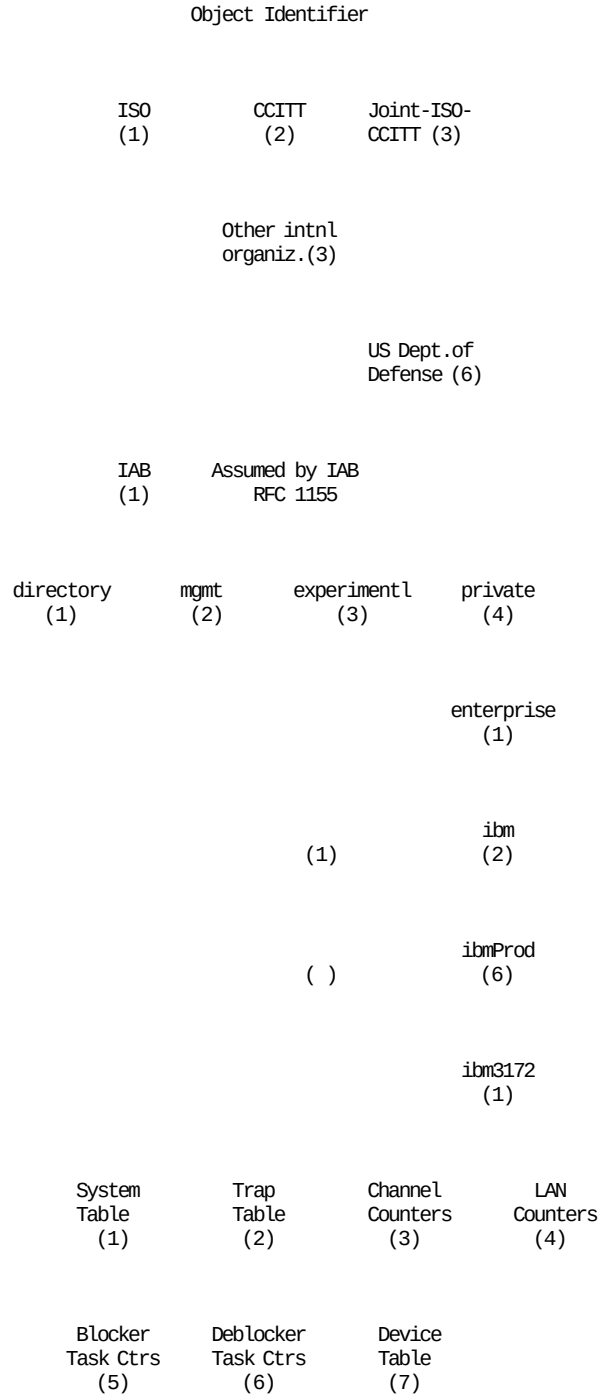


Figure 14. SMI IBM 3172 Enterprise-Specific Variables

The following is a list of the IBM 3172 enterprise-specific MIB objects with their descriptions:

- The IBM 3172 System table (ASN.1 1.3.6.1.4.1.2.6.1.1)

  Contains information about the hardware and software (*ibm3172Descr*), the contact person (*ibm3172Contact*), the physical location (*ibm3172Location*) and number of network interfaces (*ibm3172ifNumber*).

  Following is an example of the *ibm3172Descr* variable content:

  ```
  3172 MODEL 001,
  SERIAL NUMBER 000001234,
  3172 Interconnect Ctlr Program 020100,
  PROGRAM NUMBER 5601433
  ```

  Following is the related extract from the "MIB_DESC DATA" file:

  ```
  *-----------------------------------------------------------------*
  * MIB Variable name | ASN.1 notation         | Type        | TTL  *
  *-----------------------------------------------------------------*
  ibm3172Descr          1.3.6.1.4.1.2.6.1.1.1.  display       900
  ibm3172Contact        1.3.6.1.4.1.2.6.1.1.2.  display       900
  ibm3172Location       1.3.6.1.4.1.2.6.1.1.3.  display       900
  ibm3172ifNumber       1.3.6.1.4.1.2.6.1.1.4.  number          1
  ```

- The IBM 3172 Trap table (ASN.1 1.3.6.1.4.1.2.6.1.2)

  Identifies the trap settings for the interfaces (*ibm3172ifTrapEnable*).

  ```
  *-----------------------------------------------------------------*
  * MIB Variable name | ASN.1 notation         | Type        | TTL  *
  *-----------------------------------------------------------------*
  ibm3172ifTrapEnable   1.3.6.1.4.1.2.6.1.2.1.   number          1
  ```

- The IBM 3172 Channel Counters table (ASN.1 1.3.6.1.4.1.2.6.1.3)

  Identifies the inbound/outbound octets and blocks for each subchannel pair connected to TCP/IP.

  ```
  *-----------------------------------------------------------------*
  * MIB Variable name | ASN.1 notation         | Type        | TTL  *
  *-----------------------------------------------------------------*
  ibm3172ifInChanOctets    1.3.6.1.4.1.2.6.1.3.1.  counter       1
  ibm3172ifOutChanOctets   1.3.6.1.4.1.2.6.1.3.2.  counter       1
  ibm3172ifInChanBlocks    1.3.6.1.4.1.2.6.1.3.3.  counter       1
  ibm3172ifOutChanBlocks   1.3.6.1.4.1.2.6.1.3.4.  counter       1
  ```

- The IBM 3172 LAN Counters table (ASN.1 1.3.6.1.4.1.2.6.1.4)

  Identifies the inbound/outbound octets and frames, transmission errors and discarded frames for each LAN interface.

  ```
  *-----------------------------------------------------------------*
  * MIB Variable name | ASN.1 notation         | Type        | TTL  *
  *-----------------------------------------------------------------*
  ibm3172ifInLANOctets     1.3.6.1.4.1.2.6.1.4.1.  counter       1
  ibm3172ifOutLANOctets    1.3.6.1.4.1.2.6.1.4.2.  counter       1
  ibm3172ifInLANFrames     1.3.6.1.4.1.2.6.1.4.3.  counter       1
  ibm3172ifOutLANFrames    1.3.6.1.4.1.2.6.1.4.4.  counter       1
  ibm3172ifInLANErrors     1.3.6.1.4.1.2.6.1.4.5.  counter       1
  ibm3172ifOutLANErrors    1.3.6.1.4.1.2.6.1.4.6.  counter       1
  ibm3172ifInLANDiscards   1.3.6.1.4.1.2.6.1.4.7.  counter       1
  ibm3172ifOutLANDiscards  1.3.6.1.4.1.2.6.1.4.8.  counter       1
  ```

- The IBM 3172 Blocker Task table (ASN.1 1.3.6.1.4.1.2.6.1.5)

  Identifies the inbound (from the LAN) octets and frames received/sent by the blocker task, the inbound frames in error and those discarded.

```
*-------------------------------------------------------------------*
* MIB Variable name | ASN.1 notation          | Type        | TTL  *
*-------------------------------------------------------------------*
ibm3172ifBlkRcvOctets    1.3.6.1.4.1.2.6.1.5.1.  counter       1
ibm3172ifBlkXmitOctets   1.3.6.1.4.1.2.6.1.5.2.  counter       1
ibm3172ifBlkRcvFrames    1.3.6.1.4.1.2.6.1.5.3.  counter       1
ibm3172ifBlkXmitBlocks   1.3.6.1.4.1.2.6.1.5.4.  counter       1
ibm3172ifInBlkErrors     1.3.6.1.4.1.2.6.1.5.5.  counter       1
ibm3172ifInBlkDiscards   1.3.6.1.4.1.2.6.1.5.6.  counter       1
```

- The IBM 3172 Deblocker Task table (ASN.1 1.3.6.1.4.1.2.6.1.6)

  Identifies the outbound (from the channel adapter) octets and frames received/sent by the deblocker task, the outbound frames in error and those discarded.

```
*-------------------------------------------------------------------*
* MIB Variable name | ASN.1 notation          | Type        | TTL  *
*-------------------------------------------------------------------*
ibm3172ifDblkRcvOctets   1.3.6.1.4.1.2.6.1.6.1.  counter       1
ibm3172ifDblkXmitOctets  1.3.6.1.4.1.2.6.1.6.2.  counter       1
ibm3172ifDblkRcvBlocks   1.3.6.1.4.1.2.6.1.6.3.  counter       1
ibm3172ifDblkXmitFrames  1.3.6.1.4.1.2.6.1.6.4.  counter       1
ibm3172ifOutDblkErrors   1.3.6.1.4.1.2.6.1.6.5.  counter       1
ibm3172ifOutDblkDiscards 1.3.6.1.4.1.2.6.1.6.6.  counter       1
```

- The IBM 3172 Device table (ASN.1 1.3.6.1.4.1.2.6.1.7)

  Identifies the devices associated with this interface.

```
*-------------------------------------------------------------------*
* MIB Variable name | ASN.1 notation          | Type        | TTL  *
*-------------------------------------------------------------------*
ibm3172ifDeviceNumber  1.3.6.1.4.1.2.6.1.7.1.  counter       1
```

In the above descriptions:

- The term ***inbound*** denotes the direction from the LANs or to the host.
- The term ***outbound*** denotes the direction from the host or out to the LANs.

Please refer to *IBM TCP/IP V2 R2 for VM: User's Guide - Appendix E* for a more detailed description of these variables.

3172 Standard MIB Variables

In addition to the previously described 3172 enterprise-specific variables, some standard MIB variables are supported for each 3172 LAN adapter connected to the TCP/IP network. They are highlighted in Figure 15.

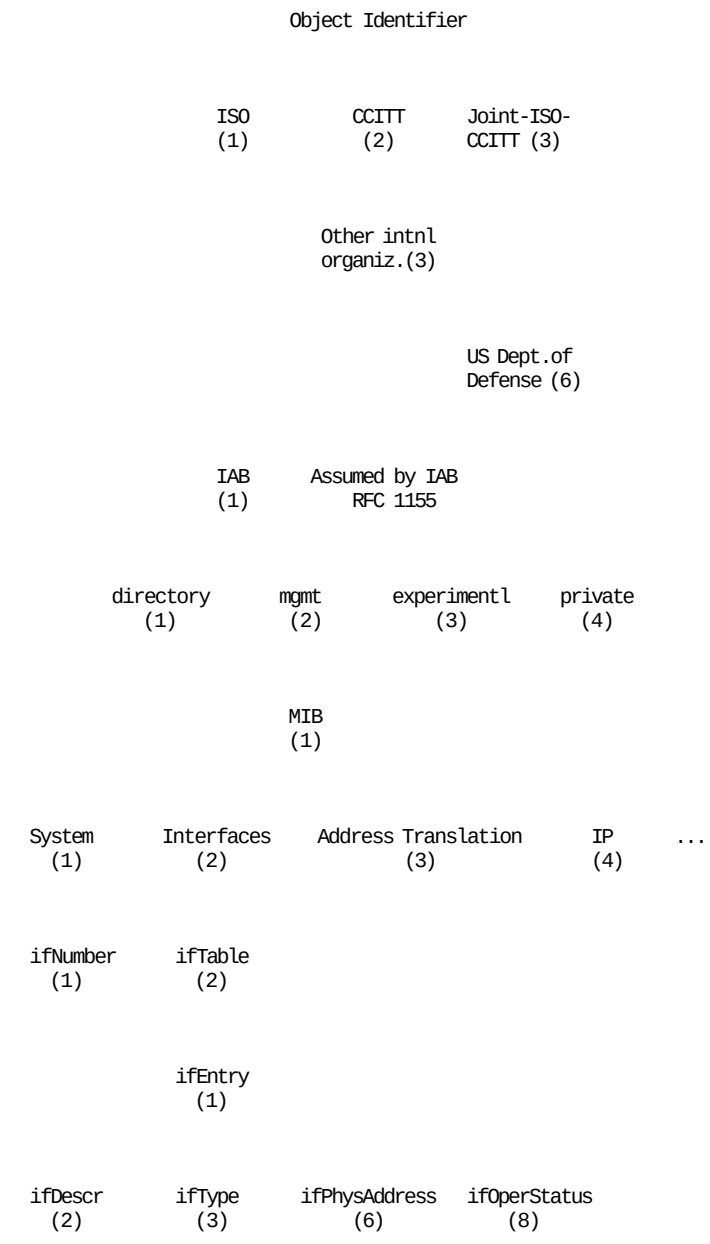


Figure 15. SMI IBM 3172 Supported Standard MIB Variables

Following is a description of these variables:

- *ifDescr*

  Contains the text description of the LAN adapter associated with the interface whose IP address is specified in the *SNMP GET* request. These are hard-coded phrases:

  ″Interconnect Controller Token-Ring Adapter″

  ″Interconnect Controller Ethernet Adapter″

  ″Interconnect Controller PC Network Adapter″

  ″Interconnect Controller FDDI Adapter″

- *ifPhysAddress*

  Contains the physical address of the interface.

- *ifType*

  Contains the type of LAN adapter associated with the interface.

- *ifOperStatus*

  Contains the operational status (″Up″, ″Down″ or ″Testing″) of the LAN adapter associated with the interface.

Referencing 3172 Variables

The 3172 variables are referenced by a single element instance identifier.

- The 3172 system variables (*ibm3172Descr*, *ibm3172Contact*, *ibm3172Location*, *ibm3172ifNumber*) pertain to an entire 3172, and their instance identifier number is assigned in the order of those *LCS DEVICE* statements of the ″PROFILE TCPIP″, which include the *NETMAN* keyword (see Section 2.6.5, “3172 SNMP Configuration” on page 87 for more details about how to configure the 3172 Network Management capability). No instance identifiers are assigned for non-LCS devices or for LCS devices without the *NETMAN* keyword coded. For example, given the following configuration:

  ```
  DEVICE LCS1 LCS cuu NETMAN
  LINK TR1 IBMTR 0 LCS1
  LINK EN1 ETHERNET 1 LCS1

  DEVICE LCS2 LCS cuu
  LINK EN2 ETHERNET 0 LCS2
  LINK EN3 ETHERNET 1 LCS2

  DEVICE LCS3 LCS cuu NETMAN
  LINK TR2 IBMTR 0 LCS3
  LINK TR3 IBMTR 0 LCS3
  ```

  ibm3172Descr.1 describes the first 3172 (LCS1), while *ibm3172Descr.2* describes the third 3172 (LCS3).

- For the other variables, which are interface specific, the instance identifier is the one which is used to reference the MIB Interfaces table entry for that interface, that is, *ifIndex*. For example, given the following configuration:

```
DEVICE LCS1 LCS cuu NETMAN
LINK TR1 IBMTR 0 LCS1
LINK EN1 ETHERNET 1 LCS1

DEVICE LCS2 LCS cuu
LINK EN2 ETHERNET 0 LCS2
LINK EN3 ETHERNET 1 LCS2

DEVICE SNADVM18 SNAIUCV SNALINK RAIATC1 SNALNKA
LINK SNALMV18 IUCV 2 SNADVM18

DEVICE LCS3 LCS cuu NETMAN
LINK TR2 IBMTR 0 LCS3
LINK TR3 IBMTR 1 LCS3
```

  The instance identifier associated with each interface is:

TR1	1
EN1	2
EN2	3
EN3	4
SNALMV18	5
TR2	6
TR3	7

  1. An *SNMP GET ibm3172ifLANCounters.2* command would retrieve information about *EN1*.
  2. An *SNMP GETNEXT* command would then retrieve information about *TR2* (the devices *LCS2* and *SNADVM18* which do not support the *NETMAN* command interface are skipped over and the *ibm3172ifLANCounters.6* variable is returned).
  3. An *SNMP GET ibm3172ifLANCounters.4* command would be responded to with a *NO SUCH NAME* answer (the corresponding interface *SNALMV18* belongs to a device which does not support the *NETMAN* command interface).

1.3.8 Network DataBase System (NDB)

1.3.8.1 Overview

The Network DataBase System feature (NDB) is an RPC-based client-server application which allows a client workstation to query a remote SQL/DS relational database using the Structured Query Language (SQL), through the TCP/IP network.

As shown on Figure 16, this new feature included in IBM TCP/IP Version 2 Release 2 for VM consists of:

- A client application (NDB client) to be installed and run on either a RISC System/6000* or a Sun Microsystems workstation
- A server application (NDB server) to be installed and run in a virtual machine on the VM system where the SQL/DS database to be accessed resides.

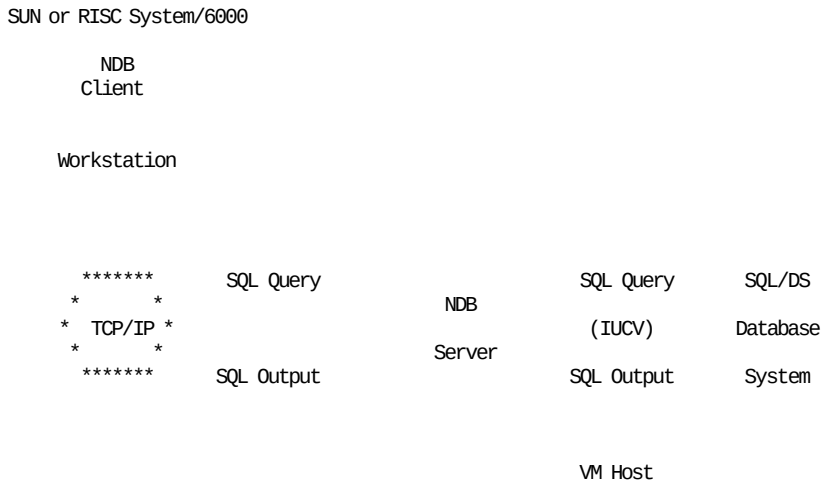


Figure 16. Network Database System

Limitations

SQL has many data types. However, the most frequently used are integers and characters. Currently, the data conversion process implemented in the Network DataBase System handles data type INT, SMALLINT, CHAR, VARCHAR and FLOAT. They can be nulls or without nulls.

The present implementation of NDB allows only R/O access to an SQL/DS table; that is, the NDB server will accept only *select* and *view* statements and will reject any statement which would change the content of a table.

RACF in not supported. That is, when a workstation issuing NDB commands is prompted for a password, the password must be the one from the VM directory.

1.3.8.2 Implementation before APAR PN17869 on NDB is Installed

Please refer to Section 1.3.8.3, “Enhancements in IBM TCP/IP Version 2 Release 2 for VM with APAR PN17869 on NDB Installed” on page 39 for more details about the NDB server implementation if you have this PTF installed.

1. **The NDB client application** is delivered in source code and must be downloaded to the client workstation where it is intended to be used (please refer to Section 2.16.3, “Installing the NDB Client” on page 127 for details about the required installation tasks).

   It is implemented under the form of a command (*ndbclnt*) whose basic format allows sending a single SQL query statement to the remote NDB server residing on host *<hostname>*:

   - Either interactively:

     ```
     ndbclnt <hostname> †<SQL statement>†
     ```

   - Or from a ″C″ application program:

     ```
     ndbclnt (†<hostname>†, †<SQL statement>†)
     ```

   The implementation also allows you to interactively send multiple queries using the following format:

   ```
   ndbclnt <hostname> †begin†
   <SQL statement 1>
   <SQL statement 2>
   ...
   end
   ```

2. **The NDB server application**, like the other TCP/IP servers implemented in IBM TCP/IP Version 2 Release 2 for VM, runs in a disconnected virtual machine (NDBSERVE). It is delivered in both source and object code, and its installation under VM may require a partial re-compilation using the C/370 compiler (please refer to Section 2.16.1, “Installing the NDB Server before APAR PN17869 on NDB” on page 120 for details about the required installation tasks).

Like any RPC-based application, the Network DataBase System requires the presence of a Portmapper** that permits the NDB client to look up the port number of the remote NDB server.

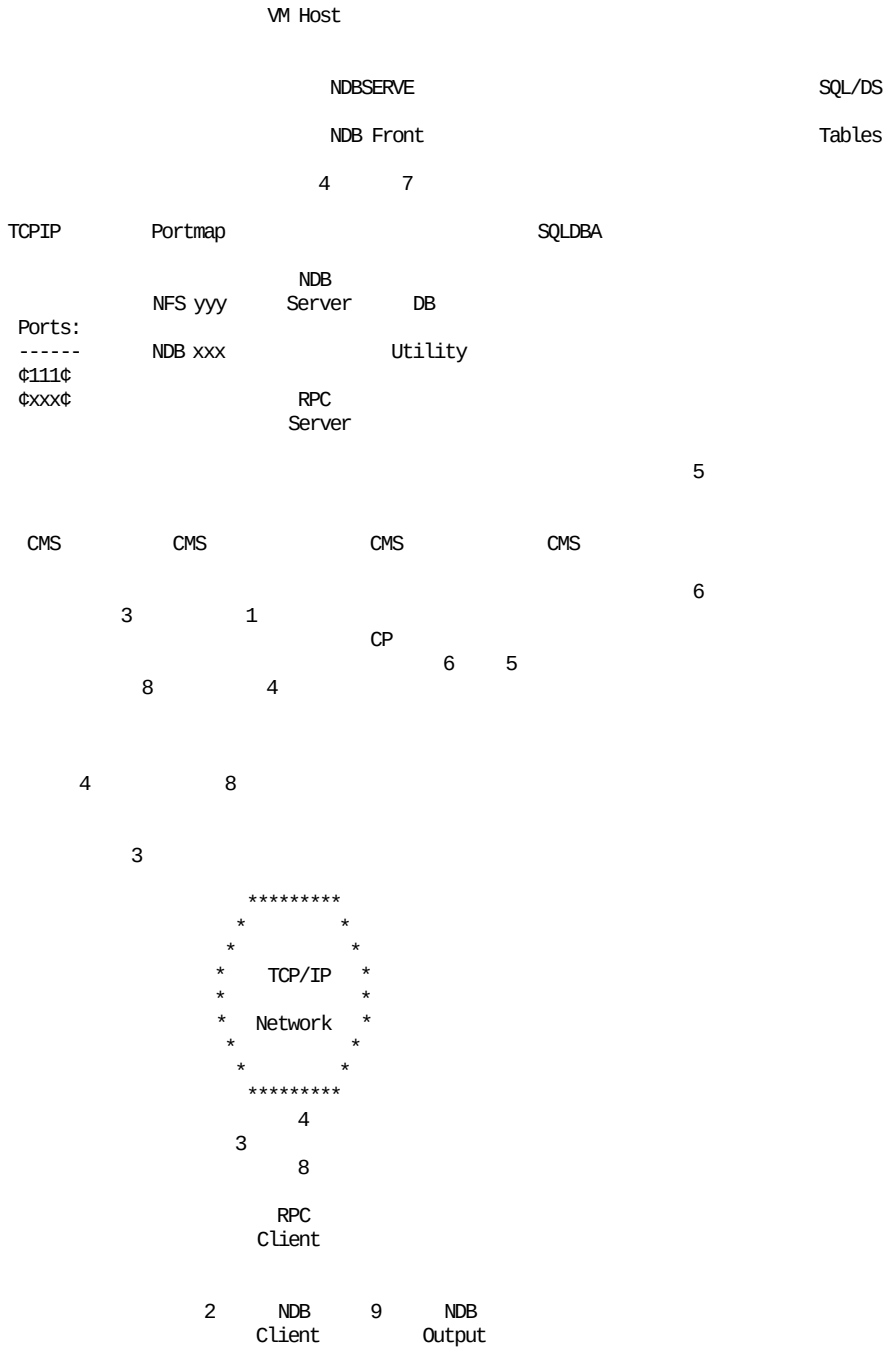


Figure 17. Network Database System Structure and Flow

As shown in Figure 17, a basic NDB client/server interaction involves the following steps:

1 As part of its startup, the NDB server registers its programs and procedures to the Portmapper which is run, on a VM system, by the PORTMAP virtual machine. The Portmapper assigns the ports which are necessary to handle the access to the NDB server's programs and procedures, updates its table with the corresponding port mapping entries (*portmaps*) and returns the port information to the NDB server.

Now that the NDB server knows the ports it will be listening on, the NDBSERVE virtual machine can complete its startup.

2 A user (or an application program) issues the *ndbclnt* command containing the SQL query to be sent to the remote NDB server.

3 The NDB client program parses the request and passes it to the client stub (RPC client) through the RPC call interface. The client stub sends a request (RPC call message) to the remote host′s Portmapper well known port (111), asking for the port to use in order to access the remote NDB server. The Portmapper returns the relevant port number in an RPC reply message.

4 Using the port information received in the previous step, the client stub program sends the SQL query to the NDB server in an RPC call message.

The NDB server processes the request that it receives from the RPC server through the RPC call interface, according to the SQL query type and passes it to the NDB front, which is responsible for data conversion and security/password checking.

5 The DB utility performs the actual access to the required SQL/DS table through the SQLDBA database service virtual machine, to retrieve the data.

6 7 8 The retrieved data is sent back through the same path in an RPC reply message, to the client stub program, which passes it to the NDB client program.

9 By default, the NDB client program writes the retrieved data into a flat file called ″NDB OUTPUT″.

1.3.8.3 Enhancements in IBM TCP/IP Version 2 Release 2 for VM with APAR PN17869 on NDB Installed

With the above implementation, if one NDB client starts a Unit Of Work, the NDB server virtual machine will be unavailable for other users until the Unit Of Work has been terminated by the first user. A Unit Of Work is either a single SQL statement or a series of SQL statements surrounded by a *BEGIN* and an *END* statement. If the client is an interactive user, a Unit Of Work may last for more minutes - blocking the NDB server virtual machine for that period.

PTF for APAR PN17869 on NDB enhances the capabilities of the Network DataBase System such that multiple Network DataBase server machines are now supported, thus allowing to serve more NDB clients concurrently. To avoid having the NDB clients try one NDB server RPC program number after the other in order to find a free server, the implementation is based on a scheduler virtual machine, called the NDB PortMap manager server (PORTSRV). This server has an RPC program number associated to it, that all the NDB clients know. When an NDB client wants to start a Unit Of Work with an NDB server, it contacts the NDB PortMap manager first to get the RPC program number of a free NDB server. Using the received RPC program number, it then connects to the corresponding NDB server.

Thus, if you decide to have more than one NDB server, you will have to start (in the ″PROFILE TCPIP″) at least three virtual machines: PORTMAP, PORTSRV and as many NDB servers as you want (NDBSRV1, NDBSRV2, etc).

The role of each required virtual machine is:

- PORTMAP: the Network DataBase System requires, like any other RPC-based application, the presence of the Portmapper. The Portmapper is the one which receives the initial message from the client and will return the port of PORTSRV. It will also return the port of a ″free″ NDB server. PORTMAP runs the ″PORTMAP MODULE″.
- PORTSRV: Each NDB server will be assigned a different program number by the PortMap manager. The NDBCLNT command is sent from the client to the server as a Unit Of Work (UOW). As long as the end statement has not been issued, the port manager will consider that the NDB server machine is busy. The PortMap manager will keep track of the ″free/busy″ status of each NDB server and will return the program number of a ″free″ NDB server to any NDB client which requests a connection to an NDB server. PORTSRV runs the ″PORTSRVS MODULE″.
- NDBSRV1, NDBSRV2: These are the NDB server machines. They are the ones actually querying the SQL database. They are started using the ″PORTCLNT MODULE″ which, in turn, calls the ″NDBSRVS MODULE″.

Figure 18 shows the flow of the Network DataBase System in its new implementation.

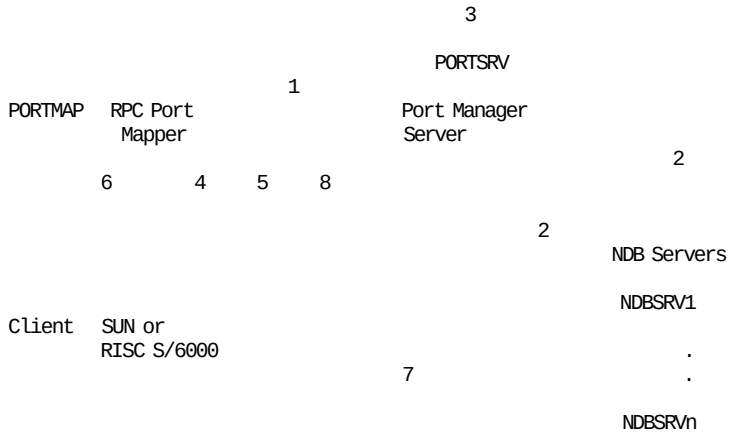


Figure 18. Network Database System Structure and Flow

The flow inside the NDB server itself did not change. Please refer to Figure 17 on page 38 for more information.

1 After having started the Portmapper (PORTMAP), the first virtual machine you start, is the NDB PortMap manager (PORTSRV) which will, during its startup, get a socket and register a fixed RPC program number (X′20000050′) with the Portmapper. It will then wait for Remote Procedure Calls.

2 The next virtual machine you start, is the Network DataBase server (NDBSRV1). It will immediately issue an RPC call to the NDB PortMap manager (PORTSRV) to request an RPC program number, which the NDB PortMap manager will assign to that NDB server.

You may start more NDB server virtual machines (NDBSRV2, etc.). They will each contact the NDB PortMap manager to request a unique RPC program number.

3 On return from the NDB PortMap manager, each NDB server will get a socket and register it's RPC program number with the Portmapper. The NDB servers will then wait for a Remote Procedure Calls to come in.

4 When the NDB client starts up, it knows the RPC program number of the NDB PortMap manager (it has been hard coded into the client code). It will, via the Portmapper running on the NDB server host, get the port of the NDB PortMap manager.

5 It will then send a request to the NDB PortMap manager for the RPC program number of a free NDB Server.

6 When the client has received the RPC program number of a free NDB server (for example NDBSRV2), it will again via the Portmapper get the port of this NDB server.

7 When the client has received the port number of NDBSRV2, it will initiate a session with it.

8 When the client has finished its session with NDBSRV2, it again contacts the NDB PortMap manager with a message indicating it is done with the NDB server. The NDB PortMap manager now knows, that this NDB server (NDBSRV2) is free for new requests.

As far as the client platform is concerned, there is no change in the way the SQL database is queried.

1.3.9 Socket API Enhancements

IBM TCP/IP Version 2 Release 2 for VM includes some enhancements in its implementation of the socket API:

- A simplified procedure to perform the transfer of a socket from one program to another
- A new feature intended to improve the performance of some UDP-based socket applications.

1.3.9.1 Socket Transfer Procedure

The *givesocket()* call is used by a program that wishes to transfer control of a socket to another program, which, in turn, uses the *takesocket()* call to take control of the socket.

Typically but not necessarily, the program giving the socket is a *master* program, and the program taking the socket is a *slave* program spawned by the master.

Original Socket Transfer Procedure

In the original V2R1 release, the following steps were necessary to perform the transfer of a socket:

1. The master calls *getclientid()* to obtain its client ID, and passes it, along with the descriptor of the socket to be given, to the slave via the slave′s startup parameter list.
2. The slave calls *getclientid()* to obtain its client ID, and passes it back to the master, via a user-defined mechanism.
3. The master calls *givesocket()*, specifying the slave′s client ID and the descriptor of the socket to be given. It then tells the slave, via a user-defined mechanism, that the socket has been given.
4. The slave calls *takesocket()*, specifying the master′s client ID and socket descriptor (see step 1 above). It then tells the master, via a user-defined mechanism, that the socket has been taken.
5. The master closes the given socket.

Steps 2, 3 and 4 presuppose the existence of a ″user-defined mechanism″ for passing information between the master and the slave.

Clientid Structure Definition

The socket transfer procedure described above does not require knowledge of the individual fields of the *clientid* structure. When using the new procedure described below under ″Simplified Socket Transfer Procedure″, your program will refer to individual fields of this structure which is defined in ″SOCKET H″, as follows:

Clientid Structure

```
struct clientid {
   int domain;
   char name[8 ;
   char subtaskname[8 ;
   char reserved[20 ;
};
```

Simplified Socket Transfer Procedure

With IBM TCP/IP Version 2 Release 2 for VM, the following steps suffice to perform the transfer of a socket:

1. The master calls *givesocket()*, with:
   - The *clientid* parameter pointing to a structure filled in as follows:

     domain AF_INET (or 2)

     name Slave′s virtual machine name, left-justified and padded with blanks. The slave may run in the same virtual machine as the master, in which case this field is set to the master′s virtual machine name.

     subtaskname Blanks

     reserved Binary zeroes

   - The *d* parameter specifying the descriptor of the socket to be given.

   Then the master calls *getclientid()* to obtain its client ID, and passes it, along with the descriptor of the socket to be given, to the slave via the slave′s startup parameter list.
2. The slave calls *takesocket()*, specifying the master′s client ID and socket descriptor.
3. Asynchronously with step 2, the master uses *select()* to test the given socket for an exception condition. When *select()* reports that an exceptional condition is pending, the master calls *close()* to free up the given socket.

1.3.9.2 UDP Bulkmode Interface Feature

Overview

In order to communicate with its peer through a TCP/IP network, a TCP/IP application built upon the socket interface provided with IBM TCP/IP Version 2 Release 1 for VM, and running in a virtual machine on a VM system, sends/receives data to/from the TCPIP virtual machine using the IUCV mechanism.

Up to V2R1, when data transfer calls are issued to a socket by the application, the associated IUCV data transfer operations which occur between the application and the TCPIP virtual machine actually transfer one datagram per operation. A new feature called the **UDP Bulkmode Socket Interface** is included in IBM TCP/IP Version 2 Release 2 for VM, which allows the datagrams to be

queued on the application side of the IUCV connection and allows the transfer of multiple datagrams to/from the TCPIP virtual machine in a single IUCV operation.

Note: The application still sends/receives datagrams one-at-a-time using the existing data transfer socket calls (*send(), recv(), sendto(), recvfrom(), sendmsg(), recvmsg(), read()* and *write()*). The description above only refers to the transfer of a burst of datagrams on the IUCV connection between the application and the TCPIP virtual machine.

Important

This feature only applies to datagram-type sockets (*SOCK_DGRAM*).

Implementation

- **Initialization**

  A socket is placed in bulkmode using a new socket call, *setibmsockopt()*, and the associated structure called *ibm_bulkmode_struct* included in the ″SOCKET H″ header file. It is possible to configure up to two queues for each socket: one for the incoming datagrams and one for the outgoing datagrams. The configurable options are:

  – The size of the queue(s)

  – Whether the socket library will be testing addressability to the application′s message buffer on each socket call

  – Whether the data is actually moved or whether pointers are used, for outgoing messages.

- **Incoming Datagrams**

  Once a socket has been set into bulkmode for receives, the normal socket calls are used to receive the datagrams. One datagram is received in the application per socket call, as before.

  For the *recv(), recvfrom(), recvmsg(), and read()* socket calls, a queue of pending datagrams is kept on the application side of the IUCV interface. When one of these calls is issued, this queue is checked first. If it is not empty, the socket call draws from that queue. If it is empty, an IUCV request is sent to the TCPIP virtual machine asking it to transfer as many datagrams as will fit in the application-side queue, that are pending to be received on that socket. They are passed over in a single IUCV operation; the oldest one is returned to the caller, and the application-side queue is replenished. Subsequent calls of this type will draw from the application-side queue until it becomes empty. Then, a new IUCV request will be sent to the TCPIP virtual machine.

- **Outgoing Datagrams**

  Once a socket has been set into bulkmode for sends, the normal socket calls are used to send the datagrams.

  For the *send(), sendto(), sendmsg(), and write()* socket calls, a separate queue is allocated on the application side of the IUCV interface. As socket calls of this type are issued, the datagrams will be queued up, either by copying the data to a buffer, or by saving pointers to the datagrams (this is controlled by the *b_movedata* flag of the *ibm_bulkmode_struct* structure; see ″Structure Used to Initialize Bulkmode″ later in this chapter). The application

can queue up multiple datagrams, and when there are no more to send out, it can issue a new socket call, *ibmsflush()*, to cause the queued datagrams to be sent over to the TCPIP virtual machine in a single IUCV operation.

The application-side queue will be flushed when one of the following occurs:

- The application-side send queue is full
- The socket is closed
- An *ibmsflush()* socket call is issued
- Another *setibmsockopt()* call is issued.

- **Monitoring**

  For monitoring purposes, three parameters can be read from the socket control block, using the new *getibmsockopt()* socket call:

  1. The maximum number of bytes that can be allocated as a send queue:

     This number is the same as the size of the TCPIP data buffer. In bulkmode, when you configure a send queue, one data buffer is allocated on the TCPIP side of the IUCV interface. This is to receive datagrams transferred from the application to TCPIP. You will not be allowed to configure a send queue on the application side which is larger than the TCPIP data buffer size.

  2. The number of IUCV requests that have been issued to TCPIP to transfer incoming datagrams, that is, as the result of *recv(), recvfrom(), recvmsg(), and read()* socket calls on the bulkmode socket:

     This can be used if you are interested in monitoring the effectiveness of the bulkmode socket interface feature.

  3. The number of IUCV requests that have been issued to TCPIP to transfer outgoing datagrams, that is, as the result of *send(), sendto(), sendmsg(), and write()* on the bulkmode socket:

     This can also be used if you are interested in monitoring the effectiveness of the feature.

Benefits

By allowing datagrams to be queued on the application side of the IUCV interface, and making it possible to transfer multiple datagrams per IUCV operation, the Bulkmode Socket Interface can improve the performance of some socket applications in the following areas:

- Reduction of the CPU usage (the overhead of the IUCV operation is spread across multiple datagrams)
- Possible increase of the data transfer throughput rate (depending on the particular environment).

The amount of improvement will depend mainly on the message size used by the application, and on the application protocol used. Applications which tend to use small or moderate size messages and which use a protocol that allows datagrams to be sent and received in bursts, will benefit more from this feature.

The parameters used to initialize the bulkmode have been made available to the socket application programmer to allow flexibility in initializing and using this option.

- **Benchmark**

  Sample send/receive benchmark programs with multiple packets were executed with and without the bulkmode option, in a dedicated system environment with the following parameters:

```
Host               : 3090-300J
Workstation        : RISC System/6000
Number of packets  : 10,000
Packet Size        : 1460 Bytes
CPU Monitoring tool: RMFMON
```

  The CPU time was monitored during the transfer. The results showed 60-65% reduction in CPU time for bulkmode transfer, as compared to the non-bulkmode transfer.

  The average queue sizes observed were:

```
6-8  datagrams/IUCV operation on recvfrom() calls (TCPIP --  appl.)
5-12 datagrams/IUCV operation on sendto()   calls (appl. --  TCPIP)
```

- **Hints**

  It may be that your application is not suited to using bulkmode for sends (for example, queueing up the datagrams may introduce undesired latency). If this is the case, you should indicate that a send queue is not to be configured by specifying 0 for the value of *b_max_send_queue_size* (see below *"*Structure Used to Initialize Bulkmode*"*) when you issue the *setibmsockopt()* call to initialize the bulkmode.

  Using bulkmode for sends is generally desirable if you are using a windowing/ACK scheme over the datagrams, or, in general, whenever multiple datagrams (possibly for different destinations) are sent over the same socket in quick succession.

  Using bulkmode for receives should operate transparently.

Structure Used to Initialize Bulkmode

A UDP socket can be configured to operate in this mode using a socket library routine called *setibmsockopt()*. A parameter of this routine is the structure called *ibm_bulkmode_struct*, which controls the options available to the application for this mode. This structure is defined in the header file, ″SOCKET H″ as follows:

ibm_bulkmode_struct

```
/*
 * Structure used for manipulating IBM bulkmode datagram sockets.
 */
#define SO_BULKMODE      0x8000 /* set/get options for IBM bulkmode  */
                                /* sockets.                          */
#define SO_NONBLOCKLOCAL 0x8001 /* don¢t block if local queue is     */
                                /* empty.                            */

struct  ibm_bulkmode_struct {
        int     b_onoff;          /* Option on/off                   */
        int     b_max_receive_queue_size; /* Maximum receiving queue */
                                          /* size (in bytes).        */
        int     b_max_send_queue_size; /* Maximum sending queue size */
                                       /* (in bytes).               */
        int     b_move_data;      /* For outbound sockets:           */
           /* non-zero: The data is moved into buffers in the queue. */
           /*           The client¢s buffers can be reused right away*/
           /* zero:     Pointers to the data are saved in the queue. */
           /*           The buffers should not be reused until the   */
           /*           queue has been flushed (generally by issuing */
           /*           an ibmsflush()).                             */
        int b_teststor;    /* For either inbound or outbound sockets:*/
           /* non-zero: The address of the message buffer and the    */
           /*           message buffer itself is checked for address-*/
           /*           ability during each socket call.  The error, */
           /*           EFAULT, is set if there is an exception.     */
           /* zero:     The above checking is not done by the socket */
           /*           library routines. If there is an addressing  */
           /*           exception, the normal runtime error handling */
           /*           environment will produce a message.  For     */
           /*           examples of this, see the BULKMODE.README    */
           /*           file.                                        */
           /*                                                        */
           /* The following are returned by getibmsockopt():         */
        int b_max_send_queue_size_avail; /* The max. send queue size */
                                   /* (in bytes) that we can set by   */
                                   /* b_max_send_queue_size option on */
                                   /* setibmsockopt().                */
        int b_num_IUCVs_sent;      /* Number of actual IUCVs issued in*/
                                   /* sending datagrams to TCPIP.     */
        int b_num_IUCVs_received; /* Number of actual IUCVs issued in*/
                                   /* receiving datagrams from TCPIP. */
};
```

- ***setibmsockopt()***

  The syntax is as follows:

```
int setibmsockopt(s, level, optname, optval, optlen)
int s;        /* The socket descriptor.                          */
int level;    /* SOL_SOCKET                                      */
int optname;  /* Either SO_BULKMODE or SO_NONBLOCKLOCAL.          */
char *optval; /* Points to option data.                          */
              /* When optname is SO_BULKMODE, this should point  */
              /* to an ibm_bulkmode_struct structure.            */
              /* When optname is SO_NONBLOCKLOCAL, this should   */
              /* point to a zero or non-zero integer to control  */
              /* whether the socket library returns EWOULDBLOCK  */
              /* if the application-side queue is empty.  This   */
              /* allows the application to gobble up whatever    */
              /* datagrams are in it¢s local queue and act on    */
              /* them.                                           */
int optlen;   /* Specifies the length of the option data.        */
```

  Note:

  - Specifying a value of 0 for *b_max_receive_queue_size* causes queueing not to be performed on inbound datagrams.
  - Specifying a value of 0 for *b_max_send_queue_size* causes queueing not to be performed on outbound datagrams.

- ***getibmsockopt()***

  The *getibmsockopt()* call returns:

  - The current settings of the parameters
  - The maximum setting that can be used for the *b_max_send_queue_size* on a *setibmsockopt()* call
  - The actual number of IUCV operations that have been executed on behalf of the socket application for each of the bulkmode send and receive queues.

  Its syntax is as follows:

```
int getibmsockopt(s, level, optname, optval, optlen)
int s;        /* The socket descriptor.                          */
int level;    /* SOL_SOCKET                                      */
int optname;  /* SO_BULKMODE                                     */
char *optval; /* Points to option data.                          */
              /* This should point to an ibm_bulkmode_struct     */
              /* structure.                                      */
int *optlen;  /* Points to the length of the option data.        */
```

- ***ibmsflush()***

  The syntax is as follows:

```
int ibmsflush(s)
int s;        /* The socket descriptor.                          */
```

Reminder

For outbound sockets, the application-side queue is flushed (that is, its contents are sent over to the TCPIP virtual machine in one IUCV operation), if any of the following occurs:

- An *ibmsflush()* call is issued on the socket
- The queue is full and another send-type socket call is issued
- The socket is closed
- Another *setibmsockopt()* call is issued.

Sample Source Code to Initialize the Bulkmode

Following is a sample piece of code which initializes bulkmode for sending and receiving datagrams:

Sample

```
#include  socket.h
#include  tcperror.h

 { struct ibm_bulkmode_struct bulkstr;
   int optlen, rc;
   optlen = sizeof(bulkstr);
   rc = getibmsockopt(s, SOL_SOCKET, SO_BULKMODE, &bulkstr, &optlen);
   if (rc  0)
     { tcperror(†on getibmsockopt()†);
       exit(-1);
     }
   fprintf(stream,†%d byte buffer available for outbound queue.\n†,
           bulkstr.b_max_send_queue_size_avail);
   bulkstr.b_max_send_queue_size = bulkstr.b_max_send_queue_size_avail;
   bulkstr.b_onoff = 1;
   bulkstr.b_teststor = 0;
   bulkstr.b_move_data = 1;
   bulkstr.b_max_receive_queue_size = 65536;
   rc = setibmsockopt(s, SOL_SOCKET, SO_BULKMODE, &bulkstr, optlen);
   if (rc  0)
     { tcperror(†on setibmsockopt()†);
       exit(-1);
     }
 }
```

Note:

- The socket calls that receive datagrams in your program (*recvfrom(), recv(), read(), or recvmsg()*) do not need to be changed.
- If you are using a queue for sending datagrams (by specifying a non-zero value for *b_max_send_queue_size*, then you should modify your code to call *ibmsflush()* at appropriate points, such as after you send a burst of datagrams that is normally followed by a pause.

1.3.10 RISC System/6000 IP Connection

This new connection type is fully described in the *Block Multiplexer Channel Adapter: User's Guide and Programming Reference* manual.

1.3.10.1 Overview

The Block Multiplexer Channel Adapter connectivity allows the system unit (RISC System/6000) to communicate with an S/370 or S/390 host. The support includes a user-written protocol across the S/370 channel, similar to CTC, or the TCP/IP protocol suite that requires IBM TCP/IP Version 2 Release 2 for VM. The support of IBM TCP/IP Version 2 Release 2 for VM is achieved via the high-performance Common Link Access to Workstation (CLAW) protocol. This protocol improves the performance on the S/370 processor by reducing the number of I/O interrupts to the CLAW host. Both CLAW and normal (non-CLAW) modes are supported and are part of the configuration options using SMIT. The non-CLAW mode is similar to the Channel-To-Channel (CTC) protocol. This connectivity allows the system unit to be used as a gateway between the S/370 VM host and the downstream networks that consist of LANs or WANs. AIX Version 3.2 or later of the operating system is required for this support.

Warning

Please note that PTF UN18074 and PTF UN19085 are required on IBM TCP/IP Version 2 Release 2 for VM to support the RISC System/6000 IP connection.

1.3.10.2 Supported S/370 or S/390 Systems

The following S/370 or S/390 systems are supported:

Table 1. Supported Systems

Host System	Maximum Speed (Mbps)	3044 Support
3090	4.5	Yes
9221	4.5•	Yes
9121	4.5	Yes
9021	4.5	Yes

Note:

- The system unit only supports the 4.5 MBps parallel channel on the 9221. The 3 MBps Block Multiplexer Channel is not supported.

The system unit can attach (by means of the 9034 ESCON Converter Model 1) to an ESCON environment, which includes an S/370 or S/390 host. This connection increases the maximum distance between the system unit and the S/370 or S/390 from 400 feet (122 meters) to 3 km, maintaining the maximum speed of 4.5 MBps.

The 3044 Channel Extender Model 2 is also supported by the system unit. The 3044 allows the channel to be extended up to 3 km. The maximum speed is 4.5 MBps.

1.3.10.3 Supported Features

The Block Multiplexer Channel adapter supports the following channel features:

- Data Streaming

  This allows data to be transferred on the channel without interlocking. This mode maximizes the channel data transfer rate. The data streaming mode supports host channels with speeds of up to 4.5 MBps.

- High-Speed DC Interlock (DCI)

  This feature permits data transfer to take place at a higher data rate than that achievable when only service in and service out are used.

The following channel features are *not* supported:

- Byte Multiplex mode
- Selector Channel mode
- Bus Extension feature
- I/O Error Alert feature
- Dynamic Reconnection feature
- XA Multipath.

In addition, the system unit does *not* support the Remote Power Interface.

1.3.10.4 Supported Configurations

The system unit supports at most two Multiplexer Channel Adapters, allowing the attachment of two channel interfaces. The two channel interfaces may be attached to the same host or to two different hosts. In either case, the channel attachments are treated independently; thus, one channel interface has no knowledge of the other channel interface. The various configurations are:

- Normal mode connection between an S/370 or S/390 processor and a system unit. The protocol is similar to CTC. The maximum distance between the S/370 host and the system unit is 400 feet (122 meters).
- Dual attachment configurations: a system unit with two Block Multiplexer Channel Adapters may be connected to a single host or to two different hosts. The maximum distance between the S/370 host and the system unit is 400 feet (122 meters).
- Direct channel attachment between S/370 or S/390 hosts and a system unit: the system unit connected to the channel is the machine used to communicate with the host. The user can be connected to the system unit by means of an ASCII terminal, an X-station on a local area network, or remotely logged in from a LAN. The maximum distance between the S/370 host and the system unit is 400 feet (122 meters).
- Direct channel attachment between S/370 or S/390 hosts and a system unit using the 3044 Channel Extender: the only difference with the previously described configuration is that the 3044-Fiber Optic Channel Extender units allow the channel distance to extend to a maximum of 3 km.
- Attachment to ESCON using the 9034 converter: if a 9034 is installed, it requires a dedicated ESCON channel. No other ESCON control unit may be attached; however, other parallel channel control units may be added through the same 9034.

Please refer to Section 2.7, “RISC System/6000 CLAW Connection” on page 89 for more information about installing this configuration.

Warning

1. To run in CLAW mode under VM, the virtual machine issuing the CLAW channel programs (usually TCPIP) must have *DIAG98* enabled.
2. To run a second level system that builds CLAW channel programs:
   - The second level system **must run V=R**

     or
   - The first level system **must be VM/ESA Release 2**. Only VM/ESA Release 2 has support for properly translating CLAW channel programs issued from V=V virtual machines. VM/ESA Release 1.1 does not have this support.

1.4 Product Evolution Summary Table

Table 2 below lists the new functions and enhancements included in TCP/IP V1 for VM and in Release 1 and 2 of TCP/IP V2 for VM.

Table 2 (Page 1 of 2). Supported Functions Summary Table

New Function or Enhancement	TCP/IP V1	TCP/IP V2R1	TCP/IP V2R2
USER APPLICATION PROTOCOLS			
TELNET	♦	♦	♦
FTP	♦	♦	♦
TFTP	♦	♦	♦
SMTP - MX record - Secure gateway - Asynchronous interface to name server - SMSG interface - Mail header customization	♦	♦ • • •	♦ Δ Δ Δ • •
NFS - RACF - CMS6 minidisks - User exits - SMSG interface	♦	♦ • • •	♦ Δ Δ Δ •
REXEC		♦	♦
LPR/LPD		♦	♦
Network Database System			♦
SYSTEM APPLICATION PROTOCOLS			
DNS - SMSG interface - UDP recursion - Intermediary caching - Zone transfer - DIG, NSLOOKUP - UDP port queue limit removal	♦	♦ • • • •	♦ Δ Δ Δ Δ • •
SNMP - 3172 subagent		♦	♦ •
RouteD		♦	♦

Table 2 (Page 2 of 2). Supported Functions Summary Table

New Function or Enhancement	TCP/IP V1	TCP/IP V2R1	TCP/IP V2R2
APPLICATION PROGRAMMING INTERFACES			
BSD Sockets - BSD4.3 full set - C language - IUCV domain (single host) - Simplified socket transfer procedure	♦	♦ • • •	♦ Δ Δ Δ •
X Window - Version 11.4 - GDDMXD - OSF/Motif toolkit - R1.1.2	♦	♦ • • •	♦ Δ Δ Δ •
RPC - RPCGEN	♦	♦ •	♦ Δ
NCS		♦	♦
Kerberos		♦	♦
SYSTEM USABILITY			
NLS - SBCS/DBCS		•	Δ
Softcopy documentation		•	Δ
VM/XA exploitation			•
VM/ESA support			•
CONNECTIVITY			
3172 FDDI LAN attachment			♦
3745 Ethernet and ACF/NCP V6 IP routing			♦

Note:

♦ denotes a new function included in the corresponding level of TCP/IP for VM.
♦ denotes a function already included in the previous level of TCP/IP for VM.
• denotes an enhancement included in the corresponding level of TCP/IP for VM.
Δ denotes an enhancement already included in the previous level of TCP/IP for VM.

1.5 IBM TCP/IP Version 2 Release 2 for VM Interoperability Summary

The following table summarizes the existing possibilities between different IBM platforms.

Please note that:

- SMTP applications use 7-bit ASCII code.
- The SNMP monitor function requires NetView VM.
- Kerberos includes the authentication and ticket-granting servers.
- For NDB, the client function is also available for Sun Microsystems workstations.
- TN3270, TN5250, VT100, VT2x0, ANSI and LINE are the different emulators available with TCP/IP. The line Telnet indicates the presence of the Telnet protocol.
- Other solutions to provide Telnet emulators are not part of this table. Such solutions may be to:
  - Have an OEM product installed on the host together with IBM TCP/IP Version 2 Release 2 for VM.
  - Have a 3174 Establishment Controller with RPQ 8Q0935 installed. Please refer to page 218 for more information.
  - Use the AIX-X Windows 3270 Emulator.

Table 3. Client/Server Relationships

Systems	VM V2R2		MVS V2R2		DOS V2.0		AS/400 V2		OS/2 V1.2.1		AIX 3.2	
Functions	Client	Server	Client	Server	Client	Server	Client	Server	Client	Server	Client	Server
Telnet	•	•	•	•	•		•	•	•	•	•	•
TN3270	•	•	•	•	•		•	•	•		•	
TN5250							•	•			•	
VT100					•		• [4]	• [4]	•		•	•
VT2x0					•				•		•	•
ANSI					•				•	•		
LINE	•	•	•	•	•		•	• [7]	•	•	•	•
FTP	•	•	•	•	•	•	•	•	•	•	•	•
TFTP	•				•	•			•	•	•	•
SMTP	•	•	•	•	•	• [3]	•	•	•	•	•	•
REXEC	•	•	•	[8]	•				•	•	•	•
RSH	•	•			•				•	•	•	•
Name Server	•	•	•	•	•		•		•		•	•
NDB		•		•							•	
X-Windows	• [5]		• [5]			• [6]	♦			•	•	•
RouteD	•		•		•				•		•	
NFS		•		•	•				•	•	•	•
SNMP	•	•	•	•	♦		♦		•	•	•	•
PING	•	•	•	•	•	•	•	•	•	•	•	•
NETSTAT	•		•				•		•		•	
LPR/LPD	•	•	•	•	•				•	•	•	•
Finger					•				•		•	•
Time/Daytime					•						•	•

Note:

• : Available now
♦ : Statement of direction

[3] With a POP2 Server running on a UNIX/AIX machine

[4] Available at the end of 92

[5] With GDDM V2R2 only

[6] Via the HummingBird Communications Ltd.** products

[7] Using an API

[8] Equivalent function is provided with FTP and the job submission facility

1.6 TCP/IP Implementation in VM

TCP/IP under VM uses many disconnected machines for the different TCP/IP components. In general, each TCP/IP application is implemented in at least one disconnected machine. The communication to the outside world is done via the TCPIP virtual machine, which is the TCP/IP main task. All servers (TCP/IP applications) communicate with the TCPIP virtual machine within the local VM system via the CP functions VMCF or IUCV.

Figure 19 shows only a part of all available servers. It is not necessary to configure and start them all. For example if your VM system will always act a a FTP client (that is a CMS user will always initiate a FTP session), there is no need to start the FTPSERVE virtual machine.

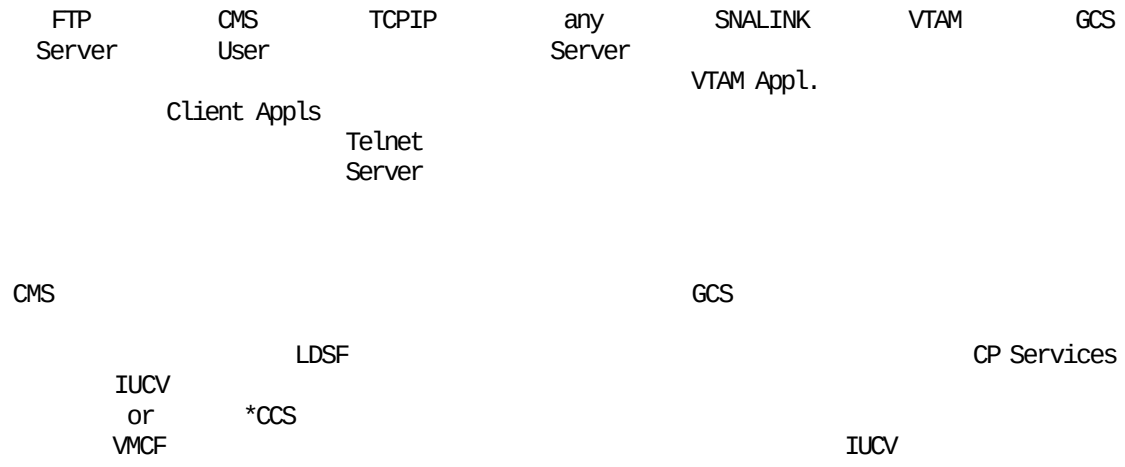


Figure 19. Structure of IBM TCP/IP Version 2 Release 2 for VM

TCP/IP for VM is implemented in the following way:

- ***TCPIP virtual machine***

  The TCPIP virtual machine handles all the hardware (physical interfaces) used for TCP/IP communication (except SNA and X.25) and also handles all TCP/IP requests from servers or clients attached to TCP/IP for VM. The Telnet server is implemented within the TCPIP virtual machine itself. It uses Console Communication Services (*CCS) to create line mode VM connections for remote users and Logical Device Support Facility (LDSF) to deal with CMS users. The TCPIP virtual machine uses the CP Virtual Machine Communication Facility (VMCF) function or the CP Inter User Communication Vehicle (IUCV) facility to manage its communication with all servers and users on its VM system.

- ***FTPSERVE virtual machine***

  The FTPSERVE virtual machine serves client requests coming in via VMCF from the TCPIP virtual machine. The TCPIP virtual machine itself gets its requests from anywhere in the network and delivers them to the appropriate server. The file server is just an example for many other servers, such as NFS, REXEC, etc.

- ***CMS user***

  Regular CMS users execute the client applications in their own virtual machine. These applications use VMCF to communicate via the TCPIP virtual machine to their final destination, a server anywhere in the network. The Telnet client uses the CP facilities *CCS for the line mode, or LDSF for the 3270 mode.

- ***SNALINK virtual machine***

  The SNALINK virtual machine allows TCP/IP to use an SNA backbone network. The TCPIP virtual machine sends IP datagrams to SNALINK in order to be routed through the SNA backbone network. SNALINK is a VTAM* application running under the control of GCS, also controlling VM VTAM.

  Another example of VTAM application running under the control of GSC is X25IPI that allows TCP/IP to use an X.25 network.

- ***VTAM virtual machine***

  VTAM handles the SNA backbone network, it does not communicate with TCP/IP directly, but it is required for the SNALINK and X25IPI virtual machines.

1.7 TCP/IP Requirements for VM

IBM TCP/IP Version 2 Release 2 for VM has specific requirements[9] for the CPU, network attachments, disk storage, virtual disks and program products.

1.7.1 Hardware Environment

IBM TCP/IP Version 2 Release 2 for VM is designed to operate on any IBM System/370* or System/390* processor that supports VM/SP, VM/SP HPO, VM/XA or VM/ESA and is equipped with:

1.7.1.1 CPU/DASD

- 7MB of user storage.
- 40MB direct access storage.
- One tape drive for installing IBM TCP/IP Version 2 Release 2 for VM.
- One or more terminals using IBM 3270 protocols and maintenance functions.
- One System/370 channel. If using one of the ES/9000* family of Enterprise System Architeture/390* (ESA/390*) processors or the ES/9370* with the Integrated Communications Processor, a System/370 channel is not required.

[9] Refer to *IBM TCP/IP V2 R2 for VM: Planning and Customization* for a complete list of prerequisites.

1.7.1.2 Network Attachments

- IBM ES/9000 or ES/9370 with the Integrated Communications Processors and one or more of the following:
  - X.25 adapter
  - Token-Ring Network adapter
  - IEEE 802.3 LAN adapter.

    Note: The NFS feature is not supported in this configuration.
- IBM 8232 LAN Channel Station.
- IBM 3172 Interconnect Controller Model 1 with one or more of the following:
  - 4/16 Mbps Token-Ring adapter
  - Ethernet/IEEE 802.3 adapter
  - PC Network (broadband/baseband) adapter.
- IBM 3172 Interconnect Controller Model 2 with:
  - 100 Mbps Fiber Distributed Data Interface (FDDI) adapter (feature #2250).
- IBM 37xx Communication Controller.
- IBM 3745 Communication Controller with:
  - Ethernet LAN adapter (features #4780 and #4781).
- HYPERchannel A220 Processor Adapter (P/N 42990007).
- Channel-to-Channel

  The following connections are supported by PVM V1R4 (5748-RC1).
  - IBM 3044
  - IBM 3088
- IBM RISC S/6000 Block Multiplexer Channel Adapter

1.7.2 Software Environment

IBM TCP/IP Version 2 Release 2 for VM is designed to operate with the software components listed below.

1.7.2.1 Operating System

IBM TCP/IP Version 2 Release 2 for VM is supported by:

- VM/SP Release 5 and 6 with HPO (5664-173) or without HPO (5664-167)
- VM/XA SP Release 2 or 2.1 (5664-308)
- VM/ESA Version 1 Release 1 (370 and ESA feature) or Release 1.1 (5684-112).

1.7.2.2 Additional Components

These components are optional, but required for many important functions, especially the IBM C/370 Runtime Library which is a prerequisite for many clients and servers written by IBM in C/370.

- To use any program developed in C, that interfaces directly with IP, UDP or TCP, one of the following is required:
  - IBM C/370 Library V1R2 (5688-039) or V2 (5688-188)
  - IBM C/370 Compiler V1R2 (5688-040) or V2 (5688-187).
- To run a primary or secondary Name Server, or the Network Database System (NDB) feature, the following is required:
  - SQL/DS Version 2 Release 2 (5688-004) with PUT tape 9105 or later.
- The X Window System GDDM interface support requires one of the following:
  - GDDM/VM V2 (5664-200)
  - GDDM/VMXA Version 2 Release 2 (5684-007).
- When using the VM/XA exploitation mode for the TCPIP virtual machine, or any program written in Pascal that interfaces directly to IP, UDP or TCP, one of the following is required:
  - IBM VS Pascal Version 1 Release 2 Library (5668-717)
  - IBM VS Pascal Version 1 Release 2 Compiler and Library (5668-767).
- For SNAlink LU0 interface or X.25 interface support, the following programs are required with IBM 37XX Communication Controller:
  - X.25 NPSI Version 3 Release 4 (5688-035), or later
  - ACF/NCP Version 5 Release 3 (5668-738), or later
  - ACF/VTAM Version 3 Release 2 (5664-280), or later.
- For 3745 Ethernet LAN adapter support, the following is required:
  - ACF/NCP V6 (5688-231).
- To use the SNMP monitor support and work with NetView, the following is required:
  - For VM/ESA, NetView V2R2 for VM/ESA (5756-051)
  - For other VM Releases, NetView V1R3 (5664-204), or higher.
- To link two TCP/IP virtual machines using VM Passthrough:
  - The PVM virtual machine on the TCP/IP host must use PVM Version 1 Release 4 (5748-RC1).
- To use the IBM 3172 Interconnect Controller FDDI or SNMP support, the following is required:
  - IBM 3172 Interconnect Controller Program Version 2 Release 1 (5601-433).
- For file access protection with NFS or FTP, the following is required:
  - RACF Release 1.8 (5740-XXH), or later.

Please refer to the *IBM TCP/IP V2 R2 for VM: Planning and Customization* manual and to the *Program Directory for use with IBM TCP/IP Version 2 Release*

2 for VM for the most current information about the hardware and software requirements.

Note: If you do not plan to use an SNA (with SNALINK) or an X25 (with X25IPI) network, VTAM is not required to use TCP/IP on a VM system. The Telnet server do not use (unlike his MVS counterpart) VTAM: all screens logging on into a VM system will be under control of CP.

Any screen using a 3270 flow will be able to issue any VM command when prompted with the VM logo. Such commands as **LOGON, DIAL VTAM, DIAL PVM ...** can be used, thus giving access to hosts that do not have TCP/IP installed.

Chapter 2. Installation and Tailoring

The installation of IBM TCP/IP Version 2 Release 2 for VM consists of three steps:

1. ***Preinstallation:*** Before you begin the actual installation, you must understand your network configuration. IBM TCP/IP Version 2 Release 2 for VM offers many services; you may not require all of the services, but you will have to identify (a service) in order to understand the hardware and software requirements.
2. ***Installation:*** This basically consists of installing the base tape and running the verification procedure. You may also install the optional products and run the verification procedure again.
3. ***Configuration:*** The configuration is the most complex part. You must modify the various configuration files according to your specific requirements.

2.1 Installation Process Enhancements over V2R1

To allow more flexibility, the entire installation process of IBM TCP/IP Version 2 Release 2 for VM has been improved over the previous release of the product in the areas of:

1. Its structure
2. Its implementation
3. Its documentation.

2.1.1 Structure

The previous releases required the product to be installed from the TCPMAINT user ID and required the servers′ user IDs and minidisks to be defined in a ″hard-coded″ way in the VM directory prior to the installation. The ″I5735FAL EXEC″ procedure used to force the TCP/IP-related user IDs off the system, to gain R/W access to their minidisks, on which all the executable components and configuration files were then installed.

With V2R2, you are no longer required to install the product from the TCPMAINT user ID, and the ″I5835FAL EXEC″ no longer tries to force the related user IDs off the system. The targets of the installation process are now controlled by a *configuration file* called ″5735FAL CONFIG″, which allows:

- The coexistence of multiple releases
- The selection of names which are consistent with your site naming conventions
- The installation of only a subset of the servers.

The automatic invocation of the CMS macro editor (XEDIT) to enter the servers′ passwords into the ″PROFILE TCPIP″ file and the automatic invocation of the ″V5735FAL EXEC″ verification procedure have been suppressed.

A new minidisk called the *server common* disk has been introduced, to consolidate all possible executable modules and configuration files for the servers. This eliminates the need of having to wander through the whole collection of the servers′ minidisks during the configuration process.

2.1.2 Implementation

The installation of IBM TCP/IP Version 2 Release 2 for VM relies upon the ″I5735FAL EXEC″ procedure. Following is a list of its major improvements over the previous release:

- It no longer tries to force the TCP/IP-related user IDs off the system.
- It validates the availability of all the target disks before proceeding (if all conditions are not met, it writes the relevant error messages to the ″I5735FAL ERRLOG″ error log file, provided the installation work disk is accessed in R/W mode, and terminates).
- It uses the target disk address in the *LINK* statement if the target user ID is the same as the installing user ID.
- It starts spooling the console of the user ID it is running in (*SPOOL CONSOLE START TO **), so that no meaningful messages are missed (the console will not be closed if it has been detected as started before the installation procedure actually tries to start it).
- It asks the user to select between installation configuration files with filetype CONFIG and SCONFIG, if both types are found in search order (SCONFIG being the filetype of the sample installation configuration file copied from the tape during the preinstallation process).
- It requests permission to attempt formatting a disk if the *ACCESS* command towards this disk fails with *RC=100* (this will be done for all the minidisks except the owner's 191).
- It issues numbered messages which are documented in Chapter 7 of the *Program Directory* (see Section 2.1.3, “Documentation” for more information).
- It will attempt to move the appropriate files after the installation completes, if the installation is not performed from the owner user ID.
- It will erase the product identifier file only after a successful installation.
- It includes control of the tape files it is positioned on to make sure that they are consistent with the ongoing installation step.

2.1.3 Documentation

Most of the documentation about the installation of IBM TCP/IP Version 2 Release 2 for VM has been moved from the previous *Installation and Maintenance* manual to the *Program Directory*. Therefore, and since the *Installation and Maintenance* manual did not contain much information about the product maintenance, this manual has been renamed *Planning and Customization*.

The major improvements in the documentation of the installation of IBM TCP/IP Version 2 Release 2 for VM are listed below:

- the Sample Source Programs tape contains sample directory definitions which are intended to assist you in defining the TCP/IP-related virtual machines, without having to manually enter this information from the printed samples provided in Chapter 3 of the *Planning and Customization* manual.
- Chapter 6 of the *Program Directory* now contains all the information about the installation of IBM TCP/IP Version 2 Release 2 for VM.
- Chapter 7 of the *Program Directory* now contains the explanation of the messages and return codes issued by the installation process, together with

the associated System Action and User Response. Three categories of messages are issued:

Prompts
Informational
Error.

- Appendix A of the *Program Directory* contains a copy of the sample ″5735FAL SCONFIG″ installation configuration file, which provides:
  - Complete reference of the default user IDs and minidisk addresses
  - For each user ID and minidisk, information about its purpose, whether it is optional, and under which conditions it might be required.

2.2 Preinstallation

Chapter 6 of the *Program Directory for use with IBM TCP/IP Version 2 Release 2 for VM* describes the preinstallation tasks in detail. Please refer to this document which is delivered with the product tapes, and which contains the most current information about how to install IBM TCP/IP Version 2 Release 2 for VM.

This bulletin is not intended to provide a copy of the *Program Directory*. Instead, to help you "navigate" in the preinstallation process, following is a picture which summarizes the content of Chapter 6.1 of the *Program Directory*. To make reference to the actual *Program Directory* easy, the picture refers directly to the Chapters which Chapter 6.1 consists of.

```
STEP 1 : Decide on an Installation Methodology (Chapter 6.1.1 of the †Program Directory†)

The main questions to answer in this step are:

   Does your installation require to run concurrent releases?

         operating at the same time?

         having dedicated hardware attached to the TCPIP server?


   Does your installation require the use of all the functions shipped with V2R2?

   ==  if not : read Chapter 6.2.6 of the †Program Directory†
                 entitled †Performing a partial installation†


   Do the default user ID names supplied with the product conform
   to your site naming conventions?

   ==  if not : read Chapter 6.2.3 of the †Program Directory†
                 entitled †Implications of assigning different user ID names†


   Are you a current user of TCP/IP for VM?

   ==  if not : read Chapter 6.1.1.1 of the †Program Directory†

   ==  if yes : read Chapter 6.1.1.2 of the †Program Directory†



STEP 2 : Obtain the sample definition files     (Chapter 6.1.2 of the †Program Directory†)

File 3 of the †VM Samples† tape contains:

   A set of sample VM directory definitions for the TCP/IP-related virtual machines.
   All the files have the filetype †DIRMODEL†.

   A sample †5735FAL SCONFIG† installation configuration file.

Load these files onto the A-disk of the installing user ID using VMFPLC2.
```

Figure 20. Preinstallation Tasks Steps 1 and 2

```
STEP 3 : Define/Update the virtual machines definitions (Chapter 6.1.3 of the P.D.)


The †Program Directory† lists the differences between V2R1 and V2R2.

Note:    TCPMAINT has 3 new minidisks: 2C1, 591 and 5C4.

         the new NDB feature requires the definition of the NDBSERVE user ID to run
         the SQL/DS RPC server, and, being an RPC application,
         it requires the PORTMAP server.

         outstanding R/W links to the servers¢ disks at install time
         will cause the installation to fail.

         use of existing directory definitions requires the addition of a link
         to the new †server common† disk (TCPMAINT 591).

         if minidisk passwords are used, the minidisks which are targets of the
         installation process (those owned by the servers and the owner user ID)
         need to have READ, WRITE and MULTIPLE passwords.

         some servers¢ virtual machine minimum storage sizes have been increased:

                NAMESRV  : 6MB
                REXECD   : 3MB
                FTPSERVE : 4MB

After deciding which virtual machines you wish to install, use the sample directory
definitions to update your VM system directory, or use the above information to update
your existing definitions.
Refer also to Chapter 3 of the †Planning and Customization† manual.



STEP 4 : Customize the installation configuration file (Chapter 6.1.4 of the P.D.)


==  If the following conditions are met:

       the default names and minidisk virtual address assignments are satisfactory.

       you do not wish to exclude any subset of the server virtual machines from
       the installation process.

       the default minidisk password values of *PROMPT* (which imply that the installer
       should be prompted for all the access passwords of the minidisks which are targets
       of the installation process) are acceptable.

    The sample †5735FAL SCONFIG† unloaded during step 3 can be used without modification,
    and you may proceed with the installation tasks (see next paragraph).

==  If not, use Chapters 6.1.4.1 and 6.1.4.2 of the †Program Directory† to
    update the sample installation configuration file according to your requirements.
```

Figure 21. Preinstallation Tasks Steps 3 and 4

2.3 Installation

Chapter 6 of the *Program Directory for use with IBM TCP/IP Version 2 Release 2 for VM* describes the installation tasks in detail. Please refer to this document which is delivered with the product tapes, and which contains the most current information about how to install IBM TCP/IP Version 2 Release 2 for VM.

This bulletin is not intended to provide a copy of the *Program Directory*. Instead, to help you "navigate" in the installation process, following is a picture which summarizes the content of Chapter 6.2 of the *Program Directory*. To make reference to the actual *Program Directory* easy, the picture refers directly to the Chapters which Chapter 6.2 consists of.

```
STEP 1 : Base feature tape installation


The base feature should be installed first.
This step is described in different Chapters of the †Program Directory†,
depending on the media the product is delivered on:

   Feature codes 5871 and 5782  (6250 BPI reel or 3480 cartridge): Chapter 6.2.1

   Feature codes 5870 and 5784  (1600 BPI reel or 1/4† cartridge): Chapter 6.2.2


Perform the following steps:

        Ensure that all the minidisks that are targets of the installation process
        are available for linking in the †M† access mode.
        This includes: the servers¢s disks
                       the client common disk (owner¢s 592)
                       the server common disk (owner¢s 591)

        Check that there are no copies of the Product Identifier files (I5735FAL 022* *)
        stored on any of the disks currently accessed by the installer¢s user ID.

        Ensure that the installer¢s user ID does not already use filemode Z,
        which will be used by the installation EXEC to access the target minidisks.

        If you modified the installation configuration file, ensure that the customized
        version that you created during the preinstallation step 4 is available.

        Unload the first 2 tape files of the Base Feature tape (labeled †VM BASIC†)
        onto the installation workdisk, using VMFPLC2.

        Read Chapter 6.2.4 of the †Program Directory†
        entitled †Installation EXEC implementation characteristics†.

        Start the installation process, using I5735FAL.

        Successful loading of all tape files to their proper minidisks will result
        in the installation EXEC terminating with return code 888.
        Any other return code implies that an error condition prevented the installation
        from completing successfully. Refer to Chapter 7 of the †Program Directory†
        for an explanation of the return codes and associated messages.
```

Figure 22. Installation Tasks Step 1

```
STEP 2 : Other feature tape installation       (Chapter 6.2.5 of the †Program Directory†)

The other feature tapes are labeled:    VM TEXT     (text decks)
                                        VM SOURCE   (source files)
                                        VM SAMPLES  (sample programs source)
                                        VM NFS      (Network File System)
                                        VM KERB     (Kerberos)
                                        VM SOFTCOPY (Softcopy documentation)

The installation steps for these features are similar to those for the base tape.
```

Figure 23. Installation Tasks Step 2

Note: A successful installation is indicated by receiving the 888 return code.

Be sure you have no target disk accessed in write mode.

If you performed partial installations, be aware that each successful partial installation will erase the Product Identifier file (I5735FAL 022021B). For the next partial installation you will have to load it again using the VMFPLC2 command.

2.4 Configuration

2.4.1 Configuration Files

The TCP/IP software is very flexible. At startup time, the configuration parameters for TCP/IP and all its services (server processes) are read from configuration files. All of them are supplied by IBM as samples, but some of them must be modified by you in order to reflect your environment.

The configuration of IBM TCP/IP Version 2 Release 2 for VM is done by modifying these configuration files. There are a few required files and many optional files depending on your configuration. The following table gives you an overview of the configuration files which need your attention.

Table 4. Configuration Files Used by IBM TCP/IP Version 2 Release 2 for VM

File Name	Origin	Used by	Purpose
PROFILE TCPIP	TCPIP 191	TCPIP	main configuration file
TCPRUN EXEC	TCPMAINT 591	Servers	controls the startup of the servers
exit EXEC	TCPMAINT 591	TCPIP and all Servers	software and hardware setups
PROFILE EXEC	TCPIP or Server 191	TCPIP and all Servers	do not modify this file
TCPIP DATA	TCPMAINT 592	Clients and Servers	Clients and Servers configuration file
HOSTS LOCAL	TCPMAINT 592	all Clients and Servers	flat table for name to IP address translation
SMTP CONFIG	TCPMAINT 591	SMTP	SMTP configuration file
PW SRC	SNMPD 191	SNMPD	community names of SNMP agent
SNMPTRAP DEST	SNMPD 191	SNMPD	list of managers receiving the traps sent by the SNMP agent
ETC GATEWAYS	ROUTED 191	ROUTED	external gateways description
X25IPI CONFIG	TCPMAINT 591	X25IPI	X.25 configuration file
NSMAIN DATA	NAMESRV 191	NAMESRV	name server configuration file
MASTER DATA	NAMESRV 191	NAMESRV	name server resource records

Please Read This !

Depending on your environment, you will have to create or modify some of these configuration files. Please read the *IBM TCP/IP V2 R2 for VM: Planning and Customization* manual to build these important files.

2.4.2 Server Startup

2.4.2.1 Overview

IBM TCP/IP Version 2 Release 2 for VM introduces a new architected way of handling the startup of a server (including TCPIP itself), based on three EXEC procedures:

1. The server's standard "PROFILE EXEC"
2. The server's "exit EXEC"
3. The common server "TCPRUN EXEC".

Important

All of your customization must be done in the server's "exit EXEC".

The main objectives of this architecture are:

- To provide a standardized way of accomodating the site environment
- To improve the environment checking prior to invoking the server's module
- To provide a standard startup sequence across all servers.

Let us now have a more detailed look at the three files mentioned above.

2.4.2.2 The "PROFILE EXEC" Procedure

The "PROFILE EXEC" procedure is invoked through the CMS IPL process and is used to:

- Spool the console to the default owner
- Provide *LINK* and *ACCESS* to the common server disk (owner's 591) and the common client disk (owner's 592)
- Call the server's "exit EXEC" targeting the PRELUDE subroutine (if the "exit EXEC" is available) and check the results
- Invoke the "TCPRUN EXEC" through the program stack with the appropriate arguments.

2.4.2.3 The "exit EXEC" Facility

This is where you must include all the elements which are necessary to adapt the virtual machine environment for a particular server to your configuration requirements.

The "exit EXEC" consists of three subroutines respectively named PRELUDE, POSTLUDE and ABORT. It is called using the following format:

Profile Exec User Exit

```
exit-exec  PRELUDE
           POSTLUDE
           ABORT rc command
```

exit-exec is the name of the "exit EXEC" procedure associated with a specific virtual machine. One sample "exit EXEC" is copied to the owner's 591 minidisk

during the installation process, for each of the servers, using the following names:

TCPIPXIT EXEC for the TCPIP server
LPDEXIT EXEC for the LPD server
FTPDEXIT EXEC for the FTP server
SNMPDXIT EXEC for the SNMP agent
SNMPQXIT EXEC for the SNMP query engine
NAMESXIT EXEC for the Name server
ROUTEXIT EXEC for the ROUTED server
SMTPEXIT EXEC for the SMTP server
REXECXIT EXEC for the REXECD server and slaves
PORTMXIT EXEC for the Portmap server
NDBSEXIT EXEC for the Network Database System server
VMNFSXIT EXEC for the Network File System server
KERBEXIT EXEC for the Kerberos authentication server
KADMEXIT EXEC for the Kerberos administrator
NCSGLXIT EXEC for the NCS Global Location Broker
NCSLLXIT EXEC for the NCS Local Location Broker

The arguments of the *exit-exec* call are:

Argument	Description
PRELUDE	Used to invoke the PRELUDE exit subroutine which allows you to change the virtual machine environment before the ″PROFILE EXEC″ begins the environment tailoring for the given server. You may use it to: • *LINK* and *ACCESS* local or maintenance minidisks • *LINK* and *ACCESS* runtime libraries • *ATTACH* the communications devices • Modify the OWNER, COMMAND and PARMS global variables • Notify the operations or support people • Interactively prompt the user.
POSTLUDE	Used to invoke the POSTLUDE exit subroutine which allows you to check and, if necessary, change the virtual machine environment before ″TCPRUN EXEC″ invokes the server′s module. You may use it to: • Inspect and modify the virtual machine environment • Display the final virtual machine environment • Notify the operations or support people • Modify the COMMAND and PARMS global variables It is better to do that here instead of doing it in the PRELUDE exit subroutine, because ″TCPRUN EXEC″ sets the global variables. • Interactively prompt the user.
ABORT	Used to invoke the ABORT exit subroutine which notifies the ″exit EXEC″ that an unrecoverable error has been encountered and that the server startup cannot continue.

You may use it to:

- Notify the operations or support people
- *LOGOFF* the server

  Since TCPIP will ″re-logon″ the server, this will cause a loop, so it would be better to find what is wrong and fix it.

rc is the CMS return code from the failing command and *command* is the failing command.

2.4.2.4 The ″TCPRUN EXEC″ Procedure

The ″TCPRUN EXEC″ procedure is invoked by the ″PROFILE EXEC″ and is used to:

- Spool the console to the specified owner
- Set up the common execution environment (*SET EMSG, ECMODE, SET LDRTBLS, GLOBAL LOADLIB, DEF STOR,* etc.)
- Set up server-specific environment (which server is going to be run)
- Call the server's ″exit EXEC″ targeting the POSTLUDE subroutine (if the ″exit EXEC″ is available) and check the results
- Invoke the server's module using the arguments set by the ″PRELUDE″ or the ″POSTLUDE″ subroutine of the ″exit EXEC″.

Figure 24 on page 74 illustrates the above description and shows the typical sequence of a server startup in IBM TCP/IP Version 2 Release 2 for VM.

2.4.2.5 Typical Sequence of a Server Startup

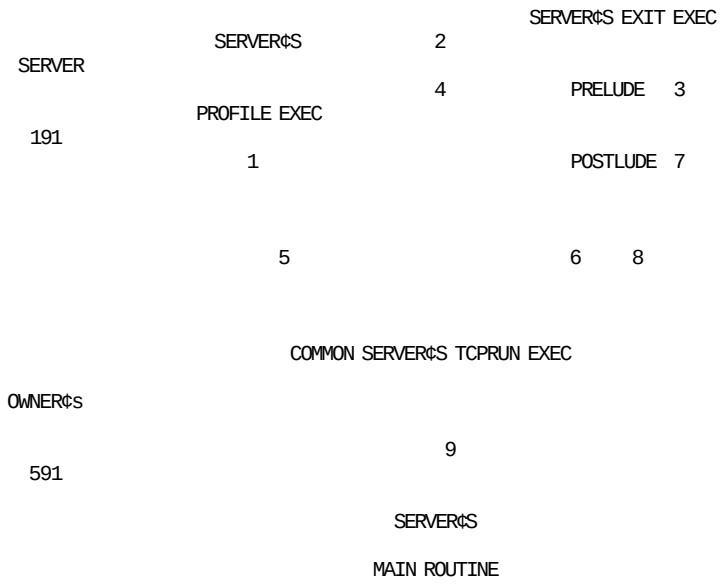


Figure 24. The Sequence of a Server Startup

1 - When the server is logged on, the ″PROFILE EXEC″ read from the server′s 191 minidisk is automatically started.

2 - The ″PROFILE EXEC″ calls a new procedure called the server′s *exit EXEC* (read from the common server disk), targeting its PRELUDE subroutine.

3 - The PRELUDE subroutine is responsible for the *LINK* and *ACCESS* to the necessary disks (C/370, SQL/DS, etc.). You should set your PF keys and attach the necessary communications hardware, here as well.

4 - After executing the PRELUDE subroutine, the *exit EXEC* returns control to the server′s ″PROFILE EXEC″.

5 - The ″PROFILE EXEC″ calls the ″TCPRUN EXEC″ procedure which is the common standard procedure used to start up all the servers and which is read from the common server disk.

6 - The ″TCPRUN EXEC″ procedure calls the server′s *exit EXEC* once again, this time targeting its POSTLUDE subroutine.

7 - The POSTLUDE subroutine is responsible for setting the options of the server startup (that is, the parameters of the command used to load the server′s main routine).

8 - The POSTLUDE subroutine returns control to the ″TCPRUN EXEC″ giving the selected options to invoke the server′s module.

9 - Now the ″TCPRUN EXEC″ has the necessary environment set up for the server and the server′s main routine is started.

2.4.3 Configuring the TCPIP Server

2.4.3.1 File ″PROFILE EXEC″ on TCPIP 191

Important

This EXEC should not be modified because all your system′s setups should be done using the ″TCPIPXIT EXEC″ file.

2.4.3.2 File ″TCPRUN EXEC″ on Owner′s 591

Important

This EXEC should not be modified because all your system′s setups should be done using the ″TCPIPXIT EXEC″ file.

However the ″TCPRUN EXEC″ shipped with the tapes does not log off the virtual machines when they exit their main routine with no error return code. That is the case when the netstat cp ext command is issued from TCPMAINT is order to stop the TCP/IP environment. All the machines (TCPIP, SNALNKA, NAMESRV, etc) will stop but will remain logged on. That may not be a problem for the ″server machines″ (FTPSERVE, SNALNKA, NAMESRV, etc.) because the TCPIP virtual machine will force/autolog them during its own startup, but it is annoying for the TCPIP virtual machine itself. Following is a section of the ″TCPRUN EXEC″ which has been modified to issue the LOGOFF command when a server exits its main routine and when the virtual machine was running disconnected. If the server machine was stopped by an operator who was logged on to the server virtual machine it will remain logged on. Numbers on the left are line numbers.

```
00241   say ¢Command †¢tcpstart¢† ended¢, /* Issue stop message. */
00242       ¢with return code¢ rc¢.¢
00243 /* Start of modifications */
00244       Select
00245          when linesize()  =0 then  /* TCPIP not DSC */
00246          say ¢Terminating server startup¢ ,
00247          ¢at your request with Rc:¢ rc ¢!!¢
00248           otherwise         /* TCPIP DSC */
00249           †CP LOGOFF†
00250        End /* end of Select */
00251 /* End of modifications */
```

2.4.3.3 File ″TCPIPXIT EXEC″ on Owner′s 591

IBM TCP/IP Version 2 Release 2 for VM is usually started as a disconnected virtual machine. The ″TCPIPXIT EXEC″ is the ″PROFILE EXEC″ user exit procedure for the TCPIP virtual machine (see also Section 2.4.2.3, “The ″exit EXEC″ Facility” on page 71). It defines the correct environment for the TCP/IP program before it is started. The physical devices for the network links are bound to the TCPIP virtual machine with this procedure.

The following is an example of what can be coded in this file (numbers on the left are line numbers):

```
00117 ¢Q V STOR¢              1
00118 ¢Q LOADLIB¢
00119 ¢Q TXTLIB¢
00120 ¢VARY ON 104-107¢       2
00121 ¢ATTACH 104 * 104¢
00122 ¢ATTACH 105 * 105¢
```

```
00123 ¢ATTACH 106 * 106¢
00124 ¢ATTACH 107 * 107¢
```

1 Here we just ask for the virtual storage, the loadlibs and the txtlibs in use.

2 The Token-Ring interface is varied online and attached to the TCPIP virtual machine.

2.4.3.4 File ″PROFILE TCPIP″ on Owner′s 591

During initialization of the TCPIP virtual machine, system operation and configuration parameters are read from this configuration file. TCP/IP tries to read a file ″node_id TCPIP″. If not there, a file called ″PROFILE TCPIP″ must exist; otherwise, TCPIP will terminate. Physical characteristics, local internet addresses, TCP/IP services and many other parameters are referenced here. This file is referred to as ″PROFILE TCPIP″ in this book. You can also change certain system parameters while TCP/IP is running with the OBEYFILE command. The hardware addresses used in the device statements must match the addresses used in the *VARY ONLINE* and *ATTACH* commands of the ″TCPIPXIT EXEC″ PRELUDE subroutine. A sample file that you can modify is provided with the distribution tape.

The following is an undocumented facility that allows the network administrator to restrict the use of TCP/IP. All users listed between the RESTRICT statements will not be allowed to use TCP/IP even if they have a link to the right minidisks.

Let′s suppose you have the following entry in your ″PROFILE TCPIP″:

```
 ......
OBEY
    OPERATOR TCPMAINT SNMPD SNMPQE ROUTED REXECD
ENDOBEY
RESTRICT
    CNMESA
ENDRESTRICT
 ......
```

The CMS user ID CNMESA will not be allowed to use TCP/IP, and will receive an error message when he issues any TCP/IP command, as shown in the following scenario.

```
id
CNMESA   AT RALYESA  VIA RJE      06/25/92 20:12:08 EDT      THURSDAY
Ready; T=0.01/0.01 20:12:08
q disk
LABEL  VDEV M  STAT  CYL TYPE BLKSIZE  FILES  BLKS USED-(%) BLKS LEFT
CNM191 191  A   R/W   10 3380 4096        15         50-03       1450
TMN592 592  B   R/O   75 3380 4096       554       7106-63       4144
TMT591 591  D   R/O   40 3380 4096       109       3423-57       2577
S-DISK 190  S   R/O   70 3380 4096       290       6833-65       3667
Y-DISK 19E  Y/S R/O  200 3380 4096      1543      19070-64      10930
Ready; T=0.01/0.01 20:12:11
ping fred
BeginTcpIp failed: You are not authorized to issue this command
Ready(00008); T=0.03/0.05 20:12:16
```

To dynamically alter the RESTRICT list, you have to create a file "OBEY TCPIP" (for example) which will contain the entire new RESTRICT list and issue the OBEYFILE command:

```
type OBEY TCPIP A

RESTRICT          1
 BILL
 CNMESA
ENDRESTRICT

Ready;
obeyfile obey tcpip a      2
VM TCP/IP Obeyfile
Requesting TCPIP to accept ¢OBEY TCPIP *¢ on TCPMAINT 191  ...
File ¢OBEY TCPIP *¢ has been read and obeyed
Ready;
type OBEY TCPIP A

RESTRICT             3
ENDRESTRICT

Ready;
obeyfile obey tcpip a        4
VM TCP/IP Obeyfile
Requesting TCPIP to accept ¢OBEY TCPIP *¢ on TCPMAINT 191  ...
File ¢OBEY TCPIP *¢ has been read and obeyed
Ready;
```

1 User IDs BILL and CNMESA are in the RESTRICT list.

2 Ask TCPIP to read the file "OBEY TCPIP" on TCPMAINT.191. TCPIP must be authorized to link the TCPMAINT.191. Once the "OBEY TCPIP" file has been read and obeyed, BILL and CNMESA cannot use TCP/IP any more.

3 This empty RESTRICT list will allow all the CMS users to use TCP/IP.

4 Once the file "OBEY TCPIP" has been read and obeyed all users can use TCP/IP including BILL and CNMESA.

Warning

Generic user IDs are not supported; that is, restrict statements with user IDs such as CNM* are invalid.

2.4.3.5 File "TCPIP DATA" on Owner's 592

All TCP/IP clients and servers read this file in order to learn their environment in the local TCP/IP system. You must change this file to reflect the local host name, the domain name and, if used, the address of your name server. The other statements in this file need not be altered if you use the installation defaults. A sample file that you can modify is provided with the distribution tape.

You may have a copy of this file on your 191 minidisk. All the client programs that you will run under your CMS user ID will then use your "TCPIP DATA" file. This may be very useful for debugging purposes such as checking which name server is used, or to temporarily use another name server, or to trace a process.

2.4.3.6 File ″HOSTS LOCAL″ on Owner′s 592

It is not easy to remember the internet addresses for many TCP/IP hosts, so it is much easier to use names instead. This file contains the names and the internet addresses for the hosts you use most often. The file is only locally known. If you have a lot of TCP/IP hosts in your environment this may not be a convenient solution, because you would have to keep an up-to-date ″HOSTS LOCAL″ file on each host. One central name server would be a much better solution.

The following example shows some host entries with multiple internet addresses and multiple symbolic names.

File ″HOSTS LOCAL″

```
HOST : 9.67.38.66  : VM14,RAL9390   ::::
HOST : 9.67.38.67  : VM15,RAL9360   ::::
HOST : 9.67.38.65  : VMESA    ::::
```

The ″HOSTS LOCAL″ file is the source for the ***MAKESITE*** program which produces two new files. These two files are actually used by the TCP/IP functions and should be placed on the TCPMAINT 592 minidisk.

Like the ″TCPIP DATA″ file these two files may be located on your 191 minidisk. All the client programs will use these files instead of those located on TCPMAINT.592. This can be useful if you want to give a host a different name than the one written in the common hosts file.

2.5 Network Routing - ROUTED

The most complex part of configuring a TCP/IP network is to establish the routing tables.

If you are not familiar with IP routing architecture please refer to the *TCP/IP Tutorial and Technical Overview - GG24-3376-02*.

Establishing a routing table can be done in two different ways on a VM system:

- ***Static routing:*** this means that you will define the routes yourself. This is part of the configuration steps when you customize TCP/IP. This implies that you know the address of every network you want to communicate with and how to get there. That is, you must know the address of the first router on the way. In most configurations static routing is sufficient to provide efficient routing. If your configuration is such that you do not have any backup route to remote networks, there is no need for dynamic routing.
- ***Dynamic routing:*** this means that a protocol will build the routing tables for you. Today, TCP/IP for VM only supports the Routing Information Protocol (RIP) which runs into the ROUTED virtual machine. Using such a protocol you do not need to know anything about the networks or the routers. Be aware that this will increase the traffic on the network since transfer of routing tables will occur periodically (the whole routing table is transmitted every 30 seconds).

Network routing is more transparent if you use dynamic routing because topology changes are transmitted over the network. However, the network will need some time to converge. In the meantime parts of the network may be

unreachable. This is due to the propagation of route updates together with the time needed, in every router on the way, to compute a new routing table.

2.5.1 Routing Tables

A routing table is the place where routing information is kept. It can be considered as a collection of pairs: *Destination network - First router on the way*.

TCP/IP maintains one internal routing table. Its content can be displayed using the NETSTAT GATE command.

A very basic output may be:

```
netstat gate
VM TCP/IP Netstat V2R2
Known gateways:

NetAddress   FirstHop   Link     Pkt Sz  Subnet Mask     Subnet Value
----------   --------   ------   ------  -----------     ------------
9.0.0.0       direct    ILALTRN  2000    0.255.255.192   0.67.38.64
Ready; T=0.12/0.31 13:27:53
```

Where:

- 9.0.0.0 is the local directly attached network.
- direct means that the network is directly connected (not accessed via a router). If a router was used its IP address would be displayed here.
- ILALTRN is the link used to access the network.
- 2000 is the packet size used on that network.
- 0.255.255.192 is the subnet mask used.
- 0.67.38.64 is the subnet value for this network.

If you are not using ROUTED you must define all routing information in the ″PROFILE TCPIP″ file following the GATEWAYS statement and the BSDROUTINGPARMS statements must be commented out. The routing information becomes static and can only be changed using the OBEYFILE command.

If you use ROUTED (dynamic routing), you only need to define the physical interfaces to the directly connected networks in the ″PROFILE TCPIP″ file following the BSDROUTINGPARMS statement. If you have gateways that do not exchange RIP datagrams, the ROUTED virtual machine must be informed that these gateways exist. This is done in the file ″ETC GATEWAYS″.

Warnings

- Unlike many other TCP/IP implementations (UNIX/AIX, OS/2, DOS), all the directly attached networks MUST be coded in the ″PROFILE TCPIP″.
- You code them either with the GATEWAY statement or the BSDROUTINGPARMS statement.
- The ROUTED virtual machine needs access to the C/370 minidisk. This is done using the ″ROUTEXIT EXEC″ file.

2.5.2 Routing with the GATEWAY Statement

If you do not use the ROUTED server, the BSDROUTINGPARMS statements must be commented out. To set up the routing table, it is important to understand the TCP/IP network architecture. The network classes and subnetting must be fully understood.

Please refer to *TCP/IP Tutorial and Technical Overview - GG24-3376-02* for explanations about IP routing and IP addressing schemes. Below are some sample definitions.

The GATEWAY statement in "PROFILE TCPIP"

```
GATEWAY
; Network  First_Hop  Link     P_Size Subn_Mask      SUBN_VALUE
 192.1.2    =         ETH1     1000   0                        1
 9          =         ILALTRN  1000   0.255.255.192 0.67.38.64  2
 9        9.67.38.65  ILALTRN  1000   0.255.255.192 0.67.32.64  3
 128.1      =         SNALMV20 2000   0.0.255.0     0.0.1.0     4
 9.67.38.33 =         SNALVM14 2000   HOST                      5
```

1 A class C network (Network → 192.1.2) can be reached directly (First_Hop → =) via the link named ETH1. There is no subnetting used (Subn_Mask → 0) and the maximum packet size is set to 1000 bytes (P_Size → 1000).

2 A class A network (Network → 9) can be reached directly via the link named ILALTRN. Subnetting is used (Subn_Mask → 0.255.255.192).

3 A class A network can be reached via a gateway (First_Hop → 9.67.38.65), subnetting is also used. For a given network all equal subnet masks must be used if you have multiple routing entries.

4 A class B network (Network → 128.1) can be reached directly via a point to point SNALINK link named SNALMV20. Subnetting is used, eight bits (Subn_Mask → 0.0.255.0) for subnetting are added to the 16 bits of the class B network.

5 A single host can be reached via a point to point SNALINK link. Subnetting is meaningless for this definition.

Warning

- IBM TCP/IP V1 for VM and MVS only supports static routing. If such routers are present in a network they should be included in the "ETC GATEWAYS" file.

2.5.3 Routing with ROUTED

The BSDROUTINGPARMS statement is used to define the characteristics of each physical link. The GATEWAY statement may be used for specific connections however. This information is used by the ROUTED server, and is not needed if you are not running ROUTED. ROUTED automatically builds the routing table of your host from the information found in the "PROFILE TCPIP" (the HOME statement tells ROUTED the IP addresses of the directly attached networks; the BSDROUTINGPARMS statement is used to determine the point-to-point connections and to identify the links from which routing information should be received), and from information received from other routers. The table is automatically built by

ROUTED. Entries are also automatically deleted after a timeout period (180 seconds) because ROUTED will then consider that the network has become unreachable. With TCP/IP for VM you can use SNALINK to establish point-to-point connections. If the remote end of the point-to-point SNALINK connection is not running ROUTED (that is the case if TCP/IP V1 is installed), ROUTED will delete the routing entry for this connection after 180 seconds. To avoid this, code the characteristics of an SNALINK connection that does not exchange RIP datagrams in the GATEWAY statement and do not include it in the BSDROUTINGPARMS statement.

The following example shows a typical BSDROUTINGPARMS statement for a gateway with two physical interfaces on which RIP datagrams arrive and an SNALINK connection to an MVS host without ROUTED running:

The BSDROUTINGPARMS statement in "PROFILE TCPIP"

```
;
 BSDROUTINGPARMS FALSE
;    Link      Maxmtu   Metric  Subnet_Mask      Dest_Addr
     ILALTRN   1000     0       255.255.0.0      0           1
     SNALVM15  2000     0       255.255.255.240  9.67.32.35  2
 ENDBSDROUTINGPARMS
;
 GATEWAY
; Network     First_Hop  Link     P_Size
  9.67.32.33  =          SNALMV20 2000   HOST                3
```

1 A directly linked network (Dest_Addr → 0) can be reached via a link named ILALTRN. A subnet mask of 255.255.0.0 is associated with this link. The maximum packet size (Maxmtu) is set to 1000 bytes and will be used only for the locally attached network (BSDROUTINGPARMS FALSE). ROUTED will be aware on the IP address of this directly attached network because it can be deduced from the subnet mask together with the Home address coded for this link.

2 A directly linked host (Dest_Addr → 9.67.32.35) running ROUTED can be reached via a point-to-point SNALINK link named SNALVM15. The subnet mask is ignored for a point-to-point link.

3 The directly connected host 9.67.32.33 does not exchange RIP datagrams. Therefore, it must not be put in the BSDROUTINGPARMS statement; that is, it must be excluded from the dynamic routing process. However, because knowing the physical characteristics of a link is vital to TCP/IP, it must be coded in the GATEWAY statement.

Warning

- Routes coded with the GATEWAY statement will not be added to ROUTED′s internal routing tables. These routes will not be broadcast to other RIP servers.
- You cannot have a GATEWAY entry for a link that is defined in the BSDROUTINGPARMS. Or, let′s say, it′s useless to do so because, when a link is ″monitored″ by ROUTED, all routing updates and deletions will be done by ROUTED. Therefore a static entry coded for a remote network accessed via a non-RIP router will be deleted by ROUTED after the usual timeout, because ROUTED will assume that the network is unreachable since no routing information has been received for this network.

The ROUTED virtual machine expects to receive routing information via the physical links defined in the BSDROUTINGPARMS statement.

When the host where ROUTED is running is hooked to a token-ring or an Ethernet, ROUTED will never delete the route to the local network because ROUTED will receive its own broadcasts. This is not the case if a point-to-point link is used and the remote host is not running RIP. ROUTED won′t see its own packet and the routes via this link will be removed after the timeout period. After a few minutes, the routing entries are gone, but the NETSTAT DEVLINKS command shows that the link is still connected. This *route deletion process* is part of the RFC and cannot be changed. Such hosts must be in the ″ETC GATEWAYS″ file.

As said previously, some routers may not be running RIP but they may be needed to access remotely connected networks. Since they won′t broadcast their routes, the local ROUTED cannot be aware that these remote networks exist. As said previously you cannot have a GATEWAY entry for a network accessed via a link that ROUTED is using. The right way to define those remote networks is to define them in the file ″ETC GATEWAYS″.

The ″ETC GATEWAYS″ file

```
net  9.67.32.96 gateway 9.67.32.19 metric 2 passive   1
host 9.67.32.2  gateway 9.67.32.18 metric 2 passive   2
net  4.0.0.0    gateway 9.67.32.19 metric 3 external  3
```

1 There is a network 9.67.32.96 connected to a router which is not capable (passive) of sending RIP datagrams. This network can be reached by sending the datagrams to the router 9.67.32.19 (first hop). Two routers (metric 2) are needed to send the datagram to the final destination.

2 A single host 9.67.32.2 is connected to a non RIP gateway. This host can be reached by sending the datagrams to the gateway 9.67.32.18.

3 The gateway that is directly connected to network 4.0.0.0 uses RIP and therefore broadcasts its route. The network must not be available to the local system; therefore the route is excluded (external) and will not be added to the routing table thus preventing all local users from accessing network 4.0.0.0.

Warning

- Make sure that you include all non-RIP gateways in the ″ETC GATEWAYS″ file.
- The text in the ″ETC GATEWAYS″ file is case sensitive and must be lowercase.
- Hosts with only SNALINK connections may require the ***-s*** option in the ***ROUTED*** command. This forces the ROUTED server to supply routing information regardless of whether it is acting as an internetwork gateway.
- ROUTED must be in the OBEY list (OBEY statement in the file ″PROFILE TCPIP″) in order to use raw sockets (sockets on the IP layer) to add, delete and update routes.
- If you use ROUTED you must not have identical IP addresses on any of your links referenced in the BSDROUTINGPARMS statement. RIP is not designed to handle this correctly.

2.5.4 ROUTED Limitations

Routing protocols are discussed in detail in *TCP/IP Tutorial and Technical Overview - GG24-3376-02*. Please refer to this publication for more information about the terms used in the following:

- RIP is not designed to provide the best route based on the link speed or reliability.
- RIP uses a vector distance algorithm and chooses routes based on a single metric (hop count). Therefore RIP does not provide any load balancing on routes. There is only one route to a destination network. If two routes lead to the same destination, the one with the lower hop count is selected; this is not necessarily the faster one.
- RIP is not designed to manage networks where legitimate hop counts approach 16, because for RIP, a hop count of 16 means that the network is unreachable.
- IBM VM ROUTED implementation does not support *triggered updates*; that is, whenever a router changes a metric for a route it will not send immediately an update message.
- IBM VM ROUTED implementation supports *simple split horizon*: routes learned from a router won′t be advertised back.
- IBM VM ROUTED implementation does not support *split horizon with poisoned reverse*: routes learned from a router won′t be advertised back with a hop count of 16.

2.6 Simple Network Management Protocol (SNMP)

In Figure 5 on page 10 there are three virtual machines required to operate a full (agent and monitor) SNMP environment. If you have multiple VM TCP/IP systems in your environment, you need only one SNMP monitor, but each VM TCP/IP system needs an agent.

NetView Version 1, Release 2 is required for the SNMP monitor function if you only plan to use the CLIST provided (or your own).

NetView Version 1, Release 3 is required for the SNMP monitor function if you plan to use fullscreen panels provided (they are written in REXX).

2.6.1 Configuring the SNMP Agent (SNMPD)

The SNMP Agent runs into the SNMPD virtual machine. Link to the C/370 minidisk must be provided using the ″SNMPDXIT EXEC″.

2.6.1.1 Community Names File

The SNMPD virtual machine (agent) needs some additional tailoring. You must first define who is authorized to send requests to the SNMP agent, in order to query the status of the TCP/IP network. The file containing the authorized monitors is called the community file ″PW SRC″ residing on SNMPD 191.

A line in the community file has the following format:

community_name network mask

The IP address of an incoming SNMP request is logically ANDed with the ***mask***. The result of the logical ANDing process is compared with the ***network***. If they match, the community names are compared. If the community names match, the request is accepted.

Community File ″PW SRC″

```
IBM             9.0.0.0       255.0.0.0         1
ITSC            9.67.38.0     255.255.255.0     2
ROLF            9.67.32.28    255.255.255.255   3
```

1. The community name IBM is valid for every monitor querying the agent from within network 9.
2. The community name ITSC can be used only within network 9.67.38.
3. The community name ROLF is restricted to the workstation ROLF itself.

Warning

The file ″PW SRC″ is case sensitive. Please note that commands entered from the NetView command line are translated to uppercase. Those entered from either the REXX panels provided with IBM TCP/IP Version 2 Release 2 for VM (to be used from NetView), or from any other monitor (AIX NetView/6000, OS/2) are not.

2.6.1.2 Trap Destination File

TRAPs are unsolicited messages that are sent by an SNMP agent to an SNMP network management station. An SNMP TRAP contains information about a significant network event. The management application running at the management station interprets the TRAP information sent by the SNMP agent. You must also define the monitors which should receive the traps sent by the SNMPD agent.

The following TRAPs are generated by an SNMP agent in the TCP/IP environment for VM:

- COLD_START
- AUTHENTICATION FAILURE
- LINK_UP
- LINK_DOWN

The file containing the internet addresses of the network management monitors which should receive the TRAPs, is called the TRAP destination file. Its name is ″SNMPTRAP DEST″ and it resides on SNMPD 191.

A line in the TRAP destination file may have the 2 following formats:

IP_address UDP Name UDP

TRAP Destination File ″SNMPTRAP DEST″

```
VM14          UPD
RS60001        UPD
9.67.38.71    UPD
```

The SNMP agent is started by TCP/IP at startup time or manually. Make sure that SNMPD is in the AUTOLOG list of the file ″PROFILE TCPIP″.

2.6.2 Configuring the SNMP Query Engine (SNMPQE)

The SNMPQE virtual machine needs no additional customization. Make sure that SNMPQE is in the AUTOLOG list of the file ″PROFILE TCPIP″.

2.6.3 Configuring NetView as an SNMP Monitor

The NetView virtual machine needs a link to the TCPMAINT 592 minidisk where all required files reside. Add the access of the TCPMAINT 592 minidisk to the NetView startup procedure ″PROFILE GCS″. Also add the load library ″SNMPLIB LOADLIB″ to the GLOBAL LOADLIB statement in the startup procedure.

Register the SNMPIUCV task and the SNMP command processor in the NetView configuration files. This is described in more detail in Chapter 20 of the *IBM TCP/IP V2 R2 for VM: Planning and Customization* manual.

2.6.4 MIB_DESC DATA File

Information that can be obtained from SNMP agents is defined by the MIB. The MIB defines objects, such as packet counts and routing tables, which are relevant to a TCP/IP environment (see Section 1.2.4, “Simple Network Management Protocol (SNMP)” on page 7).

The ″MIB_DESC DATA″ file defines the human-readable names for MIB variables. When you issue an *SNMP GET*, *GETNEXT*, or *SET* command and specify the variable name using the human-readable format, the SNMP query engine uses the ″MIB_DESC DATA″ file to map this name to the corresponding standard ASN.1 notation, before sending the SNMP request PDU to the agent.

The distributed ″MIB_DESC DATA″ file contains the text names as defined in RFC 1156. In addition to these, the following variables have been added:

- MIB-II variables from RFC 1158.
- IBM 3172 enterprise-specific MIB variables (please refer to 1.3.7.3, “3172 Network Management Variables” on page 29 for a complete list).
- Some IBM-defined variables in the enterprise-specific branch of the MIB (1.3.6.1.4.1.2.2 is ibmResearch). These IBM specific variables are:

```
* IBM SNMP agent DPI UDP port
* Queried by a DPI subagent to find out which TCP port it must use
* to communicate with the agent.
*
DPI_port                1.3.6.1.4.1.2.2.1.1.    number          2
*
* IBM †ping† round-trip-time table
* Basically used by a IBM DPI sample call PINGENG. minRTT is used
* when a NetView operator issues the SNMP Ping command.

RTTaddr                 1.3.6.1.4.1.2.2.1.3.1.  internet        60
minRTT                  1.3.6.1.4.1.2.2.1.3.2.  number          60
maxRTT                  1.3.6.1.4.1.2.2.1.3.3.  number          60
aveRTT                  1.3.6.1.4.1.2.2.1.3.4.  number          60
RTTtries                1.3.6.1.4.1.2.2.1.3.5.  number          60
RTTresponses            1.3.6.1.4.1.2.2.1.3.6.  number          60
#
# DPISAMPL C is a sample SNMP-DPI subagent and supports the following
# objects or sends them along with enterprise specific traps.
# DPISAMPL C is shipped with TCP/IP VM V2R2.
#
DPI_sample                1.3.6.1.4.1.2.2.1.4.     table          5
DPI_sampleNumber          1.3.6.1.4.1.2.2.1.4.1.   number         5
DPI_sampleOctetString     1.3.6.1.4.1.2.2.1.4.2.   string         5
DPI_sampleObjectID        1.3.6.1.4.1.2.2.1.4.3.   object         5
#
# XGMON/SQESERV does not allow you to specify empty (so use empty string)
#
DPI_sampleEmpty           1.3.6.1.4.1.2.2.1.4.4.   string         5
DPI_sampleInetAddress     1.3.6.1.4.1.2.2.1.4.5.   internet       5
DPI_sampleCounter         1.3.6.1.4.1.2.2.1.4.6.   counter        5
DPI_sampleGauge           1.3.6.1.4.1.2.2.1.4.7.   gauge          5
DPI_sampleTimeTicks       1.3.6.1.4.1.2.2.1.4.8.   ticks          5
DPI_sampleDisplayString   1.3.6.1.4.1.2.2.1.4.9.   display        5
#
# XGMON display element override. Allows you to use one SNMP Query Engine
# to which several clients connect.
```

```
#
deOverride          1.3.6.1.4.1.2.2.1.5.1.        display        1
#
# These are used by the two-way communication between NetView/390
# and XGMON. This allows a XGMON operator to use the netview.g
# program and issue NetView commands on NetView/390. The response
# comes back to XGMON and is displayed in a VGM window.
#
NVcmd               1.3.6.1.4.1.2.2.1.5.2.        display        1
NVreply             1.3.6.1.4.1.2.2.1.5.3.        display        10
NVmsgop             1.3.6.1.4.1.2.2.1.5.4.        display        1
#
# Test
#
test                1.3.6.1.9.1.                 display        10
```

The ″MIB_DESC DATA″ file must be accessible by the SNMPQE virtual machine.

It can be customized according to your own taste. Just duplicate the line you want to use with a different human-readable name (for example to use French names):

```
sysDescr            1.3.6.1.2.1.1.1.        display        900
Système             1.3.6.1.2.1.1.1.        display        900
sysObjectID         1.3.6.1.2.1.1.2.        object         900
Objet               1.3.6.1.2.1.1.2.        object         900
sysUpTime           1.3.6.1.2.1.1.3.        ticks          1
Début               1.3.6.1.2.1.1.3.        ticks          1
```

2.6.5 3172 SNMP Configuration

On the 3172 side:

The TCP/IP Network Management attributes support is a new function included in V2.1 of the IBM 3172 Interconnect Controller Program (5601-433). It is automatically included in any 3172 program load generated by the 3172 Operator Facility and it is only activated via specific commands sent from the host.

The 3172 Network Management function supports *SNMP GET*, *SNMP SET* and *SNMP TRAP* operations.

- However the SNMP agent included in TCP/IP for VM does not support *SNMP SET* operations.
- As the 3172 trap reporting function can only be enabled by explicitly setting the corresponding interface flag (*ibm3172ifTrapEnable* variable) to 01 and since the VM agent implementation does not allow any *SNMP SET* operation, no 3172-specific *SNMP TRAP* will be sent to the host.
- Problems occurring during the operation of the 3172 LAN interfaces will be detected by the TCPIP virtual machine and reported as generic *LINK_UP* and *LINK_DOWN SNMP TRAP* PDUs through the VM host′s SNMP agent.

On the VM host side:

TCP/IP V2R2 for VM only supports *SNMP GET* and *SNMP GETNEXT* operations to request/retrieve the 3172 enterprise-specific MIB variables. These requests will be answered only by those 3172 devices connected to the VM host′s TCP/IP and

whose *DEVICE* definition in the ″PROFILE TCPIP″ file includes the keyword ***NETMAN***. Following is an example of such a definition:

```
DEVICE LCS1 LCS cuu NETMAN
LINK TR1 IBMTR 0 LCS1
LINK EN1 ETHERNET 1 LCS1
```

The ″MIB_DESC DATA″ file has been updated in TCP/IP V2R2, to include the mappings between the textual names and the ASN.1 notation of the 3172-specific MIB variables, and thus to allow the retrieval of those variables using NetView and the TCP/IP Query Engine interface (SNMPQE).

2.7 RISC System/6000 CLAW Connection

2.7.1 RISC System/6000 Definitions

The following is the SMIT panel showing the definitions of the 3270 Block Multiplexer Adapter:

```
     Change/Show Characteristics of a 370 Parallel Channel Adapter


Type or select values in entry fields.
Press Enter AFTER making all desired changes.

[TOP                                              [Entry Fields
  370 Parallel Channel                           cat0
  Description                                    370 Parallel Channel A
  Status                                         Available
  Location                                       00-05
  Receive data transfer OFFSET                   [0                    +#
  Channel SPEED                                  [0                    +#
  NUMBER of transmit buffers                     [26                   +#
  NUMBER of receive buffers                      [26                   +#
  Transmit buffer SIZE                           [4096                 +#
  Receive buffer SIZE                            [4096                 +#
  STARTING subchannel address                    [0x60                 +
  NUMBER of subchannel addresses                 [2                    +#
  CLAW Mode
  [MORE...6

F1=Help           F2=Refresh        F3=Cancel         F4=List
Esc+5=Undo        F6=Command        F7=Edit           F8=Image
F9=Shell          F10=Exit          Enter=Do
```

```
  CLAW Mode
          Host NAME                                [HOST                 +
          Adapter NAME                             [PSCA                 +
          subchannel set                           [0x60                 +
  Online/Offline SWITCH                            [online               +
  Online/Offline INDICATOR                          online
  Apply change to DATABASE only                     no                   +
[BOTTOM

F1=Help           F2=Refresh        F3=Cancel         F4=List
Esc+5=Undo        F6=Command        F7=Edit           F8=Image
F9=Shell          F10=Exit          Enter=Do
```

These RISC System/6000 definitions are to be related to the VM definitions on page 90.

2.7.2 VM Definitions

2.7.2.1 System Definitions.

The S/370 or S/390 host system must be configured to properly support the system unit. It must use the appropriate IOCP definition and the proper VM definition. The system unit is similar to a 3088 CTC adapter, and should be defined as such. Please refer to *Block Multiplexer Channel Adapter: User's Guide and Programming Reference* for definition examples.

The definitions used in the ITSC Raleigh configuration were:

```
CNTLUNIT CUNUMBR=0E6,PATH=0E,SHARED=N,UNIT=3088,
         UNITADD=((60,2)),PROTOCL=S
IODEVICE ADDRESS=(E60,2),UNIT=3088,CUNUMBR=0E6
```

2.7.2.2 TCPIP Definitions

As far as IBM TCP/IP Version 2 Release 2 for VM is concerned, some definitions are needed in the "PROFILE TCPIP":

- A ***DEVICE*** statement:

  Syntax

  DEVICE *device_name CLAW address HOST PSCA NONE read_buffers write_buffers read_size write_size*

  - *device_name*: Name of the device. Maximum length 16 characters. The same name must be specified in the LINK statement.
  - *CLAW*: keyword indicating that the CLAW mode is used.
  - *address*: The hexadecimal subchannel address for communication. It must be even, and automatically a pair is assigned.
  - *HOST*: Name of the host system. This name must match the Host Name field of the *Change/Show Characteristics of a 370 Parallel Channel Adapter* SMIT panel.
  - *PSCA*: Name of the system unit. This name must match the Adapter Name field of the *Change/Show Characteristics of a 370 Parallel Channel Adapter* SMIT panel.
  - *NONE*: reserved.
  - *read_buffers*: Decimal number of buffers to allocate to the read channel program. The default is 20. Each of the buffers uses real storage.
  - *write_buffers*: Decimal number of buffers to allocate to the write channel program. The default is 20. Each of the buffers uses real storage.
  - *read_size*: Size of the read buffers. It must be less than or equal to the transmit buffer size specified in the system unit. The default is 4096.
  - *write_size*: Size of the write buffers. It must be less than or equal to the receive buffer size specified in the system unit. The default is 4096.

- A ***LINK*** statement:

  Syntax

  LINK *link_name IP 0 device_name*

  - *link_name*: unique assigned link name.
  - *IP*: keyword.
  - *device_name*: comes from the corresponding device statement.

- A ***GATEWAY*** statement:

  Syntax

  GATEWAY *network first_hop link_name max_packet_size ...*

  - The GATEWAY statement is the same as for any other device or link.

- *max_packet_size*: The value must not exceed the *write_size* specified on the DEVICE statement.

Please refer to Appendix E, “Configuration Listings for VM System RALYESA (VMESA)” on page 287 for an example of the TCPIP definitions.

2.8 FDDI LAN Attachment Support

The definition of an IBM 3172 FDDI LAN in IBM TCP/IP Version 2 Release 2 for VM consists of a standard *LCS DEVICE* statement and of a new *FDDI LINK* statement, as follows:

```
DEVICE LCSFDDI1 LCS cuu
LINK FDDILAN1 FDDI 0 LCSFDDI1
```

The maximum frame size used for transmission on the FDDI LAN is specified in the *GATEWAY* statement (static routing definition) or the *BSDROUTINGPARMS* statement (dynamic routing definition) of the ″PROFILE TCPIP″. This value must be set according to the *RFC 1188* specification (see Section 1.3.2.4, “FDDI and TCP/IP” on page 21), that is, 4352 bytes.

2.9 3745 ELA and ACF/NCP V6 IP Router Support

Figure 25 on page 93 shows an abstract of the definition files and their relationships in a basic configuration.

Note: Not all the details about the new macros and parameters of the NDF generation for the ACF/NCP V6 IP router are provided here, and the reader should refer to the relevant ACF/NCP V6 documentation, as soon as it is available. However, very useful information can already be found in the ITSC bulletin *ACF/NCP V6 - Planning and implementation Guide - GG24-3785*.

The parts of the definitions which are specific to the NCP V6 IP router are:

1. **On the TCP/IP side**:

   Up to V2R1, SNALINK uses two parallel type 0 LU sessions to communicate with another SNALINK partner, one for sending and one for receiving. In V2R2, SNALINK can be configured to use only one type 0 LU session for a connection, as required by the NCST implementation. This is achieved by setting the *SessionType* parameter of the *SNALINK* start command to the value *SINGLE* (see ″PROFILE GCS″ of the SNALNKA virtual machine in Figure 25 on page 93).

2. **On the ACF/NCP side**:

   Please refer to the 3745 - NCP V6 IP router definition - NDF sample in Figure 25 on page 93:

   - Ethernet adapter

     Each Ethernet port will be defined to VTAM as an SNA line and a type 1 PU. *LANTYPE=DYNAMIC* specifies that this Ethernet interface supports both the Ethernet V2 (DIX) and the IEEE 802.3 DLCs. *INTFACE=(ENET1,1492)* defines the unique interface name required for each Ethernet line and is used to associate IP route entries with this line, and the largest data unit that can be sent over this interface (MTU).

   - NCST Logical Unit

     The *NCST GROUP* defines the connectionless transport LUs and their sessions′ characteristics. *INTFACE=(HOST1,1492)* defines the name which is used to associate IP route entries, local (home) IP addresses and subnet masks with the sessions that will transport the IP traffic through the SNA network, and the largest data unit that can be sent over this interface (MTU). *REMLU=(SNALKH1,MODEIP)* defines the SNALINK partner LU name and a mode table entry to be used for the NCST-SNALINK session.

   - Home addresses and IP routing table entries

     New NDF statements are required to define the NCP home list (*IPLOCAL*) and the routing table (*IPGATE*). Their meaning may easily be related to the *HOME* and *GATEWAY* statements respectively of a ″PROFILE TCPIP″ file. *NEXTADDR=0* is used for direct routing. For indirect routing, the *NEXTADDR* parameter specifies the IP address of the next router on the path towards the destination network.

     Note: There is no support, in ACF/NCP V6, for the dynamic update of the IP routing table (that is, using RIP for example). All of the routing entries must be pre-defined in the NCP definition list, and any change will require the initialization of a new NCP load module in the 3745.

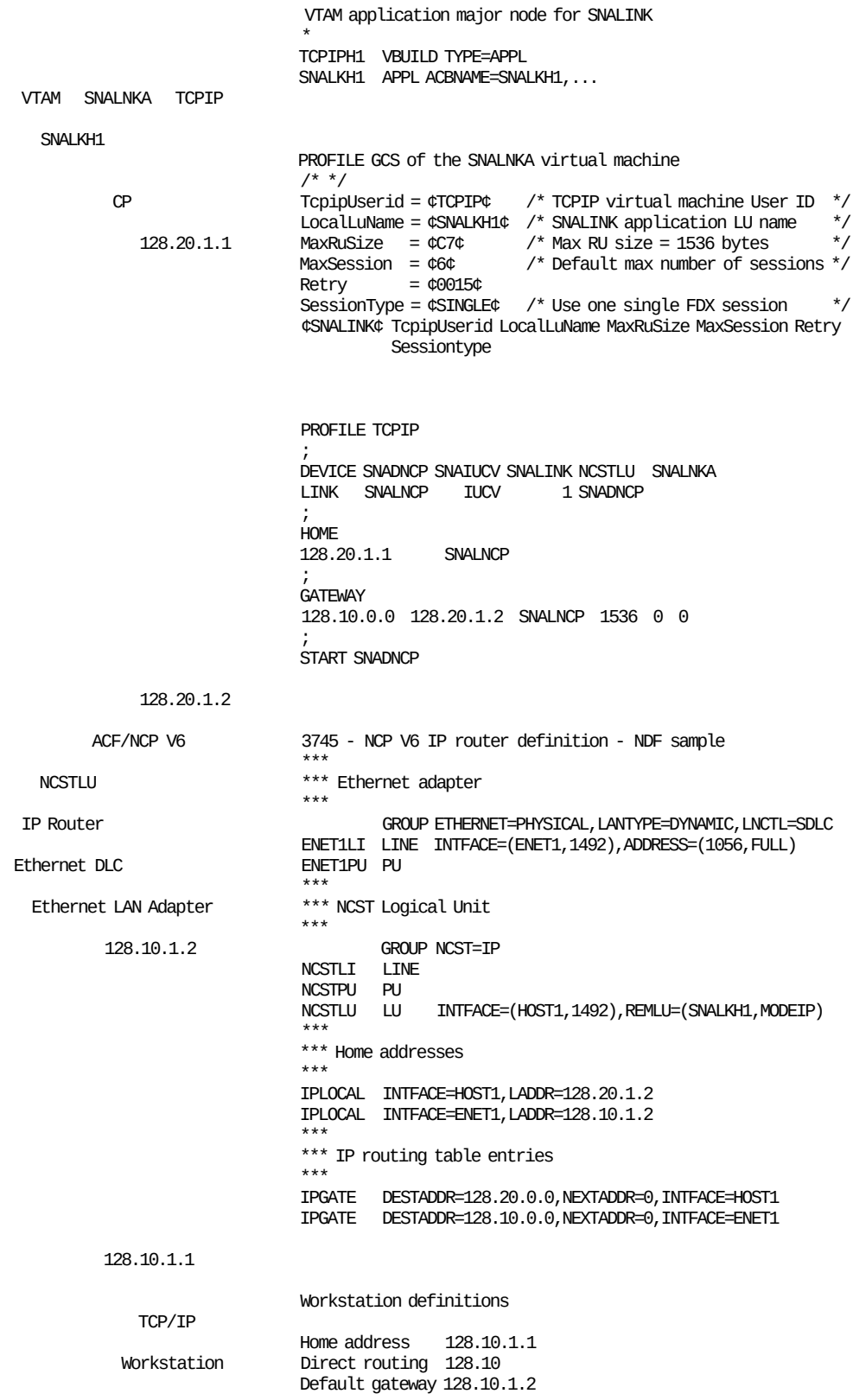


Figure 25. 3745 IP Routing Systems Definitions

2.10 File Transfer Protocol (FTP)

The File Transfer Protocol (FTP) server allows you to transfer files between your local host and a foreign host that supports TCP/IP. When invoked, FTP establishes a connection to a foreign host's FTP server. After you have identified yourself to the foreign FTP server, you will be able to retrieve information about the status of the foreign server, list directories, transfer files, delete or rename files, enter CMS commands and exit the FTP environment when an error occurs.

The FTP server virtual machine (user ID FTPSERVE) is started by the "TCPRUN EXEC" procedure. TCPRUN is invoked by the "PROFILE EXEC" residing on the FTPSERVE 191 minidisk.

TCPRUN starts the FTP server using the default values. However you can specify different startup parameters by modifying the global variable *PARMS* in the "FTPDEXIT EXEC" procedure which is located on the TCPMAINT 591 minidisk.

Please refer to 2.20, "RACF Considerations" on page 171 for RACF considerations.

2.10.1 Multiple FTP Servers

Multiple FTP servers can be configured to increase the throughput for concurrent FTP usage. It should only be considered when throughput rates become slow due to FTP congestion. The following example shows how to set up the "PROFILE TCPIP" for three FTP servers:

```
AUTOLOG
   FTPSERVE password
   FTPSERV2 password
   FTPSERV3 password
PORT
   21 TCP FTPSERVE
   21 TCP FTPSERV2
   21 TCP FTPSERV3
   20 TCP FTPSERVE NOAUTOLOG
   20 TCP FTPSERV2 NOAUTOLOG
   20 TCP FTPSERV3 NOAUTOLOG
```

To allow each FTP user to access port 20, which is the FTP port used for data transfer, you must reserve this port and tell the TCPIP virtual machine not to expect a passive open on it.

Note: Be aware that all VM limitations apply to the TCP/IP environment. For example multiple write access to the same minidisk is not recommended. This means, for example, that if you want to write a file to a VM minidisk (using the PUT command from a workstation) which is accessed in write mode by another VM user ID, the FTP server won't be able to link that minidisk in write mode and the write operation will fail. Another consequence is that you may not always see the file with a file mode of 0.

Multiple FTP servers is a solution even if you are not concerned about throughput. In VM a minidisk is accessed and is given a file mode which is a letter. Obviously the number of letters available is limited: a single FTP server may not be able to access more than 20 minidisks for file transfer purposes because it already has an A minidisk (its own), accesses some system minidisks (Y and S minidisks), and accesses TCPMAINT minidisks. So if you expect to have

more than 20 FTP sessions concurrently, you will have to define another FTP server in your VM system.

2.10.2 Using the Shared File System

With IBM TCP/IP Version 2 Release 2 for VM the Shared File System is supported by the FTP client *only*. The Shared File System (or SFS) is a part of CMS. Please refer to the *CMS Shared File System Administration* manual for more information about SFS. SFS lets users organize their files into groups known as directories and selectively share those files and directories with other users. To do this, a collection of minidisks is assigned to a single virtual machine. This virtual machine is called a file pool server machine. The collection of minidisks is known as a file pool. A file pool contains the files for many users, not just one. The file pool server manages all the files within the file pool. Each file pool is given a name. The name of the file pool we used was: SFSRSC1. Using the create directory command, we created directories and subdirectories. The following is the sequence used to access the different directories and to retrieve files from a RISC System/6000.

```
Ready; T=0.01/0.01 16:59:36
q accessed
Mode  Stat  Files  Vdev  Label/Directory
A      R/W    104  191   DEB191
B      R/O    758  301   GDDM23
C      R/O    213  5FF   RALDSK
D      R/O     97  591   TCP591
E      R/O    569  592   TCP592
S      R/O    358  190   CMS-R7
Y/S    R/O   3213  19E   Y-DISK
Z      R/O   4925  19F   Z-DISK
Ready; T=0.01/0.01 16:59:42
access . f        1
Ready; T=0.01/0.01 16:59:57
listdir f         2
Fm Directory Name
F  SFSRSC1:DEBULOIS.
-  SFSRSC1:DEBULOIS.RS6000
-  SFSRSC1:DEBULOIS.RS6000.GET
-  SFSRSC1:DEBULOIS.RS6000.PUT
Ready; T=0.01/0.01 17:00:12
access .rs6000.get g       3
Ready; T=0.01/0.01 17:00:26
access .rs6000.put h       3
Ready; T=0.01/0.01 17:00:41
q accessed
Mode  Stat  Files  Vdev  Label/Directory
A      R/W    104  191   DEB191
B      R/O    758  301   GDDM23
C      R/O    213  5FF   RALDSK
D      R/O     97  591   TCP591
E      R/O    569  592   TCP592
F      R/W      1  DIR   SFSRSC1:DEBULOIS.           4
G      R/W      0  DIR   SFSRSC1:DEBULOIS.RS6000.GET
H      R/W      0  DIR   SFSRSC1:DEBULOIS.RS6000.PUT
S      R/O    358  190   CMS-R7
Y/S    R/O   3213  19E   Y-DISK
Z      R/O   4925  19F   Z-DISK
Ready; T=0.01/0.01 17:00:50
```

1 Access the top level directory with a file mode of F.

2 List the directories and subdirectories created via the create directory command.

3 Access subdirectories with different file modes. From now on, these subdirectories will be considered as separate minidisks.

4 The directories can be seen via the q accessed CMS command.

Now the subdirectories have been accessed and we can use FTP to transfer files in a rather straightforward way:

```
Ready; T=0.01/0.01 17:57:42
ftp 9.67.38.73
VM TCP/IP FTP V2R2
Connecting to 9.67.38.73, port 21
220 rs60003 FTP server (Version 4.1 Sat Nov 23 12:52:09 CST 1991) ready.
USER (identify yourself to the host):
root,
   USER root
331 Password required for root.
Password:
   PASS ********
230 User root logged in.
Command:
cms listdir      1
Fm Directory Name
F  SFSRSC1:DEBULOIS.
-  SFSRSC1:DEBULOIS.RS6000
G  SFSRSC1:DEBULOIS.RS6000.GET
H  SFSRSC1:DEBULOIS.RS6000.PUT
Command:
lcd g            2
Local directory mode is ¢G¢
Command:
get .Xdefaults X.rs60003AIX
   PORT 9,67,38,145,4,95
200 PORT command successful.
   RETR .Xdefaults
150 Opening data connection for .Xdefaults (754 bytes).
226 Transfer complete.
779 bytes transferred. Transfer rate 1.32 Kbytes/sec.
Command:
quit
   QUIT
221 Goodbye.
Ready; T=0.20/0.24 17:08:21
listf * * g
X        RS60003A G1
```

1 Using the cms subcommands, we are able to check the way subdirectories are accessed.

2 The local directory (that is, the one where the file will be stored) is changed to DEBULOIS.RS6000.GET which was accessed with file mode G.

2.11 Simple Mail Transfer Protocol (SMTP)

The SMTP virtual machine can be configured to operate as a mail gateway between TCP/IP network sites and RSCS network sites. This means that, for example, PROFS users can exchange mail with UNIX** workstations via the VM TCP/IP SMTP gateway.

In conjunction with the name server, SMTP can also use MX records to direct the server to deliver mail to an alternate host.

The SMTP server has an interface with the VM Send Message facility (SMSG), which allows the querying of SMTP mail delivery queues and statistics, and provides a set of privileged commands for system administration tasks.

With the new mail header customization, users can change the rules used to re-write header addresses and specify desired header address transformations.

The number of simultaneously-delivered pieces of mail can be limited, to save system resources.

2.11.1 Configuring SMTP

The SMTP server is started by the ″TCPRUN EXEC″ procedure which resides on the server common disk (TCPMAINT 591). TCPRUN is invoked by the ″PROFILE EXEC″ on SMTP server′s 191 disk. TCPRUN starts the SMTP server using the default values. However, you can specify different startup parameters for the SMTP server by modifying the global variable *PARMS* in the ″SMTPEXIT EXEC″ procedure and specifying different parameters for the SMTP command. Please refer to the *IBM TCP/IP V2 R2 for VM: Planning and Customization* manual for more information on how to modify the ″SMTPEXIT EXEC″.

The SMTP program needs a configuration file at startup time. A sample file comes with the installation tape, which works fine for the gateway function. This file, called ″SMTP CONFIG″, may be modified to meet the requirements of your environment.

IBM TCP/IP Version 2 Release 2 for VM has changed some configuration statements and introduced some new ones:

- *ALTRSCSDOMAIN* has the same characteristics as *RSCSDOMAIN* and allows the specification of an alternate domain name for the RSCS network to which the SMTP server belongs.
- *BADSPOOLFILEID* replaces the old *LOOPINGMAIL* statement and specifies the user ID on the local system, to which the SMTP server should transfer unreadable spool files and looping mail. The specified user ID has to be on the same system as the SMTP server, since this will use the *CP TRANSFER* command.
- *ONDISKFULL* allows the specification of a sequence of CP commands to be executed by the SMTP server when its 191 minidisk is filled up beyond a given threshold. The SMTP server may behave unpredictably when its A-disk is full, and this statement is intended to help prevent such a situation.

For example, with the following statements:

```
ONDISKFULL  80  90  -90  -80  ACTIONS
  BEGIN CP MSG TCPMAINT *MESSAGE* END
ENDACTIONS
```

When the SMTP server's A-disk is filled up above 80%, and then above 90%, and then drops below 90% and 80%, the owner's user ID will be notified with the following message:

```
date  time  SMTPuserID at SMTPnodeID - Disk Above/Below n Percent Full
```

This statement can be used to execute any sequence of CP commands, and you can define different statements specifying different actions, each associated with a specific percentage.

- *OUTBOUNDOPENLIMIT* prevents the SMTP server from using too many system resources. It allows limiting the maximum number of simultaneous TCP connections over which the SMTP server will actively deliver mail. The default value is *infinity*, which means no limit.
- *SMSGAUTHLIST* specifies the user IDs which are allowed to issue privileged *SMSG* commands to the SMTP server. See Section 2.11.9, "SMSG Interface to SMTP" on page 106 and Section 3.4.5, "SMSG Interface to SMTP" on page 184 for details about the *SMSG* interface to the SMTP server.
- *WARNINGAGE* specifies the number of days after which a copy of the mail is returned to the sender, indicating that the mail has thus far been undeliverable, and that the SMTP server will continue to attempt delivery for the number of days specified by the *RETRYAGE* statement.

  For example, the following statements:

  ```
  RETRYAGE = 3
  WARNINGAGE = 1
  ```

  mean, from the SMTP server perspective: "I am going to try to send your mail for 3 days. After 1 day, I will warn you that it has not yet been delivered".
- *REWRITE822HEADER* specifies whether the SMTP server should rewrite the headers of the pieces of mail **arriving from the RSCS side of the SMTP mail gateway or generated inside the gateway itself, to be sent to the TCP/IP network** and controls the way the rewrite is performed.

  Up to V2R1, minimal rewriting was done if the *REWRITE822HEADER* statement was used, but the process was hard-coded and was not documented. See Section 2.11.10, "SMTP Mail Headers" on page 107 for more details.
- *MAILER* specifies the address of a virtual machine in the RSCS network, which can receive mail in *BATCH* SMTP format. In IBM TCP/IP Version 2 Release 2 for VM:

  1. The keywords *PUNCH | NETDATA* have been added to specify whether the MAILER virtual machine can accept either PUNCH format or NETDATA format spool files.

2. The keywords *SOURCEROUTES | NOSOURCEROUTES* have replaced the keywords *NEW | FOLDNOSOURCEROUTE | OLD* to specify whether the MAILER virtual machine accepts or does not accept BATCH SMTP header addresses with source route information.

Refer to the *IBM TCP/IP V2 R2 for VM: Planning and Customization* manual for details on how to modify the ″SMTP CONFIG″ file.

2.11.2 SMTP Domain Name Resolution

The SMTP server can be configured to use either a domain name server or the local site tables. To use the domain name server, configure the ″TCPIP DATA″ file with the internet address of one or more name servers. If the ″TCPIP DATA″ file does not point to any name server, the local site tables are used.

Warning

However, if the SMTP server is configured to use name servers, SMTP does not use the site tables even if there is no response from the defined name servers.

2.11.3 SMTP NOTE and SENDFILE EXECs

A new version of the ″NOTE EXEC″ and ″SENDFILE EXEC″ procedures is included for sending electronic mail using SMTP.

The ″NOTE EXEC″ and ″SENDFILE EXEC″ files are installed as ″NOTE EXECTCP″ and ″SENDFILE EXECTCP″, and reside on the TCPMAINT 592 minidisk. You may rename them respectively to ″NOTE EXEC″ and ″SENDFILE EXEC″ to use them for SMTP. If you do this, be sure that *all* your RSCS node IDs have an entry in the ″SMTPRSCS HOSTINFO″ file. If a destination node is not found in this file it will be assumed to be a TCP/IP node. Be aware that, if you rename ″NOTE EXECTCP″ to ″NOTE EXEC″ and ″SENDFILE EXECTCP″ to ″SENDFILE EXEC″, they may be modified by some VM maintenance thus preventing you from using them to send notes to TCP/IP nodes.

That's a good reason not to replace the CMS versions of NOTE and SENDFILE. Let's assume you renamed them to ″NOTETCP EXEC″ and ″SENDFTCP EXEC″. According to *IBM TCP/IP V2 R2 for VM: Planning and Customization* that's all you have to do but, unfortunately, that's not true. The following is the right sequence:

1. Rename ″NOTE EXECTCP″ to ″NOTETCP EXEC″ and ″SENDFILE EXECTCP″ to ″SENDFTCP EXEC″.
2. Modify the ″NOTETCP EXEC″ to call another note profile (let's call it NOTETCP), and to get a new GLOBALV (let's call it NOTETCP again).

   ```
   ¢GLOBALV SELECT $¢userid ¢GET NOTETCP¢
   upper notetcp

   parse value notetcp ¢PROFILE PROFTCP SHORT LOG NOACK NOTEBOOK ALL¢,
      with             .       profile  fmt   log ack   keep     keepfn .
   ```

3. Issue the following command to set a new GLOBALV with the right parameters:

   ```
   GLOBALV SELECT $userId SETLP NOTETCP PROFILE PROFTCP LONG LOG NOACK NONOTEBOOK ALL
   ```

   Except for the ″PROFTCP″ keyword, all other keywords can be changed to suit your own taste. The SETLP means (among other things) that the global

variable will be stored in the file "LASTING GLOBALV" which is read during the IPL of CMS. You won't have to issue this `GLOBALV ...` command each time you log on.

4. Copy the usual "PROFNOTE XEDIT" as "PROFTCP XEDIT".
5. Modify your "PROFTCP XEDIT" to use "SENDFTCP EXEC" instead of SENDFILE EXEC:

```
00030 ¢COMMAND SET SYNONYM SEND 4 COMMAND CMS SENDFTCP ( NOTE¢
00031 ¢COMMAND SET SYNONYM CANCEL 6 COMMAND CMS NOTE     ( CANCEL¢
00032 ¢COMMAND SET PF01 COMMAND HELP CMS NOTE¢
00033 ¢COMMAND SET PF13 COMMAND HELP CMS NOTE¢
00034 ¢COMMAND SET PF05 COMMAND CMS SENDFTCP  ( NOTE¢
00035 ¢COMMAND SET PF17 COMMAND CMS SENDFTCP  ( NOTE¢
```

6. You are ready to go if you want to send notes (just type `NOTETCP`) or files with the following command: `SENDFTCP Filename Filetype Filemode to joe at paris`. If you want to use SENDFTCP via a full screen panel (with the command `SENDFTCP`), you will have to modify the file "X$SEND$X XEDIT" :
7. In the file "X$SEND$X XEDIT" change all the SENDFILE statements to SENDFTCP.

You should leave the EXECs on the TCPMAINT 592 minidisk, so that every TCPIP user has access to them.

As said before the *NOTETCP* and *SENDFTCP* commands need to know if the recipient can be reached via RSCS or not. For this purpose, they read the "SMTPRSCS HOSTINFO" file, which contains the list of all the RSCS nodes. A utility called *SMTPRSCS* creates this file using the RSCS configuration file "RSCS CONFIG" as input. The "SMTPRSCS HOSTINFO" file must be on a user-visible minidisk, preferably on the TCPMAINT 592 minidisk.

The following command sequence creates the file and places it on the correct minidisk:

1. CP LINK TCPMAINT 592 592 MW
2. ACC 592 B
3. CP LINK RSCS 191 699 RR
4. ACC 699 C
5. SMTPRSCS RSCS CONFIG C
6. COPY SMTPRSCS HOSTINFO A = = B (REPL

Warning

The SMTP virtual machine always needs write access to its 191 minidisk. Make sure that there is enough space on the minidisk and that SMTP has write access to it.

The ″SMTPRES EXEC″ allows you to check whether a recipient is an RSCS node or a TCP/IP one. To use the ″SMTPRES″ facility in an interactive way, you need to modify it:

```
00206    answer=space(answer)
00207    if cmdfunc=¢COMMAND¢ then
00208 /*  if erc=0 then push answer - Won¢t say anything */
00209     if erc=0 then say answer   /* Modified */
00210     else nop
00211    else exit answer
```

The syntax of the command and the answer are:

```
smtpres resolve xxx wtscpok    1
XXX WTSCPOK RSCS
smtpres resolve xxx fred       2
xxx fred TCP
Ready; T=0.09/0.15 14:34:24
smtpres resolve xxx whynot     3
xxx whynot TCP
```

1 Query for an RSCS node ID listed in the ″STMPRSCS HOSTINFO″ file.

2 Query for a TCP/IP node ID (not listed in the ″STMPRSCS HOSTINFO″ file).

3 Query for a fancy node ID (not listed in the ″STMPRSCS HOSTINFO″ file). SMTP will assume it is a TCP/IP node ID.

Warning

Generic node IDs (for example: SMA*) are not supported and won't have any entry in the ″SMTPRSCS HOSTINFO″ file. They will be considered as TCP/IP nodes; that is, the files sent to them will be spooled to SMTP not to RSCS.

An easy way to solve this problem is to copy the ″RSCS CONFIG″ onto your A-disk and to add dummy entries in this file to add the RSCS nodes you want to send mail to. The ″SMTPRSCS MODULE″ does not look for valid RSCS entries: it will process the LINK and ROUTE statements and write an entry in the ″SMTPRSCS HOSTINFO″ file. This workaround can be applied either on the SMTP gateway or on the end node. The following are examples.

Sample dummy ″RSCS CONFIG″ file:

```
LOCAL     MLVFSC5
LINK MLVFSC6
LINK MLVFSC2
LINK ral9390
LINK ral9360
ROUTE FRAHON2  BDLVMA
ROUTE MLVFSC6  MLVFSC0
ROUTE DOM*     RSCSLAN
```

Executing the ″SMTPRSCS MODULE″ on this file:

```
smtprscs rscs config a
SMTPRSCS V2R2
Generic nodeid not supported, ignoring DOM*   1
Warning:  Duplicate entry: MLVFSC6            2
Entries in RSCS Hash Table:  6
Length of Hash Table:        11
Average Seeks per Entry:     1.167
Hash Table is:               54.5% Full
Ready; T=0.01/0.03 10:59:18
```

1 Generic node ID not supported. It will not appear in the ″SMTPRSCS HOSTINFO″ file. All mail sent to an RSCS node ID whose first three characters are SMA, will be sent to the SMTP gateway.

2 Node ID MLVFSC6 appears two times in the ″RSCS CONFIG″ file. That's not a problem.

″SMTPRSCS HOSTINFO″ file:

```
MLVFSC2
FRAHON2
MLVFSC5
MLVFSC6

RAL9360


RAL9390
```

Warning

Only the first node ID of a ROUTE statement will have an entry in the ″SMTPRSCS HOSTINFO″ file. For example an entry such as:

```
ROUTE FRAHON2  BDLVMA
```

Will result in a FRAHON2 entry. The RSCS node ID BDLVMA won't be present in the ″SMTPRSCS HOSTINFO″.

2.11.4 SMTP Gateway

A gateway is a host that is used to exchange mail between an RSCS network and a TCP/IP network. An SMTP gateway needs to have TCP/IP and SMTP up and running. A VM host on the RSCS network (we called it an end node) does not need TCP/IP. This end node will use the SMTP gateway to send/receive mail to/from the TCP/IP network.

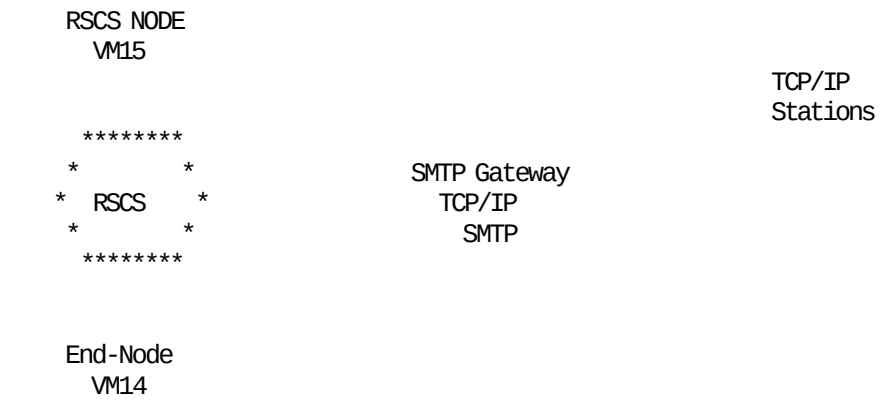


Figure 26. SMTP Gateway and End-Node

VM14: VM14 is an end node. This means it is able to send mail to the TCP/IP network through the SMTP gateway (Please refer to Section 2.11.7, "Configuring an SMTP End Node" on page 104 for more information about what is needed to configure an end node), and to send mail to a pure RSCS node using the RSCS network. TCP/IP is not installed on an end node.

VM15: VM15 is a pure RSCS system. TCP/IP is not installed. VM15 is able to send mail to any VM system using the RSCS network. It cannot send mail to the TCP/IP workstations.

SMTP Gateway: Acts as a mail gateway between the RSCS and the TCP/IP network. TCP/IP is installed.

TCP/IP Workstations: These workstations have TCP/IP installed and use SMTP to send mail to any RSCS node.

2.11.5 Configuring an SMTP Restrict Gateway

The term ″SMTP restrict gateway″ is not a technical term. It's used only to differentiate the SMTP gateway from the secure SMTP gateway discussed in Section 2.11.6, "Configuring an SMTP Secure Gateway" on page 104.

A restrict gateway is defined using the ″SMTP CONFIG″ file, by uncommenting the ″restrict″ statement. The following is a basic example:

```
RESTRICT RETURN   1
;   charming@ourvm.our.edu  2
;   charming@OURVMX         3
;   charming@ourvm*         4
  *@WTSCPOK       5
ENDRESTRICT       6
```

1 When receiving a file from restricted user ID, SMTP can take different actions. Using the `RETURN` statement the file will be sent back to the originator. Please refer to *IBM TCP/IP V2 R2 for VM: Planning and Customization* for more information. These definitions apply to VM systems only (you cannot restrict the use of such a gateway for TCP/IP workstations).

2 Mail from the TCP/IP network sent by the user ID `charming` from the VM host `ourvm.our.edu` will be returned.

3 Same as 2 but for the RSCS network.

4 Generic entry.

5 Generic entry (all mail from node ID WTSCPOK will be returned).

6 End of the *RESTRICT* statement.

With such an ″SMTP CONFIG″ file a restricted user ID will receive the following messages when attempting to use the SMTP gateway:

```
From RAL9390:          SMTP    * Rejecting Spool File 1227 (8598)
From RAL9390:          SMTP    * DEBULOIS@WTSCPOK on Restricted List
```

Warning

The Restrict and Secure options are mutually exclusive.

2.11.6 Configuring an SMTP Secure Gateway

To enable the SMTP secure gateway mode, you must add the SECURE statement to the ″SMTP CONFIG″ file. When operating in Secure Gateway mode, only those RSCS addresses in the SMTP security table (file ″SMTP SECTABLE″) are authorized to send or receive mail.

The following is an example of SMTP SECTABLE security table:

```
*  userid    nodeid    nickname    primary_nick?    primary_mbox?
*
   DEBULOIS  MLVFSC1       TCP1          Y                Y
```

According to the previous table the following translation will be done (assuming the Gateway *hostname* is SMTP-GW and the domain name is IBM.COM):

- Mail received from the following RSCS addresses will be rewritten to the following TCP/IP addresses (from primary_mbox):

  DEBULOIS at MLVFSC1 TCP1@SMTP-GW.IBM.COM

- Mail sent to the following TCP/IP address will be forwarded to the RSCS address (from primary_nick):

  TCP1@SMTP-GW.IBM.COM DEBULOIS at MLVFSC1

IBM TCP/IP V2 R2 for VM: Planning and Customization mentions some other combinations between *primary_nick* and *primary_mbox* but we have been unable to make them work.

Warning

Once your gateway is configured as a secure SMTP gateway, you must define the local SMTP users (that is the ones on the gateway itself) in the ″SMTP SECTABLE″.

2.11.7 Configuring an SMTP End Node

An end node is a host that uses an SMTP gateway to exchange mail between an RSCS network and a TCP/IP network. This host does not require to have TCP/IP installed. Typically, it is an RSCS node. The standard ″NOTE EXEC″ and ″SENDFILE EXEC″ shipped with CMS are not able to send mail/files to a TCP/IP host. Therefore you will need to send (from the SMTP gateway) some files to the end node. Those files are:

- NOTETCP EXEC (or ″NOTE EXEC″ if you did not change the name)

- SENDFTCP EXEC (or ″SENDFILE EXEC″ if you did not change the name)
- SMTPRES EXEC
- SMTPQUEU EXEC
- SMTPSEND EXEC
- NOTE HELPCMS (optional - Help file)
- SENDFILE HELPCMS (optional - Help file)
- PROFTCP XEDIT (or ″PROFNOTE XEDIT″ if you did not change the name)
- X$SEND$X XEDIT
- SMTPRSCS HOSTINFO. This file is required, since it's the way your end node will be able to decide if the mail has to be sent to the SMTP gateway or to the RSCS network.

To be able to send mail to the RSCS and TCP/IP network, the following sequence must be performed:

- In the ″NOTETCP EXEC″, the ″SENDFTCP EXEC″ and in the ″X$SEND$X XEDIT″ files, change all the `smtp_nodeid=locnode` entries to `smtp_nodeid=¢RAL9390¢` assuming that RAL9390 is the RSCS node ID of your SMTP gateway.
- Check the ″SMTPRSCS HOSTINFO″ file. The file you transferred from the gateway may not have all the RSCS entries known by your end node. If it's the case send the ″SMTPRSCS MODULE″ file from the gateway to the end node and process the ″RSCS CONFIG″ file with it. Doing this you will be sure that all the RSCS nodes defined in your end node will be in the file ″SMTPRSCS HOSTINFO″. Thus, the new ″NOTETCP EXEC″ will keep on sending mail to RSCS if the destination node is RSCS, and to the SMTP gateway if the destination is not found in the ″SMTPRSCS HOSTINFO″ file.

2.11.8 Using MX Records

The basic idea behind MX records is to send the mail as close as possible to the final destination. The destination host may currently be inactive, for example, because it is far away in another time zone. SMTP needs a synchronous connection to deliver the mail, but due to the different time zones, two systems might never be active at the same time. These two hosts could never exchange mail. Using MX records allows the SMTP server to deliver the mail to an alternate host, which is more available. SMTP tries to deliver the mail to the host with the lowest count. If the host is not available now, it tries the host with the next lowest count.

The following example shows MX records to direct the mail to an alternate host if the station *vm14* is not active. These records will be coded in the ″MASTER DATA″ file of the NAMESRV virtual machine. Please refer to Section 2.18, “Domain Name Server” on page 142 for a complete description of the file ″MASTER DATA″.

MX Records

```
vm14      IN  A      9.67.38.66
          IN  MX  0  vm14
          IN  MX  2  vm15
          IN  MX  4  vmesa
ral9390   IN  CNAME  vm14
```

2.11.9 SMSG Interface to SMTP

The CP SMSG command provides an interface to the SMTP virtual machine to:

- Query the operating statistics or the mail delivery queues of the SMTP virtual machine:

  Format of SMSG command

  ```
  SMsg    SMTP    HElp
                  QUeues
                  STats
  ```

  Only these three commands may be issued by any CMS user, and only these three will be displayed as a result of the *SMsg SMTP HElp* command.

  Note: The old *SMTPQUEU* command still exists and works the same as before (that is, it sends a batch file to the SMTP server and receives a response file back). The new *SMsg SMTP QUeues* command returns the same kind of information, but using a different format.

- Perform privileged system administration tasks:

  Format of privileged SMSG command

  ```
  SMsg    SMTP    HElp
                  CLosecon
                  REboot
                  SHutdown
                  TRace
                  NOTrace
                  DEbug
                  NODebug
  ```

  Privileged user SMSG commands are only accepted from users specified in the *SMSGAUTHLIST* statement of the ″SMTP CONFIG″ file.

  If the SMTP server is defined in the *AUTOLOG* list of the TCPIP virtual machine, it will be automatically restarted after having been stopped. Therefore, you should not define the SMTP server in the *AUTOLOG* list, if you intend to use the *SMsg SMTP SHutdown* command.

  The *SMsg SMTP TRace | NOTrace* and *SMsg SMTP DEbug | NODebug* commands respectively turn the tracing and debugging capabilities on/off, without having to shut the SMTP server down to edit the ″SMTP CONFIG″ file and then bring it back up again.

See Section 3.4.5, “SMSG Interface to SMTP” on page 184 for examples of the use of the SMSG interface to SMTP.

2.11.10 SMTP Mail Headers

Electronic mail has a standardized syntax for text messages that are sent across networks. Messages have an envelope and a content. Fields in the envelope are in a rigid format and are also called headers. These headers contain all the information necessary to send and receive messages across networks.

The standard syntax is described in *RFC 822 - Standard for the Format of ARPA Internet Text Messages*. This standard does not dictate the internal formats used at specific sites.

The *REWRITE822HEADER* statement is used to control the way SMTP performs a rewrite of RFC 822 headers.

The rules which describe the rewrite of the RFC 822 header are described in the ″SMTP RULES″ file, which consists of two sections:

field definition This section contains the names of the header fields whose addresses are to be rewritten.

rule definition This section contains the rewrite rules.

With this new feature included in IBM TCP/IP Version 2 Release 2 for VM, you will now be able to specify how RFC 822 mail headers are rewritten, when mail passes from an RSCS network to the SMTP network.

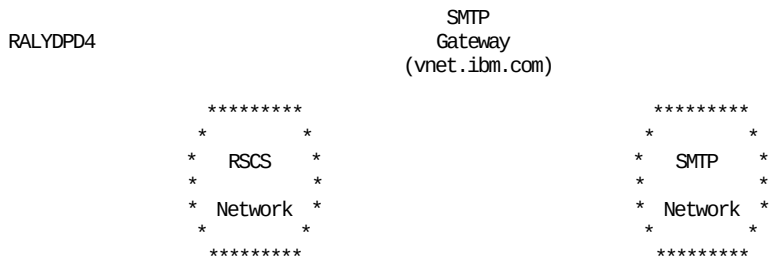


Figure 27. Rewriting RFC 822 Headers

In Figure 27, let us assume that I send mail from *lesia@ralydpd4*. The RSCS address information *lesia@ralydpd4* has no meaning to an internet user who would like to send an answer back using SMTP. Therefore I would set up the SMTP mail gateway so that it rewrites this address as *lesia@ralydpd4.vnet.ibm.com* which the internet user can understand.

Let us build the corresponding sample rewrite rule. The basic format of a rewrite rule is:

alias: before-address-pattern = after-address-pattern

where *alias* specifies which field in the header will be affected by the rewrite rule. To have the rule applied only to the *from* part of the mail header, first we specify *FromAlias='From'* in the field definition section, and second we set the *alias* field of the rewrite rule to *FromAlias*.

Now the simple rule to perform the required conversion is:

A '@' AnyRSCSHostName = A '@' AnyRSCSHostName'.'RSCSDomain

which will change any given *userID@nodeID* to *userID@nodeID.vnet.ibm.com*, if we assume that the *nodeID* belongs to the IBM RSCS network.

AnyRSCSHostName and *RSCSDomain* are predefined keywords for use in the ″SMTP RULES″ file:

- *AnyRSCSHostName* matches any RSCS host name defined in the ″SMTPRSCS HOSTINFO″ file
- *RSCSDomain* matches the name of the RSCS network as defined in the *RSCSDOMAIN* statement of the ″SMTP CONFIG″ file.

The resulting ″SMTP RULES″ file would look like:

Sample simple ″SMTP RULES″

```
Field Definition Section

FromAlias = ¢From¢

Rule Definition Section

FromAlias:  A ¢at¢ AnyRSCSHostName  =   A ¢@¢ AnyRSCSHostName¢.¢RSCSDomain
```

The *alias* field in the rewrite rule is optional. If it is omitted, then the rule(s) will apply to the header fields defined under *DefaultFields* in the Field Definition Section.

If the SMTP gateway finds no customized ″SMTP RULES″ file, it will by default use a predefined set of rules whose content depends on the gateway being generated as *SECURE* or non-*SECURE*. The *DefaultFields* of the field definition section of both default sets, defines that the default rules will affect the addresses under the following RFC 822 header fields:

From:
Resent-From:
Reply-To:
Resent-Reply-To:
Return-Path:
Sender:
Resent-Sender:
To:
Cc:
Bcc:

The implementation allows you to perform union, difference and intersection operations on the header fields in the field definition section, to create whatever subset of the header fields you would like under a single *alias* name. It also allows you to use simple or nested *IF-THEN-ELSE* statements in the Rule Definition Section.

Refer to *IBM TCP/IP V2 R2 for VM: Planning and Customization* for more information on how to define the ″SMTP RULES″ file.

Reminder

Mail headers passing from the SMTP network to the local system or RSCS network are not affected.

2.12 Remote Execution (REXEC) - Remote Shell (RSH)

The REXECD server implements the Remote Execution Command protocol (REXEC) and the Remote SHell protocol (RSH). The REXECD virtual machine will receive the *rexec* requests on port 513 and the *rsh* requests on port 514.

The REXEC protocol is used to issue commands on the remote host. The VM-specific feature is the ability to use slave machine(s), which are started by the REXECD virtual machine during its own startup, instead of using the user's virtual machine. This will reduce the time for executing the command since no AUTOLOG command is needed when a slave machine is used. If no slave machines are identified in the file "REXECXIT EXEC", the REXECD server will try to AUTOLOG a virtual machine RSLAVE1.

Slave machine(s) are identified by the following line:

```
 .....
globalv = ¢GLOBALV SELECT TCPRUN¢      /* Readability enhancement.*/
owner = ¢¢                             /* For example, ¢TCPUSR8¢.*/
parms = ¢-R -s VMUSER14 PRTYUIOG¢      /* Optional server arguments.*/
 .....
```

The *-R* means that RACF will be used when a password is needed; that is, when a user's virtual machine is used (a slave machine will be used when the keyword *guest* is entered for both the user ID and the password from the client station).

If the RACF option is not specifed, the workstation will have to specify the password of the VM directory.

PRTYUIOG is VMUSER14's password. On a VM/ESA system we had to remove the password to have REXECD autolog the slave machine.

The slave machine (VMUSER14) must have a link to TCPMAINT.592 and *must* have the same "PROFILE EXEC" as REXECD. Therefore the easiest way to define a slave machine is to define a virtual machine that has a link to REXECD.191 as its own 191.

However the slave machine *must not* have the same "REXECXIT EXEC" that the REXECD virtual machine has. The parms line must have no arguments (the *-R* and *-s* options cannot be processed by the slave machine).

The easiest way to solve this problem is to copy the "REXECXIT EXEC" onto the slave's 191 minidisk and to modify it to have no arguments after the statement parms. You may have a slave machine with no disk in write mode (actually, it does not need one because it may access REXECD.191 as its own 191). Then you will have to modify the common "PROFILE EXEC" on REXECD.191 in order to use another *exit* for the slave machine.

Following is an example of ″PROFILE EXEC″ which assumes that the exit called for the slave is ″SLAVEXIT EXEC″ (numbers on the left are line numbers):

```
00056 /* Modifications are starting right after the first comments  */
00057 /* Another exit (SLAVEXIT) will be called if the userID       */
00058 /* executing this PROFILE EXEC is not REXECD                  */
00059
00060 ¢IDENTIFY (LIFO¢                                  1
00061 Parse upper pull userid .                        2
00062 Select
00063   When userid=¢REXECD¢ then userexit=¢REXECXIT¢   3
00064   Otherwise userexit=¢SLAVEXIT¢                  4
00065 End
00066
00067/* End of modifications */
00068 /*************************************************************/
00069 /* Initialize common variables.
00070 /*************************************************************/
00071 /* userexit = ¢REXECXIT¢    Name of user exit EXEC. */  5
00072 /*                          checked and set before  */
```

1 Find who is executing the ″PROFILE EXEC″ and stack the result.

2 Retrieve the user ID from stack.

3 If the user ID is REXECD the ″PROFILE EXEC″ user exit name is set to REXECXIT.

4 If the user ID is not REXECD the ″PROFILE EXEC″ user exit name is set to SLAVEXIT. This is safe because the only user ID which has access to REXECD.191 as its own 191 is the slave machine.

5 Previous entry commented out.

Please Note

- REXECD needs access to the C/370 libraries. The ″REXECXIT EXEC″ should be customized to provide this access. Also, if you intend either to change the start options of the server or run a different module name of the server, this file should be used as well.
- The requested command must be on an accessible minidisk; that is, the slave machine or a user's virtual machine must be able to find the command on one of its minidisks. In order to use Remote Execution the user must have a valid user ID and password on the system where his command is to run.
- When a user's virtual machine is used to execute commands, this user ID must not be logged on, or in disconnected mode, or the autolog request will fail.
- RACF is supported by the VM REXECD server.

Please refer to Section 2.20, “RACF Considerations” on page 171 for RACF considerations.

2.12.1 How to Use the REXEC Protocol

The following tests were performed from an OS/2 platform.

2.12.1.1 Without RACF

REXEC has been started without any option. Therefore RACF was not used, and REXECD autologged a slave machine with the user ID RSLAVE1.

- To use a private user ID (TCPMAINT for example), you need to specify the password which is coded in the VM directory:

```
[C:TCPIPETC]rexec vmesa -l tcpmaint -p tcpmaint †Q T†
TIME IS 16:24:32 EDT WEDNESDAY 07/08/92
CONNECT= 00:00:06 VIRTCPU= 000:00.16 TOTCPU= 000:00.29
```

- To use the slave machine (RSLAVE1), you need to specify the keyword guest for both the user ID and the password:

```
[C:TCPIPETC]rexec vmesa -l guest -p guest †Q T†
TIME IS 16:24:36 EDT WEDNESDAY 07/08/92
CONNECT= 00:02:06 VIRTCPU= 000:00.16 TOTCPU= 000:00.33
```

2.12.1.2 With RACF

- To use a private user ID (TCPMAINT for example), you need to specify the RACF password:

```
[C:TCPIPETC]rexec vmesa -l tcpmaint -p anchor1 †Q T†
TIME IS 16:30:22 EDT WEDNESDAY 07/08/92
CONNECT= 00:00:06 VIRTCPU= 000:00.16 TOTCPU= 000:00.29
```

- To use the slave machine (RSLAVE1), you need to specify the keyword guest for both the user ID and the password:

```
[C:TCPIPETC]rexec vmesa -l guest -p guest †Q T†
TIME IS 16:30:42 EDT WEDNESDAY 07/08/92
CONNECT= 00:00:06 VIRTCPU= 000:00.16 TOTCPU= 000:00.29
```

2.12.2 How to Use the RSH Protocol

All tests were performed from an OS/2 platform.

2.12.2.1 Without RACF

- To use a private user ID (TCPMAINT) for example, you need to perform the following sequence:

```
[C:TCPIPETC]set USER=TCPMAINT
[C:\TCPIP\ETC]rsh vmesa †Q T†
TIME IS 16:24:32 EDT WEDNESDAY 07/08/92
CONNECT= 00:00:06 VIRTCPU= 000:00.16 TOTCPU= 000:00.29
```

  The TCPMAINT user ID will be autologged with no need for you to specify a password on the client workstation.

  The USER variable can be reset with the command set USER=.

- To use the slave machine (RSLAVE1), you need to perform the following sequence:

```
[C:TCPIPETC]set USER=GUEST
[C:\TCPIP\ETC]rsh vmesa †Q T†
TIME IS 16:24:32 EDT WEDNESDAY 07/08/92
CONNECT= 00:00:06 VIRTCPU= 000:00.16 TOTCPU= 000:00.29
```

2.12.2.2 With RACF

This is not the right way to use the RSH protocol since the RSH protocol is not supposed to let you use a password. However, in a VM environment with RACF installed it is possible to use the RSH protocol. This can be useful when the workstation only supports the RSH protocol (not the REXEC protocol).

- To use a private user ID (TCPMAINT) for example, you need to perform the following sequence:

```
[C:TCPIPETC]set USER=TCPMAINT
[C:\TCPIP\ETC]rsh vmesa -l anchor1 †Q T†
TIME IS 16:24:32 EDT WEDNESDAY 07/08/92
CONNECT= 00:00:06 VIRTCPU= 000:00.16 TOTCPU= 000:00.29
```

  The TCPMAINT user ID will be autologged but you have to specify the RACF password on the client workstation.

- To use the slave machine (RSLAVE1), you need to perform the following sequence:

```
[C:TCPIPETC]set USER=GUEST
[C:\TCPIP\ETC]rsh vmesa †Q T†
TIME IS 16:24:32 EDT WEDNESDAY 07/08/92
CONNECT= 00:00:06 VIRTCPU= 000:00.16 TOTCPU= 000:00.29
```

  The slave machine will be used and you have no need to specify a password.

2.13 Network File System (NFS)

The Network File System (NFS) allows a client system to accesss a CMS minidisk as if it were local to the client. Unlike FTP and TFTP, NFS allows users to execute programs, and to create and edit files.

NFS continues to be available as a separate feature. To improve the NFS server's performance, write operations are more streamlined and modified to use multiple-block ***BLOCKIO***, which is a fast VM Control Program (CP) system service. Additional information about MOUNTS is accessible through the SMSG to NFS interface. The internal trace was improved to give you a more extensive collection of records and trails.

Note: NFS for VM provides server functions only. NFS does not provide CMS users with the functions that would allow a CMS user to reference remote files.

Not all functions that are defined by the NFS protocol are supported by the CMS file system. *Make_Directory*, *Symbolic_Link*, and *Remove-Directory* commands are not supported by the VM NFS server.

There is limited support for the link NFS request. The intent of this support is to accommodate clients that use link as a temporary step during a file rename or similar operation. Link information is maintained in volatile storage by the NFS server, and it is not used when the reply to a *Read_Directory* or *Lookup-File* request is constructed.

The access control processes associated with mount requests have been combined into a single module. Up to V2R1, you had to build the VMNFS module according to the desired access control. In V2R2 this can be done with a command line option.

Please refer to 2.20, “RACF Considerations” on page 171 for RACF considerations. The *PARMS* global variable in the ″VMNFSXIT EXEC″ POSTLUDE subroutine (see Section 2.4.2.3, “The ″exit EXEC″ Facility” on page 71) can be set to one of the following values:

D cause the VMNFS module to write debugging-oriented messages to the console and into the ″VMNFS LOG″ file

G cause the VMNFS module to record all RPC requests received and replies sent into the ″VMNFS LOG″ file

R use RACF access control

A use Autolink access control.

The VMNFS virtual machine (C/370 application) requires access to the IBM C/370 Library during execution.

Installation of VMNFS is rather easy. The only thing you have to take care of is the access to the C libraries. The ″VMNFSXIT EXEC″ may be customized to add a link/access to the C minidisk and to enable VMNFS to receive SMSG requests (if you want to use the SMSG interface).

Following is an example of what may be added to the file ″VMNFSXIT EXEC″ (numbers on the left are line numbers):

```
00117 ¢LINK C370 191 193 RR RC370¢
00118 ¢ACC 193 Z¢
00119 ¢SET MSG ON¢
00120 ¢SET SMSG ON¢
```

Note: Make sure that the C/370 minidisk is not linked with a virtual address in the range 200 - 5FF because VMNFS uses this range.

Warning

1. NFS uses UDP as its protocol, which is not reliable. The loss of an IP datagram may not be noticed by an application program.
2. The 9370 Ethernet LAN adapter is not supported for use with NFS.
3. Be sure that PORTMAP virtual machine is started. This is required since NFS uses RPC. PORTMAP needs an access to the C/370 library.
4. Be aware that all VM limitations apply to the TCP/IP environment. For example multiple write access to the same minidisk is not recommended. This means, for example, that if you access a VM minidisk in write mode, no other VM user ID will be able to link that minidisk in write mode and the writing will fail for the other VM user ID. If another VM user ID already has a link to a VM minidisk before you issue the `mount` command to mount that minidisk, you won′t be able to have write access to that minidisk.

2.13.1 SMSG Interface to NFS

The VMNFS SMSG command allows a CMS user to communicate with the VMNFS virtual machine. This communication is necessary in the following situations.

- You want the VMNFS virtual machine to detach a minidisk. Please note that with IBM TCP/IP Version 2 Release 2 for VM, VMNFS now detaches a CMS

minidisk when the command which unmount a minidisk is issued on the client workstation.

```
smsg vmnfs m detach tcpmaint.191
MSG FROM VMNFS   : M RC=0 from detaching TCPMAINT.191
Ready; T=0.01/0.02 11:28:00

smsg vmnfs m detach tcpmaint.191
MSG FROM VMNFS   : M Disk TCPMAINT.191 is not currently linked.
Ready; T=0.01/0.02 11:25:56
```

- A change is made to a minidisk that is linked read-only by VMNFS and the NFS server must be told to discard any buffered disk blocks, so that the next client requests use of the current disk data.

```
smsg vmnfs m refresh tcpmaint.191
MSG FROM VMNFS   : M Disk TCPMAINT.191 refreshed.
Ready; T=0.01/0.01 11:23:00

smsg vmnfs n refresh tcpmaint.191
Ready; T=0.01/0.01 11:24:30
```

- VMNFS maintains information and usage data about clients. This information may be accessed and displayed with the *SMsg VMNFS Query M | L* command, where:

  M gives information about mounts
  L gives information about linked disks

  The displayed information pertains to all clients as a whole:

```
smsg vmnfs m query

MSG FROM VMNFS   : M VM NFS server start time 15Nov91 11:06:20.

MSG FROM VMNFS   : M   118 RPC (0 duplicate XID),  2 SMSG, 102 *BLOCKIO

MSG FROM VMNFS   : M   0 null, 9 getattr, 0 setattr, 93 lookup, 2 read, 0 write

MSG FROM VMNFS   : M   1 create, 0 remove, 1 rename, 0 link, 2 readdir, 2 statfs

MSG FROM VMNFS   : M   4 mount, 3 mountpw, 0 mountnull, 0 unmount, 0 unsupported

MSG FROM VMNFS   : M End of reply.
Ready; T=0.01/0.04 11:22:19
```

Please refer to the *IBM TCP/IP V2 R2 for VM: Planning and Customization* for more information about the SMSG interface to NFS.

Note: The NFS feature of IBM TCP/IP Version 2 Release 2 for VM provides exit routines to validate the SMSG commands, provides file encryption and processes the results of minidisk *LINK* commands. The ″NFS H″ file contains definitions and parameter lists needed to code these exit routines.

2.13.2 NFS Server Problem Determination

If you get the message Access denied, check that the in-addr.arpa domain has been defined in your Name server when the NFS server uses a Name Server. Please refer to 2.18, “Domain Name Server” on page 142 for more informations.

The NFS server has several features for problem determination. If the optional parameter **g** is specified when the *VMNFS* command is invoked, the ″VMNFS

LOG A″ file is created. This file contains the calls and responses processed by NFS.

The ***d*** parameter writes messages to the console and the log file.

An internal trace table is maintained and updated in response to calls to the ***trace*** or ***tracev*** functions.

The SMSG command TWRITE will write the internal trace table to a disk file. The TVPRINT utility program decodes some of this file′s data into a human readable format. Refer to the *IBM TCP/IP V2 R2 for VM: Planning and Customization* manual to see more about these debugging features.

2.14 Remote Printing (LPD)

The LPR client establishes a TCP connection to an LPD server in order to send the file to be printed to that server. The LPD server then looks for the printer named in the LPR command and tries to print out the document. The printer itself is not dedicated to TCP/IP, it may receive output from other sources (via RSCS) too. The LPD server in IBM TCP/IP Version 2 Release 2 for VM offers some nice features beyond the usual implementation. Three types of line printer connections are supported:

- Local printers
- Remote RSCS printers (RSCS connected)
- Remote TCP/IP printers (TCP/IP connected).

The local printers are physically attached to the VM system. Any printer available through the RSCS network can be accessed, with the remote RSCS printer support. This printer could be attached to an MVS system without a TCP/IP connection. Any printer controlled by another LPD server can be made available to a VM TCP/IP user.

Beyond the print support it is also possible to send PUNCH records. This allows, for example, sending Job Control (JCL) to an MVS system, which is connected to the VM system via NJE and RSCS.

2.14.1 The Profile Exit for LPSERVE

The LPD server is invoked by the ″TCPRUN EXEC″ which resides on the common server disk (TCPMAINT 591). TCPRUN is invoked by the ″PROFILE EXEC″ on the server′s 191 minidisk. TCPRUN starts the LPD server using default values. However you can specify different startup parameters by modifying the global variable *PARMS* in the ″LPDEXIT EXEC″ procedure. Please refer to the *IBM TCP/IP V2 R2 for VM: Planning and Customization* for more information on how to customize the ″LPDEXIT EXEC″.

2.14.2 Configuring LPD

The tailoring of LPD consists of preparing the configuration file ″LPD CONFIG″. A sample on the LPSERVE 191 minidisk is provided. It should be modified to meet your requirements. The basic structure of this file is as follows:

- SERVICE printer_name
  - type (LOCAL, REMOTE, RSCS)
  - options

SERVICE defines the printer name, which will later be used in the ***LPR*** command to print a file via TCP/IP. The following statements define the actual printer, where it is located and its characteristics.

The following printer configuration file shows some example definitions.

File "LPD CONFIG"

```
SERVICE LOCPRT PRINTER                  1
   LOCAL
   FILTERS f l p
   LINESIZE 132
   PAGESIZE 66
;
SERVICE PRT3287 PRINTER                 2
   RSCS SPOOL=TO RSCSV2
   TAG=PRT3287
   FILTERS f l p
   LINESIZE 132
   PAGESIZE 66
OBEY TCPMAINT                           3
DEBUG
;
SERVICE ROLF PRINTER                    4
  REMOTE LPT1@ROLF
```

1. A locally attached printer named LOCPRT is defined with these statements. Specific printer characteristics are set using the LINESIZE and PAGESIZE options.
2. A remote printer named PRT3287, accessible via RSCS, is defined here. The options are used by LPD to direct the output to a specific printer using the **cp tag** command **CP TAG DEV PRT PRT3287** and the **cp spool** command **CP SPOOL PRT TO RSCSV2**.
3. The OBEY statement tells LPD who can use the SMSG interface to LPD.
4. The remote printer ROLF is actually controlled by another LPD server. On the remote system (OS/2 system ROLF) the printer is known as LPT1.

Note: The configuration files for the ITSC environment can be found in Appendix C.5, "File "LPD CONFIG" on LPSERVE 191" on page 280 and Appendix D.4, "File "LPD CONFIG" on LPSERVE 191" on page 285. The configuration files include the definitions for the printer and punch devices in Figure 8 on page 16.

Warning

- You must define the name of the VM node where LPD is running either in the "HOST TABLES" or to the name server, because the LPD server tries to find its own address using this name. For instance, on system VM14 we needed to have an entry called RAL9390 on the name server. You can find your system node name by issuing the CMS IDENTIFY command.
- The *IBM TCP/IP V2 R2 for VM: Planning and Customization* manual mentions some more parameters for LPSERVE. We have been unable to make them work. Those parameters are:
  - RACF
  - EXIT: no usable parameter are passed to the EXEC.
  - FAILEDJOD MAIL: no mail was sent.

2.14.3 The LPR Command

Four commands are available in order to set up or use the client printing facility on VM:

- LPRSET: allows you to define the remote print server and the printer that will be used. This setup can be permanent or for this session only.
- LPRM: allows you to cancel a job on a remote print server.
- LPQ: allows you to list the printer queue on a remote printer.
- LPR: allows you to print queue on a remote printer. Many parameters have been added with IBM TCP/IP Version 2 Release 2 for VM. Please refer to *IBM TCP/IP V2 R2 for VM: User's Guide* for more information.

2.15 SNA Connections (SNALINK)

The purpose of the SNALINK virtual machine is to use an existing SNA backbone to route TCP/IP packets between VM and MVS systems. Only virtual machines under the control of the GCS virtual operating system can issue VTAM macros. Therefore, a separate virtual machine SNALINK is needed. The SNALINK disconnected machine runs a VTAM application program, which communicates with another SNALINK application on the remote side. The TCPIP virtual machine will communicate with SNALINK using IUCV. SNALINK uses SNA LU type 0 sessions.

A new module has been shipped with TCP/IP V2R2 for VM which allows SNALINK to operate either in dual mode, or in single mode (SessionType statement in the ″PROFILE GCS″).

- When operating in dual mode (mode used prior to TCP/IP V2R2), SNALINK opens two SNA sessions for each remote LU with which it communicates, one for sending and one for receiving.
- When operating in single mode, SNALINK opens one full-duplex session. This allows the support of the 3745 Ethernet adapter.

Some new SNALINK functions are available as well:

- Delay retry (retry statement in the ″PROFILE GCS″) for VTAM sense code 0857003, 080A000, 087D0001.
- MAX SNA sessions has been changed (was 6 in the previous release). This is now a user option (MaxSession statement in the ″PROFILE GCS″).
- Either side of the SNA session can start the connection.
- Retry OPEN ACB at startup time.

An SNALINK connection is a point-to-point type of connection; that means it differs from token-ring, Ethernet or X.25. LAN and X.25 connections allow communication with multiple partners on one physical interface. The routing is based on the network part of the internet address. A gateway connected to two LANs needs two internet addresses, because the address comes from the connection to the network itself and not from the host. Each additional LAN connection needs an additional internet address on a different network or subnet. This is also recommended for an SNALINK connection, but it is not a requirement. The SNALINK home address may have the same home address as the LAN address. The routing for a point-to-point (SNALINK) connection is based on the complete internet address.

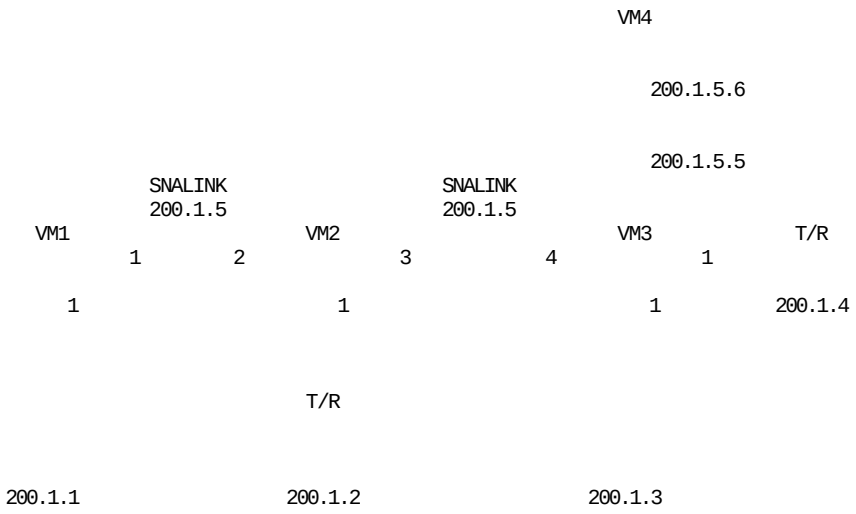


Figure 28. SNALINK Connections

In Figure 28, the systems VM1 - VM3 have at least one LAN connection and one or more SNALINK connections. An additional network, or subnet, for each of the SNALINK connections is not required; this saves IP addresses if your network becomes larger and larger.

Note: Both VM and MVS TCP/IP Version 1 do not allow you to define a single host in their routing definitions. If VM3 runs TCP/IP Version 1, the address of host VM4 must have a different network part.

Warning

If ROUTED controls a link (including an SNALINK) it assumes that each link has a unique IP address. Otherwise it will not work correctly.

2.15.1 Installing SNALINK

To install the SNALNKA virtual machine you must do the following (please also refer to *IBM TCP/IP V2 R2 for VM: Planning and Customization*):

1. Define the SNALNKA virtual machine.
   - In the VM directory, MAXCONN must be greater than MaxSession.
   - Virtual machine storage should be a minimum of 2MB. CP class B is required. Add 130KB per connection.
2. Give SNALNKA access to GCS.
   - In the GCS configuration file you must define

     ```
     AUTHUSER NAME=SNALINK_userId
     ```

3. Define the SNALINK application in the VM/VTAM files.
   - SNALINK provides its own BIND parameters, so it does not assume or require any particular LOGMODE entries.
   - EAS value should be two times the number of MaxSession as coded in ″PROFILE GCS″.
   - Code AUTHEXIT=YES.

4. Customize the ″PROFILE GCS″ of SNALNKA.
   - Use the largest MaxRuCode you can (maximum is 32KB). Be sure both VTAM (MAXBFRU) and NCP (MAXDATA, MAXBFRU, TRANSFR, BFRS) can handle it.
5. Update the ″PROFILE TCPIP″. Please refer to Section 2.15.2, “Configuring SNALINK” for an example.

2.15.2 Configuring SNALINK

The following types of definitions are required:

- **Device and Link Definitions:** The file ″PROFILE TCPIP″ defines and names these resources at startup time. The remote LU name is referenced in the DEVICE statement and it has to match the VTAM definitions on the remote side.
- **Routing Entries and Home Address:** The newly defined link now needs an internet address (HOME statement) and a description of what can be reached via this link (routing entries).
- **VTAM Definitions:** SNALINK is a VTAM application, so there is an application major node required, defining the SNALINK application to VTAM. The name of this application has to match the local LU name used in the file ″PROFILE GCS″ for the SNALINK virtual machine.

Two IBM TCP/IP Version 2 Release 2 for VM systems are interconnected via an SNALINK connection. Both hosts define each other as a host only. An alternative could be to define the remote host as a gateway to a remote network.

```
            System VM14                                    System VM15

SNALINK virtual machine                                    SNALINK virtual machine
  PROFILE GCS                                                PROFILE GCS

   LocalLuName = ¢AD114TC1¢                    LocalLuName = ¢AD115TC1¢
   ...                                         ...


VTAM virtual machine                                       VTAM virtual machine
  AD114MTC VTAMLST                                           AD115MTC VTAMLST

   AD114TC1 APPL ACBNAME=AD114TC1              AD115TC1 APPL ACBNAME=AD115TC1
   ...                                         ...


TCPIP virtual machine                                      TCPIP virtual machine
  PROFILE TCPIP                                              PROFILE TCPIP


     DEVICE SNADVM15 SNAIUCV SNALINK AD115TC1 SNALNKA  DEVICE SNADVM14 SNAIUCV SNALINK AD114TC1 SNALNKA
     LINK   SNALVM15    IUCV  2 SNADVM15               LINK   SNALVM14    IUCV  2 SNADVM14


     HOME                                              HOME
     9.67.32.49   SNALVM15                               9.67.32.50   SNALVM14

     BSDROUTINGPARMS FALSE  (or GATEWAY stmt.)         BSDROUTINGPARMS FALSE  (or GATEWAY stmt.)
      SNALVM15 2000 0 255.255.255.255 9.67.32.50          SNALVM14 2000 0 255.255.255.255 9.67.32.49
      ....                                                ....
     ENDBSDROUTINGPARMS                                ENDBSDROUTINGPARMS

     START SNADVM15                                    START SNADVM14
```

Figure 29. SNALINK Cross Reference

2.16 The Network Database System (NDB)

Preliminary notice

- The following description assumes that you are familiar with the SQL/DS product, especially with the information contained in:

  SQL/DS: Database Planning and Administration for VM/SP and VM/ESA - SH09-8017

  SQL/DS: Database Services Utility for VM/SP and VM/ESA - SH09-8051

  SQL/DS: Application Programming for VM/SP and VM/ESA - SH09-8019.

- All the tests on the ITSC Raleigh network were done using SQL/DS V2R2 with PUT 9105 installed.
- Please be aware that the required C Compiler is: **IBM C/370 Compiler Version 1 Release 2, Licensed Program (5688-040).**

To configure an NDB client/server environment, two different sets of tasks have to be completed: one on the host side (NDB server), and the other on the workstation side (NDB client).

2.16.1 Installing the NDB Server before APAR PN17869 on NDB

To install the NDB server, you have to:

1. Install the NDBSERVE user ID which will run the NDB server application in disconnected mode.

   This should have been taken care of during the preinstallation of IBM TCP/IP Version 2 Release 2 for VM. Please refer to Section 2.2, “Preinstallation” on page 66 for more information on how to create the TCP/IP servers′ user IDs.

2. Customize the NDBSERVE user ID to reflect your current layout.

   The NDB server is invoked by the ″TCPRUN EXEC″ which resides on the common server disk (TCPMAINT 591). TCPRUN is invoked by the ″PROFILE EXEC″ on the server′s 191 minidisk. TCPRUN starts the NDB server using default values. However you can specify different startup parameters by modifying the global variable *PARMS* in the ″NDBSEXIT EXEC″ procedure. Please refer to the *IBM TCP/IP V2 R2 for VM: Planning and Customization* for more information on how to customize the ″NDBSEXIT EXEC″.

   Make sure that the PORTMAP virtual machine is started before NDBSERVE.

3. Register the NDBSERVE user ID to the SQL/DS database machine and load the default or newly created access module DBUTIL2 into the SQL/DS database.

   A pre-compiled access module ″DBUTIL2 ACC_OUT″ is delivered as part of the NDB feature. Since it has been compiled in a CMS user ID named TCPBUILD, and since this information is hardcoded in the access module, it can be stored in the database only by a CMS user ID which has the same name.

   Therefore there are two options for this step:

   - Use the delivered DBUTIL2 access module:

To be able to do this, you have to create a user ID named TCPBUILD on your VM system (if you do not already have one). This option is described in Section 2.16.1.2, “Working with the TCPBUILD User ID” on page 122.

- Work with the NDBSERVE user ID:

  In that case you will have to recompile the DBUTIL2 access module using the C/370 compiler, before storing it in the SQL/DS database. This option is described in Section 2.16.1.1, “Working with the NDBSERVE User ID.”

2.16.1.1 Working with the NDBSERVE User ID

1. When working with the NDBSERVE user ID, you first have to install the C/370 compiler and preprocess and compile the ″DBUTIL2 SQC″ program, using the following command:

   TCPOBJCT DBUTIL2

2. The following files are output by the **TCPOBJCT** command:

   DBUTIL2 C the source code of the preprocessed program

   DBUTIL2 TEXT the object code of the preprocessed program

   DBUTIL2 ACC_OUT the access module

   Note: The user ID in which this compilation takes place will be hardcoded in the ″DBUTIL2 ACC_OUT″ database access module.

3. Logon as a SQL user who has the DBA authority (that is, the authorization to perform database administration functions) and issue the following commands (or ask your site database administrator):

   - GRANT CONNECT TO NDBSERVE

     To grant the authorization to access the SQL/DS database, to the NDBSERVE user ID.

   - GRANT ALL ON *table* TO NDBSERVE

     Where *table* specifies the tables in an existing database that you want the remote NDB clients to access through the NDB server. This grants all the privileges on the specified tables to the NDBSERVE user ID.

   - GRANT DBA TO NDBSERVE

     To grant the database administration authority to the NDBSERVE user ID, which is necessary for it to be able to store the access module in the database (step 4 below) and which is required during its operation, to enable it to connect the VM user ID entered by the NDB client to the database.

4. Logon to NDBSERVE and issue the following commands:

   - SQLINIT DBNAME(SQLDBA)

     To register the NDBSERVE user ID to the database machine (SQLDBA).

Important

When SQL/DS is not installed in a DCSS you will get an ARIMSG error. To avoid this, issue the following commands:

ACCESS 195 Q

COPYFILE SQLUSER PROFILE Q SQLUSER EXEC A

EXEC SQLUSER

- NDBINIT NDBSERVE

  To store the access module in the SQL/DS database.

  Note: ″NDBINIT EXEC″ assumes that ″DBUTIL2 ACC_OUT″ is on your A minidisk. Modify the FILEDEF entry if needed to change the filemode.

5. Logon as a SQL user who has the DBA authority (that is, the authorization to perform database administration functions) and issue the following command (or ask your site database administrator):

   - GRANT RUN ON NDBSERVE.DBUTIL2 TO PUBLIC

     To grant the privilege of running the DBUTIL2 program, to all the users.

Now the SQL/DS database is ready to accept SQL query statements sent by remote NDB client workstations through the NDBSERVE interface.

2.16.1.2 Working with the TCPBUILD User ID

As said before, when you compile the ″DBUTIL2 SQC″ source, the user ID from which the compilation took place is hardcoded in the ″DBUTIL2 ACC_OUT″ database access module. Therefore the difference between this option and the previous one resides in the necessity of storing the delivered access module using the CMS user ID in which it was compiled.

When working with the TCPBUILD user ID the following steps are necessary to set up the access module:

1. Logon as a SQL user who has the DBA authority (that is, the authorization to perform database administration functions) and issue the following commands (or ask your site database administrator):

   - GRANT CONNECT TO NDBSERVE

     To grant the authorization to access the SQL/DS database, to the NDBSERVE user ID.

   - GRANT ALL ON *table* TO NDBSERVE

     Where *table* specifies the tables in an existing database that you want the remote NDB clients to access through the NDB server. This grants all the privileges on the specified tables to the NDBSERVE user ID.

   - GRANT DBA TO TCPBUILD

     To grant the database administration authority to the TCPBUILD user ID, which is necessary for it to be able to store the access module in the database (step 4 below).

   - GRANT DBA TO NDBSERVE

     To grant the database administration authority to the NDBSERVE user ID, which is required to enable it to connect the VM user ID entered by the NDB client to the database.

2. Logon to TCPBUILD and issue the following commands:
   - SQLINIT DBNAME(SQLDBA)

     To register the TCPBUILD user ID to the database machine (SQLDBA).

     Important

     When SQL/DS is not installed in a DCSS you will get an ARIMSG error. To avoid this, issue the following commands:

     ACCESS 195 Q

     COPYFILE SQLUSER PROFILE Q SQLUSER EXEC A

     EXEC SQLUSER

   - NDBINIT TCPBUILD

     To store the access module in the SQL/DS database.

     Note: ″NDBINIT EXEC″ assumes that ″DBUTIL2 ACC_OUT″ is on your A minidisk. Modify the FILEDEF entry if needed to change the filemode.

3. Logon as a SQL user who has the DBA authority (that is, the authorization to perform database administration functions) and issue the following commands (or ask your site database administrator):
   - GRANT RUN ON TCPBUILD.DBUTIL2 TO PUBLIC

     To grant the privilege of running the DBUTIL2 program, to all the users.

   - REVOKE DBA FROM TCPBUILD

     To revoke the database administration authority from TCPBUILD, since it is not needed any more.

4. Logon to NDBSERVE and issue the following command:
   - SQLINIT DBNAME(SQLDBA)

     To register the NDBSERVE user ID to the database machine (SQLDBA).

     Important

     When SQL/DS is not installed in a DCSS you will get an ARIMSG error. To avoid this, issue the following commands:

     ACCESS 195 Q

     COPYFILE SQLUSER PROFILE Q SQLUSER EXEC A

     EXEC SQLUSER

Now the SQL/DS database is ready to accept SQL query statements sent by remote NDB client workstations through the NDBSERVE interface.

2.16.2 Installing the NDB Server after APAR PN17869 on NDB

Please refer to Section 1.3.8.3, “Enhancements in IBM TCP/IP Version 2 Release 2 for VM with APAR PN17869 on NDB Installed” on page 39 for an enhancements overview.

Warning

The code we installed and tested at the ITSC Raleigh was pre-packaging code which included some debugging information which may not appear in the code as distributed in the PTF for APAR PN17869 on NDB. The following shows the ″PROFILE EXEC″ for the PORTSRV virtual machine and for the NDB server machines that we actually created for our tests. As the ones which will be delivered in the PTF may differ, please refer to the appropriate documentation at installation time.

″PROFILE EXEC″ for the PORTSRV virtual machine (the PORTSRV virtual machine runs the PORTSRVS module):

```
 .....
¢ACCESS 591 B  ¢ /* TCPMAINT mdisk */
¢ACCESS 592 C  ¢ /* TCPMAINT mdisk */
 .......
¢GLOBAL LOADLIB EDCLINK IBMLIB¢  /* Load the LOADLIBs */
¢GLOBAL TXTLIB IBMLIB EDCBASE¢   /* Load the TXTLIBs */
¢SET LDRTBLS 25¢               /* More Loader tables than the default */
If linesize() =0 Then Do      /* if logged in */
  Say ¢¢
  Say ¢>>>>>>>>>>> To start the Server type: PORTSRVS              ¢
  Say ¢¢
  Exit
End
Else ¢PORTSRVS¢        /* If DSC that is, AUTOLOGed */
Exit 0
```

"PROFILE EXEC" for the NDBSRV1 virtual machine (the NDB server machine runs the PORTCLNT module):

```
 .....
¢ACCESS 591 B  ¢ /* TCPMAINT mdisk */
¢ACCESS 592 C  ¢ /* TCPMAINT mdisk */
 .......
¢GLOBAL LOADLIB EDCLINK IBMLIB¢
¢GLOBAL TXTLIB IBMLIB EDCBASE¢
¢SET LDRTBLS 25¢
 .....
¢CP LINK SQLDBA 195 195 RR¢    /* Link the SQL mdisk */
¢SQLUSER¢                  /* For the language and to avoid ARIxxx */
 ........
If linesize() =0 Then Do      /* If not DSC */
  Say ¢¢
  Say ¢>>>>>>>>>>> To start the Server type: NDBSTART              ¢
  Say ¢¢
  Exit
End
Else do             /* If DSC that is, AUTOLOGed */
     ¢IDENTIFY (LIFO¢
     Parse upper pull me ¢AT¢ node .
¢PORTCLNT¢ node me ¢NDBSRV¢
End
Exit 0
```

"NDBSTART EXEC" for the NDB server. This EXEC is used to manually start the NDB server:

```
/*  */
¢IDENTIFY (LIFO¢
Parse upper pull me ¢AT¢ node .
¢PORTCLNT¢ node me ¢ndbsrv¢
Exit rc
```

As far as SQL is concerned the steps described in Section 2.16.1.1, "Working with the NDBSERVE User ID" on page 121 and in Section 2.16.1.2, "Working with the TCPBUILD User ID" on page 122 are still valid. The "DBUTIL2 ACC_OUT" module we received had been created from the TCPMAINT user ID. You can check which user ID was used to create the "DBUTIL2 ACC_OUT" module you received, by browsing the module: on the first line you will see the user ID which was used at creation time.

Once the SQL authorities had been granted as stated in the sections mentioned earlier, the module was loaded into the SQL database and authorization was given to all users to run the DBUTIL2 program.

The following is the sequence which was performed (please refer to Section 2.16.1.1, “Working with the NDBSERVE User ID” on page 121 for the first steps):

1. NDBINIT TCPMAINT

   To store the module in the SQL/DS database. The following is a part of the output. Highlighted are the messages you should consider to check if everything is okay:

```
ndbinit tcpmaint
 ...........
ARI0870I ENTER: COMMAND TERMINATED BY SEMICOLON OR EXIT TO END
------  RELOAD PROGRAM(TCPMAINT.DBUTIL2)REPLACEKEEPINFILE(DBSFILE);
ARI0852I RELOAD PROGRAM PROCESSING STARTED.
ARI0868I DNAME=DBSFILE RECFM=FB RECSZ=80 BLKSIZE=80
ARI8011I PROGRAM TCPMAINT.DBUTIL2 WAS REPLACED
ARI0853I RELOAD TCPMAINT.DBUTIL2 SUCCESSFUL.

ARI0870I ENTER: COMMAND TERMINATED BY SEMICOLON OR EXIT TO END
------  EXIT;
ARI8997I ...BEGIN COMMIT WORK PROCESSING.
ARI0811I ...COMMIT OF ANY DATABASE CHANGES SUCCESSFUL.
ARI0809I ...NO ERROR(S) OCCURRED DURING COMMAND PROCESSING.
ARI0808I DBS PROCESSING COMPLETED: 07/04/92 19:42:20.
ARI0660I LINE-EDIT SYMBOLS RESTORED:
         LINEND=# LINEDEL=¢ CHARDEL=@ ESCAPE=† TABCHAR=ON
ARI0796I END SQLDBSU EXEC: 07/04/92 19:42:20 EDT
```

2. Using ISQL you can check that the module is stored in the SQL/DS database via the command `SELECT * FROM SYSTEM.SYSPROGAUTH WHERE CREATOR=¢TCPMAINT¢`. The following is the output:

```
GRANTOR   GRANTEE   CREATOR   PROGNAME           TIMESTAMP     RUNAUTH
--------  --------  --------  -----------------  ------------  -------
TCPMAINT  TCPMAINT  TCPMAINT  DBUTIL2            B14WB6NQ3LBL  G
* END OF RESULT *** 1 ROWS DISPLAYED ***COST ESTIMATE IS 1***************
```

3. From a user who has the DBA authority, you have to grant the privilege of running the DBUTIL2 program to all the users. This is achieved via the command `GRANT RUN ON TCPMAINT.DBUTIL2 TO PUBLIC`. The following is the output of the command `SELECT * FROM SYSTEM.SYSPROGAUTH WHERE CREATOR=¢TCPMAINT¢` once the privilege has been granted:

```
GRANTOR   GRANTEE   CREATOR   PROGNAME           TIMESTAMP     RUNAUTH
--------  --------  --------  -----------------  ------------  -------
TCPMAINT  PUBLIC    TCPMAINT  DBUTIL2            B18BPP2FN8W2  Y
TCPMAINT  TCPMAINT  TCPMAINT  DBUTIL2            B14WB6NQ3LBL  G
* END OF RESULT *** 2 ROWS DISPLAYED ***COST ESTIMATE IS 1**************
```

Now the SQL/DS database is ready to accept SQL query statements sent by remote NDB client workstations.

Once all the requested virtual machines are started you can check if their startup is complete with the command `rpcinfo`. Since all these applications are RPC-based, their port information can be retrieved using this command.

The following is the output you will get when PORTMAP, PORTSRV and one NDB server machine are started:

```
rpcinfo -p vmesa
   program vers proto   port
    100000    2   udp    111        PORTMAP
    100000    2   tcp    111        PORTMAP
 536870992    1   udp   1081        PORTSRVS
 536870992    1   tcp   1030        PORTSRVS
 536870944    1   udp   1099        NDBSERVE
 536870944    1   tcp   1037        NDBSERVE
Ready;

rpcinfo -n 1099 -u vmesa 536870944
program 536870944 version 1 ready and waiting
Ready;
```

The *Name* field is a comment we have added for a better understanding. It is not actually part of the *rpcinfo* output.

Note

Even with APAR PN17869 on NDB installed, you can still use a single NDB server machine.

Please refer to Chapter 7, “Using TCP/IP between VM and UNIX/AIX” on page 217 for more information about the use of the NDB feature.

2.16.3 Installing the NDB Client

To install the NDB client program (no matter if APAR PN17869 on NDB has been installed on the host or not), you have to:

1. Download the NDB client source files to the Sun Microsystems or RISC System/6000 workstation on which you intend to use it.

   The source files reside on the client common disk (TCPMAINT 592):

   - For a RISC System/6000:
     - NDBCI C
     - NDBCANCR C
     - NDBCLNT C
     - NDB_CLNT C
     - NDB_XDR C
     - NDB H
   - For a SUN platform:
     - NDBCI C
     - SDBCANCR C to be renamed to ndbcancr.c during the file transfer
     - NDBCLNT C
     - SDB_CLNT C to be renamed to ndb_clnt.c during the file transfer
     - SDB_XDR C to be renamed to ndb_xdr.c during the file transfer
     - NDB H

Warnings

- Transferring the client modules may be done using FTP. Do not use the binary option since the client code in not shipped in binary format. This means that a translation from EBCDIC to ASCII is needed, and it will be performed on the VM platform. Be sure you use the default translation table shipped with IBM TCP/IP Version 2 Release 2 for VM. Do not use any of the national translation tables.
- File names should be translated to lowercase during the file transfer.

2. Compile the source and build the NDB client module.

   All source programs for the NDB client are written in ″C″. Remember that you can use the NDB client command *ndbclnt*:

   - Interactively on the command line
   - Imbedded in a ″C″ application program

   The syntax of the cc command to use NDB interactively is:

   ```
   cc -o modname ndbci.c ndbcancr.c ndb_clnt.c ndb_xdr.c
   ```

   Where *modname* is the name of the resulting module (ndbclnt for example).

   Please refer to *IBM TCP/IP V2 R2 for VM: Planning and Customization* for the syntax of the cc command to use NDB in a C application program.

2.16.4 Usage Notes

When you issue the *ndbclnt* command at the client workstation, the NDB server will first prompt you for a VM user ID and a VM password. You have to enter a user ID/password pair defined in your VM system′s directory. This is because in the NDB implementation, when a SQL query is sent by an NDB client, NDBSERVE actually lets the SQL/DS database system believe that the access is requested by a CMS user ID, which impersonates the NDB client. Therefore all the restrictions which apply in a standard VM SQL/DS environment, when a SQL user requires access to a database, apply here also. For example, two NDB clients will not be allowed to use the same VM user ID/password concurrently.

Note

RACF is not supported by the present implementation of the NDB feature.

When you accept the default output file, the data the NDB server sends back in response to your SQL queries is written into a flat file called ″NDB OUTPUT″ on the client workstation.

A positive return code during your session with the NDB server is not an error. *RC=25* merely means that there is more data to be sent to you. Enter *continue* to receive it or *end* to terminate the session.

2.17 X Window System

2.17.1 Overview

The X Window System is a portable network-based graphical windowing system that was developed at the Massachusetts Institute of Technology (MIT) in 1984. Often known simply as X, it has developed through several releases since its inception, the latest being X Version 11 Release 5. It is now accepted as the default industry standard for a windowing system and is implemented by a number of platform vendors including IBM*, DEC**, Hewlett-Packard**, Sun** and At&T**.

The reasons for the wide acceptance of X by the industry and the important role it is playing in the development of applications include:

- X is a network-transparent graphical user interface (GUI). Network transparency means that X applications can run on one host and direct their output to a display either connected to the same host, or to another host. To the user, an X workstation looks like it is connected to many different hosts at the same time.
- X applications are independent of vendor and device. Since the application only needs to communicate with the X interface it does not need to know the details of any particular workstation display′s hardware or operating system. As long as an X application is able to establish a connection to the workstation, it can use all the capabilities of the base windowing system on that workstation. The workstation′s hardware and system software is hidden by the X protocol, which means an application running on one vendor′s platform can use another vendor′s workstation that supports X.
- As a GUI, X provides some advantages to the end user:
  - Easy to learn
  - Easy to use
  - Intuitive operation

The implementation of X consists of two components and this provides the independence of the applications from the devices and the hardware. They are:

- The X client
- The X server

The X server is a program that runs on a workstation that controls a display, a keyboard and a mouse. The application is called the X client. This model is probably not what you would first expect when you consider other well-known client/server relationships such as TELNET or FTP. However, it does make sense when you understand the relationship between the X application and the workstation. Remember that every time the application wants to use the screen for output, or the keyboard for input, it has to ask the X server.

The X client can make requests of any X server on the network for display, mouse, or keyboard services. The host on which the application runs need only operate with an X Window System that supports the X client service. Similarly, the workstation need only provide the X server support. It is not necessary for every platform participating in an X Window network to provide both server and client support.

Figure 30 on page 130 illustrates the components of an X Window System.

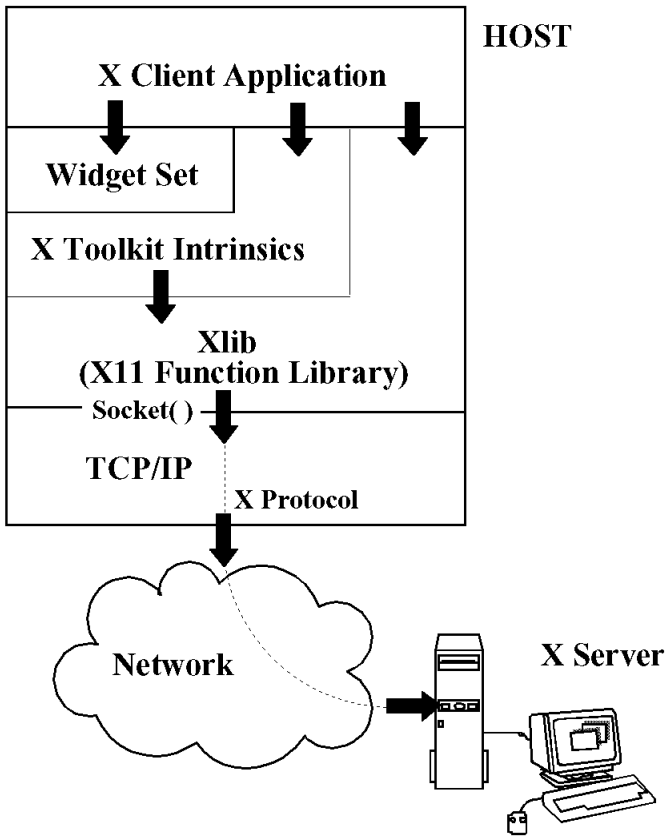


Figure 30. The Components of an X Window System

2.17.1.1 Client

The client is the actual application and is designed to employ a graphical user interface to display its output. It is usually written by the end user but may be included as part of the X Window System software.

In order to support an X client application a platform needs to provide the appropriate function libraries that allow the application to communicate across the network to the X server. When the application is written using the C programming language, the function library is called Xlib. It provides nearly 300 functions that can be called by the application to perform tasks that can include:

- Create, move, resize, stack and delete windows
- Draw lines rectangles, arcs and polygons
- Employ fonts, color maps, graphic images and cursors
- Perform a wide variety of other functions

A graphic display is not required on the client host because the client can request this of the X server. In addition, the client application does not need to be concerned with the devices provided by the server. For example, since the X Server handles the screen, the client does not need to know about the specifications of the output screen. The same application can send output to either a high resolution color screen or to a small black and white screen.

X Tool Kit: In theory Xlib is the only library that is required to run an X application. Xlib provides the base or primitive blocks required to create the X client application. However, it is quite a complex task and may require many lines of code to produce even the simplest of application windows using Xlib alone.

In order to make X client application programming much easier, a set of high-level, preprogrammed functions known as an X toolkit, or Xt is provided on most X client platforms. One Xt function usually translates into several Xlib calls and there are hundreds of standard Xt functions that are available. Xt functions are also called ′intrinsics′.

Widgets: In creating an X client application, the programmer must create and plan to manage user interface objects that the user at the workstation will manipulate to drive the application. These objects are referred to as widgets.

It is appropriate to consider widgets as a set of procedures or functions that are at a level higher than those provided with the X toolkit. Widgets are comprised of Xt functions. Examples of widgets include:

- Push buttons
- Scroll bars
- Text boxes
- Pull-down menus
- Dialog boxes

Usually an X toolkit will include a few of the more common widgets that might be used in an X client application. However it is more likely that the application developer will purchase a Widget Set separately from a vendor. Widget sets are developed by vendors as collections of preprogrammed graphics objects that are common to many X client applications. Examples of available widget sets are:

- The Athena(*) Widget Set from MIT
- The OSF/Motif(**) Widget Set from the Open Software Foundation (OSF).
- The Open Look(*) Widget Set from AT&T

Although the widgets in these sets perform the same base functions, the difference between the sets is the *look and feel.*

2.17.1.2 Server

The X server is the composite partner of the X client in an X Window network. The server is the program on the workstation that controls the screen, and handles the keyboard and the mouse (or other input devices) for one or more X clients. The X server is responsible for output to the display, the mapping of colors, the loading of fonts and the keyboard mapping.

The X server workstation must provide a bit-mapped graphic display terminal which allows each individual picture element (or pixel) on the screen to be accessed and used to display a specific color or shade of grey. The pixels are the elements that are used to construct a graphic image such as a window on the screen.

Generally, the results of the application output are displayed on the system where the client is running. In this case the client and the server reside on the one platform. Often however, you may wish to run the X client application on a remote machine that is better suited to the task, and just display the output at

your local workstation. The application on the remote machine would be the client and your local workstation would be the server.

The client identifies the server by using an X client display variable. It has the syntax:

HOST:SERVER.SCREEN

where:

HOST specifies the host, or network node on which runs the X server system. For UNIX systems (**), when *HOST*=unix, the output is directed to the local console.

SERVER specifies the server or display number (starting with 0) which represents the resources controlled by the one server program. The term server and display can be used interchangeably when discussing X Windows. A display may be composed of multiple screens, but the screens share only one keyboard and pointer. Since most workstations have only one keyboard and one pointer, they are classified as having only one display. If a host has several displays, each is assigned a number (beginning with 0) when the X server for that display is started.

SCREEN specifies the screen number, where a screen is a physical terminal. If there is only one physical screen this is omitted. It is assumed to be 0.

Typically X server programs run on personal computers, high performance UNIX workstations or special X terminals which may have the server code downloaded from the another host or stored in read-only memory (ROM).

Note

The X server is not part of IBM TCP/IP Version 2 Release 2 for VM.

Window Manager: The window manager is an important part of the X server. It is the basis for providing a graphical interface to the user that is controlled by a point and click device such as a mouse.

A window manager allows users to manipulate the application windows at the X server display. Functions supported include:

- Move windows
- Resize windows
- Move the input focus from one window to the next
- Alter the stacking of the windows
- Iconization of windows

2.17.2 Implementation

The IBM IBM TCP/IP Version 2 Release 2 for VM provides an X Windows client service. This support is automatically shipped with the product. This means that VM can support an application, known as the X client, that can communicate with a display provided by an X Window server or X server. The X client application runs in a CMS virtual machine communicates with the X server using the X protocol across a TCP/IP network.

The X server provides access to resources such as a screen, a keyboard, a mouse, fonts, and graphics. It accepts requests from the X client application and sends user input back to the X client. Each X client can make requests of multiple X servers and each X server can service multiple X clients. Examples of X Window server platforms that would interoperate with an VM X client application might be OS/2, AIX(*) or DOS.

IBM TCP/IP Version 2 Release 2 for VM provides two methods for supporting X client applications under VM:

1. X Window System GDDM(*) support
2. X Window System API for user written applications.

2.17.2.1 X Window System GDDM Support

The X Window System GDDM interface supports graphics display output from the IBM Graphical Data Display Manager (GDDM) to be displayed at an X server station.

When the X Window System GDDM interface is activated it translates the data stream created by GDDM into the X protocol and transmits it to the X server for display.

2.17.2.2 X Window System API

It is possible to write an X client application and run it in a VM/CMS environment. The VM X Window System provides an application program interface (API) allowing an application to make calls to the API to create the X protocol to have output displayed on an X server. When writing this application, the user need only be concerned with the X client API and the syntax of the API calls.

Application programs written using the X Window System must be written in C and require the following corequisites:

- IBM C for System/370, Compiler Licensed Program (Program Number 5688-040)
- IBM C for System/370, Library Licensed Program (Program Number 5688-039)

This environment under VM/CMS is illustrated in Figure 31 on page 134.

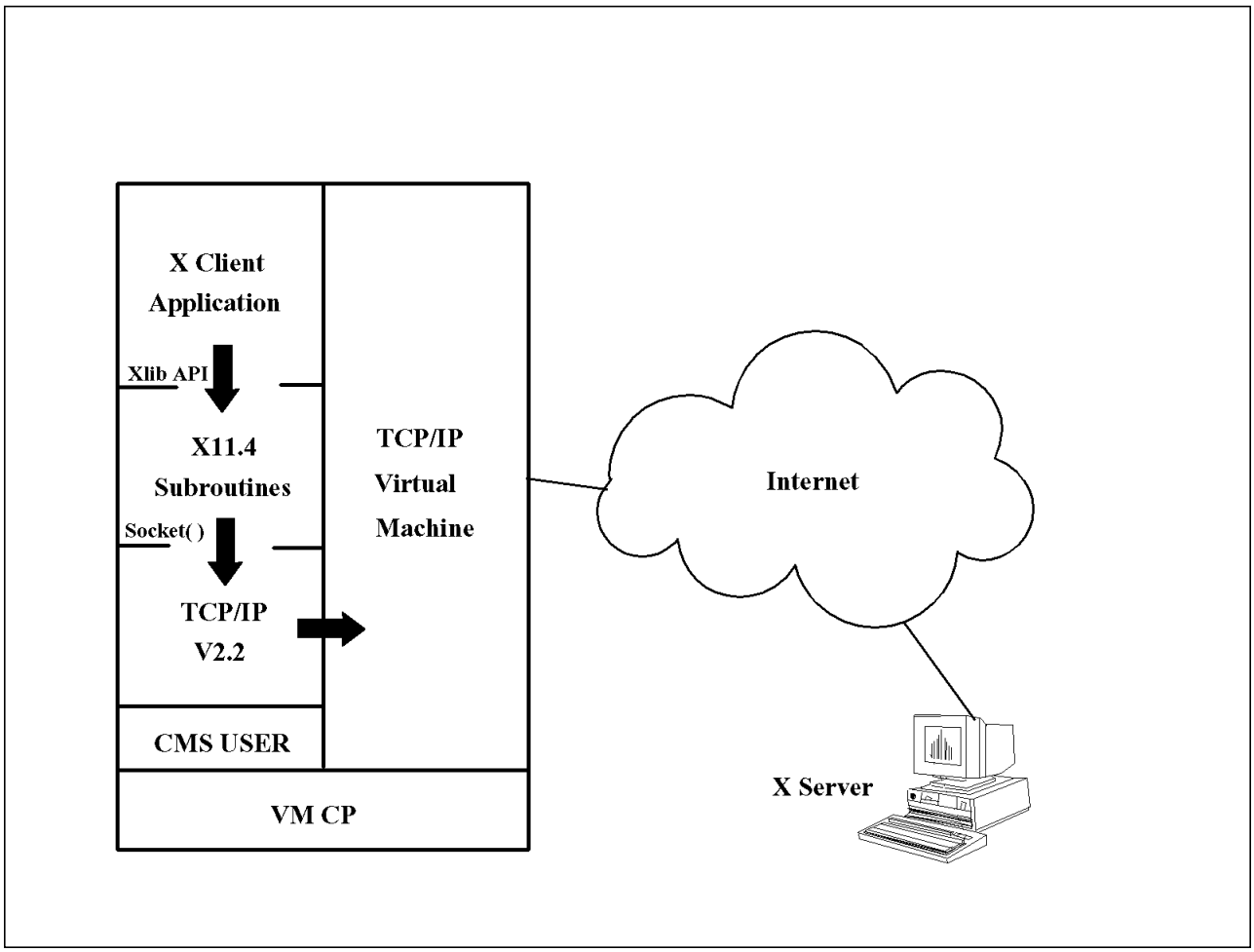


Figure 31. VM X Window System API

The X Window System under VM is is equivalent to the MIT defined X Window System Version 11 Release 4. It provides:

- X11LIB TXTLIB which provides the Version 11 X Window System subroutines:
  - X Window System Subroutines
  - Extension Routines which permit the creation of extensions to the core Xlib functions with the same performance characteristics.
  - MIT Extensions to the X Window System. As with MVS, the AIX extensions are not supported by the X Window System API under VM. The following MIT extensions are supported:
    - SHAPE
    - MITMISC
    - MULTIBUF
  - Associate Table functions which supports the association of user data structures with X Window resources.
  - Miscellaneous Utility Routines that are a common set of functions that have in the past proved useful to application writers.
  - X Authorization routines used to deal with X authorization data in X clients.
- OLDXLIB TXTLIB which provides Version 10 X Window System compatible routines.

- XTLIB TXTLIB (X Toolkit library intrinsics).
- XAWLIB TXTLIB (The Athena widget set).
- XMLIB TXTLIB (OSF/Motif based widget set).
- Header files required for compiling X client applications.
- Header files required for compiling X client applications using the OSF/Motif based widget set.
- Standard MIT X client applications.
- Sample X client applications.

The application calls XOpenDisplay() to start communication with an X server on a workstation. This invokes Xlib code to open a communication path called a socket to the X server and sends the appropriate X protocol to initiate client-server communication.

The X protocol used between the Xlib code and an X server uses an ISO Latin-1 encoding for character strings, while an application running under VM/CMS will use EBCDIC. The Xlib code will automatically translate character strings to and from EBCDIC and ISO Latin-1 as required using internal translate tables.

Under VM/CMS, external names must be eight characters or less. Under the X Window System, the X client API supports names of routines and data sets that often exceed this limit. In order to support the X client API under VM/CMS all X client application names longer than eight characters are remapped to unique names using the C compiler preprocessor. This name remapping is found in a file called X11GLUE H, which is automatically included in a user written X client application when the standard X header file called XLIB H is included.

2.17.2.3 Creating an X Client Application under VM/CMS

In order to write an application that uses the X Windows protocol under VM/CMS, it is necessary to understand the X client API. To gain a better understanding of this API, IBM provides the source and object code of three sample X client application programs, XSAMP1, XSAMP2, and XSAMP3 with TCP/IP for VM. In addition, IBM also provides some standard X client applications developed by MIT. These provide the best way to experiment with and learn about the X client API under VM.

2.17.2.4 X Window System Toolkit

IBM provides an X toolkit with the X Window subsystem for VM. The purpose of the toolkit is to simplify the design and reduce the time to code X client applications by providing a set of common user interface functions often called intrinsics. These functions are composed of many of the functions that are to be found in Xlib. By using the toolkit intrinsics, the application programmer will not need to go down to the low level of coding the Xlib routines to define the mechanisms for handling the interaction between the X server and the X client.

For a full list of the intrinsics supplied with X toolkit for VM please refer to the *mpref.* manual.

2.17.3 Installing the X Window System

The following is a guide to installing the X Window System under VM. The description is divided into two parts as follows:

1. Installation Verification for the X Window System API
2. Installing the X Window System GDDM Interface

It is assumed that the base TCP/IP product has been installed on your VM system and uses the minidisk addressing convention as described in *TCP/IP Version 2 Release 2 for VM: Planning and Customization.* This convention means that TCPMAINT controls the following minidisks:

- TCPMAINT user A disk (191)
- Client code disk (592)
- Server code disk (591)
- Service (PTF) disk (2C1)
- Source code disk (5C3)
- Samples disk (5C4)

2.17.3.1 Installation Verification for the VM X Window System API

The X Window system code is installed on the TCPMAINT client code disk at address 592. All that needs to be done is to verify that the API is ready to accept calls from an X client application.

IBM provides three sample programs with TCP/IP for VM. They are identical to those provided with MVS and are called XSAMP1, XSAMP2 and XSAMP3. In order to execute them they need to be compiled and link-edited as follows:

1. You need to have access to the following disks:

   - TCPMAINT 592 minidisk for TCP/IP client code
   - TCPMAINT 5C4 minidisk for the sample programs
   - The minidisk that has the C/370 Compiler

   If you have the required authority you can access a disk under VM by typing the following command at the CMS prompt:

   LINK *machine xxx yyy* RR
   ACC *yyy z*

   where *machine* is the virtual machine that owns the minidisk, *xxx* is the address of the minidisk, *yyy* is the virtual address that the minidisk will be known to your virtual machine and *z* is the mode. For the following discussion we have assumed that *xxx* equals *yyy*.

2. On the 5C4 minidisk you will find the source:

   - XSAMP1 C
   - XSAMP2 C
   - XSAMP3 C

3. Compile and link-edit XSAMP1 by doing the following:

   a. Set the LOADLIB and TXTLIB search order using the following CMS global commands:

      SET LDRTBLS 25

      GLOBAL LOADLIB EDCLINK

      GLOBAL TXTLIB X11LIB COMMTXT EDCBASE IBMLIB CMSLIB

b. Compile XSAMP1 C using the following command:

   CC XSAMP1 (DEFINE(IBMCPP)

   This creates XSAMP1 TEXT on your A disk.

c. Link-edit XSAMP1 using the following command:

   CMOD XSAMP1

   This creates XSAMP1 MODULE on your A disk.

4. Ensure that the X server has authorized the VM host as an X client.
5. Identify the target X server display using the following CMS global command.

   GLOBALV SELECT CENV SET DISPLAY *X client display variable*

   where the *X client display variable* is the variable that the X client must use to access the X server. It has the format *host:0.0* where *host* is the internet address of the X server and *0.0* represents the *target server.server screen*. For example, when using our OS/2 machine as the X server, it had an internet address 9.67.38.89. we typed the command:

   GLOBALV SELECT CENV SET DISPLAY 9.67.38.89:0.0

6. Run XSAMP1 by simply typing XSAMP1 at the CMS command line.

   XSAMP1 opens a display at the X server for sixty seconds and then closes it.

7. The process for compiling and link-editing XSAMP2 is the same as for XSAMP1 except for setting the LOADLIB and TXTLIB search order:

   SET LDRTBLS 25

   GLOBAL LOADLIB EDCLINK

   GLOBAL TXTLIB XAWLIB XTLIB X11LIB COMMTXT EDCBASE IBMLIB CMSLIB

   This is because XSAMP2 uses some of the intrinsics in the Xt library and some of the widgets from the Athena widget set.

8. When you compile XSAMP2 you will see the following warning:

   WARNING EDC0244 XSAM2 C D1:106 External name fallback_resources has been truncated to FALLBACK.

   This is normal.

9. When you execute XSAMP2 you will see a ″HELLO WORLD″ window. The best way to end XSAMP2 is to close down the window at the X server.
10. The process for compiling and link-editing XSAMP3 is the same as for XSAMP1 and XSAMP2 except for setting the LOADLIB and TXTLIB search order:

    SET LDRTBLS 25

    GLOBAL LOADLIB EDCLINK

    GLOBAL TXTLIB XMLIB XTLIB X11LIB COMMTXT EDCBASE IBMLIB CMSLIB

    This is because XSAMP3 uses some of the intrinsics in the Xt library and some of the widgets from the OSF/Motif widget set.

11. When you execute XSAMP3 you will see a button window. The best way to end XSAMP3 is to close down the window at the X server.

Note:

Ensure that your virtual machine has sufficient virtual storage for the compile and link-edit steps. Otherwise these steps will fail with a severe error:

Virtual Storage Exceeded.

2.17.3.2 Installing the VM X Window System GDDM Interface

The X Window System GDDM interface must be installed and activated in order for a GDDM application to send its output to an X server for display using the X protocol.

When installing the X Window System GDDM interface, the GDDM shared segment needs to be reinstalled. A shared segment is a mechanism that VM uses to make executable modules available in a single image for use by multiple users. If a shared segment was not used, each user would need a copy of the executable code running within their own virtual machine. Using a segment saves VM resources and greatly improves the performance of applications under VM.

The GDDM shared segment is for GDDM modules and improves the performance of GDDM applications. It needs to be reinstalled with the X Window System GDDM interface to allow the interface modules to access the shared segment.

Note: For our VM system we opted to make a copy of the GDDM shared segment for use by the X Window System GDDM interface modules. This is because if we reinstalled the existing named GDDM shared segment we would need to reassemble the bootstrap modules for each of the products that use GDDM.

If you elect to make a new named copy of the GDDM shared segment then you will need to zap the modules for those products that will be used with the new GDDM shared segment for the X Window System GDDM interface. This needs to be done to point the module at the new named GDDM shared segment that must be accessed for the X Window System GDDM interface.

Installation Steps The following steps provide a guide to installing the X Window System GDDM interface under VM:

1. Ensure that you have access to the TCPMAINT 592 minidisk on which you will find the following files:
   - GDDMXD TXTLIB which is the executable code.
   - INSTGDXD EXEC which is the installation exec.
   - GDDMXD EXEC which activates and deactivates the X Window System GDDM interface.
   - COMMTXT TXTLIB which is the common TCP/IP text library.
   - X11LIB TXTLIB which is the X Window function libary.
   - GDXALTCS PSS is an alternative character set for APL2.
   - GDXAPLCS SAMPMAP is a sample keyboard mapping for APL2.
2. Ensure that you are linked to the GDDM minidisk and have access to the following files:
   - ADMBLSEG EXEC
   - ADMGLIB TXTLIB
   - ADMNLIB TXTLIB
   - ADMPLIB TXTLIB

3. Ensure that you are linked to the C/370 minidisk and have access to the C/370 runtime library, IBMLIB TXTLIB.

4. Before running the installation exec, you need to define the GDDM shared segment that will be available to the X Window System GDDM interface. You need to know the name and the spool location of the existing GDDM shared segment. This will be installation dependent and can be determined by typing the following command from the CMS command line:

   Q NSS NAME *filename* MAP

   where *filename* is the name of the existing saved shared segment. This will give you the beginning and end segment locations for the shared segment.

   For example the segment name that was defined when GDDM was installed for VM on our machine was ADMXA230. We issued the command

   Q NSS NAME ADMXA230 MAP

   which produced the output as illustrated in Figure 32.

   Note: You need to have a CP privilege class E to execute this command.

```
FILE FILENAME FILETYPE MINSIZE  BEGPAG ENDPAG TYPE CL #USERS PARMREGS VMGROUP
0325 ADMXA230 DCSS       N/A    04000  042FF   SR  P  00001   N/A       N/A
0443 ADMXA230 DCSS       N/A    04000  042FF   SR  A  00000   N/A       N/A
Ready; T=0.01/0.01 17:55:13
```

Figure 32. Example of the Output from the Q NSS NAME Command on VM

5. Define the new GDDM shared segment by typing the following command:

   DEFSAG *new segment name begpag-endpag* SR

   where

   - *new segment name* is the new name of the shared segment.
   - *begpag* is the segment start location.
   - *endpag* is the segment end location.
   - SR means share/read.

   You must use the *begpag* and *endpag* values that were returned from the Q NSS NAME command. Refer to the example illustrated in Figure 32. On our VM system we typed:

   DEFSEG GDDMXD 4000-42FF

   where GDDMXD is the name we choose for the GDDM shared segment to be reinstalled.

6. The X Window System GDDM interface is actually installed by invoking the exec INSTGDXD. Start this at the CMS command prompt.

7. During the execution of INSTGDXD you will be need to respond to prompts as follows:

   a. Which are the products for which you want saved segments installed? The options are GDDM, PGF, IMD, IVU. You only need to choose GDDM.

   b. What is the name of the saved segment? Enter the name you choose when you defined the saved segment using the DEFSEG command. In our installation we used GDDMXD.

   c. Do you want to include the Image Symbol Editor routines? Respond YES.

d. Do you want to include the Image Processing routines? Respond YES.

The INSTGDXD performs the following:

a. Builds the following load modules and places them on your A disk.

- GDXLIOX0 MODULE
- GDXCLOSE MODULE
- GDXADML1 MODULE
- KEYCODE MODULE

b. Builds the demonstration load modules GXDEMO1 through to 6 and places them on your A disk.

c. Builds the GDXBLSEG EXEC and invokes it to create the GDDM shared segment.

8. Ensure that INSTGDXD runs with no errors. While it is running INSTGDXD provides you with messages to indicate the step that is currently executing. An example of the last messages you should see for a successful run are illustrated in Figure 33.

```
*** Segment name: GDDMXD   Start address: 04000000...

*** Loading TEXT decks....
Saved segment GDDMXD was successfully saved in fileid 0444.
*** GDDM DCSS build complete.

*** ADMBLSEG complete for chosen products
Ready; T=3.00/4.51 17:57:50
```

Figure 33. Example of a Successful Run for INSTGDXD

9. You can check that the GDDM shared segment has been installed successfully by issuing the following command:

   Q NSS NAME *new segment name* MAP

   where *new segment name* is the name of the GDDM shared segment just created by INSTGDXD. Notice that the class of this segment is A which means available.

10. Copy the modules built by INSTGDXD from your A disk across to a minidisk such as the TCPMAINT 592 disk so that they may be commonly accessed.

Installation Verification: Use the following steps as a guide to verifying the installation of the X Window System GDDM interface for VM.

1. IBM provides some sample programs that allow you to test the X Window System GDDM interface under VM. Before running these programs you need to do a number of tasks. In fact each time you want to have a GDDM application display its output through the X Window System GDDM interface to an X server you must do these same tasks as follows:

   a. Ensure you have access to the TCPMAINT 592 minidisk, the GDDM minidisk and the minidisk that holds the C/370 libraries.

   b. For VM/XA systems only, enter the following command from the CMS command prompt:

      SET STORECLR ENDCMD

This ensures that GETMAIN requests by the X Window System GDDM interface code are processed correctly.

c. Activate the X Window System GDDM interface by issuing the following command:

   GDDMXD ON

   You should see the message **GDDMXD/VM active**.

d. Set the following CMS global variables by issuing these three commands:

   SET LDRTBLS 25

   GLOBAL LOADLIB EDCLINK

   GLOBAL TXTLIB ADMNLIB GDDMXD ADMPLIB ADMGLIB X11LIB COMMTXT IBMLIB EDCBASE CMSLIB

e. Identify the target X server display using the following CMS global command.

   GLOBALV SELECT CENV SET DISPLAY *X client display variable*

   where the *X client display variable* is the variable that the X client must use to access the X server. It has the format *host:0.0* where *host* is the internet address of the X server and *0.0* represents the *target server.server screen*. For example, when using our OS/2 machine as the X server, it had an internet address 9.67.38.89. we typed the command:

   GLOBALV SELECT CENV SET DISPLAY 9.67.38.89:0.0

2. Ensure that the X server has authorized the VM host as an X client.
3. In order to interact with the sample programs using the X server keyboard you will need to alter the keyboard mapping for the enter key. We discovered that the default keyboard mappings provided with the X Window Systems for both OS/2 and AIX had the Enter key set to the keysymname *RETURN*. You need to remap this key keysymname to *EXECUTE*.
4. Execute the sample programs GXDEMO1, GXDEMO2, GXDEMO3 GXDEMO4, GXDEMO4A, GXDEMO5 and GXDEMO6 from the CMS command prompt. Each program displays a series of frames that you can progress through by pressing the Enter key. The application will close the window at the X server and end after the last frame.

Note:

The GXDEMOx programs will not open a window at the X server and will simply display output to your 3270 terminal if you have opted to reinstall the GDDM shared segment with a new name. This is because the GXDEMOx programs are built by INSTGDXD before the new GDDM shared segment is installed which means that they point to the existing named shared segment.

To fix this you need to zap the GXDEMOx modules to point to the new named shared segment.

2.18 Domain Name Server

RFC 1034 and RFC 1035 explain the domain name system in full detail; refer to these documents for a complete description.

TCP/IP Tutorial and Technical Overview - GG24-3376-02 also contains a lot of details about the domain name server and its implementation.

IBM TCP/IP V2 for VM allows the use of a flat name space or the use of the domain name system.

The use of a flat name space or the use of the domain name system is chosen within the ″TCPIP DATA″ file via the ″NSINTERADDR″ statement. If the statement is omitted or commented out, TCPIP will use the host table (flat name space). If this statement exists, then, depending on the NSINTERADDR statement, TCPIP will use either a local name server (`NSINTERADDR 14.0.0.0`, `14.0.0.0` is the loopback address) or one (or more) remote name servers. This provides a total of five configurations.

Note

The 14.0.0.0 address is a valid one but you will have trouble (if you code `NSINTERADDR 14.0.0.0` in your ″TCPIP DATA″ file) if you use the `DIG` or `NSLOOKUP` command. This is because the address 14.0.0.0 will, probably, not be coded in your ″MASTER DATA″ file.

The easiest solution to solve this problem (assuming the address of your local name server is 9.67.38.65) is to code `NSINTERADDR 9.67.38.65` in the ″TCPIP DATA″ file.

- **Flat Name Space**: Each host in the network needs a sequential file for symbolic-name to IP-address translation (actually, this file is used for IP-address to symbolic-name translation as well). This file (″HOSTS LOCAL″) must be converted using the `MAKESITE` utility. The two resulting files will be used by TCPIP to perform mappings; these two files should reside on a minidisk accessed by all the clients. Most of the protocols (the only exception is SMTP) will use these host files when no response is received from a name server when it exists.

  An example of ″TCPIP DATA″ for a flat name space could be:

  ```
  TCPIPUSERID TCPIP
  ;
  RALYESA: HOSTNAME RALYESA
  ;
  ; DOMAINORIGIN  ITSC.RALEIGH.IBM.COM
  ; NSINTERADDR 14.0.0.0
  ; NSPORTADDR 53
  ; RESOLVEVIA UDP
  ; RESOLVERTIMEOUT 30
  ; RESOLVERUDPRETRIES 1
  ```

  All the NS entries have been commented out. No name server will be used.

- **Using the Domain Name System**, four possibilities exist:

  – **No local name server**: The VM host will then query remote name server(s) for the address resolution. The addresses of the remote name

server(s) are indicated in the ″TCPIP DATA″ file. Connections to remote name servers are attempted in the order they appear in this configuration file.

An example of ″TCPIP DATA″ could be:

```
TCPIPUSERID TCPIP
;
RALYESA: HOSTNAME RALYESA
;
DOMAINORIGIN  ITSC.RALEIGH.IBM.COM  1
NSINTERADDR 9.67.38.98              2
NSINTERADDR 9.67.38.66              3
NSPORTADDR 53                       4
RESOLVEVIA UDP                      5
RESOLVERTIMEOUT 30                  6
RESOLVERUDPRETRIES 1                7
```

- 1 the domain name. Will be appended to any name not ending with a dot.
- 2 IP address of the first name server that will be contacted.
- 3 IP address of the second name server that will be contacted (only if no answer is received from the first one).
- 4 Port used.
- 5 Protocol used.
- 6 Number of seconds the resolver (the client program) waits until a response is received.
- 7 Maximum number of times the resolver should try to connect to the name server.

Using this kind of configuration any query will be sent to a remote name server.

– **Caching-only name server**: A name server is running into the local VM system but this name server does not have an SQL database. All queries will be sent to remote name server(s) but answers will be saved in memory (cached). Subsequent queries for the same domain name will be retrieved from the cache, thus reducing network traffic. Such a name server only needs the IP addresses of the remote name server(s).

  The NSINTERADDR statement of the file ″TCPIP DATA″ points to the local address.

  Please refer to Section 2.18.1.1, “Configuring a Caching-Only Name Server” on page 145 for more details.

– **Full-function name server**: A name server is running on the local VM host. It needs an SQL database to store data, that is to store the mappings between symbolic-name and IP-addresses.

  The data is held in sets called zones. Each zone is the complete database for a particular ″pruned″ subtree of the domain space. When a name server ″owns″ a zone it becomes *authoritative* for this zone. That means that it can answer queries from other name servers. There are two types of full-function name servers:

  1. Primary name server: The zone this name server is responsible for is ″local″. It has not been transferred from another name server. It was loaded into SQL tables from a ″MASTER DATA″ file located on the same VM system. Please refer to Section 2.18.1.2, “Configuring a Primary Name Server” on page 146 for more details.

2. Secondary name server: The zone this name server is responsible for has been transferred from another name server. It was not loaded into SQL tables from a ″MASTER DATA″ file located on the same VM system. A secondary name server may be considered as a backup name server in case the primary one fails. Please refer to Section 2.18.1.3, “Configuring a Secondary Name Server” on page 155 for more details.

A unique name server may act as a primary name server for a zone, and as a secondary name server for another zone.

The name server, once queried for an address, will cache the results in memory thus reducing database accesses. IBM TCP/IP Version 2 Release 2 for VM provides different levels of caching (please refer to *IBM TCP/IP V2 R2 for VM: Planning and Customization* for more information). This data may be incomplete, but improves the performance of the retrieval process when non-local data is repeatedly accessed. Cached data is eventually discarded by a timeout mechanism.

Warning

Keep this caching facility in mind because it may cause some problems from time to time. Suppose you have two SQL tables, ITSC0 and ITSC1. In the currently used table ITSC0 a new host name is missing, so you add it in ITSC1, load ITSC1 into the SQL database via the `NSDBLOAD` command and issue the `SMSG NAMESRV FLIPTABLE ITSC` to use ITSC1.

When you `PING` the new host it will not work if you ″pinged″ it previously using the table ITSC0. This is because the negative answer has been cached and all subsequent queries are answered from the cache. To solve this, the following sequence is required:

1. Using ITSC0 you (or somebody else in the network) ″pinged″ *newhost*. The negative answer is returned and cached.
2. Load newhost into ITSC1 using the `NSDBLOAD` command.
3. `SMSG NAMESRV FLIPTABLE ITSC` to use ITSC1.
4. `SMSG NAMESRV PURGE ALL` to purge caching entry.
5. `PING newhost` is working now.

Like most implementations, IBM TCP/IP V2 for VM implements what is called a *stub resolver*. That means the caching process will take place within the name server, not in the CMS user virtual machine.

IBM TCP/IP Version 2 Release 2 for VM includes some enhancements in its name server implementation:

- Two new commands for querying name servers from a CMS user ID: **DIG** and **NSLOOKUP**. Please refer to Section 2.18.3.3, “The DIG Command” on page 164 and Section 2.18.3.4, “The NSLOOKUP Command” on page 166 for more details about the use of these commands.
- An option to remove the queue limit from all open UDP ports.

  Normally, TCP/IP will queue up to a maximum of 21 incoming datagrams on an open UDP port. If there are already 21 datagrams queued and another datagram is received, it will be discarded. This limit keeps a malfunctioning program from monopolizing all the envelopes, and generally doesn′t pose a problem for programs that receive input at a moderate rate.

If you run a program that is affected by the queue limit, you can now specify *NOUDPQUEUELIMIT* on the *ASSORTEDPARMS* statement of the "PROFILE TCPIP".

Like all *ASSORTEDPARMS* parameters, a misspelling of this parameter will not be flagged by TCPIP. Therefore, after adding it to the "PROFILE TCPIP", you should verify that TCPIP displays the message:

```
No limit on incoming UDP datagram queue
```

during its startup.

Warning

Although this has been implemented as a requirement on the name server, *NOUDPQUEUELIMIT* removes the queue limit from **all** open UDP ports.

2.18.1 Configuring a Name Server

2.18.1.1 Configuring a Caching-Only Name Server

The following are the definitions needed:

PROFILE TCPIP

```
AUTOLOG
   .....
   NAMESRV  RacfPsw     ; DOMAIN NAME SERVER
   ....
ENDAUTOLOG
   ....
PORT
   ....
53 TCP NAMESRV             ; domain name  Server
53 UDP NAMESRV             ; domain name  Server
   ....
```

TCPIP DATA

```
TCPIPUSERID TCPIP
;
RALYESA: HOSTNAME RALYESA
;
DOMAINORIGIN  ITSC.RALEIGH.IBM.COM
NSINTERADDR 9.67.38.65              1
NSPORTADDR 53
RESOLVEVIA UDP
RESOLVERTIMEOUT 30
RESOLVERUDPRETRIES 1
```

1 Your local address. The resolver is then aware that the name server to contact is local.

NSMAIN DATA

```
   ....
CACHINGONLY  NOSQL CACHE A     ; Sample CACHINGONLY data file
   ....
```

The PRIMARY and the SECONDARY statements must be commented out.

NOSQL CACHE A

```
itsc.raleigh.ibm.com         86400  IN  NS   vm14.itsc.raleigh.ibm.com
vm14.itsc.raleigh.ibm.com    86400  IN  A    9.67.38.66
```

This file contains only domain names and their corresponding name server(s).

All queries for domain itsc.raleigh.ibm.com will be sent to VM14.ITSC.RALEIGH.IBM.COM whose internet address is 9.67.38.66.

2.18.1.2 Configuring a Primary Name Server

The following is a step-by-step implementation of a primary name server using a SQL/DS database.

Please refer also to Section 2.18.2.2, “Updating SQL Tables” on page 157 about how to update SQL tables dynamically and to Section 2.18.2.1, “Adding SQL Tables” on page 156 about how to add SQL tables.

1. **File ″NAMESXIT EXEC″ on TCPMAINT 591**

   The startup procedure expects to access the SQL/DS production minidisk at address 593. Either add a link statement in the directory entry or add a link and an access statement to the ″NAMESXIT EXEC″ before the access. If you want to be able to use the NSDBLOAD command from the NAMESRV user ID then load the appropriate loadlibs (EDCLINK, IBMLIB).

   For administrative purposes you sometimes need to call the ISQL program from the NAMESRV user ID. To be able to achieve this, perform the following sequence (this assumes that the name of the SQL machine is SQLDBA):

   a. LINK SQLDBA 195 195 RR: Link the right minidisk.

   b. ACCESS 195 Q: Access the minidisk as Q.

   c. COPY SQLUSER PROFILE Q = EXEC A: Copy the file ″SQLUSER PROFILE″ as ″SQLUSER EXEC″.

   d. DET Q: Detach the minidisk (do not release it).

   e. SQLUSER: Execute the SQLUSER EXEC.

   f. SQLINIT DB(SQLDBA): This assumes that SQLDBA is the name of your SQL database. This can be seen when the SQL machine (SQLDBA) starts.

   You are now ready to use ISQL from NAMESRV. These steps are not required to have the name server running because they are included in the file ″NAMESXIT EXEC″ which is executed when the NAMESRV user ID acts as a domain name server. The ″NAMESXIT EXEC″ allows you to specify the name of the SQL database and the language you will use.

Following is an example of the ″NAMESXIT EXEC″ (numbers on the left are line numbers):

```
00117 ¢CP LINK SQLDBA 195 593 RR¢          1
00118 ¢ACCESS 593 B¢
00119 ¢GLOBAL LOADLIB EDCLINK IBMLIB¢      2

/* Some comments here */

00128 USESQL = 1                           3
00129 database = ¢SQLDBA¢                  4
00130 language = ¢AMENG¢                   5
00131
00132 if usesql then
00133   do
00134     ¢set language¢ language ¢(add ari user¢  6
00135     ¢sqlinit db(¢database¢)¢                 7
00136   end
```

1 Link and access the SQL minidisk.

2 Loads the EDCLINK and IBMLIB (where the required module IBMBLIIA is found) loadlibs. Only needed if you want to issue ISQL commands from the NAMESRV virtual machine.

3 If set to 1 the SQL commands `set language ...` and `sqlinit ..` will be issued for you. This is the easiest way to set up NAMESRV to use the SQL database.

4 Database name. Change it if required.

5 Language used. Change it if required.

6 Required to avoid some `ARIxxx` error messages when ISQL is invoked. You can also copy the file ″SQLUSER PROFILE″ as ″SQLUSER EXEC A″ and execute it. These two solutions give the same result (the ″SQLUSER EXEC″ issues the `set language ..` command).

7 Only required once. Creates the `ARISISBT MODULE` and `ARISRMBT MODULE` on NAMESRV.191 minidisk. If the `SQLINIT DB(SQLDBA)` has been issued previously, this line may be removed from the ″NAMESXIT EXEC″ (this causes no problem if you leave it).

2. **File ″NSMAIN DATA″ on NAMESRV 191**

The system parameters for the name server are specified in this configuration file. The name server module (″NSMAIN MODULE″) reads and interprets the configuration file at startup time. In this file you define the mode of operation for the name server and the name of the SQL/DS tables that you will generate later on. The name server module is invoked by the ″TCPRUN EXEC″.

The following example shows a name server configuration file and some statements that relate to other definitions for your specific environment.

File "NSMAIN DATA"

```
PRIMARY ITSC.RALEIGH.IBM.COM ITSC          1
PRIMARY 38.67.9.IN-ADDR.ARPA ARPA          2
SECONDARY IBM.COM IBM 9.67.38.17           3
NEGATIVECACHING
STANDARDQUERYCACHE     100
INTERMEDIARYQUERYCACHE  50
INVERSEQUERYCACHE        5
DATABASEQUERYCACHE       5
LRUTIME                300
HOSTNAMECASE          UPPER                4
DOMAINNAMEPORT           53
UDPRETRYINTERVAL          5
; MSGNOH
SMSGUSERFILE    VALIDUSR EXEC A
TRACE                QUEUE
```

1 The primary zone name (ITSC.RALEIGH.IBM.COM) for this name server and the name of the SQL/DS table where the data can be found.

2 The primary zone name (38.67.9.IN-ADDR.ARPA) for this name server and the name of the SQL/DS table where the data can be found.

3 The secondary zone name (IBM.COM), the name of the SQL/DS table where the zone data will be transferred and the internet address of the primary zone owner. If your name server is not acting as a secondary one this line must be commented out.

4 The data in your tables will be either lowercase or uppercase; you will be asked which when you load the tables.

Warning

- If you load the SQL table in lowercase, and you specified UPPER in the file "NSMAIN DATA", you won't be able to use the name server.

3. **File "MASTER DATA"**

   The master file is a text file that contains Resource Records (RRs) in text form. The contents of a zone can be expressed in the form of a list of RRs. Loading the RRs into the SQL/DS table is done with the NSDBLOAD command.

   Note: This file may reside on a different minidisk from NAMESRV.191. If this is the case, you will have to specifiy the complete file name within the NSDBLOAD command. For example: NSDBLOAD ISTC MASTER DATA E if the file is located on the minidisk accessed with filemode E.

   The "MASTER DATA" file should be responsible for ONE zone only. In the ITSC configuration we had two master files: one for the zone ITSC.RALEIGH.IBM.COM and one for the IN-ADDR.ARPA domain. Please refer to the examples provided on pages 152 and 153.

   If you need additional details refer to RFC 1033, which provides guidelines for a domain administrator in operating a domain server and maintaining their portion of the hierarchical database.

The RRs have the following general format:

RR Format

```
name        ttl    class   type  data
```

Parameter	Description
name	The name field defines what domain name applies to the given RR. In some cases the name field can be left blank and it will default to the name field of the previous RR.
ttl	TTL stands for Time To Live. It specifies how long a domain resolver should cache the RR before it throws it out and asks a domain server again.
class	The class field specifies the protocol group, usually IN.
type	The type field specifies what type of data is in the RR (SOA,NS,A, etc.).
data	The data field is defined differently for each type and class of data.

The most important RR types which are needed to create a master file are explained here.

Start of Authority (SOA) Record Type

The SOA record designates the start of a zone. The zone ends at the next SOA record.

SOA Type

```
name        ttl    IN SOA   origin person  (
                   serial
                   refresh
                   retry
                   expire
                   minimum )
```

Parameter	Description
name	Is the domain name of the zone.
ttl	Is the time to live in seconds.
origin	Is the name of the host on which the master zone file resides.
person	Is a mailbox for the person responsible for the zone. It is formatted like a mailing address but the ″@″ sign that normally separates the user from the host name is replaced with a dot.
serial	Is the version number of the zone file. It should be incremented anytime a change is made to data in the zone.
refresh	Is how long, in seconds, a secondary name server should check with the primary name server to see if an update is needed. A good value here would be one hour (3600).
retry	Is how long, in seconds, a secondary name server is to retry after a failure to check for a refresh. A good value here would be 10 minutes (600).

expire Is the upper limit, in seconds, that a secondary name server is to use data before it expires for the lack of getting a refresh. A good value is 3600000, about 42 days.

minimum Is the minimum number of seconds to be used for TTL values in RRs. A minimum of at least one day is a good value here (86400).

Name Server (NS) Record Type

The NS record lists the name of a machine that provides domain service for a particular domain.

NS Type

```
name        ttl    IN NS    server
```

Parameter	Description
name	Is the domain name of the zone.
server	Is the name of the host that provides the domain service.

Internet Address (A) Type

The A record lists the internet address of a host.

A Type

```
host        ttl    IN A     address
```

Parameter	Description
host	Is the name of the host.
address	Is the internet address of the host in dotted decimal form.

The in-addr.arpa Domain

The intent of this domain is to provide a guaranteed method to perform host address to host name mapping. This is required by some servers (for example the NFS server of a RISC System/6000) in order to function properly. The domain begins at IN-ADDR.ARPA and has a substructure which follows the internet addressing structure. Domain names in the IN-ADDR.ARPA domain are defined to have up to four labels in addition to the IN-ADDR.ARPA suffix. Each label represents one octet of an internet address. See the ″MASTER FILE″ on page 153.

Host addresses are represented by domain names that have all four labels specified.

Network nodes are used to hold pointers to the primary host names of routers attached to that network. Since a router is, by definition, on more than one network, it will typically have two or more network nodes which point to it. Routers will also have host level pointers at their fully qualified addresses.

Both the router pointers at network nodes and the normal host pointers at full address nodes use the PTR RR to point back to the primary domain names of the corresponding hosts.

An example could be (extracted from RFC 1035):

```
10.IN-ADDR.ARPA           PTR  MILNET-GW.ISI.EDU.   1
10.IN-ADDR.ARPA           PTR  GW.LCS.MIT.EDU.      2
18.IN-ADDR.ARPA           PTR  GW.LCS.MIT.EDU.      3
26.IN-ADDR.ARPA           PTR  MILNET-GW.ISI.EDU.
22.0.2.10.IN-ADDR.ARPA    PTR  MILNET-GW.ISI.EDU.   4
103.0.0.26.IN-ADDR.ARPA   PTR  MILNET-GW.ISI.EDU.   5
77.0.0.10.IN-ADDR.ARPA    PTR  GW.LCS.MIT.EDU.      6
4.0.10.18.IN-ADDR.ARPA    PTR  GW.LCS.MIT.EDU.      7
```

1 Router MILNET-GW.ISI.EDU pointer at network node (network 10)

2 Router GW.LCS.MIT.EDU pointer at network node (network 10)

These two routers are connected to the same network (10).

3 Router GW.LCS.MIT.EDU pointer at network node (network 18)

4 Host pointer at full address for host MILNET-GW.ISI.EDU on network 10

5 Host pointer at full address for host MILNET-GW.ISI.EDU on network 26

6 Host pointer at full address for host GW.LCS.ISI.EDU on network 10

7 Host pointer at full address for host GW.LCS.ISI.EDU on network 18

Like any other domain the IN-ADDR.ARPA domain will require an SOA Resource Record.

Here is an example of a master file for the zone ITSC.RALEIGH.IBM.COM:

Master File "MASTER IBM-COM"

```
$origin ITSC.RALEIGH.IBM.COM.                                    1
@    IN SOA RALYESA.ITSC.RALEIGH.IBM.COM   TCPMAINT.RALYESA (  2
                           901215
                           3600
                           600
                           3600000
                           86400 )
;
ITSC.RALEIGH.IBM.COM.      IN NS  RALYESA                        3
RALYESA                    IN A      9.67.38.65                  4
VMESA                      IN CNAME  RALYESA                     5
RAL9360                    IN A      9.67.38.67
VM15                       IN CNAME  RAL9360
VM14                       IN A      9.67.38.66
RAL9390                    IN CNAME  VM14
FRED                       IN A      9.67.38.98
RS60001                    IN A      9.67.38.71
RS60002                    IN A      9.67.38.72
RS60003                    IN A      9.67.38.73
MVS20                      IN A      9.67.38.133
MVS18                      IN A      9.67.38.135
RAIANJE                    IN CNAME  MVS18
;
```

1 The *$origin* record defines the part of a symbolic name (ITSC.RALEIGH.IBM.COM) that has to be appended to every symbolic name not ending with a period. Note that the character string must end in a dot. If it does not end in a dot, the previous origin is appended.

2 The Start of Authority (SOA) record defines the zone (ITSC.RALEIGH.IBM.COM) to be managed by this name server (RALYESA.ITSC.RALEIGH.IBM.COM). The person responsible for the zone is also specified.

3 The Name Server (NS) record assigns the zone (ITSC.RALEIGH.IBM.COM) to the symbolic name (RALYESA) of the responsible name server. These references can be used by the name server to query a remote name server for a specific zone. Multiple NS records may be present to reference other name servers. The entry is not required if the only name server which can be queried is the one identified in the SOA record.

4 The Address (A) records assign an internet address to a symbolic name. This symbolic name does not end with a dot; therefore, the string after the *$origin* record is appended before the record is loaded into the SQL/DS table.

5 The Canonical Name (CNAME) records assign another hostname to a previously defined hostname. For example the RALYESA host can be contacted using the name VMESA.

Here is an example of a master file for the zone 38.67.9.IN-ADDR.ARPA:

Master File "MASTER IN-ADDR"

```
$origin 38.67.9.in-addr.arpa.           6
@                           IN SOA RALYESA.ITSC.RALEIGH.IBM.COM. (  7
                            TCPMAINT.RALYESA.ITSC.RALEIGH.IBM.COM.
                            901215
                            3600
                            600
                            3600000
                            86400 )
;
38.67.9.IN-ADDR.ARPA.       IN NS     RALYESA.ITSC.RALEIGH.IBM.COM.  8
;
65                          IN PTR    RALYESA.ITSC.RALEIGH.IBM.COM.  9
67                          IN PTR    VM15.ITSC.RALEIGH.IBM.COM.
66                          IN PTR    VM14.ITSC.RALEIGH.IBM.COM.
98                          IN PTR    FRED.ITSC.RALEIGH.IBM.COM.  10
71                          IN PTR    RS60001.ITSC.RALEIGH.IBM.COM.
72                          IN PTR    RS60002.ITSC.RALEIGH.IBM.COM.
73                          IN PTR    RS60003.ITSC.RALEIGH.IBM.COM.
133                         IN PTR    MVS20.ITSC.RALEIGH.IBM.COM.
135                         IN PTR    MVS18.ITSC.RALEIGH.IBM.COM.
;
```

6 - 7 Start of the in-addr.arpa domain. This domain is defined like any other domain; the only difference is that the addresses are inverted. A new origin is defined.

8 RALYESA.ITSC.RALEIGH.IBM.COM is the name server responsible for the zone. Multiple name servers may be specified. Again, this entry is not required if the only name server able to answer is the one indentified in the SOA record.

9 The PoinTeR (PTR) records assign a name to an internet address.

10 We used the fully qualified name so that an inverse query will return the right name. This is required to use some servers which perform inverse query (based on the address) and which will compare the result with a kind of host file (the OS/2 RSH server for example).

4. **Preparing the Database**

The first time you want to use SQL/DS from your NAMESRV virtual machine you must prepare your user ID for SQL/DS.

You must have a link to the C/370 minidisk.

Issue the following command to have the right loadlibs:

```
GLOBAL LOADLIB EDCLINK IBMLIB
```

Issue the following command to prepare NAMESRV for SQL/DS:

```
SQLINIT DB(database_name)
```

The name of the database is displayed during the SQL virtual machine startup. If you do not have access to the SQL virtual machine (or to its log), ask your system administrator. An example of the command could be SQLINIT DB(SQLDBA).

Now your NAMESRV virtual machine needs sufficient authority to preprocess the application programs and acquire a dbspace. You can do it yourself if you know the SQLDBA password, or ask your system administrator to give NAMESRV temporary ***DBA*** authority during the customization.

If you know the SQLDBA password, start an ISQL session. To work interactively with SQL/DS, issue the command:

```
ISQL
```

Connect NAMESRV to SQLDBA by using the *ISQL* subcommand:

```
CONNECT SQLDBA IDENTIFIED BY sqldba_password
```

and give authority to NAMESRV by issuing the *ISQL* subcommand:

```
GRANT DBA TO NAMESRV
```

Now you no longer need ISQL, so exit it.

```
EXIT
```

When you are back in VM/CMS the installation of the tables may begin. The name server application programs need SQL/DS preprocessing, which is done by issuing:

```
NSPREP
```

The name server tables are stored in a separate dbspace. Recall the name of the SQL/DS database and issue the following command, which will create a dbspace (TCPSPACE) with a size of 5120:

```
NSACQ database_name
```

An example could be `NSACQ SQLDBA`.

If you need to drop the dbspace created by this command for any reason, issue this command:

```
DROP DBSPACE TCPSPACE
```

After this, you will have to execute the *NSACQ* command again.

Note: The command *DROP DBSPACE* is a SQL/DS command, so you will need to start ISQL before executing this command.

Create the SQL/DS tables by invoking:

```
NSTABLE filename filetype filemode
```

Multiple tables are created according to the file specified. The default is ″NSMAIN DATA A″. Two tables are created for each PRIMARY or SECONDARY entry. This will allow you to have a backup table.

Finally you can load the SQL/DS tables from the ″MASTER DATA″ file with the `NSDBLOAD` command. Recall the table name in the PRIMARY statement of the ″NSMAIN DATA″ file to execute the following command:

```
NSDBLOAD primary_table_name master_file
```

When you invoke the NSDBLOAD command you will be asked if you want to create the resource records from the master file; when you answer `yes` the

input for the SQL/DS database is created as ″NSMAIN HOSTINFO″. It is a good practice to look at this file to see if everything is okay.

Note: The default master file is ″MASTER DATA A″. The default output file is ″NSMAIN HOSTINFO A″. They can be changed using the NSDBLOAD command.

For example, in the ITSC configuration we had two master files defined: one for the zone ITSC.RALEIGH.IBM.COM and one for the zone 38.67.9.IN-ADDR.ARPA. The first file is named ″MASTER IBM-COM″ and the second one is named ″MASTER IN-ADDR″. To load them the following commands were used:

- To load ″MASTER IBM-COM″ in the ITSC table:

  ```
  NSDBLOAD NAMESRV.ITSC MASTER IBM-COM A
  ```

- To load ″MASTER IN-ARPA″ in the ARPA table:

  ```
  NSDBLOAD NAMESRV.ARPA MASTER IN-ADDR A
  ```

Here is an example of ″NSMAIN HOSTINFO″ file created during the execution of **NSDBLOAD** command with the ″MASTER DATA″ file on page 152:

File ″NSMAIN HOSTINFO″

```
ITSC.RALEIGH.IBM.COM 0 IN SOA RALYESA.ITSC.RALEIGH.IBM.COM   1
TCPMAINT.RALYESA.ITSC.RALEIGH.IBM.COM 901215    0            1
IBM.COM 0 IN NS RALYESA.ITSC.RALEIGH.IBM.COM
ITSC.RALEIGH.IBM.COM 0 IN NS RALYESA.ITSC.RALEIGH.IBM.COM
RALYESA.ITSC.RALEIGH.IBM.COM 0 IN A 9.67.38.65
VMESA.ITSC.RALEIGH.IBM.COM 0 IN CNAME RALYESA.ITSC.RALEIGH.IBM.COM
RAL9360.ITSC.RALEIGH.IBM.COM 0 IN A 9.67.38.67
VM15.ITSC.RALEIGH.IBM.COM 0 IN CNAME RAL9360.ITSC.RALEIGH.IBM.COM
VM14.ITSC.RALEIGH.IBM.COM 0 IN A 9.67.38.66
RAL9390.ITSC.RALEIGH.IBM.COM 0 IN CNAME VM14.ITSC.RALEIGH.IBM.COM
FRED.ITSC.RALEIGH.IBM.COM 0 IN A 9.67.38.98
RS60001.ITSC.RALEIGH.IBM.COM 0 IN A 9.67.38.71
RS60002.ITSC.RALEIGH.IBM.COM 0 IN A 9.67.38.72
RS60003.ITSC.RALEIGH.IBM.COM 0 IN A 9.67.38.73
MVS20.ITSC.RALEIGH.IBM.COM 0 IN A 9.67.38.133
MVS18.ITSC.RALEIGH.IBM.COM 0 IN A 9.67.38.135
RAIANJE.ITSC.RALEIGH.IBM.COM 0 IN CNAME MVS18.ITSC.RALEIGH.IBM.COM
```

1 One line (the SOA record)

The same kind of file will be created with a ″MASTER DATA″ file like the one displayed on page 153.

DBA authority for NAMESRV is no longer required. If you want to remove this privilege log on to ISQL, connect to SQLDBA and issue the following command:

```
REVOKE DBA FROM NAMESRV
```

Note: All the tests on the ITSC Raleigh network were done using SQL/DS V2R2 PUT 9105.

2.18.1.3 Configuring a Secondary Name Server

This is not very different from configuring a primary name server.

The main difference resides in the ″NSMAIN DATA″ file. The PRIMARY statement must be commented out, and the SECONDARY statement must be uncommented.

File "NSMAIN DATA"

```
; PRIMARY ITSC.RALEIGH.IBM.COM ITSC        1
SECONDARY IBM.COM IBM 9.67.38.17           2
  ...
```

1 This statement may be uncommented if your name server is acting as a primary name server for a zone and as a secondary name server for another zone.

2 The IBM.COM domain will be loaded from a remote primary name server (9.67.38.17) and will be loaded in the tables IBM0 and IBM1.

Please refer to Section 2.18.1.2, "Configuring a Primary Name Server" on page 146 for the complete sequence of the SQL commands.

2.18.2 Operating the Domain Name Server

2.18.2.1 Adding SQL Tables

In the ITSC Raleigh configuration, two VM systems have been set up in two steps. One system (VMESA) is a primary name server, the other system (VM14) is a secondary name server (it will download the SQL tables from VMESA).

1. First step:

   For this first step only one zone was defined: the ITSC.RALEIGH.IBM.COM zone.

   The "NSMAIN DATA" files were:

   - VMESA (Primary)

     ```
      ....
     PRIMARY ITSC.RALEIGH.IBM.COM ITSC
     NEGATIVECACHING
     STANDARDQUERYCACHE     100
      ....
     ```

   - VM14 (Secondary)

     ```
      ....
     SECONDARY ITSC.RALEIGH.IBM.COM ITSC 9.67.38.65
     NEGATIVECACHING
     STANDARDQUERYCACHE     100
      ....
     ```

   It's not required to have the same name for the SQL tables in the two VM systems.

   At startup VM14 downloads the ITSC SQL from VMESA into its own ITSC table.

2. Second step:

   We added two more SQL tables on the two VM systems to take care of the in-addr.arpa domain.

The ″NSMAIN DATA″ files have been changed to:

- VMESA (Primary)

```
 ....
PRIMARY ITSC.RALEIGH.IBM.COM ITSC
PRIMARY 38.67.9.IN-ADDR.ARPA ARPA
NEGATIVECACHING
 ....
```

- VM14 (Secondary)

```
 ....
SECONDARY ITSC.RALEIGH.IBM.COM ITSC 9.67.38.65
SECONDARY 38.67.9.IN-ADDR.ARPA ARPA 9.67.38.65
NEGATIVECACHING
STANDARDQUERYCACHE     100
 ....
```

The only steps neccessary to load/define those new tables are:

a. Log on to user ID with DBA authority and issue the following command: GRANT DBA TO NAMESRV. This step is not required if NAMESRV still has the DBA authority.

b. Log on to NAMESRV and issue the NSTABLE command. This will create the new tables. The output will be something like:

```
nstable nsmain data a
Using input file nsmain data a
Table ITSC0 already exists
Table ITSC1 already exists
Creating table ARPA0
Creating table ARPA1
Ready; T=0.72/1.40 12:01:22
Ready; T=0.01/0.01 12:01:22
```

Now you are ready to load the new SQL tables in the primary name server (we used the ″MASTER IN-ADDR″ file shown on page 153 for the ARPA zone and the ″MASTER IBM-COM″ file shown on page 152 for the zone ITSC.RALEIGH.IBM.COM).

2.18.2.2 Updating SQL Tables

From NAMESRV: Obviously updating can be achieved from the name server virtual machine itself using the following sequence:

1. Log on to the NAMESRV virtual machine.
2. Issue the #CP EXT command which stops the name server:

```
LOGON NAMESRV
 ICH70001I NAMESRV  LAST ACCESS AT 12:41:12 ON MONDAY, JUNE 15, 1992
There is no logmsg data
FILES: 0001 RDR, 0008 PRT,   NO PUN
RECONNECTED AT 12:55:31 EDT MONDAY 06/15/92
B

CP EXT
freeing primary and secondary links
freeing cache
12:41:25  ns:      purging the entire cache
          ns:      purging the entire cache
          ns:      purging the entire cache
          ns:      purging the entire cache
freeing SOANS list
exiting program
12:41:26  Name server shut down with CP EXT.
Command †NSMAIN † ended with return code 0.
Terminating server startup at your request with Rc: 0 !!  1
Ready; T=0.49/0.90 12:41:27
```

1 This message is issued if the "TCPRUN EXEC" has been modified as indicated on page 75.

3. Update the "MASTER DATA" file.
4. Load the updates in the SQL database via the **NSDBLOAD** command.
5. Flip the tables via the **SMSG** interface. Refer to Section 2.18.2.3, "The SMSG Interface to the Domain Name Server" on page 160 for more details.

From TCPMAINT: The previous solution had one important drawback. The name server must be stopped in order to update the tables. This may be unacceptable in many environments because it may prevent users from communicating (remember the name server is used every time a host name or an IP address must be resolved). Thus, it may be mandatory to be able to update and flip the SQL tables without stopping the name server. The two following sequences assume that the name server is up and running and that the updates will be performed from the TCPMAINT user ID.

The first sequence has not been tested at ITSC Raleigh. The second one has been successfully tested.

Please refer to Section 2.18.1.2, "Configuring a Primary Name Server" on page 146 for details about some of these commands.

First sequence:

1. From ISQL: CONNECT SQLDBA INDENTIFIED by Password
2. GRANT SELECT ON SYSTEM.SYSCATALOG TO NAMESRV WITH GRANT OPTION
3. Exit ISQL
4. Either the CSQL and ASMSQL files must be prepped from user ID NAMESRV or the "NSPREP EXEC" must be modified by adding the parameter: *userid=namesrv/password* after each *prepname=*. For example preppname(prepname=nsdbload,userid=namesrv/password
5. From ISQL: CONNECT NAMESRV IDENTIFIED BY password
6. GRANT RUN ON NAMESRV.NSDBLOAD TO TCPMAINT
7. From NAMESRV: GRANT ALL ON NAMESRV.*tables* to TCPMAINT. GRANT ALL ON NAMESRV.TCPIP TO TCPMAINT.

Second sequence:

We have also been able to update the SQL tables from the TCPMAINT user ID using the following sequence:

1. Log on to the TCPMAINT user ID.
2. Have a link/access to the SQL minidisk for TCPMAINT.
3. Get the necessary LOADLIBS and TXTLIBS: `GLOBAL LOADLIB EDCLINK, GLOBAL TXTLIB IBMLIB EDCBASE`.
4. To avoid some `ARIxxx` error messages, copy the ″SQLUSER PROFILE″ from the SQL minidisk as ″SQLUSER EXEC″. Check the language set in the ″SQLUSER EXEC″.
5. Remove the access to the SQL disk (do not detach it), and execute ″SQLUSER EXEC″. This will access the SQL disk again.
6. `SQLINIT DB(SQLDBA)`. Check the name of the SQL database.
7. `ISQL`.
8. `CONNECT SQLDBA IDENTIFIED BY Password`.
9. `Exit`. To exit ISQL.
10. Log off from the TCPMAINT virtual machine.
11. Log on to the NAMESERV virtual machine.
    - `GRANT ALL ON NAMESRV.tables to TCPMAINT`. Issue this command for all the tables you want to be able to update (ITSC0, ITSC1)
    - `GRANT ALL ON NAMESRV.TCPIP to TCPMAINT`
    - `GRANT DBA to TCPMAINT`

    This will allow TCPMAINT to update the tables. `NAMESRV.TCPIP` is a table where a list of the SQL tables currently used by the name server is kept. Here is the ouput of the ISQL command `SELECT * FROM SYSTEM.SYSCATALOG WHERE CREATOR=¢NAMESRV¢`, assuming ITSC and ARPA are the prefixes of the tables used by NAMESRV when it is acting as primary name server.

```
TNAME              CREATOR   TABLETYPE   NCOLS  REMARKS      DBSPACEN
-----------------  --------  ---------  ------  -----------  --------
ITSC0              NAMESRV   R               5                      1
ITSC1              NAMESRV   R               5                      1
TCPIP              NAMESRV   V               5  ITSC1 ARPA1   1
ARPA0              NAMESRV   R               5                      1
ARPA1              NAMESRV   R               5                      1
* END OF RESULT *** 5 ROWS DISPLAYED ***COST ESTIMATE IS 1***********
```

1 Currently used tables.

Now that all the authorities have been granted you are ready to update the tables using the following sequence (please refer to Section “From NAMESRV” on page 157 for more information):

12. Log on to TCPMAINT.
13. Update the master file(s).
14. `NSDBLOAD NAMESRV.ITSC` if you want to update the ITSC table. Do not forget the `NAMESRV` prefix.

15. Based on the table NAMESRV.TCPIP the NSDBLOAD procedure has updated the ITSC table which is not in use. In order to use the recently loaded ITSC table type: SMSG NAMESRV FLIPTABLE ITSC.

2.18.2.3 The SMSG Interface to the Domain Name Server

The VM Send Message (SMSG) interface can be used to obtain information about the name server, to debug through trace facilities, purge cache entries, change SQL tables and close the name server console log. The following command displays the available subcommands and their options.

```
smsg namesrv help

Name Server commands
--------------------
LIST (all) (NS cache) (STandard cache) (INverse cache)
PURGE (domain name) (all) - purge single or all entries
STATS  - Returns statistical information
TRACE (ALL) (SUBroutines) (STOrage) (QUEue) (ZONE) (END)
FLIPTABLE <sqltable> - Informs the NS which table to use

CLOSECON  - close console
COMMIT    - close all outstanding SQL connections
()- optional parameter <> - required parameter
```

The SMSG COMMIT command issues the SQL command COMMIT WORK RELEASE. This command causes the termination of the connection to the SQL database, so that the SQL database server can be taken down.

The syntax of the SMSG COMMIT is:

SMSG namesrv COMMIT, where *namesrv* is the name of the name server virtual machine.

We already mentioned the fliptable option to change the table that the name server is using to find information about address mapping.

Once you use the fliptable option, the name of the table in use after the flip, is stored in the SQL database. Therefore it is not necessary to repeat this command after a shutdown of the name server. The last table used will be the same after a new startup.

The following are some different examples:

```
smsg namesrv stats   1
Ready;
 From RALYESA(NAMESRV): NS start time:            Wed Jul 15 11:24:28
 From RALYESA(NAMESRV): ------------------------------------------------
 From RALYESA(NAMESRV): Total number of queries:  265    2
 From RALYESA(NAMESRV): Answers from cache:       30   (11%)   3
 From RALYESA(NAMESRV): Size of cache:            100  used: 3

smsg namesrv list  ST   5
Ready;
From RALYESA(NAMESRV): Standard query cache
From RALYESA(NAMESRV): Type Class Used Orig.TTL  Rem.TTL Entry
From RALYESA(NAMESRV): ------------------------------------------------------
From RALYESA(NAMESRV): PTR  IN       0   143613   142898 -84.38.67.9.in-addr.arpa        6
From RALYESA(NAMESRV): PTR  IN       0   143613   142899 -64.38.67.9.in-addr.arpa
From RALYESA(NAMESRV): A    IN       0   143619   142970 -localhost.itsc.raleigh.ibm.com
```

1 Asking for statistics

2 Number of queries received by the name server

3 How many were answers from the cache

5 Asking for the content of the standard cache

6 Seems that only invalid queries are kept there.

The SMSGUSERFILE command in the ″NSMAIN DATA″ file specifies the file to use when verifying whether a user is authorized for a particular SMSG command. The default file name is ″VALIDUSR EXEC″, which is located at the common server minidisk (TCPMAINT 591).

Here is an example of what you need to change on the ″VALIDUSR EXEC″ to include the TCPMAINT and MAINT virtual machines from another VM system as authorized SMSG users.

File ″VALIDUSR EXEC″

```
/*********************************************************************/
/*  Create the data table of userids that are allowed to issue SMSG  */
/*  commands to the name server.  Use data.0 to define the special   */
/*  userid †SMSGDEFAULT† that lists the commands that all users are  */
/*  allowed to issue.  Use a nodeid of †THISRSCSNODE† for users at   */
/*  at this node, which is determined by the IDENTIFY command.       */
/*                                                                   */
/*  The correct format is:                                           */
/*      data.x = †nodeid userid command(s)†                          */
/*                                                                   */
/*  The possible commands are:                                       */
/*      HELP, LIST, PURGE, STATS, TRACE, FLIPTABLE, CLOSECON, COMMIT */
/*********************************************************************/
data.0 = †THISRSCSNODE SMSGDEFAULT HELP            STATS
data.1 = †THISRSCSNODE TCPMAINT    HELP LIST PURGE STATS TRACE FLIPTABLE  1
data.2 = †RAL9390 TCPMAINT HELP LIST PURGE STATS TRACE FLIPTABLE  2
         CLOSECON COMMIT†;
data.3 = †RAL9390 MAINT HELP LIST STATS TRACE                     3
         CLOSECON COMMIT†;
```

1 Authorization for user ID TCPMAINT on the local VM system. The default authorizations are shown.

2 Authorization for user ID TCPMAINT of the VM system called RAL9390.

2 Authorization for user ID MAINT of the VM system called RAL9390. MAINT has less authority than TCPMAINT; thus, some commands have been removed from the list.

2.18.3 Checking Name Servers

2.18.3.1 Log On the NAMESRV Virtual Machine

This is a good way to check what's going on. Once logged on you can check what the name server receives and answers. Following are some examples.

Here is the answer for a *ping alfred* from workstation 9.67.38.98:

```
16:23:07  Q:    1 from 9.67.38.98      :received
                 name=alfred.itsc.raleigh.ibm.com type=1
          Q:    1 from 9.67.38.98      :answered
```

The next example is the output of a mount command issued from station 9.67.38.98 (fred) to station 9.67.38.71 (RS60001).

```
16:24:49  Q:    1 from 9.67.38.98      :received
                 name=rs60001.itsc.raleigh.ibm.com type=1
          Q:    1 from 9.67.38.98      :answered
16:24:55  Q:    2 from 9.67.38.98      :received
                 name=rs60001.itsc.raleigh.ibm.com type=1
          Q:    2 from 9.67.38.98      :answered
16:24:56  Q:    2 from 9.67.38.71      :received      1
                 name=98.38.67.9.in-addr.arpa type=12
          Q:    2 from 9.67.38.71      :answered
          Q:    2 from 9.67.38.98      :received
                 name=rs60001.itsc.raleigh.ibm.com type=1
          Q:    2 from 9.67.38.98      :answered
          Q:    2 from 9.67.38.98      :received
                 name=rs60001.itsc.raleigh.ibm.com type=1
          Q:    2 from 9.67.38.98      :returned cached entry  2
          Q:    2 from 9.67.38.98      :received
                 name=rs60001.itsc.raleigh.ibm.com type=1
          Q:    2 from 9.67.38.98      :returned cached entry
          Q:    2 from 9.67.38.98      :received
                 name=rs60001.itsc.raleigh.ibm.com type=1
          Q:    2 from 9.67.38.98      :returned cached entry
```

There we can see that the name server caches entries (2) and uses the in-addr.arpa domain (1). The NFS server of the station 9.67.38.71 (rs60001) asks for the name of the station 98.38.67.9.in-addr.arpa. If the in-addr.arpa domain was not defined the result of the mount command would have been: *Access denied by server*.

2.18.3.2 The TRACE RESOLVER

This trace command is very useful for checking the configuration of the name server(s) you are using. To have this trace enabled perform the following:

1. Copy the file "TCPIP DATA" to your minidisk.
2. Do not change anything but add a *TRACE RESOLVER* command at the end of the file:

```
 .....
RESOLVERUDPRETRIES 1
 .....
TRACE RESOLVER
;
```

Note: The file "TCPIP DATA" is used by any client application and can be customized according to your own taste. For example you may want to temporarily use a host file instead of a name server (just comment out all the NSxxxx statements) or add a *TRACE* command.

Here is an output of a ping command when the TRACE RESOLVER is enabled:

```
ping vm14
Userid of Caller:         TCPMAINT
TCP Host Name:            RALYESA
Domain Origin:            ITSC.RALEIGH.IBM.COM
Userid of TCP/IP VM:      TCPIP
Communicate Via:          UDP
OpenTimeOut:              30
MaxRetrys:                1
NSPort:                   53
Name Server  Userid:        NAMESRV
NSInternetAddress(.1.) := 14.0.0.0        1
NSInternetAddress(.2.) := 9.67.38.66
NSInternetAddress(.3.) := 9.67.38.71

Resolving Name:           VM14            2
Result from InitResolver: OK
Building Name Server Query:
* * * * * Beginning of Message * * * * *
Query Id:                 1
Flags:                    0000 0001 0000 0000
Number of Question RRs:   1
Question   1: VM14.ITSC.RALEIGH.IBM.COM A IN
Number of Answer RRs:     0
Number of Authority RRs:  0
Number of Additional RRs: 0
* * * * * End of Message * * * * *
Sending Query to Name Server at 14.0.0.0 Result: OK     3
Notification Arrived: UDP data delivered RC = OK
UDP Data Length: 84
Return from WaitForAnswer: OK
* * * * * Beginning of Message * * * * *
Query Id:                 1
Flags:                    1000 0101 1000 0000
Number of Question RRs:   1
Question   1: VM14.ITSC.RALEIGH.IBM.COM A IN
Number of Answer RRs:     1
Answer     1: vm14.itsc.raleigh.ibm.com 0 A IN 9.67.38.66    4
Number of Authority RRs:  0
Number of Additional RRs: 0
* * * * * End of Message * * * * *
HostNumber (1) is: 9.67.38.66                 5
Ping V2R2: Pinging host VM14  (9.67.38.66).  Enter #CP EXT to interrupt.
PING: Ping #1 response took 0.092 seconds. Successes so far 1.
Ready;
```

1 List of name server that may be queried (read from ″TCPIP DATA″).

2 Name queried.

3 Query sent to name server at address 14.0.0.0 (first in the list).

4 Answer is found in this Resource Record.

5 Answer is sent back.

2.18.3.3 The DIG Command

DIG is a program for querying name servers. The DIG program allows you to:

- Exercise name servers
- Gather large volumes of domain name information
- Execute simple domain name queries.

You can use DIG in command line mode, where all options are specified on the invoking command line, or in batch mode, where a group of queries are placed in a file and are executed by a single invokation of DIG.

The following is the format of the **DIG** command:

Format of the DIG Command

```
DIG @server-name domain-name    qtype qclass %comment queryoption digoption
    @server-address
```

Note: The character @ is normally used as a delete character. If you intend to execute the command above, first issue the command TERMinal CHardel OFF.

An Example of Using the DIG Command: On this first command, we will ask the name server in VMESA for information about the zone ITSC.RALEIGH.IBM.COM. VMESA is the name server responsible for that zone.

```
dig @vmesa ibm.com

;      DiG 2.0      @vmesa itsc.raleigh.ibm.com
;; -  HEADER  - opcode: QUERY , status: NOERROR, id: 4
;; flags: qr aa rd ra ; Ques: 1, Ans: 0, Auth: 0, Addit: 1
;; QUESTIONS:
;;      itsc.raleigh.ibm.com, type = A, class = IN

;; ADDITIONAL RECORDS:
itsc.raleigh.ibm.com.  0  SOA  ralyesa.itsc.raleigh.ibm.com.  tcpmaint.
ralyesa.itsc.raleigh.ibm.com. (
                        901215  ;serial
                        901215  ;refresh
                        901215  ;retry
                        901215  ;expire
                        0 )     ;minim


;; Sent 1 pkts, answer found in time: 18 msec
;; FROM: vmesa to SERVER: vmesa  9.67.38.65
;; WHEN: Thu Jul  2 18:26:53 1992
;; MSG SIZE  sent: 38  rcvd: 139

Ready;
```

The answer shows that the name server responsible for this zone is on the RALYESA host.

Now let's ask the same question of the VM14 host. VM14 is a secondary name server for the ITSC.RALEIGH.IBM.COM zone.

```
dig @vm14 itsc.raleigh.ibm.com

;      DiG 2.0      @vm14 itsc.raleigh.ibm.com
;; -  HEADER  - opcode: QUERY , status: NOERROR, id: 4
;; flags: qr aa rd ra ; Ques: 1, Ans: 0, Auth: 0, Addit: 1
;; QUESTIONS:
;;      itsc.raleigh.ibm.com, type = A, class = IN

;; ADDITIONAL RECORDS:
ITSC.RALEIGH.IBM.COM.   0     SOA   RALYESA.ITSC.RALEIGH.IBM.COM.  TCPMAINT.
RALYESA.ITSC.RALEIGH.IBM.COM. (
                        901215  ;serial
                        901215  ;refresh
                        901215  ;retry
                        901215  ;expire
                        0 )     ;minim
;; Sent 1 pkts, answer found in time: 30697 msec
;; FROM: vmesa to SERVER: vm14  9.67.38.66
;; WHEN: Thu Jul  2 18:20:32 1992
;; MSG SIZE  sent: 38  rcvd: 159

Ready;
```

The name server on the VM14 host has the same information, so we now know that the two name servers have the same information about the ITSC.RALEIGH.IBM.COM zone.

DIG has a lot of options and you should refer to *IBM TCP/IP V2 R2 for VM: User's Guide* for more information about the DIG command. The following are some examples:

```
dig @vmesa itsc.raleigh.ibm.com +noques  +noHeader +noheader +nostats

;      DiG 2.0      @vmesa itsc.raleigh.ibm.com +noques +noHeader +noheader
+nostats
; Ques: 1, Ans: 0, Auth: 0, Addit: 1
;; ADDITIONAL RECORDS:
itsc.raleigh.ibm.com.   0       SOA     ralyesa.itsc.raleigh.ibm.com.  tcpmaint
ralyesa.itsc.raleigh.ibm.com. (
                        901215  ;serial
                        901215  ;refresh
                        901215  ;retry
                        901215  ;expire
                        0 )     ;minim


Ready;
```

Let's suppose that the previous query is the kind of output you want. You can save this environment in a file "DIG ENV A" via the following command:

```
dig @vmesa itsc.raleigh.ibm.com +nostats +noHeader +noques +noheader -envsav

;     DiG 2.0      @vmesa itsc.raleigh.ibm.com +nostats +noHeader +noques +nohe
ader -envsav
; Ques: 1, Ans: 0, Auth: 0, Addit: 1
;; ADDITIONAL RECORDS:
itsc.raleigh.ibm.com.   0       SOA     ralyesa.itsc.raleigh.ibm.com.  tcpmaint
ralyesa.itsc.raleigh.ibm.com. (
                        920702  ;serial
                        920702  ;refresh
                        920702  ;retry
                        920702  ;expire
                        0 )     ;minim


Ready;
```

The next time you query the ITSC.RALEIGH.IBM.COM domain using the DIG command the same environment will be set up for you. The following is an example of DIG command after the -envsav statement has been typed together with the options mentioned earlier:

```
dig @vmesa itsc.raleigh.ibm.com

;      DiG 2.0      @vmesa itsc.raleigh.ibm.com
; Ques: 1, Ans: 0, Auth: 0, Addit: 1
;; ADDITIONAL RECORDS:
itsc.raleigh.ibm.com.   0       SOA   ralyesa.itsc.raleigh.ibm.com.  tcpmaint.
ralyesa.itsc.raleigh.ibm.com. (
                        920702  ;serial
                        920702  ;refresh
                        920702  ;retry
                        920702  ;expire
                        0 )     ;minim


Ready;
```

Warning

Before using the **DIG** command, you must make sure that your user ID has access to the IBM C/370 run-time libraries.

2.18.3.4 The NSLOOKUP Command

NSLOOKUP is a program for querying name servers. The **NSLOOKUP** program allows you to:

- Locate information about network nodes
- Examine the contents of a name server database
- Establish the accessibility of name servers.

You can use the **NSLOOKUP** interactively to issue multiple queries in a **NSLOOKUP** session, or specify an individual query on the invoking command line.

Format of the NSLOOKUP Command

```
NSLOOKUP Environment
         Domain-name Server-name
         Domain-address Server-address
```

The NSLOOKUP command has many subcommands. The SET subcommand changes the internal state information which affects the operation and the results of your queries. The parameters for the SET subcommand can be either entered on the command line, or be written in the ″NSLOOKUP ENV″ file which is read by the NSLOOKUP program. A basic example of ″NSLOOKUP ENV″ could be:

```
querytype=CNAME       1
nodebug               2
```

- 1 Type of information returned by queries. The initial value is A, address information is returned.
- 2 No debug.

NSLOOKUP, individual query: The query sent is about the domain ITSC.RALEIGH.IBM.COM and the name server queried is on the system named VMESA. The nodebug option has been specified in the ″NSLOOKUP ENV″:

```
Ready;
nslookup itsc.raleigh.ibm.com vmesa
Server:  ralyesa.itsc.raleigh.ibm.com
Address:  9.67.38.65

itsc.raleigh.ibm.com
        origin = ralyesa.itsc.raleigh.ibm.com
        mail addr = tcpmaint.ralyesa.itsc.raleigh.ibm.com
        serial = 920704
        refresh = 920704 (10 days 15 hours 45 mins 4 secs)
        retry   = 920704 (10 days 15 hours 45 mins 4 secs)
        expire  = 920704 (10 days 15 hours 45 mins 4 secs)
        minimum ttl = 0 (10 days 15 hours 45 mins 4 secs)
Ready;
nslookup fred vmesa
Server:  ralyesa.itsc.raleigh.ibm.com
Address:  9.67.38.65

Name:    fred.itsc.raleigh.ibm.com
Address:  9.67.38.98

Ready;
```

This information has been set in the ″MASTER DATA″ file.

NSLOOKUP, interactive queries: To use NSLOOKUP interactively type:

NSLOOKUP - Server_name or NSLOOKUP:

```
nslookup - vmesa     1
Default Server:  ralyesa.itsc.raleigh.ibm.com     2
Address:  9.67.38.65


fred                    3
Server:  ralyesa.itsc.raleigh.ibm.com
Address:  9.67.38.65

Name:    fred.itsc.raleigh.ibm.com
Address:  9.67.38.98               4


ls -t CNAME itsc.raleigh.ibm.com     5
Yralyesa.itsc.raleigh.ibm.com†
 raianje                       mvs18.itsc.raleigh.ibm.com   6
 ral9390                       vm14.itsc.raleigh.ibm.com
 vmesa                         ralyesa.itsc.raleigh.ibm.com
 vm15                          ral9360.itsc.raleigh.ibm.com
```

- 1 The server queried is vmesa.
- 2 ralyesa appears here because of the CNAME entry in the "MASTER DATA" file.
- 3 Asking for the address of host fred.
- 4 Answer from the name server.
- 5 Query sent is about the CNAME entries for hosts in the domain itsc.raleigh.ibm.com.
- 6 Only hosts with a CNAME entry are listed.

The output of an ls command can be written into a CMS file. This file can be viewed either from NSLOOKUP (with the view command) or from CMS:

```
nslookup - vmesa
Default Server:  ralyesa.itsc.raleigh.ibm.com
Address:  9.67.38.65


ls -t CNAME itsc.raleigh.ibm.com   out.put
Yralyesa.itsc.raleigh.ibm.com†
Received 23 records.

view out.put

 raianje                       mvs18.itsc.raleigh.ibm.com
 ral9390                       vm14.itsc.raleigh.ibm.com
 vmesa                         ralyesa.itsc.raleigh.ibm.com
 vm15                          ral9360.itsc.raleigh.ibm.com
  ls -t CNAME itsc.raleigh.ibm.com
Yralyesa.itsc.raleigh.ibm.com†
```

Another kind of query could be:

```
nslookup - vmesa
Default Server:  ralyesa.itsc.raleigh.ibm.com
Address:  9.67.38.65


ls -t PTR 38.67.9.in-addr.arpa
Yralyesa.itsc.raleigh.ibm.com†
 133                            host = mvs20.itsc.raleigh.ibm.com
 135                            host = mvs18.itsc.raleigh.ibm.com
 65                             host = ralyesa.itsc.raleigh.ibm.com
 66                             host = vm14.itsc.raleigh.ibm.com
 67                             host = vm15.itsc.raleigh.ibm.com
 68                             host = dos20.itsc.raleigh.ibm.com
 69                             host = os25.itsc.raleigh.ibm.com
 70                             host = osx25.itsc.raleigh.ibm.com
 71                             host = rs60001.itsc.raleigh.ibm.com
 72                             host = rs60002.itsc.raleigh.ibm.com
 73                             host = rs60003.itsc.raleigh.ibm.com
 81                             host = paul.itsc.raleigh.ibm.com
 86                             host = sergio.itsc.raleigh.ibm.com
 89                             host = paulb.itsc.raleigh.ibm.com
 98                             host = fred.itsc.raleigh.ibm.com

exit
Ready;
```

Warning

Before using the **NSLOOKUP** command, you must have access to the IBM C370 191 disk and issue these commands: ″GLOBAL TXTLIB IBMLIB″ and ″GLOBAL LOADLIB EDCLINK″.

Refer to *IBM TCP/IP V2 R2 for VM: User's Guide* for more information about the **NSLOOKUP** command.

2.19 VM/CMS User ID for a TCP/IP User

There are some special requirements for a user ID that enable it to use the TCP/IP client programs. The following example shows a regular VM/CMS user with access to TCP/IP.

Directory Entry "LESIA DIRECT"

```
USER LESIA LESIA 4M 6M G
 ACCOUNT 1 LESIA
 OPTION ECMODE
 IPL CMS PARM AUTOLOG
 CONSOLE 009 3215 T OPERATOR
 SPOOL 00C 2540 READER *
 SPOOL 00D 2540 PUNCH A
 SPOOL 00E 1403 A
 LINK MAINT    190 190 RR
 LINK MAINT    19E 19E RR
 LINK TCPMAINT 592 592 RR                          1
 MDISK 191 FB-512 16 5900 VM9304 MR Rpw Wpw Mpw    2
```

1. All TCP/IP client programs, important to a regular user, reside on this minidisk.
2. If the VM/CMS user wants to allow remote users to access this 191 minidisk via FTP or NFS, minidisk passwords are required (not required if you use RACF).

File "PROFILE EXEC"

```
/*****************************************************/
/* example profile exec for a tcp/ip user            */
/*****************************************************/
trace e
¢cp set pf12    retrieve¢
¢cp spool con  start to *¢
¢cp spool print rscs cl a¢
¢cp tag dev prt ralydpd4 itsc 99 sysout=a¢
¢acc 592 r¢                                          1
¢set ldrtbls 10¢                                     2
¢cruntime¢                                           3
¢global loadlib edclink¢                             4
¢global txtlib rpclib commtxt ibmlib cmslib edcbase¢  4
```

1. This statement accesses the TCP/IP minidisk. A link statement in the directory or a previous link statement in the profile is required.
2. The default number of loader tables might be too low for some clients (FTP for example).
3. The CRUNTIME procedure establishes the link and accesses the C/370 minidisk.
4. Some client programs, or your own written TCP/IP applications, may need access to the C/370 libraries. This is optional for a regular TCP/IP user, but it is required for TCP/IP programming and the execution of the TCP/IP API programs.

2.20 RACF Considerations

Effective user verification can be offered by the Resource Access Control Facility (RACF). RACF uses the user ID and a system-encrypted password to perform its identification and verification.

This section will only discuss the TCP/IP servers that can use RACF to validate user access and the way they should be customized to use RACF.

2.20.1 FTP Interface to RACF

If you use VM/SP with RACF, additional customization is required. RACF has an alternate user ID feature, which allows the FTP server to act as a surrogate (alternate) user ID for other users. This means that the FTP server can access the required disk on behalf of the user ID which is entered during the FTP session establishment. If RACF is installed on your VM system, you must modify the FTP server exit ″FTPDEXIT EXEC″ to include the *RACF* option:

```
00085 parms = ¢RACF¢                /* Optional server arguments */
```

In order to authorize FTPSERVE to access a user's minidisk one should use the ″FTPPERM EXEC″ or issue the appropriate RACF commands.

Note: the ″FTPPERM EXEC″ assumes that a VMBATCH RACF class exists and that FTPSERVE is defined to this class (the command is RDEF VMBATCH FTPSERVE UACC(NONE)). Please check this with the RACF administrator. Actually, the ″FTPPERM EXEC″ issues the following RACF commands:

```
PERMIT¢ userid ¢CLASS(VMBATCH)¢ access ¢ID(¢vm¢)¢
```

The *access* keyword may be either:

- DELETE (removes the surrogate capability for FTPSERVE).
  You can also use the FTPPERM DEL userid command.
- ACCESS(CONTROL) (allows FTPSERVE to act as a surrogate user ID). You can also use the FTPPERM ADD userid command.

This command is included in a loop so that the *vm* keyword can stand for all the FTP servers that you defined in your system. You must code all the names of the different FTP servers in the ″FTPPERM EXEC″:

```
00016 ftpserve = ¢FTPSERVE FTPSERV1 FTPSERV2¢
```

All these servers should be defined to the class VMBATCH.

When the RACF option is chosen, when FTPSERVE is started, the ″FTPDRACF EXEC″ is called. This EXEC calls the ″RACFLINK EXEC″ with the INIT option. Depending on the VM/SP release level, RACFLINK (INIT will copy either ″SURROGAT EXEC″ or ″ALTERNAT EXEC″ to ″$ALTUSER EXEC″, which will be used to perform either the surrogate or the alternate user ID. The ″ALTERNAT MODULE″ and the ″RPIVAL MODULE″ are loaded. Finally the FTP server is started with the command SRVRFTP RACF.

When a TCP/IP user establishes a connection in order to transfer files with the VM FTP server, he has to type a user ID and a password. The user ID and the password will be validated via the ″VALIDATE EXEC″ which calls the ″RPIVAL MODULE″. If the user ID and password are valid then the ″RACFLINK EXEC″ is called to perform the link.

The following is the result of a RACF query. The class queried is VMBATCH which is the class required to act as alternate. In this example FTPSERVE is authorized to act as alternate for TCPMAINT.

```
id
FTPSERVE AT RALYESA  VIA RJE      06/17/92 09:21:30 EDT      WEDNESDAY
Ready; T=0.01/0.01 09:21:30
ICH70001I FTPSERVE LAST ACCESS AT 09:15:48 ON WEDNESDAY, JUNE 17, 1992
RPITMP001I RACF/VM SESSION ESTABLISHED. TO TERMINATE ENTER †END†
RPITMP002I ENTER RACF COMMAND OR †END† TO EXIT
rl vmbatch * all

CLASS      NAME
-----      ----
VMBATCH    TCPMAINT

LEVEL  OWNER      UNIVERSAL ACCESS  YOUR ACCESS  WARNING
-----  --------   ----------------  ---- ------  -------
 00    TCPMAINT        NONE            CONTROL    NO
```

2.20.2 VMNFS Interface to RACF

If RACF is installed on your VM system, you must modify the VMNFS server exit "VMNFSXIT EXEC" to include the *RACF* option:

```
00085 parms = ¢R¢                    /* Optional server arguments */
```

Then perform the following sequence:

1. Assign the VM class neccessary to use Diagnose D4 to the VMNFS virtual machine in the CP directory. For example the VMNFS virtual machine could be assigned the VM classes: ABG.
2. The VMNFS virtual machine must be in a RACF group that has the privilege of executing surrogate requests for another user.
3. Copy the RACF "VALIDATE MODULE" or "RPIVAL MODULE" as "VERIFYPW MODULE" on VMNFS.191.
4. The part of the **MOUNT** command transmitted to the server must include the **user** keyword. For example a valid sequence (in order to use RACF) from an TCP/IP V1.2.1 for OS/2 NFS client would be:

   ```
   [C:]mount -v x: vmesa:tcpmaint.2c0,user=tcpmaint,ro
   mount: vmesa:tcpmaint.2c0,user=tcpmaint,ro
   Enter password:

   NFS Drive ¢x:¢ was attached successfully.
   ```

2.20.3 REXEC Interface to RACF

If RACF is installed on your VM system, you must modify the REXECD server exit ″REXECXIT EXEC″ to include the *RACF* option:

```
00085 PARMS = ¢-r -s RSLAVE1¢              /* Optional server arguments */
```

The `-s RSLAVE1` identifies a slave machine. It's optional. If a password was specified on our VM/ESA system, REXECD was not able to start.

Then perform the following sequence:

1. Assign the VM class neccessary to the REXECD virtual machine in the CP directory. For example the REXECD virtual machine could be assigned the VM classes A, B and G.
2. The VMNFS virtual machine must be in a RACF group that has the privilege of executing surrogate requests for another user.
3. Copy the RACF ″VALIDATE MODULE″ as ″RPIVAL MODULE″ on a minidisk accessed by REXECD. If your system already has a ″RPIVAL MODULE″ be sure it is accessed by REXECD.
4. Modify the ″VALIDATE EXEC″ in order to *NUCXLOAD* the ″RPIVAL MODULE″. If you do not *NUCXLOAD* the ″RPIVAL MODULE″, REXECD will abend. Following is a part of the ″VALIDATE EXEC″. Numbers on the left are line numbers:

```
00014 ¢QUERY CMSTYPE ( STACK LIFO¢
00015 pull . . cmstype
00016 ¢SET CMSTYPE HT¢
00017 ¢NUCXLOAD RPIVAL MODULE¢ /* REQUIRED !! */
00018 ¢RPIVAL¢ args
00019 status=rc
```

Chapter 3. Using TCP/IP between Two VM Systems

To use TCP/IP between two VM systems can be an alternative when a TCP/IP backbone network already exists and no SNA facilities are available.

3.1 File Transfer Protocol (FTP)

The following panels are an example of a typical FTP session.

3.1.1 Login Sequence with FTP

A file transfer is invoked using the FTP command. The internet address or a symbolic name can be used as a parameter to indicate the remote system. If no parameter is given, the user will be prompted to supply the remote system address. FTP then tries to establish a TCP connection to the remote TCP/IP host on the reserved port 21. The remote FTP server sends a welcome message and asks for a valid user ID and password. After verifying authorization, the user is required to supply a minidisk password for the 191 minidisk. If the password is correct, the so-called control connection is now established; the user can then use all the FTP subcommands and interact with the remote server.

```
ftp vm15
VM TCP/IP FTP V2R2
Connecting to VM15 9.67.32.19, port 21
220-FTPSERVE at RAL9360.ITSC.RALEIGH.IBM.COM, 16:19:36 EDT THURSDAY 11/15/90
220 Connection will close if idle for more than 5 minutes.
USER (identify yourself to the host):
frick
   USER frick
331 Send password please.
Password:
frick
   PASS ********
332-FRICK logged in; no working directory defined
332 to access FRICK 191, send ¢ACCOUNT minidisk-password¢
Account:
mfrick
   ACCT ********
230 Working directory is FRICK 191
Command:
```

- If RACF is installed on your VM system (and if the FTP server has been customized to use RACF), you do not need to type a password for any minidisk you want to access. Access to minidisks will be granted based on the user ID used during the connection process.
- If RACF is not installed on your VM system, you need the minidisk password in order to access a minidisk. The FTP user is automatically prompted to supply the minidisk password. It is necessary to define a link password; otherwise, the disk cannot be accessed with FTP. If the link password is set to ALL, the link is done immediately, and no password is required.

  Note: No passwords are displayed on the real screens.

3.1.2 Getting Help

3.1.2.1 Client Help

Just type help on the command line:

```
Command:
help
User-FTP understands these commands:
  ?        acct      append    ascii    binary     cd       close
  cms      debug     delete    delimit  dir        ebcdic   euckanji
  get      help      ibmkanji  jis78kj  jis83kj    lcd      locsite
  locstat  lpwd      ls        mdelete  mget       mode     mput
  noop     open      pass      put      pwd        quit     quote
  rename   sendport  sendsite  site     sjiskanji  status   struct
  sunique  system    type      user
Specify a command by any unambiguous prefix
Specify a local file by name.type.mode or name.type
Default mode is * for PUT and a current local directory for MPUT, MGET and GET
For information about a particular command, say †HELP command†
Command:
```

Note: The cms FTP command allows you to issue cms commands on the client system. Please refer to Section 5.2.2, “FTP from VM to OS/2” on page 198 for an example.

3.1.2.2 Server Help

Just type quote help on the command line. The output will obviously depend on the FTP server. If the FTP server is a VM system site the output will be:

```
Command:
quote help
   help
214-The server-FTP commands are:
214-ABOR, ACCT,*ALLO, APPE,  CWD, DELE, HELP, LIST,* MKD, MODE
214-NLST, NOOP, PASS, PASV, PORT,  PWD, QUIT, REIN,*REST, RETR
214-RNFR, RNTO, SITE, SYST, STAT, STOR, STOU, STRU, TYPE, USER
214-The commands preceded by  ¢*¢  are unimplemented
214-The data representation type may be ASCII, EBCDIC or IMAGE. The data
214-structure must be File; the mode may be Stream or Block.
214-If the connection to this server is inactive for more than
214-300 seconds, the connection will be closed.
214-File identifiers have two components: the name and the type.
214-These components are separated by a period.
214-For information about a particular command, type
214 HELP SERVER command.
Command:
```

3.1.3 Receive/Send a File

The directory of the remote minidisk can be displayed using the dir subcommand. To receive a specific file the get subcommand is used. To receive a specific file the put subcommand is used. The get and put subcommands have two parameters, the local and the remote file name. For sending and receiving files the UNIX format must be used. File name, file type and file mode are separated by periods.

```
Command:
dir
   PORT 9,67,32,18,4,15
200 Port request OK.
   LIST
125 List started OK
ALL      NOTEBOOK V         80         17          1  3/28/90 12:18:48 FRI191
N458059A STG1     F         80        783         62  5/30/89 12:08:57 FRI191
PASSTHRU DATA     V         80         46          2  6/21/89 19:03:29 FRI191
PROFILE  EXEC     V         39         26          1  3/22/90 14:41:13 FRI191
PUT      MEMO     V         73        397         17  7/12/89 10:59:49 FRI191
XI08V2R1 STG1     F         80        223         18  5/30/89 11:58:37 FRI191
250 List completed successfully.
Command:
get put.memo put.newmemo
   PORT 9,67,32,18,4,16
200 Port request OK.
   RETR put.memo
150 Sending file ¢put.memo¢
250 Transfer completed successfully.
16816 bytes transferred. Transfer rate 12.94 Kbytes/sec.
Command:
```

To transfer a file, FTP establishes a temporary additional TCP connection. The number of bytes and the throughput rates are displayed after the file is transferred successfully. If the file is very large and it takes a long time to transmit, additional messages are generated to indicate that the file transfer is still in process.

3.1.4 Access a Minidisk

Another minidisk can be accessed using the *cd* subcommand. The format is: *user_id.cuu*, where cuu is the minidisk address of the minidisk to be linked. The *account* subcommand must be used to supply the minidisk password. FTP determines itself, according to the supplied password, if the user has read or write permission.

```
Command:
cd tcpmaint.191
   CWD tcpmaint.191
332 Supply minidisk password using ¢account¢
Account:
rtcp
   ACCT ********
230 Working directory is TCPMAINT 191 (ReadOnly)
Command:
pwd
   PWD
257 †TCPMAINT.191† is working directory (ReadOnly)
Command:
```

Again, if RACF is used no password is needed.

3.1.5 Invoke FTP within a Procedure

Sometimes it is more convenient to invoke a command by a procedure which supplies all the necessary parameters to log on, send or receive a file and exit FTP without the need to manually supply this information.

The following sample commands ***FTP1*** and ***FTP2*** allow the user to do a fully automated file transfer.

The necessary parameters for FTP are provided by the FTP1 and FTP2 procedures.

FTP1 EXEC

```
/* Sample REXX procedure to invoke FTP */
/* and supply parameters               */
trace e
arg cmd file
if cmd == ¢GET¢ then
  do
    queue  †frick frick rfrick†
    queue  †get† file †(replace†
    queue  †quit†
    ¢ftp vm15¢
    exit 0
  end
if cmd == ¢PUT¢ then
  do
    queue  †frick frick mfrick†
    queue  †put† file
    queue  †quit†
    ¢ftp vm15¢
    exit 0
  end
exit 12
```

Normally the results of your FTP dialog are printed on your terminal. However, if a disk file is defined as the output device, the dialog results go into the file.

This is an example of applying the FTP subcommands from a disk file.

FTP2 EXEC

```
/* FTP Exec Interface */
¢Filedef output disk test  file¢
¢Filedef input  disk input file¢
¢ftp vm15 (exit¢
Say ¢Ready(FTP rec=¢rc¢);¢
¢Filedef * CLEAR¢
```

The following is an example ″INPUT FILE″.

″**INPUT FILE**″

```
tcpmaint PASSWORD
cd tcpmaint.591
MdiskPass
get ftpdexit.exec
quit
```

This is the ″TEST FILE″, which contains the subsequent dialog log.

″**TEST FILE**″

```
VM TCP/IP FTP V2R2
conn VM15
Connecting to VM15 9.67.32.19, port 21
220-FTPSERV3 at RAL9360.ITSC.RALEIGH.IBM.COM, 12:02:28 EDT TUESDAY 11/19/91
220 Connection will close if idle for more than 5 minutes.
USER (identify yourself to the host):
log tcpmaint tcp
   USER tcpmaint
331 Send password please.
   PASS ********
230-TCPMAINT logged in; no working directory defined
230 to access TCPMAINT 191, send ¢ACCOUNT minidisk-password¢
Command:
cd tcpmaint.591
   CWD tcpmaint.591
332 Supply minidisk password using ¢account¢
Account:
   ACCT ********
230 Working directory is TCPMAINT 591
Command:
get ftpdexit.exec
   PORT 9,67,32,17,4,20
200 Port request OK.
   RETR ftpdexit.exec
150 Sending file ¢ftpdexit.exec¢
250 Transfer completed successfully.
10796 bytes transferred. Transfer rate 10.36 Kbytes/sec.
Command:
quit
   QUIT
221 Quit command received. Goodbye.
```

3.2 Telnet

Telnet allows a VM/CMS user to log on to a remote VM system in 3270 mode via a TCP/IP network. In this situation, ***transparent mode*** is used. That means no character translation is performed, and all 3270 functions are available (including GDDM graphics and National Language Support) except the PA1 key, which is used to enter Telnet control mode.

Note: The PA1 key can be sent using the PA1 subcommand in Telnet command mode.

3.3 Remote Execution (REXEC)

REXEC is very useful for sending a single command to a remote system. Remote system name, userid, password and the command string can be passed as parameters. It is much easier to use REXEC instead of logging on via Telnet, executing the command, and logging off.

3.3.1 Execute a Command on a Remote Site

The following example shows how a VM/CMS command can be executed on a remote VM system.

```
rexec -l frick -p frick ral9360 netstat gate
VM TCP/IP Netstat V2R2
Known gateways:

NetAddress      FirstHop        Link     Pkt Sz  Subnet Mask     Subnet Value
----------      --------        ------   ------  -----------     ------------
9.0.0.0          direct         ILALTRN  1000    0.255.255.240   0.67.32.16
9.67.32.34       direct         SNALVM14 2000    HOST
9.0.0.0         9.67.32.17      ILALTRN  Default 0.255.255.240   0.67.32.80
Ready; T=0.17/0.42 18:22:43

rexec -l frick -p frick ral9360 listfile * exec
F        EXEC     A1
LNKTCPRR EXEC     A1
PROFILE  EXEC     A1
QQ       EXEC     A1
REXTST   EXEC     A1
Ready; T=0.14/0.38 18:27:46
```

To use a slave machine on the VM server side type:

```
rexec -l guest -p guest ral9390 netstat gate
```

```
rexec -l guest -p guest ral9390 listfile * exec
```

REXEC could also be used for network management purposes. Procedures to stop and start TCP/IP devices (physical links) can be initiated using the REXEC command. See also Section 10.5, “Using the OBEYFILE Command” on page 255, which explains how to use the `OBEYFILE` command.

Note

- The command sent must complete by itself; it is not possible to interact via REXEC.
- RACF is supported.
- The connection can be automated using the ″NETRC DATA A0″ file.
- When using a slave machine do not call any full screen applications (FileList, Xedit, Browse ...), or the slave machine will hang.

Following is an example of the file ″NETRC DATA A0″:

```
machine fred login fred password pass
```

With such a ″NETRC DATA A0″, sending a remote command to the host ″fred″ is very simple:

```
Ready;
rexec fred dir *.lst,

  The volume label in drive C is OS2.
  The Volume Serial Number is 25E3:5015
  Directory of C:\TCPIP\BIN

 SYS0002: The system cannot find the file specified.
Ready;
```

Warning

Most of the authorization processes are case sensitive. Issue the xedit command SET CASE M when creating the ″NETRC DATA A0″.

3.4 Simple Mail Transfer Protocol (SMTP)

SMTP is the communication vehicle for TCP/IP network users using electronic mail. Unlike PROFS*, which uses RSCS as a communication vehicle, it is not store and forward oriented.

Connections are synchronous; both parties have to be active at the same time. This can be a problem when the two parties are located in different time zones. If their systems are never active at the same time, they cannot exchange mail. To solve this problem you may use the Mail eXchange (MX) facility of a name server. Please refer to Section 2.11.8, “Using MX Records” on page 105 for more details.

A VM/CMS user can use the mailing facilities of TCP/IP combined with the usual VM/CMS mailing functions (NOTE, PROFS). With the installation of IBM TCP/IP Version 2 Release 2 for VM, TCP/IP versions of the NOTE and SENDFILE command are included as SENDFILE EXECTCP and NOTE EXECTCP files and you have to rename these files. Refer to *IBM TCP/IP V2 R2 for VM: Planning and Customization* manual and to Section 2.11.3, “SMTP NOTE and SENDFILE EXECs” on page 99 for more information on that. These commands are able to detect if the recipient is an SMTP or RSCS mailbox. They are also able to support the

usual TCP/IP SMTP addressing format using the ″@″ and ″%″ characters and the VM/CMS format as well.

For example:

- ***userid@smtp_system***
- ***userid@smtp_system.domain***
- ***userid%system@via_smtp_gateway***
- ***userid%system@via_smtp_gateway.domain***

Note: The ″@″ character is usually the ***character delete*** function, it can be changed with the CP command **TERMINAL CHARDEL** to any other character. SMTP will only be used when the recipient is not known by RSCS.

Note

If you renamed ″NOTE EXECTCP″ to ″NOTETCP EXEC″ and ″SENDFILE EXECTCP″ to ″SENDFTCP EXEC″ please read the following explanations accordingly.

3.4.1 Send a File Using the SENDFILE Command

SMTP can also be used to send a file to a recipient within a TCP/IP network. When the SENDFILE command recognizes that the recipient is on a TCP/IP network, it delivers the file via SMTP. SMTP acknowledges the delivery of the file as you see in the example.

```
sendfile profile exec a rolf
 * From SMTP: Received Spool File 2683
File PROFILE  EXEC     A1 sent  to wtcr11%ralydpd4 at vm14  on 11/09/90 11:3:54
Ready; T=0.83/1.07 11:03:56
 * From SMTP: Mail delivered to: WTCR11 at RALYDPD4
```

SENDFILE may also be used with no file name specified. The following screen will be displayed:

```
                ---------------  SENDFILE  ---------------

File(s) to be sent    (use * for Filename, Filetype and/or Filemode
                           to select from a list of files)
Enter filename :
      filetype :
      filemode :

Send files to  :

Type over YES or NO to change the options:

   NO    Request acknowledgement when the file has been received?
   YES   Make a log entry when the file has been sent?
   YES   Display the file name when the file has been sent?
   NO    This file is actually a list of files to be sent?
   YES   Send the file in NETDATA format?

   1= Help          3= Quit          5= Send          12= Cursor

====
                                                  Macro-read 1 file
```

Note: The SENDFILE command must not be used to send a binary file via a TCP/IP network. SMTP is designed for electronic mail using 7-bit ASCII text only. Use FTP to send binary files.

3.4.2 The Nickname File

VM/CMS users may build their own nickname files to ease the communication, since the correct addressing format may become very long when domain names must be used. The following nickname file contains three different parts. To understand it properly, it is necessary to know that the node names RAL9360, RALYDPD4 and CHVM1 are known by RSCS. The node names VM14 and VM15 are known by TCP/IP only.

The nickname file ″TCPMAINT NAMES″ is located on the A disk of user TCPMAINT in system RAL9390.

TCPMAINT NAMES

```
*  via rscs
:nick.tcpmaint          :userid.tcpmaint
                        :node.ral9360
:nick.wtcr11            :userid.wtcr11
                        :node.ralydpd4
*  via smtp direct delivered
:nick.tcpmv14           :userid.tcpmaint
                        :node.vm14
:nick.tcpmv15           :userid.tcpmaint
                        :node.vm15.itsc.raleigh.ibm.com
:nick.frick             :userid.frick
                        :node.vm15
*  via smtp rscs gateway function
:nick.trt               :userid.trt%chvm1
                        :node.vm14
:nick.rolf              :userid.wtcr11%ralydpd4
                        :node.vm14
:nick.lesia             :userid.lesia%ralydpd4
                        :node.ral9390.itsc.raleigh.ibm.com
```

3.4.3 Send a Note Using the NOTE Command

In the example below, the command **NOTE ROLF** was executed to send a note to WTCR11 at RALYDPD4 using the nickname ROLF which indicates to use the SMTP RSCS gateway function.

```
 TCPMAINT NOTE     A0  V 132  Trunc=132 Size=10 Line=10 Col=1 Alt=0
* * * Top of File * * *
OPTIONS: NOACK    LOG    SHORT    NOTEBOOK ALL

Date: 9 November 1990, 10:59:24 EDT
From: TCPMAINT at RAL9390
To:   wtcr11%ralydpd4 at vm14
Subject: SMTP test via gateway
 please forward to wtcr11 at ralydpd4 via RSCS.
* * * End of File * * *


1= Help      2= Add line  3= Quit   4= Tab      5= Send       6= ?
7= Backward  8= Forward   9= =     10= Rgtleft 11= Spltjoin  12= Power Input

====


                                                      X E D I T  1 File
```

3.4.4 Send a Note Using PROFS

The PROFS Extended Mail Product is an electronic mail interface for PROFS users. PROFS users can send and receive electronic mail from users on TCP/IP networks with PROFS Extended Mail. For more information about using PROFS Extended Mail, see the *PROFS Extended Mail User's Guide and Installation* manual.

A limited PROFS to SMTP interface is provided for sites that do not have PROFS Extended Mail installed. PROFS users can prepare mail for TCP/IP network recipients by including the SMTP virtual machine as one of the recipients of the mail. TCP/IP network recipients are specified through a special command within the PROFS note. The following example shows how to use PROFS without the Extended Mail feature. The TCP/IP network recipients are addressed using the ***.ddn smtp_system(userid)*** command within the note.

```
                                SEND A NOTE                              E04
Send to: ral9390(smtp)
Subject: Note to TCP/IP user
Hello SMTP,
please deliver this note to tcpmaint

 .ddn vm14(tcpmaint)
 .lf tcpip

PF1 Top  PF2 Bottom  PF3 Erase Line  PF4 Add Line  PF5 Nulls Off  PF6 Format
PF7 Send  PF8 Proofread  PF9 Help  PF10 Next  PF11 Previous  PF12 Cancel
```

3.4.5 SMSG Interface to SMTP

The SMTP server in IBM TCP/IP Version 2 Release 2 for VM now provides an interface with VM Send Message (SMSG) that allows the querying of SMTP mail delivery, statistics and provides a set of privileged commands for system administration tasks. Responses to commands are sent back to the originator via CP MSG commands (or CP MSGNOH commands if SMTP is running with CP privilege class B).

The following is the format of the general user SMSG command accepted by SMTP, and the associated output screens:

Format of SMSG Command

```
SMsg   SMTP    HElp
               QUeues
               STats
```

```
smsg smtp help
* From SMTP: Valid Commands for General Users:
* From SMTP:   QUeues   - for mail queue lengths
* From SMTP:   STats    - for operating statistics
* From SMTP:   HElp     - to get this message
* From SMTP: Valid Commands for Authorized Users:
* From SMTP:   CLosecon - to close the virtual console
* From SMTP:   REboot   - to re-IPL CMS
* From SMTP:   SHutdown - to LOGOFF the virtual machine
* From SMTP:   TRace    - to enable resolver tracing
* From SMTP:   NOTrace  - to disable resolver tracing
* From SMTP:   DEbug    - to enable session debugging
* From SMTP:   NODebug  - to disable session debugging
```

```
smsg smtp queues
* From SMTP: ----- Mail Queues -----
* From SMTP: Spool Queue:         0
* From SMTP: R: 9.67.32.18      : 3 VM14.ITSC.RALEIGH.IBM.COM
* From SMTP: Undeliverable Queue: 0
* From SMTP: --- Resolver Queues ---
* From SMTP: Process Queue:       0
* From SMTP: Send Queue:          0
* From SMTP: Wait Queue:          0
* From SMTP: Retry Queue:         0
* From SMTP: Completed Queue:     0
* From SMTP: Error Queue:         0
```

```
smsg smtp stats
* From SMTP: Last Up Time: Tue, 12 Nov 91 17:59:09 EDT
* From SMTP: Spool Files :    0  Held:    0
* From SMTP: Disk Files  :   15  Full:  01%
* From SMTP: Statistics  : 11/13  11/12  11/11  11/08
* From SMTP: From TCP    :     0      0      0      2
* From SMTP: From Spool  :     2      6      8     36
* From SMTP: BSMTP Logs  :     1      0      0      0
* From SMTP: Error Mail  :     3      0      3     11
* From SMTP: To Local    :     4      0      3     16
* From SMTP: To RSCS     :     0      0      0      0
* From SMTP: To TCP      :     0      4      6     31
* From SMTP: Passive Opns:     0      0      0      3
* From SMTP: Active Opens:     0      4      6     14
```

The following is the format of the privileged user SMSG command accepted by SMTP:

Format of privileged SMSG Command

```
SMsg   SMTP    CLosecon
               REboot
               SHutdown
               TRace
               NOTrace
               DEbug
               NODebug
```

Note: Privileged user SMSG commands are only accepted from users specified by the SMSGAUTHLIST in the ″SMTP CONFIG″ file.

```
smsg smtp trace
* From SMTP: Resolver Tracing Enabled
Ready; T=0.01/0.01 17:56:10
```

```
smsg smtp debug
* From SMTP: Session Debugging Enabled
Ready; T=0.01/0.01 17:56:26
```

The TRACE and the DEBUG commands are equivalent to the options in the ″SMTP CONFIG″ file.

3.5 Network Management (SNMP)

See Section 9.2, “SNMP in NetView” on page 237 for the VM NetView SNMP monitor function.

3.6 Remote Printing (LPR/LPD)

The ***LPR*** command is primarily used to print a file via TCP/IP. Printer name and address of the TCP/IP system are optional parameters. Using ***LPRSET***, the printer name and system name can be set for the whole VM/CMS session. Subsequent LPR commands will use the values set.

3.6.1 Print a Document via TCP/IP

This example shows the usage of the LPR and LPRSET commands.

```
lpr profile exec (host vm14 printer ITSC
Host vm14 did not accept printer name ITSC.
Ready(00036); T=0.12/0.30 10:31:52
Ready; T=0.06/0.10 10:14:58
lpr profile exec (host vm14 printer itsc
Ready; T=0.18/0.42 10:15:39
lprset
Use this form: LPRSET printer host
Ready(00024); T=0.03/0.04 10:32:25
lprset itsc vm14
Ready; T=0.03/0.04 10:32:45
lpr profile exec
Ready; T=0.18/0.44 10:33:00
```

The "LPD CONFIG" may look like :

```
SERVICE itsc PRINTER
   RSCS DEST=RALYDPD IDENTIFIER=ITSC
   FILTERS f l p
   LINESIZE 132
   PAGESIZE 66
```

Note: The printer names in the LPR or LPRSET commands are case sensitive. See also Section 5.9, "Printing Files (LPD/LPD)" on page 208; some additional examples on remote printing are provided, including an example of sending a JCL to an MVS system.

Help for the LPR and LPRSET can be obtained using the usual CMS HELP command:

```
Ready;
help lpr

 +----------+-----------------------------------------------------+
 |          |                  +       +                          |
 | LPR      | filename filetype |filemode| ( options ...            |
 |          |                   |   *    |                          |
 |          |                   +       +                          |
 |          |         +       + +    +  +           +              |
 |          | options |BURST  | |CC  |  |LINECOUNT nn|              |
 |          |         |NOBURST| |NOCC|  |          55|              |
 |          |         +       + +    +  +            +              |
 |          |         (LANDSCAPE)    (TYPE)                        |
 |          |         (PRINTER name) (HOST host)     +           +  |
 |          |         (TRACE)        (VERSION)       |POSTSCRIPT  |  |
 |          |                                        |NOPOSTSCRIPT|  |
 |          |                                        +           +  |
 +----------+-----------------------------------------------------+
```

Chapter 4. Using TCP/IP between VM and MVS

The TCP/IP implementations for VM and MVS are very similar, so there is little difference in using TCP/IP on these operating systems. On MVS all TCP/IP client commands (Telnet, FTP, SMTPNOTE, ...) are entered under the TSO command prompt.

4.1 Transferring Files (FTP)

4.1.1 FTP from MVS to VM

Please refer to Section 3.1, “File Transfer Protocol (FTP)” on page 175 for more details about the VM FTP server. The FTP protocol can be used either from the TSO command prompt (via the FTP command) or by submitting jobs.

The help of the MVS FTP client is displayed via the help command when the FTP prompt is displayed.

4.1.2 FTP from VM to MVS

Obviously what has been described in Section 3.1, “File Transfer Protocol (FTP)” on page 175 for the VM FTP client is still valid.

The file system on MVS allows users to define many parameters on their own when they want to create a file (such as the allocation size, the name of the volume where the file will be allocated, etc). The FTP server on MVS uses default values for most of these parameters, but most of them may be changed during the FTP session using the QUOTE SITE or the SITE command.

The login procedure is identical to the one described for VM. You will have to specify a RACF user ID and a RACF password in order to be able to transfer files between a VM and an MVS system.

4.1.3 Getting Help

For the VM FTP client and server help please refer to Section 3.1, “File Transfer Protocol (FTP)” on page 175.

For the MVS FTP client help just type help when the FTP command prompt is displayed.

The following is an example of some interesting commands. The FTP command QUOTE sends the second part of the command to the MVS FTP server. The outputs of these commands are meaningful for an MVS user.

Note

The AUTOMount and NOAUTOMount statements have nothing to do with NFS. They concern DFHSM which is a space manager on MVS systems.

```
Command:
quote stat
   stat
211-Server FTP talking to host 9.67.38.136, port 1026
211-User: DEBULOI  Working directory: DEBULOI
211-The control connection has transferred 1265 bytes.
211-There is no current data connection.
211-The next data connection will be actively opened
211-to host 9.67.38.136, port 1026, using
211-mode Stream, structure File, type ASCII, byte-size 8.
211-No automatic recall of migrated data sets.
211-No automatic mount of direct access volumes.
211-Data set mode. (Do not treat each qualifier as a directory.)
211-Primary allocation 1 track. Secondary allocation 1 track.
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 Record format VB. Lrecl: 256, Blocksize: 6233
Command:
quote help site
   help site
214-The SITE command sub parameters are:
214-AUTORecall    - permits migrated data sets to be recalled
214-                automatically
214-NOAUTORecall  - prevents migrated data sets from being
214-                recalled automatically
214-AUTOMount     - allows data sets on volumes that are
214-                not mounted to be mounted automatically
214-NOAUTOMount   - prevents data sets on volumes that are not
214-                mounted from being mounted automatically
214-DIrectorymode - treats each level of a data set as a directory.
214-                Only the next lower level is used for MPUT or
214-                server MGET, LS and DIR commands
214-DAtasetmode   - treats all subsequent levels as a data set
214-                (disables directory mode)
214-PRImary=value   specifies the amount of storage for the primary
214-                allocation for new data sets
214-                are listed
214-Unit=value      specifies the name of a unit for allocation
214-                If no value is given, the previous value is remov
214-                and the system default unit is used
214-FILEtype=value  specifies file type (SEQ, JES, or PIO)   1
214-                SEQ is standard sequential files
214-                and is the default, and most common.
214-                JES is MVS spool used for submitting
214-                Jobs and retrieving their output.
214-                PIO is Parallel I/O.
214 Use server STAT command to display present values.
Command:
```

Using these commands, you are able to see what default values are used and how to change them.

1 Although it is not documented, an interface with DB2* is available which allows you to submit DB2 queries from an FTP client to MVS using the FTP server on the MVS system.

The file structure between MVS and VM is different. Basically an MVS system has two kinds of files that can be handled by the FTP server: physical sequential files (Dsorg: PS) and partitioned organization data sets (Dsorg: PO). PO data sets may be considered as directories; that is, they include files (members).

Following is an example (the user ID used for logging on was DEBULOI). Following is an example of how to navigate in the MVS files organization:

```
dir
   PORT 9,67,38,136,4,5
200 Port request OK.
   LIST
125 List started OK.
Volume Unit     Date  Ext Used Recfm Lrecl BlkSz Dsorg Dsname
WTLSTG 3380K  06/09/92  1  147  FB      80  6160  PO  BIN.CLIST
WTLSTG 3380K  06/05/92  1  146  FB      80  6160  PO  BIN.JCL
WTLSTG 3380K  06/09/92  1    2  FB      80  3120  PO  ESA4.ISPPROF
WTLSTG 3380K  06/05/92  1  287  U     6144  6144  PO  XCLIENT.LOAD
WTLSTG 3380K  06/05/92  1    3  FB      80  3200  PO  XCLIENT.OBJ
WTLSTG 3380K  06/09/92  1   10  FB      80  3200  PS  XWINDOWS.DISPLAY
WTLSTG 3380K  06/08/92  1    2  FB      80  3200  PS  XWINDOWS.TEST
250 List completed successfully.
Command:
cd bin
   CWD bin
257 †¢DEBULOI.BIN.¢† is working directory name prefix.
Command:
dir
   PORT 9,67,38,136,4,7
200 Port request OK.
   LIST
125 List started OK.
Volume Unit     Date  Ext Used Recfm Lrecl BlkSz Dsorg Dsname
WTLSTG 3380K  06/09/92  1  147  FB      80  6160  PO  CLIST
WTLSTG 3380K  06/05/92  1  146  FB      80  6160  PO  JCL
250 List completed successfully.
Command:
cd jcl
   CWD jcl
257 †¢DEBULOI.BIN.JCL¢† partitioned data set is working directory.
Command:
dir
   PORT 9,67,38,136,4,8
200 Port request OK.
   LIST
125 List started OK.
 Name     VV.MM  Created     Changed     Size  Init   Mod   Id
EDCC      01.12 92/06/05 92/06/06 11:14    16   118     0 DEBULOI
EDCL      01.05 92/06/06 92/06/06 12:03    50    34     0 DEBULOI
250 List completed successfully.
Command:
```

4.1.4 Site-Dependant Commands

As indicated previously, there are a lot of attributes needed in an MVS system to define a data set. When you transfer a file (which does not already exist in the MVS system) using FTP, the FTPSERVE address space will create a file. If the defaults used do not suit you, you must use the SITE (or QUOTE SITE) command to modify them.

The following is an example:

```
site tracks            1
   SITE tracks
200 Site command was accepted
Command:
site primary=10        2
   SITE primary=10
200 Site command was accepted
Command:
site secondary=20      2
   SITE secondary=20
200 Site command was accepted
Command:
site recfm=fb          3
   SITE recfm=fb
200 Site command was accepted
Command:
site blksize=8000      4
   SITE blksize=8000
200 Site command was accepted
Command:
site lrecl=80          5
   SITE lrecl=80
200 Site command was accepted
```

1 Indicates that the space units for the next commands (2) is tracks.
2 Primary and secondary allocation
3 Record format will be fixed blocked
4 Blocksize
5 Record length.

Note: Some other FTP implementations may not know the site subcommand. But, there is usually a quote subcommand, which allows you to send any character string to the remote side.

4.1.5 Storing and Retrieving Files

After applying the site subcommands, the defined data set characteristics are used to build the data set in MVS. The dir subcommand shows that the data set characteristics are as expected.

When you login to MVS with FTP, your user ID is used to form the first level qualifier. So when you want to receive or send a data set that does not begin with your first level qualifier, you must put single quotes around the remote file name.

```
Command:
put profile.exec
   SITE FIXrecfm 80               1
200 Site command was accepted
   PORT 9,67,32,18,4,9
200 Port request OK.
   STOR profile.exec
125 Storing data set LESIA.PROFILE.EXEC
250 Transfer completed successfully.
984 bytes transferred. Transfer rate 2.11 Kbytes/sec.
Command:
dir
   PORT 9,67,32,18,4,3
200 Port request OK.
   LIST
125 List started OK.
Volume Unit     Date  Ext Used Recfm Lrecl BlkSz Dsorg Dsname
WTLSTG 3380K  09/13/90 16   16  VB     256  6233  PS  ADTK0105.DD
WTL320 3380D  11/15/90  1    2  FB      80  3120  PO  ISPFESA.ISPPROF
WTL843 3380   09/11/90  1    2  FB      80  3120  PO  ISPFOLD.ISPPROF
WTL844 3380   09/24/90  1   30  FB      80  3120  PO  NFSTEST
WTLSTG 3380K  11/15/90  1    1  FB      80  8000  PS  PROFILE.EXEC
WTLSTG 3380K  11/15/90  1    8  VA     125   129  PS  SPFLOG1.LIST
250 List completed successfully.
Command:
pwd
   PWD
257 †LESIA.† is working directory
Command:
get ¢tcpip.ralvsmv6.tcpip¢ ralvsmv6.tcpip
   PORT 9,67,32,18,4,5
200 Port request OK.
   RETR ¢tcpip.ralvsmv6.tcpip¢
125 Sending data set TCPIP.RALVSMV6.TCPIP FIXrecfm 80
250 Transfer completed successfully.
14022 bytes transferred. Transfer rate 9.51 Kbytes/sec.
Command:
```

1 The default is to issue the site command before every put command.

To disable the use of the site command do the following:

```
locstat
Trace:FALSE, Send Port:TRUE
Send Site with Put command:TRUE
HOSTS ADDRINFO * not found.
Connected to:9.67.38.135, Port:FTP control (21), logged in
Local Port:1033
Data type:a, Transfer mode:s
Record format:V
Command:
sendsite
Usage of SITE command with PUT is OFF
Command:
locstat
Trace:FALSE, Send Port:TRUE
Send Site with Put command:FALSE
HOSTS ADDRINFO * not found.
Connected to:9.67.38.135, Port:FTP control (21), logged in
Local Port:1033
Data type:a, Transfer mode:s
Record format:V
Command:
```

4.2 Logging On (Telnet)

4.2.1 Telnet from VM to MVS

Please refer to Section 3.2, “Telnet” on page 180 for more details. Transparent mode will be used.

4.2.2 Telnet from MVS to VM

Please refer to Section 3.2, “Telnet” on page 180 for more details. Transparent mode will be used.

4.3 Sending Mail (SMTP)

4.3.1 Sending Mail from VM to MVS

Please refer to Section 3.4, “Simple Mail Transfer Protocol (SMTP)” on page 181 for more details about about how to use the SMTP client function on a VM system.

4.3.2 Sending Mail from MVS to VM

This is done under the TSO command mode via the SMTPNOTE CLIST. As in VM, the SMTP address space must be running. Nicknames are not supported. Like a VM system, an MVS system may act as a mail gateway between NJE and TCP/IP network.

4.4 Executing Remote Commands (REXEC)

4.4.1 REXEC from VM to MVS

The REXEC sever is not implemented on MVS. Job submission as well as DB2 queries can be achieved via FTP.

4.4.2 REXEC from MVS to VM

Please refer to Section 3.3, “Remote Execution (REXEC)” on page 180 for more details about how to use a REXEC server on a VM host.

Chapter 5. Using TCP/IP between VM and OS/2

This chapter shows how VM and OS/2 can communicate via TCP/IP. An OS/2 TCP/IP user might primarily be considered as a TCP/IP user of a VM/CMS system but TCP/IP V1.2.1 for OS/2 now provides the OS/2 workstation with a lot of possibilities which allows OS/2 TCP/IP hosts to act as servers as well. The following server capabilities are provided:

- Telnet
- FTP
- TFTP
- X.Window
- SMTP
- NFS
- REXEC
- RSH
- SNMP
- LPD
- TALK

VM and OS/2 differ in that VM represents characters using EBCDIC and OS/2 uses ASCII. Therefore translation tables will be used to map EBCDIC codes to ASCII codes and vice versa. Please refer to Chapter 11, “National Language Support (NLS)” on page 261 for more information.

5.1 Transferring Files (TFTP)

5.1.1 TFTP from OS/2 to VM

Due to the lack of authentication of the TFTP protocol, the server function is not implemented in IBM TCP/IP Version 2 Release 2 for VM.

5.1.2 TFTP from VM to OS/2

As mentioned previously, TFTP does not provide any authentication mechanism. To avoid a remote user to TFTP files such as CONFIG.SYS to your OS/2 you will have to specify some parameters when you start the TFTP server on the OS/2 side. The files may then have a specific prefix, or be written in a specific directory.

Following is the TFTP sequence.

```
tftp fred
Command:
put profile.exec.a proto.exe
1148 bytes transferred in 1.596 seconds. Transfer rate 0.719 Kbytes/sec.
Command:
quit
Ready; T=0.17/0.45 08:31:21
```

The file transmitted will be:

- In the TFTP directory if TFTP was started by the command TFTPD C:TFTP. The file will be named C:TFTPproto.exe.

- In the "root" directory but with the **dum** prefix (the file will be named **dumproto.exe** on the OS/2 station) if TFTP is started by the command **TFTPD dum**. The file name plus the prefix must not exceed 8 characters.

5.2 Transferring Files (FTP)

5.2.1 FTP from OS/2 to VM

The sequence is the same as the one described in Section 3.1, "File Transfer Protocol (FTP)" on page 175. The screens will differ slightly but the information needed to be provided to the VM server is the same.

TCP/IP V1.2.1 for OS/2 also provides you with an FTP client which is a Presentation Manager application (**FTPPM**). You will then choose whatever is needed to actually execute the file transfer using the mouse. The main panel looks like this:

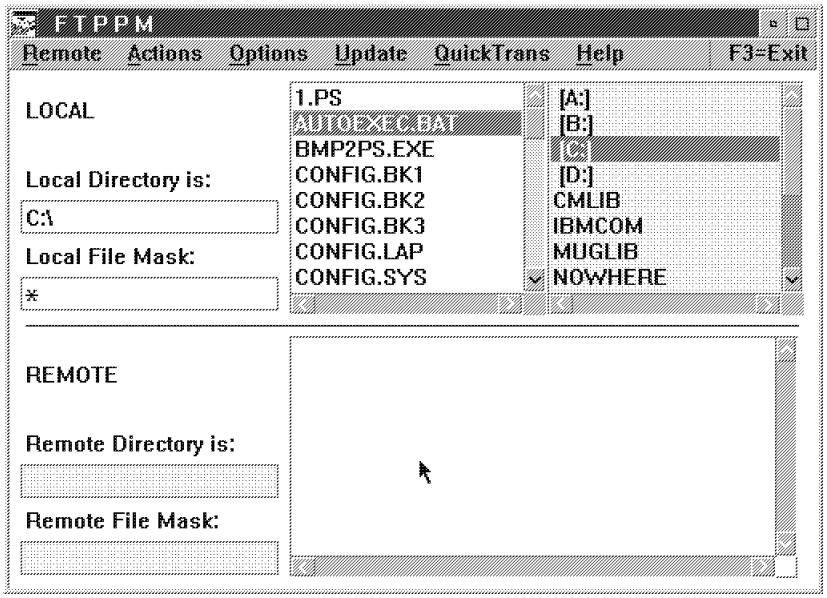


Figure 34. FTPPM Main Panel

5.2.2 FTP from VM to OS/2

The FTP client in IBM TCP/IP Version 2 Release 2 for VM allows you to access the file system of OS/2. Files can be copied in both directions with (that's the default) or without (binary mode) character translation (ASCII-EBCDIC translation). The VM FTP client may use a specific translate table to access a workstation which uses a different code page from the one the regular (STANDARD) translate table assumes.

The FTP server on the OS/2 workstation must be either:

- Started in foreground. This is done using the ICAT tool. The startup command will then appear in the file "TCPIPBINTCPSTART.CMD".
- Be listed in the file "TCPIPETCINETD.LST". This is also done using ICAT. The super daemon INETD will start the FTP server whenever a request for file transfer is received.

Authorization checking on the OS/2 side is done via the file ″TCPIPETCTRUSERS″ where user ID/password pairs are defined as well as the directories/files which can be accessed. This file can be created either using an OS/2 editor or using ICAT.

Following is a very basic example of the TRUSERS file (please note the which is required to give access to the D drive).

```
user: fred fred
rd: C:\TCPIP\ETC D:\
wr: C:\TCPIP\ETC
```

Warning

- **rd** means Read/Only
- **wr** means Write/Only

If you want to get a file, the corresponding directory must be listed in the **rd** list, not in the **wr** list.

The following FTP connection shows how a VM/CMS user can copy a file from OS/2, modify it in VM/CMS with XEDIT and copy it back to OS/2. Note the *cms subset* command which allows you to go back temporarily to CMS. The *return* exits from the CMS subset mode.

```
ftp fred
VM TCP/IP FTP V2R2
Connecting to FRED 9.67.38.84, port 21
220 fred FTP server (IBM OS/2 TCP/IP FTP Version 1.2) ready.
USER (identify yourself to the host):
fred
   USER fred
331 Password required for fred.
Password:
Password entered here
   PASS ********
230 User fred logged in.
Command:
cd tcpip\etc
   CWD tcpip\etc
250 CWD command successful.
Command:
get inetd.lst inetd.os2
   PORT 9,67,38,66,4,13
200 PORT command successful.
   RETR inetd.lst
150 Opening ASCII mode data connection for inetd.lst (20 bytes).
226 Transfer complete.
20 bytes transferred. Transfer rate 0.08 Kbytes/sec.
Command:
cms subset
CMS subset
xedit inetd.os2
Ready;
return
Command:
put inetd.os2 inetd.lst
   SITE VARrecfm
202 SITE not necessary; you may proceed.
   PORT 9,67,38,66,4,14
200 PORT command successful.
   STOR inetd.lst
150 Opening ASCII mode data connection for inetd.lst.
226 Transfer complete.
20 bytes transferred. Transfer rate 0.02 Kbytes/sec.
Command:
quit
   QUIT
221 Goodbye.
Ready; T=1.03/1.75 13:12:13
```

5.2.3 Getting Help

Help can be obtained with the same commands as the ones described in Section 3.1.2, “Getting Help” on page 176.

5.3 Logging On (Telnet)

5.3.1 Telnet from OS/2 to VM

TCP/IP V1.2.1 for OS/2 includes two Telnet clients to access IBM TCP/IP Version 2 Release 2 for VM in 3270 mode. These two 3270 emulators are called PMANT (which is Presentation Manager based) and TN3270 (which is a full-screen application). Both emulators support extended data stream and allow the TCP/IP V1.2.1 for OS/2 user to select the desired screen size.

National language is not supported yet by TCP/IP V1.2.1 for OS/2.

5.3.2 Telnet from VM to OS/2

The Telnet client in IBM TCP/IP Version 2 Release 2 for VM allows you to log on from any VM/CMS terminal to a TCP/IP V1.2.1 for OS/2 workstation as a line mode terminal. On the TCP/IP V1.2.1 for OS/2 workstation you can use any line mode application. Presentation Manager or full-screen applications are not supported by the VM Telnet client. If you enter a command to start one of these types of applications, your connection becomes locked and you cannot control it any more. The only way to recover from this situation is to drop the connection with a Telnet subcommand or go to the OS/2 workstation and end the command locally using the workstation keyboard.

The VM Telnet client may use a specific translate table to access a workstation which uses a different code page than the one the regular (TELNET) translate table assumes. Even if you choose the right translation table on the VM side, you won't have national language support (the Telnet protocol will then only use 7 bits).

```
telnet os25 (translate swisgerm
VM TCP/IP Telnet V2R2
Connecting to OS25 9.67.32.21, port TELNET (23)

Using Line Mode...

Notes on using Telnet when in Line Mode:
- To hide Password, Hit PF3 or PF15
- To enter Telnet Command, Hit PF4-12, or PF16-24


                                                        MORE...   RAL9390
```

PF4 can be used to enter the Telnet command mode, which allows you to control the connection (send Ctrl. Break, close connection, etc.). The command mode may be useful when you enter any unusual conditions.

```
 The Server will start a session.

  Please hold for a moment...

password:
login completed

[OS25:C:\ dir *.cmd

 The volume label in drive C is OS2.
 The Volume Serial Number is E2C5:AC14
 Directory of C:\

 1-19-90   5:57p       517          34  PATCH.CMD
10-25-90   7:58a        38           0  STARTUP.CMD
     2 File(s)   11247104 bytes free

[OS25:C:\ logout
```

Note: OS/2 is designed as a single user operating system. However, TCP/IP allows multiple users to work with OS/2 concurrently. Once users are logged on to OS/2, they can do whatever they like (formatting a disk). There is no user ID and password concept in OS/2; therefore, it is not possible to restrict a user to certain commands or data.

5.4 Sending Mail (SMTP)

Mail on an OS/2 system may be sent and received using plain SMTP protocol. TCP/IP V1.2.1 for OS/2 allows you to use a Presentation Manager interface (LaMail). ASCII-EBCDIC translation will be performed on the VM host.

5.4.1 Sending Mail from VM to OS/2

Please refer to Section 3.4, “Simple Mail Transfer Protocol (SMTP)” on page 181 for more detail about the NOTE and SENDFILE commands.

On the OS/2 side, SENDMAIL *must* be started. LaMail may be started in order to view, answer, and store mail using a Presentation Manager interface.

Following is an example of the LaMail window once a note has been received:

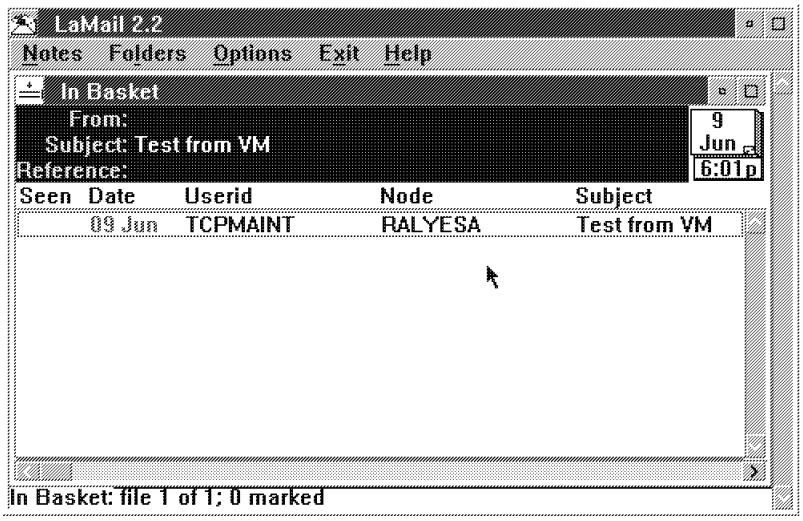


Figure 35. LaMail Window

5.4.2 Sending Mail from OS/2 to VM

This is done using LaMail from the same panel as the one shown above. Nicknames are supported.

5.5 Network Management (SNMP)

5.5.1 Querying the VM SNMP Agent from OS/2

TCP/IP V1.2.1 for OS/2 has a limited SNMP monitor, which is able to issue GET PDUs towards the VM SNMP agent. The community file (PW SRC) listed below restricts the access to the VM SNMP agent. The IP address of an incoming request to the SNMP agent is logically ANDed with the ″Mask″ and compared with the ″Network″. If the ″Community Name″ also matches, the request is accepted and will be executed. This ensures that no one outside the defined network can query the SNMP agent using a community name.

```
Community Name   Network      Mask

TCP              9.67.38.0    255.255.255.0
IBM              9.0.0.0      255.0.0.0
```

All commands are executed from an OS/2 workstation located on the network 9.67.38. The command SNMPGRP implements a simple OS/2 monitor. If you execute it without any parameters, it will list them for you like a ″help″ function. The group ″arptab″ of system VM14 is listed by the second command. The community name ″TCP″ can be used from anywhere within the network 9.67.38.

```
[C:\ snmpgrp
usage:  snmpgrp host community [sys [ip [icmp [udp [tcp [tcptab [iproute
                          [ipaddr [arptab [iftab [mediatab
sys     - get system group MIB variables
ip      - get ip group MIB variables
icmp    - get icmp group MIB variables
udp     - get udp group MIB variables
tcp     - get tcp group MIB variables
tcptab  - get tcp connection table
iproute - get ip routing table
ipaddr  - get ip address table
arptab  - get arp table
iftab   - get interface table
mediatab- get net to media table

[C:\ snmpgrp vm14 tcp arptab

ARP table
 intrf hware addr      ip addr
    1 400000114014      9.67.38.66
    1 10005aa87023      9.67.38.72
    1 10005a4f58ce      9.67.38.73
    1 400000032232      9.67.38.84
    4 310600198332      9.67.32.65
    4 310600198432      9.67.32.66
[C:\
```

TCP/IP V1.2.1 for OS/2 also allows you to:

- Issue an **SNMP** command:

```
[C:]snmp
usage:  snmp [get] [next] host community_name  textual_name
[C:\]snmp get vm14 tcp sysuptime
Value of sysuptime is 6124900

[C:\]snmp next vm14 tcp sysuptime
Value of sysContact is PHILIPPE BEAUPIED (OFFICE CC119 - EXT 2365
[C:\]
```

- Receive traps from an agent with the **SNMPTRAP** utility, which is a Presentation Manager application (the icon is usually black and will turn red whenever a trap is received).

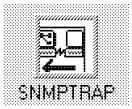


Figure 36. SNMPTrap Icon

When the window is open, information about the traps received is displayed:

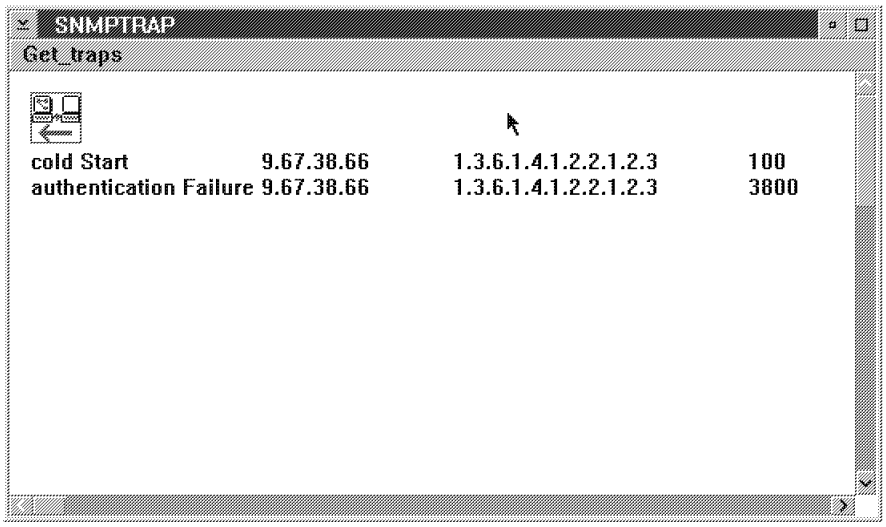


Figure 37. SNMPTrap Trap Window

5.5.2 Querying the OS/2 SNMP Agent from VM

See Section 9.2, “SNMP in NetView” on page 237 for the VM NetView SNMP monitor function.

5.5.3 Listing the Routing Table

The following example lists the routing table on VM TCP/IP maintained by ROUTED. To investigate routing problems, this command is very helpful in examining remote routing tables.

```
[C:]snmpgrp vm14 ITSC iproute

IP routing table

destination  9.67.38.64
intrf  1
m1  -1
m2  -1
m3  -1
m4  -1
gateway  9.67.38.66
type  3
proto  2
age  318922
mask  255.255.255.192
```

5.5.4 Rejected Access

The following example shows an unauthorized access to the VM SNMP agent, because the community name ″Dummy″ is not valid. An Authentication_Failure trap will be transmitted to the monitor(s) defined in the ″SNMPTRAP DEST″ file.

```
[D:\tcpip\etc snmpgrp vm14 Dummy sys

SYSTEM group
could not receive response

[D:\tcpip\etc
```

5.6 Remote SHell (RSH)

Please refer to Section 2.12.2, “How to Use the RSH Protocol” on page 111 for more information.

5.7 Remote Execution (REXEC)

5.7.1 REXEC from OS/2 to VM

Please refer to Section 2.12.1, “How to Use the REXEC Protocol” on page 110 to see how to use the VM REXECD/RSH servers on VM systems from an OS/2 platform.

5.7.2 REXEC from VM to OS/2

The REXEC client in VM/CMS may invoke any command in OS/2, but this command must not expect any interaction with the user. It is possible to start a full screen or Presentation Manager application, but the output of these applications cannot be redirected to the REXEC client. Authorization checking is done on the OS/2 using variables specified using ICAT.

```
rexec -l fred -p pass fred dir

  The volume label in drive C is OS2.
  The Volume Serial Number is 25E3:5015
  Directory of C:\TCPIP\BIN

 .             DIR       5-29-92  10:42a
 ..            DIR       5-29-92  10:42a
 CNTRL    EXE    10185   5-29-92  10:42a
 ARP      EXE    16655   5-29-92  10:42a
 BOOTP    EXE    27063   5-29-92  10:42a
 BOOTPD   EXE    42879   5-29-92  10:42a
```

The same output could have been obtained using a file ″NETRC DATA A0″ to automate the login procedure:

```
type netrc data a0

machine fred login fred password pass

Ready; T=0.01/0.03 11:06:07
rexec fred dir

  The volume label in drive C is OS2.
  The Volume Serial Number is 25E3:5015
  Directory of C:\TCPIP\BIN

.            DIR       5-29-92  10\42a
..           DIR       5-29-92  10\42a
CNTRL   EXE    10185   5-29-92  10\42a
ARP     EXE    16655   5-29-92  10\42a
BOOTP   EXE    27063   5-29-92  10\42a
BOOTPD  EXE    42879   5-29-92  10\42a
FINGER  EXE    29483   5-29-92  10\42a
FTP     EXE    92043   5-29-92  10\42a
```

Please also refer to Section 2.12.1, “How to Use the REXEC Protocol” on page 110 for some more information about the REXECD/RSH servers depending on the use RACF.

5.8 Sharing Files (NFS)

5.8.1 NFS Server on OS/2 - NFS Client on VM

The NFS client function is not available with IBM TCP/IP Version 2 Release 2 for VM.

5.8.2 NFS Client on OS/2 - NFS Server on VM

TCP/IP V1.2.1 for OS/2 includes an NFS client which allows access to the VM/CMS minidisks and makes them appear like regular OS/2 files. The access to VM/CMS files via NFS is transparent to OS/2 programs. However, it is not possible to create directories or to modify file attributes.

```
[C:]mount -v x: vm14:tcpmaint.191,rw,user=tcpmaint,record=nl
mount: vm14:tcpmaint.191,rw,record=nl
Enter password:

NFS Drive ¢x:¢ was attached successfully.

[C:\]type x:profile.exec
/*****************************************************************/
/* The PROFILE EXEC should go on TCPMAINT 191 minidisk           */
/*****************************************************************/
¢CP SET PF12    RETRIEVE¢
¢CP SPOOL CON  START TO *¢
¢SET LDRTBLS 10¢
¢EXEC TCPACC¢

[C:\]umount x:
Unmounting ¢vm14:tcpmaint.191,rw,record=nl¢...      successful.
```

Please refer to Section 2.20.2, “VMNFS Interface to RACF” on page 172 for more information about how to use RACF together with NFS on VM.

The remote file system (VM/CMS minidisk) must be attached to the local file system (OS/2) using the `MOUNT` command on the client side. This command needs some parameters defining exactly what remote file system is to be mounted and how it can be accessed locally. Using VMNFS as the remote server, the user ID, the minidisk address and the minidisk password must be supplied with the MOUNT command. The server uses this information to execute a `CP LINK` command against the remote file system (minidisk).

VM/CMS and OS/2 interpret their data differently; VM/CMS is an EBCDIC machine, and OS/2 uses ASCII. The data in a CMS file is arranged in records, whereas an OS/2 file is a stream of data using a special character combination (CR,LF) to indicate record boundaries. The mount option `record=nl` requests this data conversion. The data on the CMS minidisk is meaningful for regular VM/CMS users as well as for OS/2 users. The files stored should also meet the VM/CMS file system restrictions.

If you want to store executable OS/2 modules (.com or .exe files) and share these programs with other OS/2 network users, do not use the `record=nl` option. The data should not be translated; rather, it should be stored in binary form. The NFS server on VM TCP/IP can also handle files that do not meet the VM/CMS file name restrictions, for example, a file without a file extension. The NFS server can translate these file names transparently to the user. See also Section 7.9, “Network File System (NFS)” on page 222 for UNIX systems.

Warnings

When a minidisk is mounted in read/write mode the multiple access password must be provided if RACF is not installed. The write password will result in an error message: Access denied.

If you want to share files between OS/2 and UNIX/AIX stations you have to issue the following command from OS/2:

```
MOUNT -c -v X: ral9390:tcpmaint.191,rw,record=nl
```

The files created from OS/2 will then use LF (UNIX/AIX-like) as record boundaries, instead of CR/LF (OS/2, DOS-like).

5.9 Printing Files (LPD/LPD)

Users on an OS/2 workstation with TCP/IP running on VM and OS/2 have a very convenient way to print out VM/CMS files to a printer directly attached to OS/2, or printing out an OS/2 file on a host-attached printer.

Note: LPD and LPR are designed for line printing mode only. A page printer like a 3820 can be used, but in line mode only.

5.9.1 Print a VM/CMS File on an OS/2-Attached Printer

The example below shows how to print a VM/CMS file on an OS/2-attached printer in two different ways. The first way is a direct TCP/IP connection from VM to the OS/2 printer named LPT1. The LPR client in VM/CMS communicates directly with the LPD server on the OS/2 workstation. The second way addresses the same printer, but it uses the VM TCP/IP LPD server to print the file. The VM LPD server relays the data to the OS/2 LPD server.

```
lpr profile exec (host rolf printer lpt1
Ready; T=0.18/0.42 10:31:35
lpr profile exec (host vm14 printer rolf
Host vm14 did not accept printer name rolf.
Ready(00036); T=0.12/0.30 10:31:52
lpr profile exec (host vm14 printer ROLF
Ready; T=0.18/0.41 10:32:00
lprset
Use this form: LPRSET printer host
Ready(00024); T=0.03/0.04 10:32:25
lprset LPT1 rolf
Ready; T=0.03/0.04 10:32:45
lpr profile exec
Ready; T=0.18/0.44 10:33:00
```

Note: The printer name is case sensitive. The LPD configuration file ″LPD CONFIG″ in IBM TCP/IP Version 2 Release 2 for VM is usually in all capital letters; however, other systems may use all lowercase letters. It would help your users to set up a convention, or define the printer names in both, lower and uppercase characters.

5.9.2 Print an OS/2 File on a Host-Attached Printer

The LPR client on OS/2 is very convenient for workstations without an attached printer. An OS/2 file can be printed on a host printer with the following command.

```
[C:\SPOOL lpr -p ITSC -s VM14 c:\config.sys
Trying LPD print server vm14, device ITSC.
Sent 1554 bytes.
The entire document was sent.

[C:\SPOOL
```

The name of the printer and the name of the host acting as a printer server, can be set in the ″CONFIG.SYS″.

5.9.3 Send MVS Job Control Language (JCL) via LPR

The following command sends a file containing JCL statements to an MVS host, executes the requested program and sends the output to the VM user ID TCPMAINT on system RAL9390. The JCL is first sent to the LPD server on system VM14. The LPD server then sends the JOB to the MVS host using RSCS. The punch device JCLMVS20 must be defined in the printer configuration file ″LPD CONFIG″ on the system VM14. No TCP/IP connection between the LPD server and the MVS host is required.

```
[D:\temp lpr -p JCLMVS20 -s vm14 iefbr14.jcl
Trying LPD print server vm14, device JCLMVS20.
Sent 185 bytes.
The entire document was sent.

[D:\temp
[C:\SPOOL
```

The MVS system thinks that the JCL is sent from the user ID LPSERVE on system RAL9390 where the LPD server program runs; therefore, it will send the output back to the place of origin. The /*ROUTE statement in the JCL is used to define the destination for the output.

File "liefbr4.jcl"

```
//LESIA10 JOB (),¢LESIA¢,
//         USER=LESIA,PASSWORD=GEHEIM,
//         NOTIFY=LESIA
/*ROUTE   PUNCH RAL9390.TCPMAINT
/*ROUTE   PRINT RAL9390.TCPMAINT
//S001     EXEC PGM=IEFBR14
```

Note: There is no direct way to get the output back to the OS/2 workstation.

Chapter 6. Using TCP/IP between VM and DOS

This chapter shows how VM and DOS can communicate via TCP/IP.

Most of the output screens have already been shown in previous sections; therefore, they won't be repeated in this section.

VM and DOS differ in that VM represents characters using EBCDIC and DOS uses ASCII. Therefore translation tables will be used to map EBCDIC codes to ASCII codes and vice versa. Please refer to Chapter 11, "National Language Support (NLS)" on page 261 for more information.

6.1 Transferring Files (TFTP)

6.1.1 TFTP from DOS to VM

Due to the lack of authentication on the TFTP protocol, the server function is not implemented in IBM TCP/IP Version 2 Release 2 for VM.

6.1.2 TFTP from VM to DOS

The command is the same as in Section 5.1.2, "TFTP from VM to OS/2" on page 197. The only protection available on the DOS machine is to disable the write capability when starting the TFTP server (via the command TFTP -s -w).

Note

TCP/IP V2.0 for DOS has a TFTP server included in the Telnet client which allows you to receive files while you are in a Telnet session. Telnet *must* be called with the TELNET command.

6.2 Transferring Files (FTP)

6.2.1 FTP from DOS to VM

The sequence is the same as the one described in Section 3.1, "File Transfer Protocol (FTP)" on page 175. The screens will differ slightly but the information which needs to be provided to the VM server is the same.

Note

TCP/IP V2.0 for DOS has an FTP client included in the Telnet client which allows you to receive files while you are in a Telnet session. Telnet *must* be called with the FTELNET command.

6.2.2 FTP from VM to DOS

The FTP client in IBM TCP/IP Version 2 Release 2 for VM allows you to access the file system of DOS. Files can be copied in both directions with (default) or without (binary mode) character translation. The VM FTP client may use a specific translate table to access a workstation which uses a different code page than the one the regular (STANDARD) translate table assumes.

The sequence is the same as the one decribed earlier. Please refer to Section 5.2.2, “FTP from VM to OS/2” on page 198. Authorization checking on the DOS side is performed in the same way as OS/2 (the file ″TCPDOSETCTRUSERS″ has the same format as the OS/2 one).

The main difference is that the FTP server must be started manually on the DOS host using the FTPD command. Once the FTP server has been started in the DOS system, all the usual FTP commands are available. The following is the login sequence.

```
ftp 9.67.38.99
VM TCP/IP FTP V2R2
Connecting to 9.67.38.99, port 21
220 dosps FTP server (IBM DOS TCP/IP FTP Version 2.0.2) ready.
USER (identify yourself to the host):
fred
   USER fred
331 Password required for fred.
Password:

   PASS ********
220 User fred logged in.
Command:
dir
   PORT 9,67,38,65,4,5
200 PORT command successful.
   LIST
150 Opening ASCII mode data connection for C:\.
              8440      A          03-30-92   12:00  OS2LDR.MSG
              2294      A          05-26-92   14:43  CONFIG.BK1
                 0           DIR   06-01-92   22:00  EXCEEDW
226 Transfer complete.
Command:
```

6.2.3 Getting Help

Help can be obtained with the same commands as the ones described in Section 3.1.2, “Getting Help” on page 176.

6.3 Logging On (Telnet)

6.3.1 Telnet from VM to DOS

The Telnet server is not implemented in TCP/IP V2.0 for DOS.

6.3.2 Telnet from DOS to VM

TCP/IP V2.0 for DOS includes one Telnet client to access IBM TCP/IP Version 2 Release 2 for VM in 3270 mode.

Using the SETTERM facility you can customize the colors of the screen, remap keys,etc. National language support is provided via an external table which has the same format as the one used on VM.

6.4 Executing Remote Commands (REXEC)

6.4.1 REXEC from VM to DOS

The Remote Execution server is not available with TCP/IP V2.0 for DOS.

6.4.2 REXEC from DOS to VM

Please refer to Section 3.3, “Remote Execution (REXEC)” on page 180 about how output screens will look. On the DOS side, the connection cannot be automated using a ″NETRC″ file.

6.5 Executing Remote Commands (RSH)

6.5.1 RSH from VM to DOS

The RSH server is not available with TCP/IP V2.0 for DOS.

6.5.2 RSH from DOS to VM

- **With RACF installed on the VM system:**
  - **Using a slave user ID:**

```
[C:]rsh vmesa -l guest †Q T†
Local username: guest
TIME IS 16:34:07 EDT THUERSDAY 07/16/92
CONNECT= 00:01:24 VIRTCPU= 000:00:23 TOTCPU= 000:00:32

[C:\]rsh vmesa †Q T†
Local username: guest
TIME IS 16:38:36 EDT THUERSDAY 07/16/92
CONNECT= 00:05:54 VIRTCPU= 000:00:25 TOTCPU= 000:00:36
```

  - **Using a private user ID (TCPMAINT):**

```
[C:]rsh vmesa -l anchor1 †Q T†  1
Local username: tcpmaint
TIME IS 16:43:46 EDT THUERSDAY 07/16/92
CONNECT= 00:00:06 VIRTCPU= 000:00:16 TOTCPU= 000:00:29
```

1 anchor1 is the RACF password of TCPMAINT.

- **Without RACF installed on the VM system:**
  - **Using a slave user ID:**

```
[C:]rsh vmesa †Q T†
Local username: guest
TIME IS 16:47:52 EDT THUERSDAY 07/16/92
CONNECT= 00:01:40 VIRTCPU= 000:00:23 TOTCPU= 000:00:33
```

  - **Using a private user ID (TCPMAINT):**

```
[C:]rsh vmesa †Q T†
Local username: tcpmaint
TIME IS 16:47:18 EDT THUERSDAY 07/16/92
CONNECT= 00:00:05 VIRTCPU= 000:00:16 TOTCPU= 000:00:28
```

    The TCPMAINT user ID will be started by REXECD with no need for the DOS user to provide a password.

6.6 File Sharing (NFS)

6.6.1 VM Client - DOS Server

NFS client is not available with IBM TCP/IP Version 2 Release 2 for VM.

NFS server is not available with TCP/IP V2.0 for DOS.

6.6.2 DOS Client - VM Server

TCP/IP V2.0 for DOS has an NFS client which allows access to the VM/CMS minidisks and makes them look like regular DOS files. The access to VM/CMS files via NFS is transparent to DOS programs. However, it is not possible to create directories or to modify file attributes.

```
[C:]mount -v x: vm14:tcpmaint.191,rw,user=tcpmaint,record=nl
mount: vm14:tcpmaint.191,rw,record=nl
Enter password:

NFS Drive ¢x:¢ was attached successfully.

[C:\]type x:profile.exec
/*****************************************************************/
/* The PROFILE EXEC should go on TCPMAINT 191 minidisk           */
/*****************************************************************/
¢CP SET PF12    RETRIEVE¢
¢CP SPOOL CON  START TO *¢
¢SET LDRTBLS 10¢
¢EXEC TCPACC¢

[C:\]umount x:
Unmounting ¢vm14:tcpmaint.191,rw,record=nl¢...      successful.
```

Please refer to Section 2.20.2, “VMNFS Interface to RACF” on page 172 for more information about how to use RACF together with NFS on VM.

The remote file system (VM/CMS minidisk) must be attached to the local file system (DOS) using the MOUNT command on the client side. This command needs

some parameters defining exactly what remote file system is to be mounted and how it can be accessed locally. Using VMNFS as the remote server, the user ID, the minidisk address and the minidisk password must be supplied with the MOUNT command. The server uses this information to execute a CP LINK command against the remote file system (minidisk).

VM/CMS and DOS interpret their data differently; VM/CMS is an EBCDIC machine, and DOS uses ASCII. The data in a CMS file is arranged in records, whereas a DOS file is a stream of data using a special character combination (CR,LF) to indicate record boundaries. The mount option record=nl requests this data conversion. The data on the CMS minidisk is meaningful for regular VM/CMS users as well as for DOS users. The files stored should also meet the VM/CMS file system restrictions.

If you want to store executable DOS modules and share these programs with other DOS network users, do not use the record=nl option. The data should not be translated; rather, it should be stored in binary form. The NFS server on VM TCP/IP can also handle files that do not meet the VM/CMS file name restrictions, for example, a file without a file extension. The NFS server can translate these file names transparently to the user. See also Section 7.9, “Network File System (NFS)” on page 222 for UNIX systems.

Warnings

When a minidisk is mounted in read/write mode the multiple access password must be provided, the write password will result in an error message: Access denied.

If you want to share files between DOS and UNIX/AIX stations you have to issue the following commands from DOS TODOS or TOUNIX to have a LF or a CRLF at the end of each line.

The MOUNT -c option is not yet available on DOS. It should be available with the new realease of the product.

6.7 Printing (LPR/LPD)

6.7.1 LPR from VM to DOS

The print server is not yet available with TCP/IP V2.0 for DOS.

6.7.2 LPR from DOS to VM

Users on an DOS workstation with TCP/IP running on VM and DOS have a very convenient way to print DOS files on a host-attached printer.

Note: LPD and LPR are designed for line printing mode only. A page printer like a 3820 can be used, but in line mode only.

Please refer to Section 5.9.2, “Print an OS/2 File on a Host-Attached Printer” on page 209 for the syntax of the LPR command.

Chapter 7. Using TCP/IP between VM and UNIX/AIX

TCP/IP is the main communication vehicle on UNIX systems; therefore, it is a very convenient way to use the resources of a large VM host system. Since TCP/IP ″grew up″ in the UNIX world, these systems usually have a very rich implementation of TCP/IP facilities. This is also true for AIX, the IBM version of UNIX. Please refer to Section 1.5, “IBM TCP/IP Version 2 Release 2 for VM Interoperability Summary” on page 55 for a list of the main TCP/IP possibilities of an AIX system. Startup procedures of servers on the AIX/UNIX systems won′t be discussed here because most of them will be started by the INETD super server. Most of the testing described in the present chapter has been done with an AIX system.

VM and AIX/UNIX differ in that VM represents characters using EBCDIC and AIX/UNIX uses ASCII. Therefore translation tables will be used to map EBCDIC codes to ASCII codes and vice versa. Please refer to Chapter 11, “National Language Support (NLS)” on page 261 for more information.

7.1 Transferring Files (TFTP)

7.1.1 TFTP from AIX/UNIX to VM

Due to the lack of authentication of the TFTP protocol, the server function is not implemented in IBM TCP/IP Version 2 Release 2 for VM.

7.1.2 TFTP from VM to AIX/UNIX

No checking is performed on the UNIX/AIX side.

7.2 Transferring Files (FTP)

7.2.1 FTP from AIX/UNIX to VM

The sequence is the same as the one described in Section 3.1, “File Transfer Protocol (FTP)” on page 175. The screens will differ slightly but the information which needs to be provided to the VM server is the same.

7.2.2 FTP from VM to AIX/UNIX

As usual a user ID and a password are needed. Based on the authority of this user ID you will be granted access to a directory in read or write mode.

7.3 Logging On (Telnet)

7.3.1 Telnet from VM to AIX/UNIX

The Telnet client in IBM TCP/IP Version 2 Release 2 for VM allows you to log on from any VM/CMS terminal to an AIX/UNIX workstation as a line mode terminal. On the UNIX/AIX workstation you can use any line mode application. X Windows or full-screen applications are not supported by the VM Telnet client.

The VM Telnet client may use a specific translate table to access a workstation which uses a different code page from the one the regular (Telnet) translate table assumes. Even if you choose the right translation table on the VM side, you won't have national language support (the Telnet protocol will then use only 7 bits).

However some solutions exist which allow a VM terminal to log into an AIX/UNIX host with a VTxxx emulation:

1. Use an OEM product which must be installed in the VM system and will translate 3270 flow into VTxxx flow.
2. Have a 3174 with the RPQ 8Q0935 installed. This 3174 must be hooked to a token-ring and be defined as an X3R (DSPU). The destination host must also be hooked to the token-ring. That is, it must be either on the same token-ring, or on an Ethernet bridged/routed to the token-ring where the 3174 is, or on a remote token-ring bridged/routed to the token-ring where the 3174 is.

   This RPQ allows CUT terminals (coax connected) to have up to five Telnet sessions. The emulators supported are:

   - VT100
   - VT220
   - IBM 3101.

   The 3174 supports the following protocols:

   - IP-ICMP-ARP
   - TCP-UDP
   - Telnet Client
   - PING
   - DNS: stub resolver
   - SNMP agent.

7.3.2 Telnet from AIX/UNIX to VM

AIX systems have a Telnet 3270 emulator. With AIX V3.2 you can have full national language support when logging to a VM system. To achieve this you must set a variable in the AIX system to the EBCDIC code page of your VM host. The right command in France would be `export RM_HOST_LANG=IBM-297`.

UNIX systems may not have a Telnet 3270 emulator; they will be connected to VM in line-by-line mode. However some solutions exist for a UNIX host to be connected in 3270 mode to a VM system:

1. Use an OEM product which must be installed on the VM system and which translates VTxxx flow into 3270 flow.
2. Order from IBM the AIX-X Windows 3270 Emulator (5765-011) which is available for RISC System/6000, SUN** and Hewlett Packard** platforms. X3270 requires an X Windows interface, and is a graphical Telnet in 3270 mode. That means that, using X3270, you can Telnet to a VM system and call GDDM graphics. The only software required in VM are TCP/IP and GDDM.

7.4 Sending Mail (SMTP)

7.4.1 Sending Mail from VM to AIX/UNIX

Please refer to Section 3.4, “Simple Mail Transfer Protocol (SMTP)” on page 181 for more information about the NOTE and SENDFILE commands on VM. Mail is received on AIX/UNIX machines using the mail command. Some systems offer an X Windows interface to view and receive mail.

7.4.2 Sending Mail from AIX/UNIX to VM

Mail is sent on AIX/UNIX machines using the mail command. Some systems offer an X Windows interface to send mail.

7.5 Network Management (SNMP)

7.5.1 Querying the VM SNMP Agent from AIX/UNIX

AIX/UNIX machines provide an X Windows application in order to monitor TCP/IP networks. The AIX V3.2 implementation is called AIX NetView/6000 and provides a graphical interface as well as a communication interface with a NetView running on a VM or MVS host. It allows, for example, SNMP traps to be displayed on an NPDA screen. It allows a NetView operator to issue SNMP commands as well.

7.5.2 Querying the AIX/UNIX SNMP Agent from VM

See Section 9.2, “SNMP in NetView” on page 237 for the VM NetView SNMP monitor function.

7.6 Remote SHell (RSH)

7.6.1 RSH from AIX/UNIX to VM

With RACF used on the VM side:

We have been unable to use RSH when RACF is installed on a VM system because the AIX variable LOGNAME is sent to VM and can’t be changed on the AIX side (Read-Only). We were logged into AIX using the root user ID which was not defined in our VM system. The use of the -l option of the rsh command did not help.

Our guess is that RSH may be usable if the user ID used on the AIX platform is defined in VM.

The command should be:

```
rsh vmesa -l anchor1 "Q T"
```

Where anchor1 is the RACF password.

Without RACF used on the VM side:

Same as before. The LOGNAME Read-Only variable is transmitted to the VM host.

Our guess is that RSH may be usable if the user ID used on the AIX platform is defined in VM. The command should be:

```
rsh vmesa -l tcpmaint "Q T"
```

Where tcpmaint is the password defined in the VM directory.

7.7 Remote Execution (REXEC)

7.7.1 REXEC from AIX/UNIX to VM

Please refer to Section 3.3, "Remote Execution (REXEC)" on page 180 to see how to use the VM REXECD server on VM systems. The commands are almost the same as the ones described in Section 2.12.1, "How to Use the REXEC Protocol" on page 110.

Warning

All the following commands have been issued from an AIX V3.1.5 platform. We have been unable to use the REXEC protocol from an AIX V3.2 platform to a VM system.

With RACF used on the VM side:

- Using a private user ID (TCPMAINT):

```
 rs60001 # rexec vmesa "q t"
Name (ralyesa.itsc.raleigh.ibm.com:root): tcpmaint
Password (ralyesa.itsc.raleigh.ibm.com:tcpmaint):   1
TIME IS 14:22:58 EDT TUESDAY 07/14/92
CONNECT= 00:00:04 VIRTCPU= 000:00.16 TOTCPU= 000:00.29
 rs60001 #
```

1 The RACF password is entered here.

- Using a slave user ID:

```
 rs60001 # rexec vmesa "q t"
Name (ralyesa.itsc.raleigh.ibm.com:root): guest
Password (ralyesa.itsc.raleigh.ibm.com:guest):   2
TIME IS 14:31:14 EDT TUESDAY 07/14/92
CONNECT= 00:11:28 VIRTCPU= 000:00.23 TOTCPU= 000:00.33
 rs60001 #
```

2 guest keyword entered here.

Without RACF on the VM side:

- Using a private user ID (TCPMAINT):

```
 rs60001 # rexec vmesa †q t†
Name (ralyesa.itsc.raleigh.ibm.com:root): tcpmaint
Password (ralyesa.itsc.raleigh.ibm.com:tcpmaint):   3
TIME IS 14:22:58 EDT TUESDAY 07/14/92
CONNECT= 00:00:04 VIRTCPU= 000:00.16 TOTCPU= 000:00.29
 rs60001 #
```

3 The VM directory password is entered here.

- Using a slave user ID:

```
 rs60001 # rexec vmesa †q t†
Name (ralyesa.itsc.raleigh.ibm.com:root): guest
Password (ralyesa.itsc.raleigh.ibm.com:guest):   4
TIME IS 14:31:14 EDT TUESDAY 07/14/92
CONNECT= 00:11:28 VIRTCPU= 000:00.23 TOTCPU= 000:00.33
 rs60001 #
```

4 *guest* keyword entered here.

7.7.2 REXEC from VM to AIX/UNIX

Please refer to Section 5.7.2, “REXEC from VM to OS/2” on page 206. Most of the comments are still valid except that no ICAT is available on AIX/UNIX machines. ″Real″ user ID and passwords are used on AIX/UNIX systems.

7.8 Printing Files (LPR/LPD)

7.8.1 Printing from AIX/UNIX to VM

Once the ″LPD CONFIG″ file has been customized on VM (please refer to page 187 for an example), you need to use SMIT to define the queue and the device on the RISC System/6000.

```
                              Add a Remote Queue

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                      [Entry Fields
* NAME of queue to add                               [itsc
  ACTIVATE the queue?                                 yes                    +
  Will this become the DEFAULT queue?                 no                     +
  Queueing DISCIPLINE                                 first come first serve +
  ACCOUNTING FILE pathname                            [
* DESTINATION HOST for remote jobs                   [vm14
* Pathname of the SHORT FORM FILTER for queue        [/usr/lpd/aixshort     +/
    status output
* Pathname of the LONG FORM FILTER for queue         [/usr/lpd/aixlong      +/
    status output
* Name of QUEUE on remote printer                    [itsc
* NAME of device to add                              [itsc
* BACKEND PROGRAM pathname                           [/usr/lpd/rembak

F1=Help          F2=Refresh       F3=Cancel        F4=List
Esc+5=Undo       F6=Command       F7=Edit          F8=Image
F9=Shell         F10=Exit         Enter=Do
```

The printer itsc has been defined as the default queue. If no printer is specified with the lpr command, the file will be sent for printing to host VM14 on printer itsc. Actually, a virtual printer will be created by LPSERVE on VM14, *tagged* according to the definition in "LPD CONFIG" and sent to printer itsc on system RALYDPD using RSCS.

7.8.2 Printing from VM to AIX/UNIX

Please refer to Section 3.6, "Remote Printing (LPR/LPD)" on page 186 for the syntax of the LPR command.

7.9 Network File System (NFS)

UNIX systems usually have both NFS server and client implementations. The NFS client on UNIX can be used to access the data on the VM/CMS system like regular UNIX files. However, some restrictions apply; it is not possible to create subdirectories. As said previously the NFS client is not available with IBM TCP/IP Version 2 Release 2 for VM.

7.9.1 Using the MOUNT Command on the AIX PS/2

A VM/CMS minidisk can be mounted to an AIX PS/2 workstation with the MOUNT command. VM/CMS files on the minidisk can be read and modified in the UNIX environment like local data. However, UNIX and VM systems have different data interpretation. If the minidisk is to be used for users on both systems, the data must be in VM/CMS EBCDIC format. The ***record=nl*** option in the MOUNT command must be used.

The following example shows how a VM/CMS minidisk is mounted to the workstation. Because the data is meaningful for VM/CMS and AIX, the *record=nl* option is used. Read/write access is also requested in order to modify data.

```
aixps2 # mount vm14:frick.191,rw,password=mfrick,record=nl /mnt/vm14
aixps2 # mount
mounted                 mounted over            options          Gfs# pck
---------------------- ----------------------- -------------- ------- ---
/dev/hd3                /                       rw                   1   1 PRI
sdev/hd2                /aixps2                 rw                   3   1
/dev/hd6                /aixps2/tmp             rw                   4   1
/dev/hd1                /u                      rw                   2   1
vm14:frick.191,rw,pass /mnt/vm14                rw                65537   1 NFS
```

7.9.2 Using the MOUNT Command on a RISC System/6000

Using the MOUNT command in the AIX/6000 environment is similar to using it in the AIX PS/2 environment. The following example shows how a command file is used to invoke multiple MOUNT commands. The second MOUNT of the minidisk from system VM14 is done without the *record=nl* option; therefore, data stored on the minidisk is meaningless to a regular VM/CMS user. Typically, compiled programs are stored without translation, since they are system dependent.

```
#
# mount -o rw,soft vm14:tcpmaint.191,rw,password=mtcp,record=nl /mnt/vm14
# mount -o rw,soft vm15:tcpmaint.191,rw,password=mtcp /mnt/vm15
```

7.9.3 Using the SMIT Command on a RISC System/6000

A VM/CMS minidisk can also be mounted with the SMIT menu interface on the RISC System/6000.

```
                        Add a File System for Mounting

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

[TOP                                                   [Entry Fields
  PATHNAME of mount point                             [/mnt/vm15                /
  PATHNAME of remote directory                        [frick.191,rw,pass=mfri
  HOST where remote directory resides                 [vm15
  Mount type NAME                                     [nfs
  Use SECURE mount option?                             no                      +
  MOUNT now, add entry to /etc/filesystems or both?    now                     +
  MODE for this NFS file system                        read-write              +
  ATTEMPT mount in foreground or background            foreground              +
  NUMBER of times to attempt mount                    [                        #
  Buffer SIZE for read                                [                        #
  Buffer SIZE for writes                              [                        #
  NFS TIMEOUT. In tenths of a second                  [                        #
  internet port NUMBER for server                     [                        #
[MORE...14

F1=Help           F2=Refresh        F3=Cancel         F4=List
Esc+5=Undo        Esc+6=Command     Esc+7=Edit        Esc+8=Image
Esc+9=Shell       Esc+0=Back        Enter=Do
```

7.10 Using NDB

7.10.1 Using the SQL/DS Database from AIX to VM

Once the client C sources have been compiled and the host has been customized, the SQL/DS database can be queried from the RISC System/6000. As said in Section 1.3.8, “Network DataBase System (NDB)” on page 36, queries to an SQL/DS database on a VM system can be sent either interactively or imbedded in C application programs. Only the interactive methods will be discussed here.

1. Multiple queries

   With this method you can send multiple queries to an SQL/DS database. The SQL flow is started with the †begin† statement and is ended with the end statement. The following is an example of multiple queries:

```
#ndbclnt vmesa †begin†   1
ready to call the port manager server
result1- prognum is 20000020
please type in your Host user id ==     2
tcpmaint
Your Host id is tcpmaint.
please type in your Host password ==    2
If you want the result in a file, type in a file name;
Otherwise, hit the enter key.
Default is ndb.output ==    3

Ready to call the server
a message from host vmesa
The return code from NDB is 0
***  result- ndbrpd::
You have started your Unit of Work
It is writing to the file ndb.output now.
Input your SQL statement without any quotes
select * from system.sysprogauth where creator=¢TCPMAINT¢   4
Ready to call the server
a message from host vmesa
The return code from NDB is 0
***  result- ndbrpd::        5

cn8    TCPMAINT
cn8    PUBLIC
cn8    TCPMAINT
vn7    DBUTIL2
cn12   B18BPP2FN8W2
cn1    Y
cn8    TCPMAINT
cn8    TCPMAINT
cn8    TCPMAINT
vn7    DBUTIL2
cn12   B14WB6NQ3LBL
cn1    G
Only 360 bytes were displayed.
Look at your output file, if you want to see the entire output
It is writing to the file ndb.output now.
```

```
Input your SQL statement without any quotes
select * from system.syscatalog where tname=¢ITSC0¢  6
Ready to call the server
a message from host vmesa
The return code from NDB is 0
***  result- ndbrpd::

vn5    ITSC0
cn8    NAMESRV
cn1    R
sn1    5
vn0
sn2    10
vn8    TCPSPACE
sn6    -32767
cn1    I
ln5    35104
sn2    70
ln2    42
ln1    2
sn3    100
ln1    0
sn1    0
sn1    0
sn1    0
vi0
si1    0
Only 360 bytes were displayed.
Look at your output file, if you want to see the entire output
It is writing to the file ndb.output now.
Input your SQL statement without any quotes
end   7
Ready to call the server
a message from host vmesa
The return code from NDB is 0
You are done with your unit of work   8
ndbpnum is 20000020
ready to call the port manager server for done
OK...#
```

1 We start the client program. The begin statement MUST be between double quotes.

2 A user ID and a password (known on the VM system) are required. RACF is NOT supported. The password is the one which is coded in the VM directory.

3 The name of the file used for output can be changed. The default is ″ndb.output″.

4 We send the SQL/DS query.

Warning

This is case sensitive. If the name in the SQL database is TCPMAINT then it MUST be entered in uppercase on the workstation.

5 Part of the result is written on the screen. The complete output is written in the output file. Since this was tested with pre-release code, you may observe some differences in the way the output is displayed once APAR PN17869 on NDB is installed.

6 Another query. Note that the name of the table has been entered in uppercase.

7 The *end* statement ends the NDB client program.

8 As long as the *end* statement is not typed in, the Unit Of Work (UOW) is not complete and the NDB server virtual machine is considered as busy. That's why the multiple NDB server machines feature has been implemented.

The following is the result of the same SQL command as 6 via the ISQL interface. Due to its length it has been split into several parts:

```
TNAME              CREATOR   TABLETYPE   NCOLS  REMARKS
------------------ --------  ---------  ------  --------------------
ITSC0              NAMESRV   R              5
* END OF RESULT *** 1 ROWS DISPLAYED ***COST ESTIMATE IS 1**********


DBSPACENO  DBSPACENAME          TABID  CLUSTERTYPE    CLUSTERROW  AVGROWLEN
---------  -----------------  ------  -----------  ------------  ---------
       10  TCPSPACE           -32767  I                 35104         70
* END OF RESULT *** 1 ROWS DISPLAYED ***COST ESTIMATE IS 1*****************


    ROWCOUNT        NPAGES  PCTPAGES     NOVERFLOW  LFDTABID  LFDLINK
 -----------  ------------  --------  ------------  --------  -------
          42             2       100             0         0        0
 * END OF RESULT *** 1 ROWS DISPLAYED ***COST ESTIMATE IS 1*************


 LFDDBSPACE  TLABEL               PARENTS  DEPENDENTS  INACTIVE
 ----------  -------------------  -------  ----------  --------
          0  ?                          0           0         0
 * END OF RESULT *** 1 ROWS DISPLAYED ***COST ESTIMATE IS 1******
```

Therefore we can see that the same results are obtained either from the ISQL interface on the VM system, or from the NDB client program at the workstation.

2. Single query

Only one query can be sent. The output (on the screen or in the output file) is the same as before. The only difference is in the way the SQL command is typed. The following is part of the output (please refer to the previous option for the complete output):

```
#ndbclnt vmesa †select * from system.syscatalog where tname=¢ITSC0¢†   1
ready to call the port manager server
result1- prognum is 20000020
please type in your Host user id ==
tcpmaint
Your Host id is tcpmaint.
please type in your Host password ==

 ................

Look at your output file, if you want to see the entire output
It is writing to the file ndb.output now.
You are done with your unit of work
ndbpnum is 20000020
ready to call the port manager server for done
OK...#
```

1 Please note that the table name has been typed in uppercase. Also note the way quotes are used; the whole query is between double quotes.

In both cases (multiple or single queries), the output file looks the same. The following is the output file generated by the SQL command `select * from system.syscatalog where tname=¢ITSC0¢`:

```
Your request was select * from system.syscatalog where tname=¢ITSC0¢
The returned data from NDB server are below:
vn5     ITSC0
cn8     NAMESRV
cn1     R
sn1     5
vn0
sn2     10
vn8     TCPSPACE
sn6     -32767
cn1     I
ln5     35104
sn2     70
ln2     42
ln1     2
sn3     100
ln1     0
sn1     0
sn1     0
sn1     0
vi0
si1     0
si1     0
si1     0
```

Let's explain the different fields that appear in the "NDB.OUTPUT" file. NDB adopted an ISO standard for data representation and data conversion. This standard is Abstract Syntax Notation (ASN) with Basic Encoding Rules (BER). The following is the format of the data representation together with an entry (`vn8 TCPSPACE`) from the "NDB.OUTPUT" file:

Table 5. Format of Data Representation

Tag		**Actual Value**
Data type	Length	Value
vn	8	TCPSPACE

`vn` is the data type.

`8` is the length.

`TCPSPACE` is the value.

Let's now explain the different data types:

Table 6 (Page 1 of 2). Meaning of the Data Types

Data type	Meaning
li	long integer with indicator
ln	long integer without indicator

Table 6 (Page 2 of 2). Meaning of the Data Types	
si	short integer with indicator
sn	short integer without indicator
ci	char with indicator
cn	char without indicator
vi	varchar with indicator
vn	varchar without indicator
fi	float with indicator
fn	float without indicator

In SQL/DS, ″with indicator″ means it allows NULL characters, and ″without indicator″ means it does not allow NULL characters.

Chapter 8. Using TCP/IP between VM and OS/400

8.1 Transferring Files (FTP)

8.1.1 FTP from OS/400 to VM

Please refer to Section 3.1, “File Transfer Protocol (FTP)” on page 175 for more details about the VM FTP server.

Please refer to *Version 2 TCP/IP Guide - SC41-9875* and to *IBM AS/400 TCP/IP Configuration and Operation - GG24-3442* for more information about the FTP client on an OS/400 platform.

8.1.2 FTP from VM to OS/400

Obviously what has been described in Section 3.1, “File Transfer Protocol (FTP)” on page 175 for the VM FTP client is still valid.

8.1.3 Getting Help

For the VM FTP client and server help please refer to Section 3.1, “File Transfer Protocol (FTP)” on page 175.

The following is the help for the OS/400 FTP server:

```
Command:
quote help
   help
214-Server-FTP commands follow:
214-ABOR, ADDM, ADDV, APPE, CRTL, CRTP, CRTS, CWD
214-DELE, DLTF, DLTL, HELP, LIST, MODE, NLST, NOOP
214-PASS, PASV, PORT, PWD, QUIT, RCMD, RETR, RNFR
214-RNTO, SITE, STAT, STOR, STOU, STRU, SYST, TIME
214-TYPE
214-The data representation type may be ASCII, EBCDIC, or IMAGE.
214-Data structure must be file.  Mode can be stream (S) or block (B).
214-If this connection is not used more than 300 seconds, the session will end
214-File identifiers have three components: File, library, and member.
214-Library and file components are separated by the / delimiter.
214-File and member components are separated by the . delimiter.
214-Example:  Library/file.member.
214 For information about a specific command, enter HELP  command .
Command:
```

The only thing to note is the way file identifiers must be typed.

8.2 Logging On (Telnet)

8.2.1 Telnet from VM to OS/400

Please refer to Section 3.2, “Telnet” on page 180 for more details. Transparent mode will be used.

Following is how an OS/400 screen will be seen from a VM terminal:

```
MAIN                              AS/400 Main Menu
                                                        System:   RALYAS4B
Select one of the following:

     1. User tasks
     2. Office tasks
     3. General system tasks
     4. Files, libraries, and folders
     5. Programming
     6. Communications
     7. Define or change the system
     8. Problem handling
     9. Display a menu
    10. Information Assistant options
    11. PC Support tasks

    90. Sign off

Selection or command
===  ------------------------------------------------------------------------
------------------------------------------------------------------------------
F3=Exit   F4=Prompt   F9=Retrieve   F12=Cancel   F13=User support
F23=Set initial menu
(C) COPYRIGHT IBM CORP. 1980, 1992.
```

The following is the sequence required to send functions keys to an OS/400 during a Telnet session:

1. Hit the PA1 function key. The bottom part of the screen will be like this:

```
 Selection or command
 ===  ------------------------------------------------------------
------------------------------------------------------------------
 F3=Exit   F4=Prompt   F9=Retrieve   F12=Cancel   F13=User support
Telnet command:
 Type option number or command.
```

2. Type PA1 after the Telnet command: prompt:

```
 Selection or command
 ===  ------------------------------------------------------------
------------------------------------------------------------------
 F3=Exit   F4=Prompt   F9=Retrieve   F12=Cancel   F13=User support
Telnet command: PA1
 Type option number or command.
```

   and hit the Enter key.

3. The bottom part of the screen will be cleared, and then you can hit the function key you want to be executed.

The same principle must be applied to end a Telnet session:

1. Hit the PA1 key.
2. Type QUIT at the Telnet command: prompt.

8.2.2 Telnet from OS/400 to VM

Please refer to Section 3.2, “Telnet” on page 180 for more details. Transparent mode will be used.

Information about keyboard mapping (for example how to emulate 3270 keys such as PA1, PA2, etc.) can be found in *IBM AS/400 TCP/IP Configuration and Operation - GG24-3442*.

The Attn key will display the Telnet menu.

8.3 Sending Mail (SMTP)

8.3.1 Sending Mail from VM to OS/400

Please refer to Section 3.4, “Simple Mail Transfer Protocol (SMTP)” on page 181 for more details about about how to use the SMTP client function on a VM system.

8.3.2 Sending Mail from OS/400 to VM

Please refer to *Version 2 TCP/IP Guide - SC41-9875* and to *IBM AS/400 TCP/IP Configuration and Operation - GG24-3442* for more information.

Chapter 9. Network Management in TCP/IP

IBM TCP/IP Version 2 Release 2 for VM offers two new important functions for network management. The ROUTED server manages the local routing tables and also broadcasts them on the local network. If physical links fail, after a few minutes ROUTED may find an alternate route. The SNMP monitor and agents allow a NetView operator to display TCP/IP network information from a central point.

9.1 Alternate Routes with RIP

The system VMESA has multiple possible paths to reach a system on the token-ring network; see Figure 39 on page 267 for an overview. The most efficient way is to use the token-ring interface of the 3172 which is attached to the VMESA system. What happens when the physical token-ring interface to the system VMESA fails? This situation can be simulated by stopping the token-ring device. System VMESA then has an alternate route to the token-ring network, via a RISC System/6000 platform which has two interfaces (one on the Ethernet network and one on the token-ring network). Since the RISC System/6000 is running RouteD (which implements RIP), this alternate route may be automatically selected.

The definitions in use in the ″PROFILE TCPIP″ of the VMESA system were:

```
AUTOLOG
   ..........
   ROUTED   Password    ; FTP SERVER
   .........
ENDAUTOLOG

PORT
    .......
    520 UDP ROUTED              ; RouteD Server
    .........

; Routing information (if you are not using the ROUTED server)
GATEWAY
; NETWORK     FIRST   DRIVER    PACKET SIZE   SUBN MASK      SUBN VALUE
;              HOP
;
9.67.38.135   =     SNALMV18  DEFAULTSIZE   HOST   1
9.67.38.145   =     SNALPOK  2000           HOST   1
;
BSDROUTINGPARMS FALSE
  LINK     MAXMTU      METRIC   SUBNET MASK       DEST ADDR
;
  ETH1     DEFAULTSIZE   0        255.255.255.192   0      2
  TR1      DEFAULTSIZE   0        255.255.255.192   0      2
 ENDBSDROUTINGPARMS

   ...........
```

1 The point-to-point links are excluded from the dynamic routing process because the remote systems were not running RouteD at that time.

2 The token-ring and the Ethernet links are part of the dynamic routing process.

The following example shows how RIP can react to failing links and choose an alternate route to reach the destination.

```
netstat gate        0
VM TCP/IP Netstat V2R2
Known gateways:

NetAddress      FirstHop        Link     Pkt Sz  Subnet Mask     Subnet Value
----------      --------        ------   ------  -----------     ------------
9.0.0.0          direct         ETH1     Default 0.255.255.192   0.67.32.64  1
9.0.0.0          direct         TR1      Default 0.255.255.192   0.67.38.64  2
9.67.38.135      direct         SNALMV18 Default HOST    3
9.67.38.145      direct         SNALPOK  2000    HOST    3
Ready;
netstat devlinks
VM TCP/IP Netstat V2R2

Device LCS1             Type: LCS        Status: Ready   4
Queue size: 0           Address: 0EC2
   Link TR1             Type: IBMTR      Net number: 1

Device LCS2             Type: LCS        Status: Ready   5
Queue size: 0           Address: 0EC4
   Link ETH1            Type: ETHERNET   Net number: 1

Device SNADMV18         Type: SNA IUCV   Status: Connected
Queue size: 0           Vm Id: SNALNKA   Pgm: SNALINK      LU: RAIATC1
   Link SNALMV18        Type: IUCV       Net number: 2

Device SNADPOK          Type: SNA IUCV   Status: Connected
Queue size: 0           Vm Id: SNALNKA   Pgm: SNALINK      LU: SCXASNAL
   Link SNALPOK         Type: IUCV       Net number: 3
Ready;
obey cmd stop lcs1          6
VM TCP/IP Obeyfile
Requesting TCPIP to accept ¢OBEYCMD TCPIP *¢ on TCPMAINT 191  ...
File ¢OBEYCMD TCPIP *¢ has been read and obeyed
Ready;
netstat devlinks
VM TCP/IP Netstat V2R2

Device LCS1             Type: LCS        Status: Inactive    7
Queue size: 0           Address: 0EC2
   Link TR1             Type: IBMTR      Net number: 1

Device LCS2             Type: LCS        Status: Ready
Queue size: 0           Address: 0EC4
   Link ETH1            Type: ETHERNET   Net number: 1

 Device SNADMV18         Type: SNA IUCV   Status: Connected
 Queue size: 0           Vm Id: SNALNKA   Pgm: SNALINK      LU: RAIATC1
    Link SNALMV18        Type: IUCV       Net number: 2

 Device SNADPOK          Type: SNA IUCV   Status: Connected
 Queue size: 0           Vm Id: SNALNKA   Pgm: SNALINK      LU: SCXASNAL
    Link SNALPOK         Type: IUCV       Net number: 3
 Ready;
```

```
netstat gate
VM TCP/IP Netstat V2R2
Known gateways:

NetAddress      FirstHop        Link     Pkt Sz  Subnet Mask     Subnet Value
----------      --------        ------   ------  -----------     ------------
9.0.0.0          direct         ETH1     Default 0.255.255.192   0.67.32.64
9.0.0.0         9.67.32.85      ETH1     Default 0.255.255.192   0.67.38.64  8
9.67.38.135      direct         SNALMV18 Default HOST
9.67.38.145      direct         SNALPOK  2000    HOST
Ready;
```

0 The NETSTAT GATE command is used to display the routing entries before we stop the token-ring interface.

1 The connection to the Ethernet (Subnet Value = 0.67.32.64) is directly available (FirstHop = direct) via the ETH1 (Link = ETH1) link.

2 The connection to the token-ring (Subnet Value = 0.67.38.64) is directly available (FirstHop = direct) via the TR1 (Link = TR1) link.

3 SNALINK connections are also available.

4 The token-ring link (TR1) on device LCS1 is available.

5 The Ethernet link (ETH1) on device LCS2 is available.

6 The OBEY command is used to stop the token-ring device (LCS1).

7 NETSTAT DEVLINKS displays the status of all physical interfaces defined in the ″PROFILE TCPIP″. The device LCS1 associated to the link TR1 is now inactive. The device LCS2 which gives access to the Ethernet is available. From now on RouteD will not receive any packet from the TR1 interface. After the usual timeout (3 minutes), RouteD will delete the route to the 9.67.38.64 network (token-ring) via the TR1 interface. It will then receive a RIP packet advertising that the 9.67.38.64 network is reachable via the router 9.67.38.85 (RISC System/6000). Since it is the shortest route that will be advertised for this network (other routers such as VM14 or VM15 will advertise longer routes); it will be included in the routing table.

8 The network 9.67.38.64 (token-ring) is now accessible via router 9.67.38.85.

It took a few minutes (between 3 and 4) to adjust the routing tables to the alternate route. This is because RIP does not react on failing links, but on timeouts for responses.

The tracing example in Section 10.6, “Tracing” on page 257 shows the trace of the *PING* command for this scenario.

Now the VMESA system is able to send packets to the token-ring network. However this is not enough to provide immediate communication between all our systems. If you refer to Figure 39 on page 267 you will notice that 2 routers are available to reach the Ethernet network from the token-ring network: the VMESA system and a RISC System/6000 system. Both of them will advertise a hop count with a value of 1.

Therefore it may occur that VM14 has a routing entry for network 9.67.32.64 (Ethernet) which points to the RISC System/6000, and that VM15 has a routing entry for the Ethernet network which points to VMESA. This may happen for example if VM14 amd VM15 are not started at the same time; VM14 may receive

its first routing packet from the RISC System/6000 and VM15 may receive its first routing packet from VMESA.

Since both routers will advertise a hop count of 1, neither VM14 nor VM15 will update their routing tables.

As soon as VMESA has a route to the token-ring network via the RISC System/6000 it will be able to communicate with VM14, but you may have to wait for another minute until VM15 deletes its routing statement (Ethernet network accessed through VMESA) and adds a new one (Ethernet network reached via the RISC System/6000)

The following sequence shows that when the token-ring interface becomes available again, the routing table are updated accordingly. This process is much faster than the previous one since no timeout mechanism occurs. RouteD will receive its own broadcast packets immediately.

```
obey cmd start lcs1     1
VM TCP/IP Obeyfile
Requesting TCPIP to accept ¢OBEYCMD TCPIP *¢ on TCPMAINT 191  ...
File ¢OBEYCMD TCPIP *¢ has been read and obeyed
Ready;
netstat gate
VM TCP/IP Netstat V2R2
Known gateways:

NetAddress      FirstHop        Link     Pkt Sz  Subnet Mask     Subnet Value
----------      --------        ------   ------  -----------     ------------
9.0.0.0          direct         ETH1     Default 0.255.255.192   0.67.32.64
9.0.0.0          direct         TR1      Default 0.255.255.192   0.67.38.64   2
9.67.38.135      direct         SNALMV18 Default HOST
9.67.38.145      direct         SNALPOK  2000    HOST
```

1 Start the token-ring device.

2 The token-ring network is again a directly attached network.

9.2 SNMP in NetView

The SNMP support in NetView consists of a command processor and some sample CLISTs. A CLIST, or the NetView operator, may invoke the SNMP command in order to obtain an answer from the addressed remote SNMP agent. Information that can be obtained from SNMP agents is defined by the Management Information Base (MIB). The MIB defines objects, such as packet counts and routing tables, which are relevant to a TCP/IP environment. The objects defined by the MIB are divided into groups, with each group representing a set of management data.

9.2.1 Command Interface

The base implementation of SNMP in NetView is the SNMP command processor. It allows the NetView operator to perform SNMP requests.

9.2.1.1 SNMP GET Request

In the following example the NetView operator sends an ***SNMP GET*** request to the SNMP agent on system VM14 to show the contents of the variable SYSDESCR. The community name IBM is used to verify the requestor's authority.

```
NCCF         N E T V I E W               ND114 OPER1     12/04/90 18:22:06
 * ND114    SNMP GET VM14 IBM SYSDESCR
 + ND114    SNM050I SNMP Request 1096 from OPER1 accepted, sent to Query Engine
 = ND114
SNM040I SNMP Request 1096 from OPER1 Returned the following response:
SNM042I Variable name: 1.3.6.1.2.1.1.1.0
SNM043I Variable value type: 9
SNM044I Variable value: IBM 9377, VM/SP RELEASE 6, SERVICE LEVEL 601, VM TCP/IP
SNM044I Variable value: V2R2
SNM049I SNMP Request 1096 End of response
 ------------------------------------------------------------------------------
```

9.2.1.2 SNMP PING

SNMP in NetView has also implemented the *PING* command, which is the basic tool to test TCP/IP connectivity. The NetView operator does not need to logon to a TCP/IP system in order to test the availability of a certain host within the network.

```
NCCF         N E T V I E W               ND114 OPER1     12/04/90 18:20:05
 * ND114    SNMP PING VM15
 + ND114    SNM050I SNMP Request 1094 from OPER1 accepted, sent to Query Engine
 = ND114
SNM040I SNMP Request 1094 from OPER1 Returned the following response:
SNM042I Variable name: 1.3.6.1.4.1.2.2.1.3.2.9.67.32.19
SNM043I Variable value type: 1
SNM044I Variable value: 444
SNM049I SNMP Request 1094 End of response
 ------------------------------------------------------------------------------
The variable value in the ping command is the turnaround time in milliseconds.
```

9.2.1.3 SNMP Commands

The following SNMP commands are implemented in NetView:

- SNMP GET host_name com_name var_name
- SNMP GETNEXT host_name com_name var_name
- SNMP SET host_name com_name var_name var_value
- SNMP TRAPSON net_mask net_desired
- SNMP TRAPSOFF filter_id
- SNMP MIBVNAME asn.1_name
- SNMP PING host_name

Refer to the *IBM TCP/IP V2 R2 for VM: User's Guide* for a complete description of the syntax and meaning of these commands.

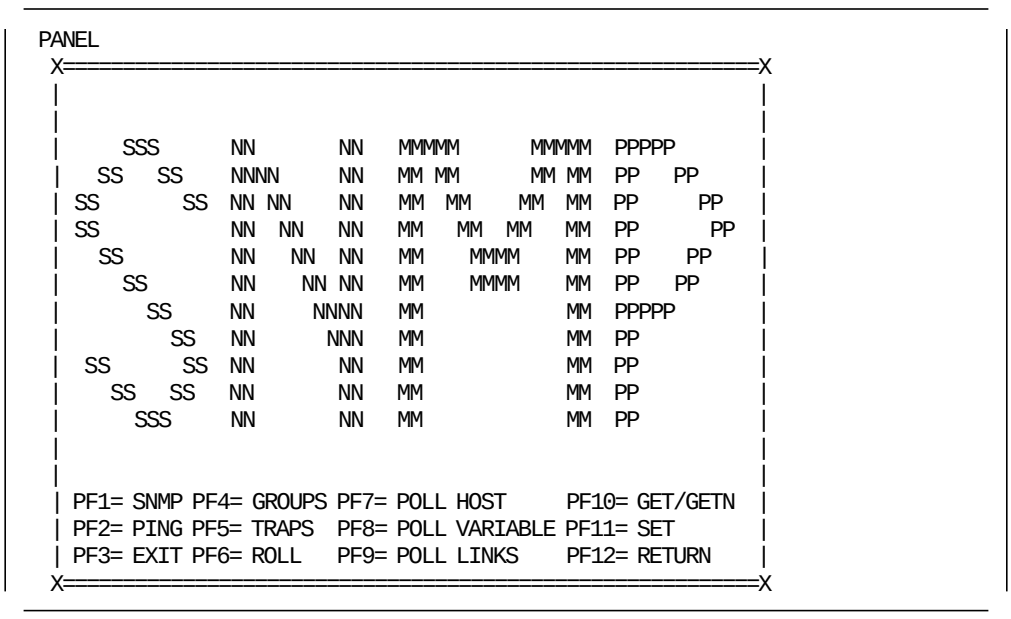
9.2.2 CLIST Interface

Two menu driven CLIST applications are supplied as samples with the installation tape: an EXEC2- and a REXX-based application. To use REXX in a VM environment you must have VM/SP Release 6 and VM NetView Release 3 with the REXX SPEs installed. The APAR numbers that correspond to this SPE are: VM36993 on CMS, VM36994 on REXX and VM36998 on GCS. Only the EXEC2 application is shown in this documentation.

To invoke the NetView Command Language panel, type: **SNMPRUN**. This will start the Master Driver panel from which various SNMP data can be gathered via a panel interface.

9.2.2.1 SNMP Master Panel

The master panel looks like this:



The valid function key selections are:

- ***PF1*** - SNMP help panel
- ***PF2*** - Ping panel
- ***PF3*** - Exit
- ***PF4*** - Group panels
- ***PF6*** - ROLL

- ***PF10*** - GET/GETN panel
- ***PF11*** - SET panel
- ***PF12*** - Return.

All other PF keys will result in no action and this same panel will be displayed.

9.2.2.2 SNMP PING Panel

When PF2 is hit, the NetView operator is presented with a panel where up to 10 host names can be entered.

```
PINGHP
 X=============================================================X
 |                                                             |
 | ENTER THE HOST NAMES YOU WISH TO PING:                      |
 |                                                             |
 | HOST:  vm14                                                 |
 |        vm15                                                 |
 |        mvs18                                                |
 |        mvs20                                                |
 |        rios                                                 |
 |        psaix                                                |
 |        s881                                                 |
 |        rolf                                                 |
 |        os25                                                 |
 |        hpux                                                 |
 |                                                             |
 |                                                             |
 | PF1 = HELP  PF4 =         PF7 =          PF10 =             |
 | PF2 =       PF5 =         PF8 =          PF11 =             |
 | PF3 = EXIT  PF6 = ROLL    PF9 =          PF12 = RETURN      |
 X=============================================================X
```

After pressing Enter, an SNMP PING command is executed for each of these host names.

When all PING requests are completed (response received or timeout) the NetView operator is presented with the PING RESULTS panel. This panel shows the minimum-round-trip-time for each of the hosts.

```
PINGHSTP
 X=============================================================X
 |                                                             |
 | PING RESULTS                                                |
 |                                                             |
 | HOST              MINRTT                                    |
 | vm14               63                                       |
 | vm15               164                                      |
 | mvs18              1366                                     |
 | mvs20             PING FAILURE                              |
 | rios               65                                       |
 | psaix              76                                       |
 | s881              PING FAILURE                              |
 | rolf               108                                      |
 | os25               114                                      |
 | hpux                                                        |
 |                                                             |
 |                                                             |
 | ERROR TEXT: unknown host                                    |
 |                                              PF12 = RETURN  |
 X=============================================================X
```

9.2.2.3 SNMP Group Panels

The Group panel allows the NetView operator to retrieve all the important TCP/IP system status indicators from any TCP/IP host running an SNMP agent. Basically, the operator can do the same as with the ***NETSTAT*** command, but he does not need to logon to that host.

When the operator hits PF4 to obtain the panel to select gathering of various MIB group information, the following panel is displayed.

```
MIBGET
 X=============================================================X
 |             QUERY SNMP GROUP/TABLE                          |
 |                                                             |
 | HOST   vm14                                                 |
 |                                                             |
 | COMMUNITY   IBM                                             |
 |                                                             |
 | SELECT GROUP  2                                             |
 |   1.  SYS      - GET SYSTEM GROUP MIB VARIABLES             |
 |   2.  IFTAB    - GET INTERFACE TABLE                        |
 |   3.  ARPTAB   - GET ARP TABLE                              |
 |   4.  IP       - GET IP GROUP MIB VARIABLES                 |
 |   5.  IPADDR   - GET IP ADDRESS TABLE                       |
 |   6.  IPROUTE  - GET IP ROUTING TABLE                       |
 |   7.  ICMP     - GET ICMP GROUP MIB VARIABLES               |
 |   8.  TCP      - GET TCP GROUP MIB VARIABLES                |
 |   9.  TCPTAB   - GET TCP CONNECTION TABLE                   |
 |  10.  UDP      - GET UDP GROUP MIB VARIABLES                |
 |  11.  EGP      - GET EGP NEIGHBOR TABLE                     |
 | PF1 = HELP    PF4 =         PF7 =          PF10 =           |
 | PF2 =         PF5 =         PF8 =          PF11 =           |
 | PF3 = EXIT    PF6 = ROLL    PF9 =          PF12 = RETURN    |
 X=============================================================X
```

The operator will input a host name or internet address, a community name and select a number from 1 - 11. A series of SNMP GETs and GET_NEXTs is issued to obtain all the MIB variable information for that particular group.

SNMP Interface Panel: The following screens display the information received when the operator selected the interface table to be displayed.

```
MIB2P
 X=============================================================X
 |        INTERFACE TABLE for vm14                             |
 | INDEX:           1                                          |
 | DESCR:           IBM CETI Token Ring adapter                |
 | TYPE:            9                                          |
 | MTU:             2048                                       |
 | SPEED:           4000000                                    |
 | PHYSADDRESS:     400000114014                               |
 | ADMINSTATUS:     1               OUTERRORS: 24              |
 | OPERSTATUS:      1               OUTQLEN:   0               |
 | LASTCHANGE:      322784                                     |
 | INOCTETS:        5998916                                    |
 | INUCASTPKTS:     2539                                       |
 | INNUCASTSPKTS:   42646                                      |
 | INDISCARDS:      0                                          |
 | INERRORS:        0                                          |
 | INUNKNOWNPROTOS: 0                                          |
 | OUTOCTETS:       1845951                                    |
 | OUTUCASTPKTS:    2645                                       |
 | OUTNUCASTPKTS:   10784                                      |
 | OUTDISCARDS:     0              PRESS PF12 TO RETURN        |
 | ERROR:                          PRESS ENTER TO CONTINUE     |
 X=============================================================X
```

OPERSTATUS 1 means that the interface is ready to pass packets.

```
MIB2AP
 X=============================================================X
 |        INTERFACE TABLE for vm14                             |
 | INDEX:           2               OUTQLEN:  0                |
 | DESCR:           IBM IP-over-SNA link                       |
 | TYPE:            1               ERROR:                     |
 | MTU:             8520                                       |
 | SPEED:           56000                                      |
 | PHYSADDRESS:                                                |
 | ADMINSTATUS:     1                                          |
 | OPERSTATUS:      2                                          |
 | LASTCHANGE:      25243424                                   |
 | INOCTETS:        1240876                                    |
 | INUCASTPKTS:     8186                                       |
 | INNUCASTSPKTS:   0                                          |
 | INDISCARDS:      0                                          |
 | INERRORS:        0                                          |
 | INUNKNOWNPROTOS: 0                                          |
 | OUTOCTETS:       1245186                                    |
 | OUTUCASTPKTS:    10608                                      |
 | OUTNUCASTPKTS:   0                                          |
 | OUTDISCARDS:     0              PRESS PF12 TO RETURN        |
 | OUTERRORS:       2378           PRESS ENTER TO CONTINUE     |
 X=============================================================X
```

OPERSTATUS 2 means that the interface is down.

SNMP ARP Table: Another useful display for analyzing connectivity problems is to receive the Address Resolution Protocol (ARP) table.

```
MIB3P
 X=============================================================X
 |       ARP TABLE for HOST vm14                               |
 |                                                             |
 |                                                             |
 | INTERFACE     PHYSADDRESS            NETADDRESS             |
 | 1             400000114014            9.67.32.18            |
 | 1             400030010002            9.67.32.21            |
 | 1             10005A4F1F0D            9.67.32.23            |
 | 1             400065732252            9.67.32.28            |
 |                                                             |
 |                                                             |
 |                                                             |
 | ERROR TEXT:                        PF8=FORWARD PF12=RETURN  |
 X=============================================================X
```

Old or incorrect entries in this table can also cause connectivity problems. The whole table can be flushed using the TRANSLATE statement with the OBEYFILE command.

SNMP Routing Table: The following panel shows the routing table of system vm14.

```
MIB4CP
 X=============================================================X
 |  IP ROUTING TABLE FOR HOST vm14                             |
 |                                                             |
 |                                                             |
 |DEST        IF   M1   M2   M3   M4   NEXTHOP     TYP  PRO   AGE   |
 |0.0.0.0     1    -1   -1   -1   -1   9.67.32.23  4    8    13944 |
 |9.67.32.16  1    -1   -1   -1   -1   9.67.32.18  3    8    14164 |
 |9.67.32.35  2    -1   -1   -1   -1   9.67.32.35  3    8    14164 |
 |9.67.32.80  1    -1   -1   -1   -1   9.67.32.23  4    8    14163 |
 |                                                             |
 |                                                             |
 |                                                             |
 | ERROR TEXT:                                PF12 = RETURN    |
 X=============================================================X
```

The meaning of the entries can be found in RFC 1156, which covers the MIB for SNMP. For example the TYP field means the ipRouteType MIB variable, which shows the type of route. ″3″ means a route to a directly connected (sub-)network and ″4″ means a route to a non-local host/network/sub-network.

9.2.3 TRAPs

The SNMP query engine can only forward those traps that it receives. Each agent has a trap destination table, which lists all the hosts that should receive the agent's traps. The ***SNMP TRAPSON*** command must be used to ask the SNMP query engine to forward SNMP traps to NetView. This command permits the specification of a filtering condition, thus enabling the query engine to perform some filtering on its own.

The following command uses no filtering at all.

```
NCCF         N E T V I E W               ND114 OPER1     12/07/90 10:58:20
 * ND114    SNMP TRAPSON
 + ND114    SNM050I SNMP Request 1175 from OPER1 accepted, sent to Query Engine
 = ND114
SNM040I SNMP Request 1175 from OPER1 Returned the following response:
SNM045I Major error code: 0
SNM046I Minor error code: 0
SNM047I Error index: 0
SNM048I Error text: no error
SNM049I SNMP Request 1175 End of response
 ------------------------------------------------------------------------------


 ???
```

The following SNMP TRAPs have been produced by stopping (***obey cmd stop snadvm16***) and starting the SNALINK connection from system VM14 to VM16. A TRAP PDU has been sent from the vm14 agent to defined monitors in the trap destination file. The SNMP query engine is one of these destinations, now decoding and forwarding the PDU to NetView via the SNMPIUCV task.

When traps arrive in NetView, NetView displays a multi-line message in the following form.

```
STATMON.BROWSE     ACTS  NETWORK LOG FOR 12/07/90 (90341) COLS 017 094  10:39
HOST: HOST14               *1*     *2*     *3*     *4*        SCROLL ==  HALF
 ---2----+----3----+----4----+----5----+----6----+----7----+----8----+----9----
 ND114 P% 11:05:31   SNM030I SNMP request 1175 received following trap:
 ND114 P% 11:05:31   SNM031I Agent Address: 9.67.32.18
 ND114 P% 11:05:31   SNM032I Generic trap type: 2
 ND114 P% 11:05:31   SNM033I Specific trap type: 0
 ND114 P% 11:05:31   SNM034I Time stamp: 82300
 ND114 P% 11:05:31   SNM035I Enterprise Object ID: 1.3.6.1.4.1.2.2.1.2.3
 ND114 P% 11:05:31   SNM036I Variable name: 1.3.6.1.2.1.2.2.1.1
 ND114 P% 11:05:31   SNM037I Variable value type: 1
 ND114 P% 11:05:31   SNM038I Variable value: 3
 ND114 P% 11:05:31   SNM039I SNMP request 1175 End of trap data

 ND114 P% 11:06:28   SNM030I SNMP request 1175 received following trap:
 ND114 P% 11:06:28   SNM031I Agent Address: 9.67.32.18
 ND114 P% 11:06:28   SNM032I Generic trap type: 3
 ND114 P% 11:06:28   SNM033I Specific trap type: 0
 ND114 P% 11:06:28   SNM034I Time stamp: 88500
 ND114 P% 11:06:28   SNM035I Enterprise Object ID: 1.3.6.1.4.1.2.2.1.2.3
 ND114 P% 11:06:28   SNM036I Variable name: 1.3.6.1.2.1.2.2.1.1
 ND114 P% 11:06:28   SNM037I Variable value type: 1
 ND114 P% 11:06:28   SNM038I Variable value: 3
 ND114 P% 11:06:28   SNM039I SNMP request 1175 End of trap data

CMD==
1=HELP 2=END 3=RET 4=TOP 5=BOT 6=ROLL 7=BCK 8=FWD 9=RPTFND 10=LFT 11=RGT 12=ALL
```

The content of a trap PDU can be found in RFC 1157. A generic trap type ″2″ indicates a link down condition, while ″3″ means link up. The variable name 1.3.6.1.2.1.2.2.1.1 (ifIndex) indicates the number of the physical interface which, in this example, is ″3″.

When you receive an ASN.1 variable name as part of a trap, you can look up the short name of the variable by using the SNMP MIBVNAME command.

```
NCCF         N E T V I E W              ND114 OPER1     12/07/90 11:09:01
 * ND114    SNMP MIBVNAME 1.3.6.1.2.1.2.2.1.1
 + ND114    SNM050I SNMP Request 1177 from OPER1 accepted, sent to Query Engine
 = ND114
SNM040I SNMP Request 1177 from OPER1 Returned the following response:
SNM042I Variable name: 1.3.6.1.2.1.2.2.1.1
SNM043I Variable value type: 9
SNM044I Variable value: ifIndex
SNM049I SNMP Request 1177 End of response
```

9.3 The NETSTAT Command

The ***NETSTAT*** command displays the network status of the local host, information about TCP/IP connections, network clients, routers, devices and the Telnet server. NETSTAT also drops connections and executes commands for users in the TCPIP virtual machine's obey list. For example, a TCP/IP server can be shut down via the NETSTAT command (NETSTAT CP FORCE FTPSERVE) or TCP/IP itself can be shut down (NETSTAT CP EXT). Usually NETSTAT is used to display information about the locally active TCP/IP system. The information available via NETSTAT is comparable to the NetView SNMP interface, and is available to any local TCP/IP user, unlike the NetView interface.

The different possiblities of the netstat command are:

```
netstat ?
VM TCP/IP Netstat V2R2
Usage: netstat information-list
Current information viewable:
ALL       - Everything about a connection
CLIENTS   - Current clients.
CONN      - Active control blocks.
GATE      - Current known gateways.
HOME      - Home address list.
INTERVAL  - Full screen, real-time, connection display
            (3278 terminal only)
UP        - Date and time tcpip was last started
TELNET    - Telnet connections and logical devices
DEVLINKS  - Devices and links
POOLSIZE  - Free pool status
SOCKETS   - Socket interface users and their sockets
ALLCONN   - With CONN or INTERVAL, shows TIME-WAIT and
            CLOSED connections
TCP       - Displays detailed information about the specified
            TCPIP server.
TEST      - Displays detailed information about the TCPIPTES
            server.
Commands available:
CP        - Issue a CP command
DROP      - Drop a TCP connection
RESETPOOL - Reset record of pool informs sent
Ready;
```

Warning

The NETSTAT command won't use a name server. In order to have hostnames displayed instead of internet addresses one should code the file "HOSTS LOCAL" and use the MAKESITE command.

9.3.1 Sample Outputs of the NETSTAT Command

The NETSTAT command can help you monitor your network in many ways. We will give you some outputs of NETSTAT commands, to give you the information necessary to interpret your own system information.

The ***NETSTAT TELNET*** command allows you to check the telnet connections. The following is an output which assumes the the host FRED is in the hosts files:

```
netstat telnet
VM TCP/IP Netstat V2R2
Internal Telnet server status:

Conn Status        Foreign Host     B out      B in  Logical device status
---- ------        ------------     -----      ----  --------------------
1000 Established   FRED              1656      1960  L001 ENABLED
1003 Listen        *                    0         0
```

The ***NETSTAT DEvlinks*** command allows you to verify the status and types of your host's devices and links.

```
netstat devlinks
netstat devlinks
VM TCP/IP Netstat V2R2

Device LCS1              Type: LCS         Status: Ready
Queue size: 0            Address: 0EC2
   Link TR1              Type: IBMTR       Net number: 1

Device LCS2              Type: LCS         Status: Ready
Queue size: 0            Address: 0EC4
   Link ETH1             Type: ETHERNET    Net number: 1

Device SNADMV18          Type: SNA IUCV    Status: Connected
Queue size: 0            Vm Id: SNALNKA    Pgm: SNALINK      LU: RAIATC1
   Link SNALMV18         Type: IUCV        Net number: 2
```

The ***NETSTAT HOme*** command shows the IP addresses of your host's interfaces, that is, the "home list".

```
netstat home
VM TCP/IP Netstat V2R2
Home address list:

Address            Link
-------            ------
9.67.32.65         ETH1
9.67.38.65         TR1
9.67.38.136        SNALMV18
```

The ***NETSTAT Interval*** command gives you the opportunity to view all your active servers at a glance. It provides information about active TCP connections. The information is updated at the specified interval of time. The valid values of the time period range between 0 and 3600 seconds, but 0, if specified, will be changed to 1. The default period is 20 seconds.

```
07/02/92          VM TCP/IP Real Time Network Monitor              00:01:18

User Id     B Out    B In  L Port Foreign Socket        State         Idle
--------    -----    ----- ------ --------------        -----------  --------
NAMESRV         0        0 DNS    *..*                  Listen        0:22:20
NAMESRV       860      626 DNS    *..*                  UDP           0:01:52
FTPSERVE        0        0 FTP-C  *..*                  Listen        0:04:50
INTCLIEN        0        0 TELNET *..*                  Listen        0:03:12
INTCLIEN     1653     1957 TELNET FRED..1043            Established   0:03:11









Refresh interval: 20 Seconds.                          TCB¢s In Use:  4
```

Warning.

If a host (FRED for example) is using the VM TCP/IP you must have the files ″HOSTS SITEINFO″ and ″HOSTS ADDRINFO″. If the remote host is found in those file, its name will be displayed using the NETSTAT INTERVAL command (This is the case for the host FRED in the previous screen). If the remote host is not found, then its IP address will be displayed. If not hosts files are found, your VM session will hang.

The ***NETSTAT SOCKet*** command provides information about each client using the socket interface. The following list gives some explanations about the content of the fields as displayed on the output screen.

Name The client′s virtual machine name.

Subtask Identifier When combined with the virtual machine name, yields a unique identifier for the client:

- For socket programs written in **C**, the EBCDIC hex representation of an address within the program is used as the subtask identifier.
- For socket programs generating their own IUCV calls, the subtask identifier is any string up to 8 characters chosen by the program.

Path id The TCP/IP virtual machine′s IUCV path number for this client′s path.

Pending call The socket call issued by this client, and which is waiting for completion:

- *readv()* calls are displayed as ″read″.

- *recvfrom()* and *recvmsg()* calls are both displayed as ″recv″.
- *write()* calls are displayed as ″write″.
- *sendmsg()* calls may be displayed as either ″send″ or ″sendto″ depending on the calling parameters.
- This field is omitted if no socket call is waiting for completion.

The socket descriptor.

- Note that more than one detail line may have the same socket descriptor.

  In this case the first is a listening stream socket, and subsequent lines with the same descriptor are TCP connections awaiting acceptance (the acceptance queue) or waiting establishment (the almost accept queue).

  See also the ″Flgs″ field.

- Detail lines with no descriptor list TCP connections whose socket has been closed but which have not yet entered *Closed* state.

Type The type of the socket:

- ″Stream″ for stream (TCP) sockets.
- ″Dgram″ for stream (UDP) sockets.
- ″Raw″ for raw sockets.
- ″DPI″ for the special SNMP DPI socket type used only by SNMP agent.

Bound to Shows the address and port that the socket is bound to, or ″not bound″.

Unbound TCP and UDP sockets are not displayed by ***NETSTAT CONN*** or ***NETSTAT INTERVAL*** command.

Connected to Shows the address and port that the socket is connected to, or ″not connected″.

State

- For TCP sockets, displays the TCP connection state. If blank, ″CONN″ column will also be blank.
- For raw sockets, displays the IP protocol number.

Flgs For TCP sockets only:

- ″L″ indicates a listening socket.
- ″A″ indicates a connection on the almost-accept queue.
- ″C″ indicates a connection on the accept queue.

Conn For TCP sockets only, displays TCPIP′s internal TCP number:

- If blank, then the ″Flgs″ column will show ″L″. This is a listening socket for which the accept queue is full, or for which TCPIP is temporarily unable to allocate resources to put a TCB in Listen state.

- Attempts to connect to the port shown in the ″Bound to″ column will be ignored, not reset, in order to let the connecting TCP retry in a few seconds rather than abort its attempt.

```
netstat socket
VM TCP/IP Netstat V2R2
Socket interface status:
#    Type    Bound to            Connected to         State        Flgs Conn
=    ====    ========            ============         =====        ==== ====
Name: ROUTED       Subtask: 00035678   Path id: 1          Pending call: select
3    Dgram   *..520              Not connected
Name: SNMPD        Subtask: 00050800   Path id: 2          Pending call: select
3    Dgram   VM14..161           Not connected
4    Stream  *..1069             *..*                 Listen       L    1004
6    DPI
Name: SNMPQE       Subtask: 0005d980   Path id: 3          Pending call: select
3    Dgram   *..1063             Not connected
4    Dgram   *..162              Not connected
5    Stream  *..1070             *..*                 Listen       L    1007
7    Raw     Not bound           Not connected        1
```

The ***NETSTAT POOLsize*** command provides information about objects in the free pool.

Object Name of the object:

- ACB - Activity Control Blocks
- CCB - Client Control Blocks
- Dat Buf - Data Buffers
- Sm dat buf - Small data buffers
- Env - Small Envelopes
- Lrg Env - Large Small Envelopes
- RCB - Raw IP Control Blocks
- SCB - Socket Control Blocks
- SKCB - Socket Interface Control Blocks
- TCB - Transmission Control Blocks
- UCB - User Datagram Protocol Control Blocks

 # alloc Number of objects allocated at startup time (includes objects in use plus objects in the free pool).

 # free Number of objects in the free pool.

Lo-Water Low water number for each object in free pool (lower number reached since TCPIP startup).

Permit size The value at which TCP/IP considers an object's free pool to be running low.

- When the free buffer pool size of an object drops below the permitted size, TCP/IP sends messages to all users in the *INFORM* list.
- Only one notification is sent regardless of the number of fluctuations in the free pool size.

```
netstat poolsize
VM TCP/IP Netstat V2R2
TCPIP Free pool status:
Object     # alloc     # free      Lo-water    Permit size
======     =======     ======      ========    ===========
ACB          1000        987         977         100
CCB           150        131         131          10
Dat buf       160        152         149          32
Sm dat buf     50         47          44           5
Env           750        750         742          75
Lrg env        50         49          48          10
RCB            50         49          49           3
SCB           256        227         224          17
SKCB          256        248         247          17
TCB           256        248         246          17
UCB           100         94          92           6
```

The ***NETSTAT RESETPool*** command resets the ″free informed flags″ (list of the users which have been informed of the free pool dropping below the permitted size), allowing notification messages to be sent.

```
netstat resetpool
VM TCP/IP Netstat V2R2
Function performed
Ready; T=0.10/0.26 12:34:33
```

Refer to *IBM TCP/IP V2 R2 for VM: User's Guide* to find more information about the **NETSTAT** command.

9.4 The OBEYFILE Command

The ***OBEYFILE*** command allows you to execute the TCP/IP configuration statements while TCP/IP is running. Primarily, OBEYFILE is used to change the TCP/IP system temporarily while TCP/IP is running. See Section 10.5, “Using the OBEYFILE Command” on page 255 for information on the usage of the OBEYFILE command.

Chapter 10. Debugging

This chapter may cover some basic problems which were discovered when installing and using the product. See the *IBM TCP/IP V2 for VM and MVS: Diagnosis Guide* for a complete description of diagnosis procedures.

10.1 Installation Problems

The installation process may be slightly different than documented in the installation manual. The ″* MEMO″ and ″* README″ files on the installation tapes always contain the most current information on how to install the product. Make sure that you read this information carefully.

10.2 Application Problems

If you have a problem with a client program (FTP, Telnet, etc.) or with one of your own applications, you need to find out if the problem arises due to a temporary communication problem. Bad response times (timeouts), link failures or lost datagrams may cause unpredictable results, especially in UDP (UDP is not a reliable protocol) based applications. Use the PING command several times to see if you have acceptable performance and reliability.

Some servers and clients have a debugging or trace mode implemented, which can be turned on or off. Usually a statement in a configuration file or a command option in the command itself turns the debugging mode on. See the *IBM TCP/IP V2 R2 for VM: Planning and Customization* manual for these debugging modes.

10.3 Connectivity Problems

The basic TCP/IP tools available to analyze connectivity problems are the ***PING*** and the ***NETSTAT*** commands, which are available to all users. You may find the failing component based on the results of these commands. Use numeric IP addresses only for basic connectivity checks. This eliminates the name resolver and host tables as potential sources of problems.

10.3.1 Basic Connectivity Check Procedure

Suppose you cannot use a certain service on a remote host; use the following procedure to identify the failing component (this procedure assumes you are using the IP addresses. If you are using a name server please refer to Section 10.4, “Debugging the Name Server” on page 253).

1. **PING** your own IP address.

   If the command fails, make sure that TCP/IP is running on the local system, and run the verification procedure (V5735FAL).

   PING another host

   - On the same physical network.

     If you get ″ping #1 timed out″ check the physical interface using the **NETSTAT DEVLINKS** command. If your interface is up, then check the

remote host (is it running, is its interface up?). Check the address you just typed.

- On another network

  If you get "Destination network is unreachable", you have to investigate your local routing tables. The following is an example:

```
netstat gate
VM TCP/IP Netstat V2R2
Known gateways:

NetAddress    FirstHop     Link     Pkt Sz  Subnet Mask     Subnet Value
----------    --------     ------   ------  -----------     ------------
9.0.0.0        direct      ETH1     Default 0.255.255.192   0.67.32.64
9.0.0.0        direct      TR1      Default 0.255.255.192   0.67.38.64
9.67.38.135    direct      SNALMV18 Default HOST
9.67.38.145    direct      SNALPOK  2000    HOST
Ready;

ping 8.7.6.5
Ping V2R2: Pinging host 8.7.6.5.  Enter #CP EXT to interrupt.
Ping request failed: Destination network is unreachable
Ready(00004);
```

  The NETSTAT GATE command indicates that no routes are defined to reach the network number 8. Thus the network unreachable message.

- If an IP router (gateway) is involved in reaching the target, PING both its near and far side addresses. If you cannot ping the far side address, make sure that your local host recognizes the gateway as a relay (check with the NETSTAT GATE command). You must have an entry for the network attached to the far end of the router. If you still cannot ping the far side address, check the remote gateway. The routing tables and the status of the physical interfaces are the key components on each gateway.

- **PING** a host beyond the gateway

  Make sure that this host is active. If you have not yet reached the final destination, check the routers on the way (check their routing tables) and the remote host (it must have a routing entry for your network).

Basically you have to check all gateways to ensure that their routing tables are correct and that their physical interfaces are operational.

10.4 Debugging the Name Server

Once you are sure you can communicate with a host using an internet address, you can set up a name server. As said previously, this will free the users to have their own hosts file and to maintain them.

Refer also to Section 2.18.3, “Checking Name Servers” on page 161 for additional information.

Basically four kinds of problems may occur:

1. You are not using a name server (but you think you are using one)
2. You are using a name server (but it is not the one you think)
3. You are using a name server (but it is not the right one)
4. You are using the right name server (but you can’t connect to the remote hosts).

What you can do is:

- For problem 1:
  - Look at the file ″TCPIP DATA″. You may have several copies of it. Check the one which is on the first accessed minidisk and look for the NSINTERADDR statement. If there is no IP address coded there, this means you are not using a name server. Thus you should have a ″HOSTS LOCAL″ file in order to contact the remote hosts.
  - Either create the ″HOSTS LOCAL″ file and use the MAKESITE command to build the files used by the TCP/IP client programs, or code the address of a name server in the ″TCPIP DATA″ file.
- For problem 2:
  - Same as problem 1. Check the address of the name server in the ″TCPIP DATA″ file.
  - If the address is incorrect, copy the ″TCPIP DATA″ file to your A minidisk and correct the NSINTERADDR statement to point to the right name server.
- For problem 3:

  The name server you are using is the one you are supposed to use, but it cannot resolve the name because it is not responsible for the name you are looking for.
  - If you are not the administator of the system, you won’t be able to correct this problem. The SQL database must be updated in order to add an entry to the right name server depending on the domain name provided.
  - You can check the definitions of your local name server either by looking at the ″MASTER DATA″ file if you can access it, or using the NSLOOKUP or DIG command. Please refer to Section 2.18.3.3, “The DIG Command” on page 164 and Section 2.18.3.4, “The NSLOOKUP Command” on page 166 for more details about the use of these commands.
- For problem 4:

  The name server you are querying is the one responsible for the host you want to connect to, but the remote host is not present in the database.
  - Almost like problem 3: The SQL database must be updated to add the entry you are looking for.

– You can bypass this problem by creating a hosts file.

Warning

Most of the problems due to a name server may be bypassed by creating a file ″HOSTS LOCAL″ and by running the MAKESITE command against it. All the client programs will look at the hosts files if they do not receive any answer from a name server; a 30 second delay is to be expected. The only exception to this rule is SMTP. **SMTP will not use host files if it has been customized to use a name server**. This is achieved through the ″MASTER DATA″ file.

10.5 Using the OBEYFILE Command

The ***OBEYFILE*** commands allows you to execute the TCP/IP configuration statements while TCP/IP is running. This is especially important in activating and deactivating physical interfaces without shutting down the whole TCP/IP system. For example, SNALINKs may generate a lot of error messages if the remote side of the link is not active. Deactivating the SNALINK device avoids these error messages.

To use the OBEYFILE command you must first create a file which contains the statements to be executed. Executing the OBEYFILE command will inform the TCPIP disconnected machine to link your minidisk, read the file and execute the statements in the file. The following REXX procedure is more convenient to use because it allows you to supply the TCP/IP statements as parameters of the REXX procedure.

OBEY EXEC

```
/*--------------------------------------------------------------------*/
/* REXX procedure to execute TCP/IP configuration statement           */
/* Syntax: OBEY CMD command_string                                    */
/*             FILE fn ft fm                                          */
/*--------------------------------------------------------------------*/
trace e
arg type cmd
link_password = ¢RTCP¢   1
if type == ¢CMD¢ then
  do
    state ¢OBEYCMD TCPIP A¢
    if rc    0 then ¢ERASE OBEYCMD TCPIP A ¢
    ¢EXECIO 1 DISKW OBEYCMD TCPIP A 1 (FINIS STRING¢ CMD
    ¢OBEYFILE OBEYCMD TCPIP A (¢link_password   1
     exit 0
  end
if type == ¢FILE¢ then
  do
    ¢OBEYFILE¢ cmd ¢ (¢link_password   1
     exit 0
  end
exit 12
```

1 If RACF is used on your VM system, you will have to remove all the link_password statement.

The SNALINK device SNADMV18 causes a lot of error messages because the partner system is not active. If this situation lasts hours or even days, the error messages may unnecessarily fill your logs. The REXX procedure OBEY is used to deactivate the SNALINK device and stop the generation of unnecessary error messages.

```
netstat devlinks
VM TCP/IP Netstat V2R2

Device ILADTRN          Type: ILANS      Status: Ready
Queue size: 0           Address: 0204
   Link ILALTRN         Type: IBMTR      Net number: 1

Device SNADVM14         Type: SNA IUC    Status: Connected
Queue size: 0           Vm Id: SNALNKA   Pgm: SNALINK     LU: AD114TC1
   Link SNALVM14        Type: IUCV       Net number: 2

Device SNADMV18         Type: SNA IUCV   Status: Issued connect
Queue size: 0           Vm Id: SNALNKA   Pgm: SNALINK     LU: RAIATC1
   Link SNALMV18        Type: IUCV       Net number: 4
Ready; T=0.33/0.68 08:48:26

obey cmd stop snadmv18

VM TCP/IP Obeyfile
Requesting TCPIP to accept ¢OBEYCMD TCPIP *¢ on TCPMAINT 191  ...
File ¢OBEYCMD TCPIP *¢ has been read and obeyed
Ready; T=0.33/0.70 08:48:52

netstat devlinks
VM TCP/IP Netstat V2R2

Device ILADTRN          Type: ILANS      Status: Ready
Queue size: 0           Address: 0204
   Link ILALTRN         Type: IBMTR      Net number: 1

Device SNADVM14         Type: SNA IUCV   Status: Connected
Queue size: 0           Vm Id: SNALNKA   Pgm: SNALINK     LU: AD114TC1
   Link SNALVM14        Type: IUCV       Net number: 2

Device SNADMV18         Type: SNA IUCV   Status: Inactive
Queue size: 0           Vm Id: SNALNKA   Pgm: SNALINK     LU: RAIATC1
   Link SNALMV18        Type: IUCV       Net number: 4
Ready; T=0.32/0.63 08:49:44
```

10.6 Tracing

IBM TCP/IP Version 2 Release 2 for VM has very extended tracing facilities. The tracing for specific events, protocols or devices can be enabled or disabled any time during normal TCP/IP operation without the need to restart TCP/IP. The output can be written to a file or to the TCP/IP console. The trace control statements can be defined in the ″PROFILE TCPIP″ configuration file, but it is much more convenient to instruct TCP/IP via the OBEYFILE command to trace a specific event for a short time in order to minimize trace output.

Warning

Traces usually produce a lot of data. Make sure that the 191 minidisk of the user ID TCP/IP is large enough to receive this data and that TCP/IP has write access to that minidisk; otherwise TCP/IP will abend.

The following command sequence produces a trace file on the 191 minidisk in the user ID where TCP/IP is running. The minidisk can be reaccessed by TCPMAINT in order to analyze the trace data.

1. Define a new trace file name using the obey command (which uses OBEYFILE).

   OBEY CMD FILE PING TRACE

2. Instruct TCP/IP to write trace output to a file, not to the screen.

   OBEY CMD NOSCREEN

3. Instruct TCP/IP to trace one or more certain events, protocols or devices. Here we want to trace *PING* commands only.

   OBEY CMD TRACE PING

4. Invoke your specific test procedure in order to produce the trace data you need. Here we ping the the RS/6000 gateway on its far side Ethernet interfaces.

   PING 9.67.32.85

5. Stop tracing after your test procedure has completed.

   OBEY CMD NOTRACE

6. A new trace file name forces TCP/IP to close the old trace file so that you can read the data without shutting down TCP/IP.

   OBEY CMD FILE X X

The following example shows a *PING* trace produced with the above mentioned procedure.

```
Ping called:
13403504:
   Accept ping request -  Ping process (from External interrupt handler)
      Client name: TCPMAINT
      Address: 9.67.32.84
      Length: 256
      Timeout: 10

DoPing sending datagram:
   version: 4
   Internet Header Length: 5 = 20 bytes
   Type of Service:Precedence = Routine
   Total Length: 276 bytes
   Identification: 1234
   Flags: May Fragment, Last Fragment
   Fragment Offset: 0
   Time To Live: 60
   Protocol: ICMP
   Header CheckSum: 9756
   Source Address: 09432022
   Destination Address: 09432054
   Data:
08 00 DA 55 00 CC 89 80 D0 B7 11 0E C6 79 92 47
F1 3C D5 F2 AA 3F 0C A9 28 95 88 69 30 A6 18 F7
49 81 DE 73 98 26 B1 DC 52 62 90 6A 22 F6 41 B0
C9 C4 C5 C5 C6 C2 DA 4B 92 75 DE 83 3A 4D 26 FD
13 BF 9F B3 59 61 8D 7D B1 89 43 11 84 03 97 9F
DB 6A D7 B8 04 5F ED 3E CA 2F 4A 42 55 E9 5C 1A
37 C6 4C 9E 2A 91 98 0A 59 0C 8E CB D8 4C 8A A2
D4 8F FE A6 4B C5 41 DE B2 21 B3 D7 6C CD F0 BC
D2 7F 61 68 59 B0 B3 4B BA 83 60 67 64 6C 3E 4C
28 F2 52 4B 1A 46 6F 4A 03 D0 46 BA 8A 36 5D C5
2C 5F C5 2C 61 BB 67 05 A6 41 FD F8 5D 08 9F 66
28 D9 EC CA 82 52 BC 71 CC F3 AD 2C 7A 24 DD D8
3E DF B0 28 8C BC 39 1E 44 79 14 3E A5 10 26 38
B9 9C CB CA A2 94 0F 42 94 74 E1 6E B3 89 44 0F
92 AE 83 6C 1F 08 DF E6 E6 DF 0B 0A E4 C8 93 F2
E7 CB 7E 6C 28 D1 1F A4 35 4C 32 B3 C8 C5 C3 D3

UpToPing processing datagram:
   version: 4
   Internet Header Length: 5 = 20 bytes
   Type of Service:Precedence = Routine
   Total Length: 276 bytes
   Identification: 21131
   Flags: May Fragment, Last Fragment
   Fragment Offset: 0
   Time To Live: 56
   Protocol: ICMP
   Header CheckSum: 56418
   Source Address: 09432054
   Destination Address: 09432022
   Data:
00 00 E2 55 00 CC 89 80 D0 B7 11 0E C6 79 92 47
F1 3C D5 F2 AA 3F 0C A9 28 95 88 69 30 A6 18 F7
49 81 DE 73 98 26 B1 DC 52 62 90 6A 22 F6 41 B0
C9 C4 C5 C5 C6 C2 DA 4B 92 75 DE 83 3A 4D 26 FD
13 BF 9F B3 59 61 8D 7D B1 89 43 11 84 03 97 9F
DB 6A D7 B8 04 5F ED 3E CA 2F 4A 42 55 E9 5C 1A
```

```
37 C6 4C 9E 2A 91 98 0A 59 0C 8E CB D8 4C 8A A2
D4 8F FE A6 4B C5 41 DE B2 21 B3 D7 6C CD F0 BC
D2 7F 61 68 59 B0 B3 4B BA 83 60 67 64 6C 3E 4C
28 F2 52 4B 1A 46 6F 4A 03 D0 46 BA 8A 36 5D C5
2C 5F C5 2C 61 BB 67 05 A6 41 FD F8 5D 08 9F 66
28 D9 EC CA 82 52 BC 71 CC F3 AD 2C 7A 24 DD D8
3E DF B0 28 8C BC 39 1E 44 79 14 3E A5 10 26 38
B9 9C CB CA A2 94 0F 42 94 74 E1 6E B3 89 44 0F
92 AE 83 6C 1F 08 DF E6 E6 DF 0B 0A E4 C8 93 F2
E7 CB 7E 6C 28 D1 1F A4 35 4C 32 B3 C8 C5 C3 D3

UpToPing: Ping was requested by TCPMAINT
UpToPing: Ping took 1.021 seconds
Switching trace to X X A
```

Chapter 11. National Language Support (NLS)

The aim of this document is not to discuss NLS in full detail. People interrested in NLS should read *TCP/IP and National Language Support - GG24-3840*.

TCP/IP is an ″any-to-any″ communication protocol for different systems. A general mechanism to negotiate the code page at time of the connection is not defined in the architecture. A client uses the services of a remote server. A server may concurrently service multiple clients residing on different systems and using different code pages. There is no way for the server to use the correct translation for each client. The client, however, could use a specific translation table as a parameter when starting the connection.

Note: A proper NLS implementation requires the ability to request a specific translation table invoking a client program.

The NLS support in IBM TCP/IP Version 2 Release 2 for VM consists of translation tables for the translation between the different code pages and the ability to choose a user-defined translation table in some client programs. The different translation tables are supplied in source. A utility allows a user to compile the source into a binary file, which is actually used by TCP/IP. If you need a special translation table, you can create your own.

Translation tables are used to map EBCDIC (VM and MVS systems) hexadecimal codes to ASCII (DOS, OS/2, UNIX and AIX sysems) hexadecimal codes and vice versa.

11.1 Using Your Country NLS Translation Table

IBM TCP/IP Version 2 Release 2 for VM contains the following NLS translation tables in source:

- AUSGER
- BELGIAN
- CANADIAN
- DANNOR
- DUTCH
- FINSWED
- FRENCH
- ITALIAN
- JAPANESE
- PORTUGUE
- SPANISH
- SWISFREN
- SWISGERM
- UK
- US

Translation tables are divided in two parts:

1. To translate from ASCII to EBCDIC

```
;
; ASCII-to-EBCDIC table for French CECP Code Page 297
; 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
;
  00 01 02 03 37 2D 2E 2F 16 05 25 0B 0C 0D 0E 0F     ; 00 ;
  10 11 12 13 3C 3D 32 26 18 19 3F 27 1C 1D 1E 1F     ; 10 ;
  40 4F 7F B1 5B 6C 50 7D 4D 5D 5C 4E 6B 60 4B 61     ; 20 ;
  F0 F1 F2 F3 F4 F5 F6 F7 F8 F9 7A 5E 4C 7E 6E 6F     ; 30 ;
                    ............
                    ............
                    ............
```

For example the ASCII code 23 will be translated into the EBCDIC code B1.

2. To translate from EBCDIC to ASCII.

```
;
; EBCDIC-to-ASCII table for French CECP Code Page 297
; 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
;
  00 01 02 03 DC 09 C3 7F CA B2 D5 0B 0C 0D 0E 0F     ; 00 ;
  10 11 12 13 DB DA 08 C1 18 19 C8 F2 1C 1D 1E 1F     ; 10 ;
  C4 B3 C0 D9 BF 0A 17 1B B4 C2 C5 B0 B1 05 06 07     ; 20 ;
  CD BA 16 BC BB C9 CC 04 B9 CB CE DF 14 15 FE 1A     ; 30 ;
  20 FF 83 84 40 A0 C6 86 5C A4 F8 2E 3C 28 2B 21     ; 40 ;
                 ................
                 ................
                 ................
```

Of course, you can modify these tables the way you want.

There are always two translation tables for each language, one for Telnet usage (for example FRENCH TELXLATE) and another for all the other applications (for example FRENCH TCPXLATE). The Telnet translation table prevents the VM Telnet client from translating special ASCII characters to EBCDIC characters which are not allowed in the 3270 data stream and could lead to unpredictable results. To use one of these translation tables you must convert it into binary with the ***CONVXLAT*** utility.

Use the following procedure to convert the source table:

1. Log on to TCPMAINT.
2. Access the disk containing the translation tables.

   ACCESS 591 C

3. Copy the standard translation table source to your A disk.

   COPY *file_name* **TCPXLATE B STANDARD = A (REPL**

4. Copy the Telnet translation table source to your A disk.

   COPY *file_name* **TELXLATE B TELNET TCPXLATE A (REPL**

5. Convert standard translation table source to binary.

   CONVXLAT STANDARD

6. Convert Telnet translation table source to binary.

   CONVXLAT TELNET

7. Copy the binary standard translation table to your B disk.

   COPY STANDARD TCPXLBIN A = = C (REPL

8. Copy the binary Telnet translation table to your B disk.

   COPY TELNET TCPXLBIN A = = C (REPL

   Note: The C disk here is the TCPMAINT.591 minidisk.

The VM/CMS user working with a TCP/IP client can usually request a specific translation table as an option when invoking the client program.

11.2 File Transfer Protocol (FTP)

Both parties of FTP (server and client) in the VM environment may use specific translation tables. FTP uses two connections to communicate with its partner, a control connection and a data connection. As the name implies, the control connection is used to send commands and replies to the partner; for this connection ASCII code is always used. The data connection is used to actually transfer data, and the code to be used can be set via the control connection.

VM Server: The server chooses its translation table at startup time. It uses the table ″SRVRFTP TCPXLBIN *″ or ″STANDARD TCPXLBIN *″ for the whole time it is running. The ″SRVRFTP″ table can be built according to your requirements and put on TCPMAINT 591.

Note: THE FTP server can only use ONE translation table. This may be a problem if you have ASCII systems which use different hexadecimal codes.

VM Client: The user invoking the FTP client program can request a specific translation table as an option in the FTP command. This table will be in use for the current invocation of the FTP client program only. If no user defined table is requested, FTP uses the table ″FTP TCPXLBIN *″ or ″STANDARD TCPXLBIN *″.

The command is: `ftp Hostname (translate french.`

Note

Full national language support is available with FTP.

11.3 Trivial File Transfer Protocol (TFTP)

IBM TCP/IP Version 2 Release 2 for VM implements a TFTP client only. Similar to FTP, the user can choose a specific translation table as an option in the TFTP command. If no user-defined table is requested, TFTP uses the table ″TFTP TCPXLBIN *″ or ″STANDARD TCPXLBIN *″.

Note

Full national language support is available with TFTP.

11.4 Telnet

Telnet must always use its own translation tables in order to prevent the translation of special ASCII characters to EBCDIC characters outside the range of displayable data for the 3270 data stream.

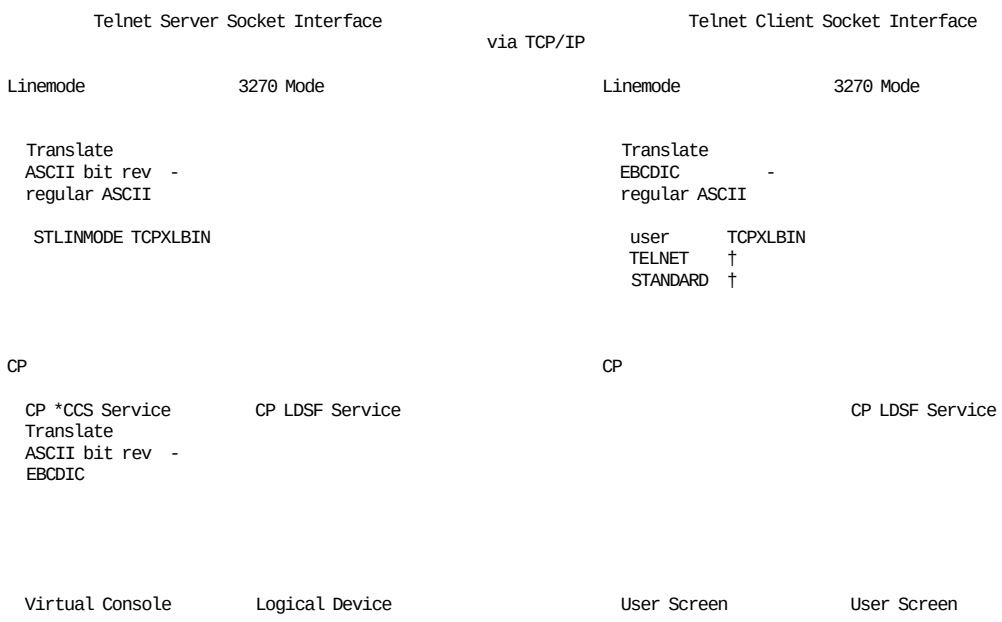


Figure 38. Telnet Data Stream Translation

Both parts of VM Telnet, server and client, recognize two modes: 3270 full-screen mode and line mode.

3270 Mode: Telnet in 3270 mode (usually called Tn3270) will be used between two S/390 hosts, or from a workstation (with Tn3270 available) to a S/390 or an AS/400 host.

Note: All IBM TCP/IP products (DOS, OS/2, AIX, VM, MVS, AS/400) include a Tn3270 client. You can also order from IBM a program called X3270 which is a 3270 Telnet emulator with graphical capabilities. That is, using X3270, you can log on to a VM or MVS host and call GDDM. NLS is available. This X3270 product includes code that can be run on AIX, SUN or HP platforms.

- VM client to VM-MVS-OS/400 and vice versa: No translation at all is done, and the connection is transparent. Extended data stream, including Programmed Symbols (PS) and NLS, is supported.
- Workstation (DOS, OS/2, AIX) to VM server:

  A translation is needed and it will be performed on the client (workstation side). Let's consider the NLS part of this emulation:

  – IBM TCP/IP V2.0 for DOS is the client:

    When Tn3270 is typed, it looks at an external table (TCPDOSETCUS.XLT is the default) which has the same format as the one on VM. This means you can download a customized translation table from VM and use it from your DOS platform (via the command: `Telnet hostname -T Table`).

  – IBM TCP/IP V1.2.1 for OS/2 is the client. No NLS support for now.

  – IBM AIX V3.2 is the client: Let's suppose you have a RISC System/6000 customized in French and that you want to access, using Tn3270, a French VM system. To get full national language support just set a variable to the EBCDIC code page. For example (in France): `SET RM_HOST_LANG=IBM-297` and the issue the `Tn3270` command.

Line Mode:

- VM client:

  Users (Telnet clients) send and receive data to be displayed on their screens in EBCDIC; therefore, EBCDIC/ASCII translation must be done on the client side. The table used for this is either a user defined alternate translation table or ″TELNET TCPXLBIN″ or ″STANDARD TCPXLBIN″. Data sent from the client are regular ASCII characters.

  Even if you use the right translation table you won't have any national language support: the Telnet client only uses 7 bits.

- VM server:

  The Telnet server uses CP *CCS services. The *CCS service sends and receives bit-reversed ASCII to and from the TCP/IP Telnet server. Bit-reversed ASCII is used only in communication between Telnet server and *CCS. EBCDIC code is not involved at all on the Telnet server side. However, there is an EBCDIC/ASCII translation done by CP, which is necessary for VM console input and output.

There is no VTxxx or ANSI emulator available with TCP/IP for VM.

This means that when you log on from a VM to a non-EBCDIC host you will be in line mode with no national language support. An IBM solution to get VTxxx emulators for 3270 terminals is to have a 3174 (hooked to a token-ring) with the RPQ 8Q0935 installed. Please refer to page 218 for more information.

11.5 Simple Mail Transfer Protocol (SMTP)

The VM/CMS user does not directly interact with SMTP. The user commands SENDFILE or NOTE send the data via SPOOL to the SMTP virtual machine which reads the file and processes it accordingly. Receiving mail via SMTP in VM/CMS is similar; SMTP sends the mail via SPOOL to the user virtual machine. For its translation purposes SMTP uses ″SMTP TCPXLBIN″ or ″STANDARD TCPXLBIN″.

Note: SMTP is a mailing protocol and, therefore, designed for text only. Binary data is not allowed.

Warning

You will have full NLS support with SMTP except for the EBCDIC code 7C (which is used by most European languages. As an example X′7C′ is **à** in French) because it is used by SMTP to distinguish the hostname and the domain name (on French systems when you look at the note header the address field is the following: JOEàVM.IBM.COM). If the 7C code is translated you will be able to send mail (but the destination host may not be able to answer) but you won't be able to receive mail because SMTP, when receiving mail, won't be able to find the hostname.

11.6 Network File System (NFS)

The VMNFS server uses ″VMNFS TCPXLBIN″ or ″STANDARD TCPXLBIN″ for its translation purposes. The mount options used by the NFS client determine if translation is required or not (record=nl).

Note

Full national language support is available with NFS.

Appendix A. TCP/IP Network in Use at the ITSC

The figure below shows the TCP/IP network used at the ITSC Raleigh. All TCP/IP hosts in this network can access any other TCP/IP host of the network. You can find the configuration files for each of these hosts in the appendix.

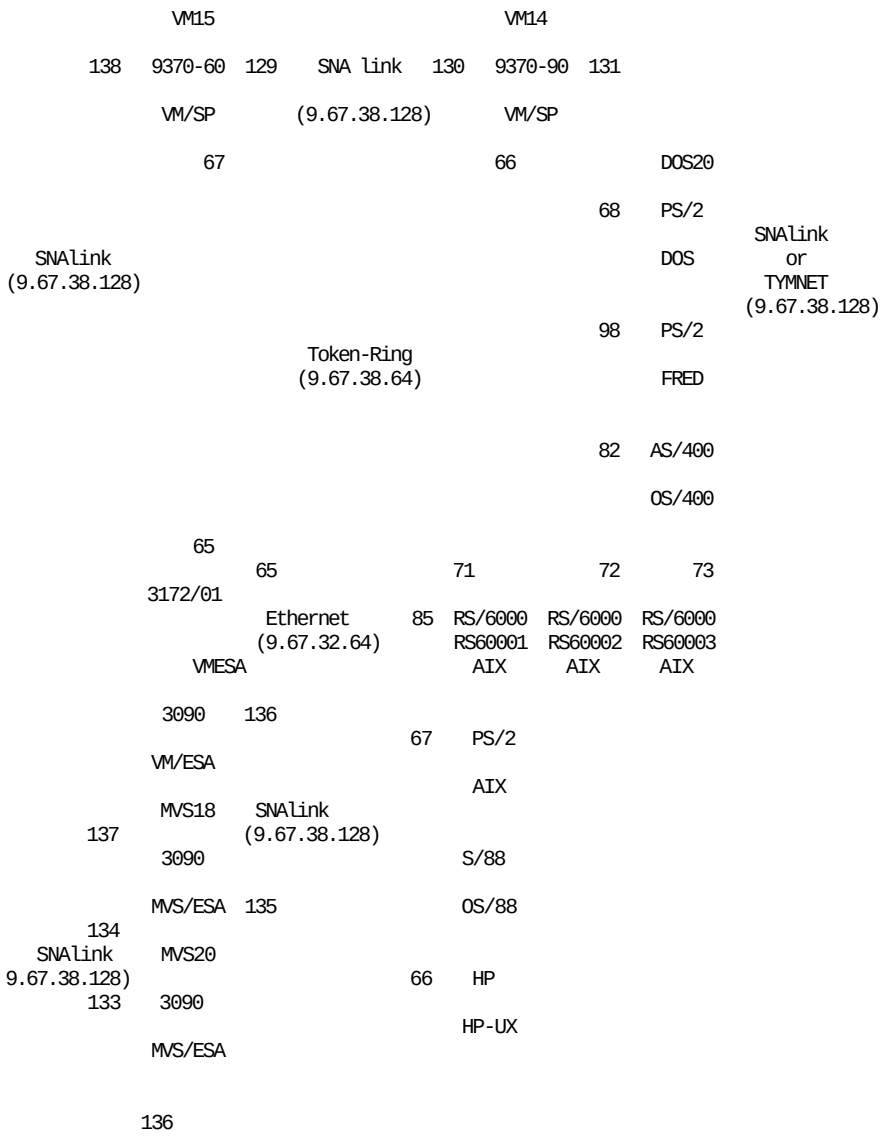


Figure 39. TCP/IP Network at the ITSC Raleigh

A.1 Subnetting

The network IP address assigned by IBM to the ITSC Raleigh is 9.67.38.0. The different subnetworks which are used to build the TCP/IP network are based on a subnet mask of 255.255.255.192, and the assigned subnets are:

```
9.67.38.64      Token-Ring LAN
9.67.38.128     SNA link
9.67.32.64      Ethernet LAN
```

Note: It is not necessary to define a separate subnet for each SNA link connection. All of the point-to-point links are independent of each other, and the IP addresses assigned to the ends of the SNA link connections may be chosen freely, provided they are consistent with the overall addressing structure. This specific assignment has been made here to allow some experiments with the SNA links being defined as *HOST* or as *NETWORK* in the routing tables.

A.2 Token-Ring Network

The token-ring network is the main LAN in the ITSC to which many additional devices are attached.

A.3 Ethernet Network

The Ethernet is an additional LAN with two possible TCP/IP gateways, the system VMESA (9.67.32.65) and the system RS60001 (9.67.32.85).

A.4 SNALINK

The SNALINK network is primarily an alternate route to reach some TCP/IP systems when the token-ring network becomes inoperative.

A.5 VM14

VM14 has three physical interfaces, one token-ring interface and two SNALINK interfaces. The SNALINK connections are primarily alternate routes if the token ring interface fails. The name server resolves all names of the domain ″ITSC.RALEIGH.IBM.COM″. This name server has been defined as a secondary one (VMESA is the primary name server). NetView is also installed and configured for SNMP in order to manage the growing TCP/IP network at the ITSC.

A.6 VM15

VM15 also has three physical interfaces, one token-ring interface and two SNALINK interfaces.

A.7 VMESA

VMESA has one SNALINK connection to MVS18 and is used as a gateway between the Ethernet (9.67.32.64) and token-ring (9.67.38.64) subnets. It also has a channel connection to a RISC System/6000 platform. It supports RIP. The name server resolves all names of the domain ″ITSC.RALEIGH.IBM.COM″.

A.8 RS60001

The RISC System/6000 is the main gateway between Ethernet (9.67.32.64) and token-ring (9.67.38.64) and it supports RIP. All available routes are broadcast via the token-ring and Ethernet links.

A.9 MVS20

MVS20 is connected via an SNALINK to VM14 and MVS18 and supports the RIP protocol.

A.10 MVS18

MVS18 is connected via an SNALINK to VM33 and MVS20 and supports the RIP protocol.

A.11 FRED

The OS/2 workstation FRED (9.67.38.98) has only one physical interface. ROUTED is used on the workstation to set up the routing table automatically. NFS client, SNMP agent, FTP, SMTP and Telnet are all available and configured on this workstation.

A.12 PSAIX

The PSAIX workstation is connected to the Ethernet only (9.67.32.67) and it uses RIP. For a workstation on our Ethernet network, it is not absolutely necessary to use RIP. One static default route to the RS60001 would allow access to the whole ITSC network.

A.13 RALYAS4B

The RALYAS4B system is an OS/400 platform connected to the token-ring (9.67.38.83) and it does not use RIP.

Appendix B. Name Server Installation Console Log

The following console log was produced creating the name server SQL/DS table ITSC on system VM15. It was created in 1990 but the messages that are displayed when the name server is installed have not changed.

```
sqlinit db(startdb)
ARI0717I START SQLINIT EXEC: 12/05/90 14:04:21 EDT
ARI0320I THE DEFAULT DATABASE NAME IS STARTDB.
ARI0796I END SQLINIT EXEC: 12/05/90 14:04:24 EDT
Ready; T=1.23/1.60 14:04:25
isql
ARI0659I LINE-EDIT SYMBOLS RESET:
         LINEND=# LINEDEL=OFF CHARDEL=OFF ESCAPE=OFF TABCHAR=OFF
ARI0662I EMSG FUNCTION VALUE RESET TO: ON.
ARI0320I THE DEFAULT DATABASE NAME IS STARTDB.
ARI7399I THE ISQL DEFAULT PROFILE VALUES ARE IN EFFECT.
ARI7079I ISQL INITIALIZATION COMPLETE.
ARI7080A PLEASE ENTER AN ISQL OR SQL COMMAND.
connect sqldba identified by sqldba
ARI7716I USER SQLDBA CONNECTED TO DATABASE STARTDB.
ARI0500I SQL PROCESSING WAS SUCCESSFUL.
ARI0505I SQLCODE = 0 ROWCOUNT = 0
ARI7945I THE CONNECT COMMAND COMPLETED SUCCESSFULLY.
grant dba to namesrv
ARI0500I SQL PROCESSING WAS SUCCESSFUL.
ARI0505I SQLCODE = 0 ROWCOUNT = 0
ARI7080A PLEASE ENTER AN ISQL OR SQL COMMAND.
exit
ARI7601I ISQL ENDED NORMALLY BY YOUR REQUEST.
ARI0657I EMSG FUNCTION VALUE RESTORED TO: TEXT.
ARI0660I LINE-EDIT SYMBOLS RESTORED:
         LINEND=# LINEDEL=¢ CHARDEL=@ ESCAPE=† TABCHAR=ON
Ready; T=1.04/1.81 14:16:20
Ready; T=0.01/0.01 14:16:20
Ready; T=0.01/0.01 14:16:20
nsprep
    21 *-* ¢global txtlib ibmlib cmslib edcbase edclib¢
             †global txtlib ibmlib cmslib edcbase edclib†
File EDCLIB TXTLIB * not found
       +++ RC(28) +++
    22 *-* ¢global loadlib edclink¢
             †global loadlib edclink†
    24 *-* parm = ¢PREPPARM(PREPNAME=NSTABLE)¢
    25 *-* parm =  PARM ¢SYSIN(NSTABLE CSQL) SYSPRINT(PRINTER)¢
    26 *-* ¢EXEC SQLPREP C¢ parm
             †EXEC SQLPREP C PREPPARM(PREPNAME=NSTABLE) SYSIN(NSTABLE CSQL) SYSP
RINT(PRINTER)†
ARI0717I START SQLPREP EXEC: 12/05/90 14:32:45 EDT
ARI0320I THE DEFAULT DATABASE NAME IS STARTDB.
ARI0663I FILEDEFS IN EFFECT ARE:
SYSIN    DISK     NSTABLE  CSQL     *
SYSPRINT PRINTER
SYSPUNCH DISK     NSTABLE  C        A1
ARISQLLD DISK     ARISQLLD LOADLIB  C1
ARI0713I PREPROCESSOR ARIPRPB CALLED WITH THE FOLLOWING PARAMETERS:
 ....... PREPNAME=NSTABLE
ARI0708I ALL SQLPREP EXEC PROCESSING COMPLETED SUCCESSFULLY.
```

```
ARI0796I END SQLPREP EXEC: 12/05/90 14:33:18 EDT
PRT FILE 1649  FOR NAMESRV  COPY 001     HOLD
    27 *-* if rc  = 0
    29 *-* parm = ¢PREPPARM(PREPNAME=NSACQ)¢
    30 *-* parm =  PARM ¢SYSIN(NSACQ CSQL) SYSPRINT(PRINTER)¢
    31 *-* ¢EXEC SQLPREP C¢ parm
             †EXEC SQLPREP C PREPPARM(PREPNAME=NSACQ) SYSIN(NSACQ CSQL) SYSPRINT
(PRINTER)†
ARI0717I START SQLPREP EXEC: 12/05/90 14:33:20 EDT
ARI0320I THE DEFAULT DATABASE NAME IS STARTDB.
ARI0663I FILEDEFS IN EFFECT ARE:
SYSIN    DISK     NSACQ    CSQL     *
SYSPRINT PRINTER
SYSPUNCH DISK     NSACQ    C        A1
ARISQLLD DISK     ARISQLLD LOADLIB  C1
ARI0713I PREPROCESSOR ARIPRPB CALLED WITH THE FOLLOWING PARAMETERS:
 ....... PREPNAME=NSACQ
ARI0708I ALL SQLPREP EXEC PROCESSING COMPLETED SUCCESSFULLY.
ARI0796I END SQLPREP EXEC: 12/05/90 14:34:18 EDT
PRT FILE 1650  FOR NAMESRV  COPY 001     HOLD
    32 *-* if rc  = 0
    34 *-* parm = ¢PREPPARM(PREPNAME=NSDBLOAD)¢
    35 *-* parm =  PARM ¢SYSIN(NSDBLOAD CSQL) SYSPRINT(PRINTER)¢
    36 *-* ¢EXEC SQLPREP C¢ parm
             †EXEC SQLPREP C PREPPARM(PREPNAME=NSDBLOAD) SYSIN(NSDBLOAD CSQL) SY
SPRINT(PRINTER)†
ARI0717I START SQLPREP EXEC: 12/05/90 14:34:20 EDT
ARI0320I THE DEFAULT DATABASE NAME IS STARTDB.
ARI0663I FILEDEFS IN EFFECT ARE:
SYSIN    DISK     NSDBLOAD CSQL     *
SYSPRINT PRINTER
SYSPUNCH DISK     NSDBLOAD C        A1
ARISQLLD DISK     ARISQLLD LOADLIB  C1
ARI0713I PREPROCESSOR ARIPRPB CALLED WITH THE FOLLOWING PARAMETERS:
 ....... PREPNAME=NSDBLOAD
ARI0708I ALL SQLPREP EXEC PROCESSING COMPLETED SUCCESSFULLY.
ARI0796I END SQLPREP EXEC: 12/05/90 14:34:38 EDT
PRT FILE 1651  FOR NAMESRV  COPY 001     HOLD
    37 *-* if rc  = 0
    39 *-* parm = ¢PREPPARM(PREPNAME=NSTHRESH)¢
    40 *-* parm =  PARM ¢SYSIN(NSTHRESH CSQL) SYSPRINT(PRINTER)¢
    41 *-* ¢EXEC SQLPREP C¢ parm
             †EXEC SQLPREP C PREPPARM(PREPNAME=NSTHRESH) SYSIN(NSTHRESH CSQL) SY
SPRINT(PRINTER)†
ARI0717I START SQLPREP EXEC: 12/05/90 14:35:04 EDT
ARI0320I THE DEFAULT DATABASE NAME IS STARTDB.
ARI0663I FILEDEFS IN EFFECT ARE:
SYSIN    DISK     NSTHRESH CSQL     *
SYSPRINT PRINTER
SYSPUNCH DISK     NSTHRESH C        A1
ARISQLLD DISK     ARISQLLD LOADLIB  C1
ARI0713I PREPROCESSOR ARIPRPB CALLED WITH THE FOLLOWING PARAMETERS:
 ....... PREPNAME=NSTHRESH
ARI0708I ALL SQLPREP EXEC PROCESSING COMPLETED SUCCESSFULLY.
ARI0796I END SQLPREP EXEC: 12/05/90 14:35:11 EDT
PRT FILE 1652  FOR NAMESRV  COPY 001     HOLD
    42 *-* if rc  = 0
    44 *-* parm = ¢PREPPARM(PREPNAME=NSDBSQL)¢
    45 *-* parm =  PARM ¢SYSIN(NSDBSQL ASMSQL) SYSPRINT(PRINTER)¢
```

```
    46 *-* ¢EXEC SQLPREP ASM¢ parm
             †EXEC SQLPREP ASM PREPPARM(PREPNAME=NSDBSQL) SYSIN(NSDBSQL ASMSQL)
SYSPRINT(PRINTER)†
ARI0717I START SQLPREP EXEC: 12/05/90 14:35:43 EDT
ARI0320I THE DEFAULT DATABASE NAME IS STARTDB.
ARI0663I FILEDEFS IN EFFECT ARE:
SYSIN    DISK     NSDBSQL  ASMSQL   *
SYSPRINT PRINTER
SYSPUNCH DISK     NSDBSQL  ASSEMBLE A1
ARISQLLD DISK     ARISQLLD LOADLIB  C1
ARI0713I PREPROCESSOR ARIPRPA CALLED WITH THE FOLLOWING PARAMETERS:
 ....... PREPNAME=NSDBSQL
ARI0708I ALL SQLPREP EXEC PROCESSING COMPLETED SUCCESSFULLY.
ARI0796I END SQLPREP EXEC: 12/05/90 14:35:58 EDT
PRT FILE 1653  FOR NAMESRV  COPY 001     HOLD
    47 *-* if rc  = 0
Ready; T=30.89/42.63 14:36:00
Ready; T=0.01/0.03 14:36:28
nsacq startdb
EXEC SQLINIT DBNAME(startdb)
ARI0717I START SQLINIT EXEC: 12/05/90 14:37:03 EDT
ARI0320I THE DEFAULT DATABASE NAME IS STARTDB.
ARI0796I END SQLINIT EXEC: 12/05/90 14:37:07 EDT
ACQUIRE PUBLIC DBSPACE NAMED NAMESRV.TCPSPACE (PAGES= 5120)
Ready; T=2.39/3.40 14:37:09
Ready; T=0.01/0.01 14:37:09
nstable
Using input file NSMAIN DATA A
Creating table ITSC0
Creating table ITSC1
Ready; T=1.29/1.99 14:49:14
Ready; T=0.01/0.01 14:49:14
nsdbload itsc
Select remarks
NO ROWS FOUND ON: Select remarks.....
Currently used TABLES:
Using sql table itsc1
Wed Dec  5 14:51:08: Do you want to create resource records from a master file
Wed Dec  5 14:51:08: (y/n)?
y
Using input file MASTER.DATA.A
Wed Dec  5 14:51:54: New origin itsc.raleigh.ibm.com

Wed Dec  5 14:51:57: Do you want to delete all resource records from db(y/n)?
n
Wed Dec  5 14:52:04: Do you want to insert the RR in the sqltable now(y/n)?
y
Create index itsc1 on itsc1
Wed Dec  5 14:52:08: Do you want translate all data to lower case?(y/n)
y
Wed Dec  5 14:52:15: The data in NSMAIN HOSTINFO A is now being added to the SQL
Wed Dec  5 14:52:17: Finished adding data to table ....
Wed Dec  5 14:52:17: Update statistics on the itsc1 table

Wed Dec  5 14:52:17: Data base update completed.

Ready; T=4.58/5.92 14:52:18
Ready; T=0.01/0.01 14:52:18
```

```
isql
ARI0659I LINE-EDIT SYMBOLS RESET:
         LINEND=# LINEDEL=OFF CHARDEL=OFF ESCAPE=OFF TABCHAR=OFF
ARI0662I EMSG FUNCTION VALUE RESET TO: ON.
ARI0320I THE DEFAULT DATABASE NAME IS STARTDB.
ARI7399I THE ISQL DEFAULT PROFILE VALUES ARE IN EFFECT.
ARI7079I ISQL INITIALIZATION COMPLETE.
ARI7080A PLEASE ENTER AN ISQL OR SQL COMMAND.
ARI0504I SQLERRP: ARIXA01 SQLERRD1: -170 SQLERRD2: 0
connect sqldba identified by sqldba
ARI7716I USER SQLDBA CONNECTED TO DATABASE STARTDB.
ARI0500I SQL PROCESSING WAS SUCCESSFUL.
ARI0505I SQLCODE = 0 ROWCOUNT = 0
ARI7945I THE CONNECT COMMAND COMPLETED SUCCESSFULLY.
revoke dba from namesrv
ARI0501I AN SQL WARNING HAS OCCURRED.
         CONNECT AUTHORITY IS STILL ACTIVE FOR USER ID(S).
ARI0505I SQLCODE = 150 ROWCOUNT = 0
ARI7080A PLEASE ENTER AN ISQL OR SQL COMMAND.
exit
ARI7601I ISQL ENDED NORMALLY BY YOUR REQUEST.
ARI0657I EMSG FUNCTION VALUE RESTORED TO: TEXT.
ARI0660I LINE-EDIT SYMBOLS RESTORED:
         LINEND=# LINEDEL=¢ CHARDEL=@ ESCAPE=† TABCHAR=ON
Ready; T=1.05/1.72 14:35:54
Ready; T=0.01/0.01 14:35:54
Ready; T=0.01/0.01 14:35:54
nsmain
VM TCP/IP Name Server V2R2
Name        Entry       Origin      Bytes        Attributes
EDCX24      00384000    00384000    000023C0    Amode 24  Non-reloc
EDCXV       002F6000    002F6000    00066EA0    Amode 24  Non-reloc
IBMBLII1    0010A4B0    0010A4B0    00005F40    Amode Any Non-reloc
NSMAIN      00020000    00020000    0009A4B0    Amode 24  Non-reloc
Using input file NSMAIN.DATA.A
Primary data in table ITSC
Negative Caching
sosqcach 100
sonscach 50
soiqcach 5
sosicach 5
LRUtime 300
hostcase UPPER
DomainNamePort 53
qqtimr 5
Using CP message no header command
smsguserfile VALIDUSR
Using NAMESRV as table owner on SQL queries
Using RSCS as RSCS name
SQL table  NAMESRV.ITSC1 has 28 records.
Threshold value is 16
Wed Dec  5 14:53:43: ns: Began TCP/IP service rc = 0
Wed Dec  5 14:53:43: ns: Domain name server initialization complete.
Wed Dec  5 15:03:11: Query from: 9.67.32.28 for rolf.itsc.raleigh.ibm.com
Wed Dec  5 15:03:11: ns: Standard Query Cache:
Wed Dec  5 15:03:11: ns:      Looking for Query
                   : cm:      qname=¢rolf.itsc.raleigh.ibm.com¢, qtype=1, qclass
=1
Wed Dec  5 15:03:11: ns:      Cache Entry NOT Found
```

```
Wed Dec  5 15:03:12: ns: Standard Query Cache:
Wed Dec  5 15:03:12: ns:      Adding an entry to the cache
Wed Dec  5 15:03:12: ns:      Query will remain cached for 999999 seconds
#cp ext
freeing primary and secondary links
freeing cache
Wed Dec  5 15:03:59: ns: Standard Query Cache:
Wed Dec  5 15:03:59: ns:      purging the entire cache
Wed Dec  5 15:03:59: ns: Intermediary Query Cache:
Wed Dec  5 15:03:59: ns:      purging the entire cache
Wed Dec  5 15:03:59: ns: Inverse Query Cache:
Wed Dec  5 15:03:59: ns:      purging the entire cache
Wed Dec  5 15:03:59: ns: Data Base Query Cache:
Wed Dec  5 15:03:59: ns:      purging the entire cache
freeing SOANS list
exiting program
Wed Dec  5 15:03:59: Name server shut down with CP EXT.
Ready; T=3.83/5.85 15:03:59
Ready; T=0.01/0.01 15:03:59
```

You may have the following output when you try to load a new SQL table:

```
nsdbload namesrv.arpa0 master in-addr a
outnfile.fn NSMAIN
infile.fn master
Select remarks
Currently used TABLES: ITSC1
Tue Jul  7 15:18:38: Do you want to create resource records from a master file
Tue Jul  7 15:18:38: (y/n)?
y
Using input file master.in-addr.a
Tue Jul  7 15:18:41: New origin 38.67.9.in-addr.arpa

Tue Jul  7 15:18:41: @ - current origin 38.67.9.in-addr.arpa

Tue Jul  7 15:18:41: Do you want to delete all resource records from db(y/n)?
y
Tue Jul  7 15:18:51: Drop index namesrv.arpa0

drop index .. sqlcode -204   1
Tue Jul  7 15:18:51: Get number of entries in table ...
Tue Jul  7 15:18:51: Do you want to insert the RR in the sqltable now(y/n)?
y
Create index namesrv.arpa0 on namesrv.arpa0
Tue Jul  7 15:18:57: Do you want translate all data to lower case?(y/n)
y
Tue Jul  7 15:19:00: The data in NSMAIN HOSTINFO A is now being added to the SQL

Tue Jul  7 15:19:00: Finished adding data to table ....
Tue Jul  7 15:19:00: Update statistics on the namesrv.arpa0 table

Tue Jul  7 15:19:00: Data base update completed.

Ready;
Ready;
```

1 This is not a problem. It just means that you have been unable to drop an index. This index can't be dropped simply because it does not exist yet since you are creating it using the **NSDBLOAD** command.

The process ends with no error.

Appendix C. Configuration Listings for VM System VM14

VM14 is an IBM 9370-90 running VM/SP Release 6 with IBM TCP/IP Version 2 Release 2 for VM.

C.1 File "PROFILE TCPIP" on TCPMAINT 591

```
ACBPOOLSIZE                  1000
ADDRESSTRANSLATIONPOOLSIZE   1500
CCBPOOLSIZE                  150
DATABUFFERPOOLSIZE           160
ENVELOPEPOOLSIZE             750
IPROUTEPOOLSIZE              300
LARGEENVELOPEPOOLSIZE        50
RCBPOOLSIZE                  50
SCBPOOLSIZE                  256
SKCBPOOLSIZE                 256
SMALLDATABUFFERPOOLSIZE      50
TCBPOOLSIZE                  256
UCBPOOLSIZE                  100

; Turn off all TCP/IP tracing

NOTRACE

; Flush the ARP table every 5 minutes

ARPAGE 5

; The SYSCONTACT and SYSLOCATION statements are used for SNMP.
;
; SYSCONTACT is the contact person for this managed node and how to
; contact this person. Used for VM agent MIB variable sysContact

SYSCONTACT
   Philippe BEAUPIED (Office CC119 - Ext 2365)
ENDSYSCONTACT

; SYSLOCATION is the physical location of this node. Used for VM
; agent MIB variable sysLocation

SYSLOCATION
   9370 MDL 90 - COMPUTER ROOM AA136
ENDSYSLOCATION

; Inform the following users of serious errors

INFORM
    OPERATOR TCPMAINT
ENDINFORM

; Obey the following users for restricted commands

OBEY
```

```
    OPERATOR TCPMAINT SNMPD SNMPQE ROUTED REXECD
ENDOBEY

; Autolog the following server machines

AUTOLOG
;    NAMESRV  TCP          ; DOMAIN NAME SERVER
;    ROUTED   TCP          ; ROUTED SERVER
;    FTPSERVE TCP          ; FTP SERVER
;    SNALNKA  TCP          ; SNALINK
;    LPSERVE  TCP          ; LP SERVER
;    NCSGLBD  TCP          ; NCS GLBD SERVER
;    NCSLLBD  TCP          ; NCS LLBD SERVER
;    PORTMAP  TCP          ; PORTMAP SERVER
;    REXECD   TCP          ; REXEC SERVER
;    SMTP     TCP          ; SMTP SERVER
;    SNMPD    TCP          ; SNMP VM AGENT VIRTUAL MACHINE
;    SNMPQE   TCP          ; SNMP VM CLIENT VIRTUAL MACHINE
;    VMNFS    TCP          ; NFS SERVER
ENDAUTOLOG

; Reserve the following ports for specific servers
; values from RFC 1060, †Assigned numbers†

PORT
    20 TCP FTPSERVE  NOAUTOLOG ; FTP Server
    21 TCP FTPSERVE            ; FTP Server
    23 TCP INTCLIEN            ; TELNET Server
    25 TCP SMTP                ; SMTP Server
    53 TCP NAMESRV             ; Domain Name Server
    53 UDP NAMESRV             ; Domain Name Server
   111 TCP PORTMAP             ; Portmap Server
   111 UDP PORTMAP             ; Portmap Server
   135 UDP NCSLLBD             ; NCS LLBD Server
   161 UDP SNMPD               ; SNMP Agent
   162 UDP SNMPQE              ; SNMPQE Agent
   512 TCP REXECD              ; REXECD Server (REXEC)
   514 TCP REXECD              ; REXECD Server (RSH)
   515 TCP LPSERVE             ; LP Server
   520 UDP ROUTED              ; RouteD Server
   750 TCP VMKERB              ; Kerberos Server
   750 UDP VMKERB              ; Kerberos Server
   751 TCP ADM_SERV            ; Kerberos Database Server
   751 UDP ADM_SERV            ; Kerberos Database Server
  2049 UDP VMNFS               ; NFS Server

; DEVICE and LINK statements

DEVICE ILADTRN ILANS       104
LINK ILALTRN IBMTR    1 ILADTRN

DEVICE SNADVM15 SNAIUCV SNALINK AD115TC1 SNALNKA
LINK   SNALVM15    IUCV  2 SNADVM15

DEVICE SNADMV20 SNAIUCV SNALINK RAKASNAL SNALNKA
LINK   SNALMV20    IUCV  3 SNADMV20

DEVICE X25A     X25ICA      9C0
LINK   X25LA    X25ICA   4  X25A
```

```
; The local host¢s internet addresses for the above interfaces

HOME
    9.67.38.66   ILALTRN
    9.67.38.130  SNALVM15
    9.67.38.131  SNALMV20
;   9.67.32.66   X25LA

; Routing information (if you are not using the ROUTED server)
; V2 allows to define HOST specific routes

  GATEWAY
;  Network     First_hop  Driver    Packet  Subn mask     Subn value
   9           =          ILALTRN   2000    0.255.255.192  0.67.38.64

TRANSLATE
; 9.67.32.65 X25ICA 3106001983 X25LA
; 9.67.32.66 X25ICA 3106001984 X25LA

; Start the interfaces

START ILADTRN
; START SNADVM15
; START SNADMV20
; START X25A
```

C.2 File "TCPIP DATA" on TCPMAINT 592

```
TCPIPUSERID TCPIP
RAL9390:   HOSTNAME  RAL9390
DOMAINORIGIN  ITSC.RALEIGH.IBM.COM
NSINTERADDR  14.0.0.0
NSINTERADDR  9.67.38.65
NSPORTADDR 53
RESOLVEVIA UDP
RESOLVERTIMEOUT 20
RESOLVERUDPRETRIES 2
; TRACE RESOLVER
```

C.3 File "HOSTS LOCAL" on TCPMAINT 592

This is pretty basic because a name server was used (VMESA).

```
HOST : 9.67.38.66 : vm14 ::::
HOST : 9.67.38.67 : vm15.ITSC.RALEIGH.IBM.COM ::::
HOST : 9.67.38.84 : fred.ITSC.RALEIGH.IBM.COM ::::
HOST : 9.67.38.65 : vmesa ::::
```

C.4 File "NSMAIN DATA" on NAMESRV 191

```
; PRIMARY watson.ibm.com WATSON
SECONDARY itsc.raleigh.ibm.com  IBM 9.67.38.65
SECONDARY 38.67.9.in-addr.arpa  ARPA 9.67.38.65
; CACHINGONLY  NSMAIN CACHE A
NEGATIVECACHING
STANDARDQUERYCACHE     100
```

```
INTERMEDIARYQUERYCACHE  50
INVERSEQUERYCACHE        5
DATABASEQUERYCACHE       5
LRUTIME                300
HOSTNAMECASE          UPPER
DOMAINNAMEPORT           53
UDPONLY
UDPRETRYINTERVAL         5
; NORECURSION
MSGNOH
SMSGUSERFILE    VALIDUSR EXEC A
TRACE                QUEUE
```

C.5 File "LPD CONFIG" on LPSERVE 191

```
; DEBUG
;
SERVICE LOCPRT PRINTER
   LOCAL
   FILTERS f l p
   LINESIZE 132
   PAGESIZE 66
;
SERVICE itsc PRINTER
   RSCS DEST=RALYDPD  IDENTIFIER=ITSC
        PRIORITY=44
   FILTERS f l p
   LINESIZE 132
   PAGESIZE 66
;
 SERVICE OS25 PRINTER
   REMOTE LPT1@OS25
;
 SERVICE ROLF PRINTER
   REMOTE LPT1@ROLF
   FILTERS f l p
   LINESIZE 132
   PAGESIZE 66
;
  SERVICE JCLMVS20 PUNCH
   RSCS DEST=RALVSMV6 IDENTIFIER=JOB
   FILTERS f l p
   LINESIZE 80
;
  SERVICE JCLMVS18 PUNCH
   RSCS DEST=RAIANJE  IDENTIFIER=JOB
   FILTERS f l p
   LINESIZE 80
;
```

C.6 File "PW SRC" on SNMPD 191

Same community name both in lower and upper case to be sure NetView operator can query the agent, either from the panels or from the command prompt.

```
itsc 9.67.38.0 255.255.255.0
ITSC 9.67.38.0 255.255.255.0
```

C.7 File "SNMPTRAP DEST" on SNMPD 191

```
VM14 UDP
RS6001 UDP
```

C.8 File "SMTP CONFIG" on SMTP 191

```
PORT 25                 ; port to accept incoming mail on
POSTMASTER  TCPMAINT    ; where mail addressed to postmaster is spooled
LOOPINGMAIL TCPMAINT    ; where looping mail is spooled to
LOG SPOOL               ; log all mail delivered in SMTP LOG file
INACTIVE 180            ; timeout for inactive connections
FINISHOPEN 120          ; timeout for opening up TCP connections
RETRYAGE 3              ; keep retrying mail delivery for 3 days
RETRYINT 20             ; retry mail delivery every 20 minutes
MAXMAILBYTES 524288     ; largest mail to accept over a TCP connection
RESOLVERRETRYINT 20     ; Retry pending name resolutions every 20 minutes
RCPTRESPONSEDELAY 60    ; How long to delay RCPT TO: response when waiting
                        ; for an address resolution.
TEMPERRORRETRIES 0      ; How many times to retry temporary deliver errors
                        ; default, 0, means retry for RETRYAGE days;
                        ; otherwise the mail is returned after this number
                        ; of deliver attempts.
; DEBUG                 ; Normally not used, causes debug information to
                        ; be written to the SMTP DEBUG file.
;
;  Configuration for a typical RSCS to TCP/IP mail gateway.
;
GATEWAY              ; ACCEPT MAIL FROM AND DELIVER MAIL TO RSCS HOSTS
RSCSDOMAIN RAL9390      ; PSEUDO DOMAIN NAME OF ASSOCIATED RSCS NETWORK
LOCALFORMAT PUNCH       ; local recipients receive mail in Punch format
RSCSFORMAT NETDATA      ; RSCS RECIPIENTS RECEIVE MAIL IN PUNCH FORMAT
REWRITE822HEADER YES    ; Only set to no if you do not want SMTP to
                        ; rewrite the 822 headers on all mail passing
                        ; from RSCS to TCP through gateway.
;
; ALTTCPHOSTNAME is used to specify an alternative fully qualified host
; name by which SMTP will know the local host.  Mail sent to users at
;  hostname  are treated as if they were local users.
;
; ALTTCPHOSTNAME  hostname
;
;
; Use MAILER option if you run with Columbia Mailer.  Specify NEW
; if Columbia Mailer 2.03B or newer; OLD if prior to version 2.0.
; FOLDnoSOURCEroute if mailer is at level between OLD and NEW.
; LOCAL, RSCS, and UNKNOWN specify conditions under which mail will
; be forwarded to the MAILER - see Installation and Maintenance Manual
; for further details.
;
; MAILER MUSER@MNODE OLD LOCAL RSCS UNKNOWN
;
;
; Restriction Lists
;
RESTRICT RETURN         ; return mail from restricted users
charming@ourvm.our.edu ; Don¢t accept any mail from Prince Charming
charming@OURVMX         ; via RSCS or TCP network.
```

```
charming@ourvm*        ; This line takes place of previous two lines!
*@castle               ; Don¢t accept mail from anyone at node: castle
ENDRESTRICT
;
;
; Use the SECURE statement if this SMTP machine is to run as an SMTP
; to RSCS Secure Gateway.  Only users in the data file SMTP SECTABLE
; will be allowed to send mail, all other mail will be returned or
; rejected.  Note that the file SECURITY MEMO will be sent to RSCS
; users that are not authorized to use the gateway.
;
; SECURE
;
;
; Use the KANJI statement if this SMTP machine is to perform Kanji
; code conversion on the mail.  Consult the Installation and Maintenance
; manual for the necessary parameters.
;
; KANJI   parameters
;
```

C.9 File "AD114MTC VTAMLST"

```
AD114MTC VBUILD TYPE=APPL
*********************************************************************
*     TCPIP  ACB                                                    *
*********************************************************************
AD114TC1 APPL ACBNAME=AD114TC1,AUTH=(ACQ,NVPACE),AUTHEXIT=NO,EAS=12,   +
               PARSESS=YES,SONSCIP=YES,VPACING=0
AD114TC2 APPL ACBNAME=AD114TC2,AUTH=(ACQ,NVPACE),AUTHEXIT=NO,EAS=12,   +
               PARSESS=YES,SONSCIP=YES,VPACING=0
AD114TCX APPL ACBNAME=X25IPI,PRTCT=X25IPI,AUTH=(ACQ),                  +
               PARSESS=YES,EAS=20
```

Appendix D. Configuration Listings for VM System VM15

VM15 is an IBM 9370-60 running VM/SP Release 6 with IBM TCP/IP Version 2 Release 2 for VM.

D.1 File "PROFILE TCPIP" on TCPMAINT 591

```
ACBPOOLSIZE                  1000
ADDRESSTRANSLATIONPOOLSIZE   1500
CCBPOOLSIZE                  150
DATABUFFERPOOLSIZE           160
ENVELOPEPOOLSIZE             750
IPROUTEPOOLSIZE              300
LARGEENVELOPEPOOLSIZE        50
RCBPOOLSIZE                  50
SCBPOOLSIZE                  256
SKCBPOOLSIZE                 256
SMALLDATABUFFERPOOLSIZE      50
TCBPOOLSIZE                  256
UCBPOOLSIZE                  100

NOTRACE

ARPAGE 5

SYSCONTACT
   Philippe BEAUPIED (Office CC119 - Ext 2365)
ENDSYSCONTACT

SYSLOCATION
   9370 MDL 60 - COMPUTER ROOM AA136
ENDSYSLOCATION

INFORM
    OPERATOR TCPMAINT
ENDINFORM

OBEY
    OPERATOR TCPMAINT SNMPD SNMPQE ROUTED REXECD
ENDOBEY

AUTOLOG
    NAMESRV  TCP
    ROUTED   TCP
    FTPSERVE TCP
    FTPSERV2 TCP
    FTPSERV3 TCP
    SNALNKA  TCP
    LPSERVE  TCP
    PORTMAP  TCP
    REXECD   TCP
    SMTP     TCP
    SNMPD    TCP
    SNMPQE   TCP
    VMNFS    TCP
ENDAUTOLOG
```

```
PORT
    20 TCP FTPSERVE  NOAUTOLOG
    20 TCP FTPSERV2  NOAUTOLOG
    20 TCP FTPSERV3  NOAUTOLOG
    21 TCP FTPSERVE
    21 TCP FTPSERV2
    21 TCP FTPSERV3
    23 TCP INTCLIEN
    25 TCP SMTP
    53 TCP NAMESRV
    53 UDP NAMESRV
   111 TCP PORTMAP
   111 UDP PORTMAP
   135 UDP NCSLLBD
   161 UDP SNMPD
   162 UDP SNMPQE
   512 TCP REXECD
   514 TCP REXECD
   515 TCP LPSERVE
   520 UDP ROUTED
   750 TCP VMKERB
   750 UDP VMKERB
   751 TCP ADM_SERV
   751 UDP ADM_SERV
  2049 UDP VMNFS

DEVICE ILADTRN ILANS       204
LINK ILALTRN IBMTR    1 ILADTRN

DEVICE SNADVM14 SNAIUCV SNALINK AD114TC1 SNALNKA
LINK   SNALVM14    IUCV  2 SNADVM14

DEVICE SNADMV18 SNAIUCV SNALINK RAIATC1  SNALNKA
LINK   SNALMV18    IUCV  3 SNADMV18

HOME
    9.67.38.67   ILALTRN
    9.67.38.129  SNALVM14
    9.67.32.138  SNALMV18

  GATEWAY
;  Network     First_hop  Driver    Packet  Subn mask     Subn value
   9           =          ILALTRN   2000    0.255.255.192  0.67.38.64

START ILADTRN
; START SNADVM14
; START SNADMV18
```

D.2 File ″TCPIP DATA″ on TCPMAINT 592

```
TCPIPUSERID TCPIP
RAL9390:   HOSTNAME  RAL9390
RAL9360:   HOSTNAME  RAL9360
DOMAINORIGIN  ITSC.RALEIGH.IBM.COM
NSINTERADDR  9.67.38.65
NSINTERADDR  9.67.38.66
NSPORTADDR 53
```

```
RESOLVEVIA UDP
RESOLVERTIMEOUT 30
RESOLVERUDPRETRIES 1
; TRACE RESOLVER
```

D.3 File "HOSTS LOCAL" on TCPMAINT 592

Again, this file is basic because name servers were used.

```
HOST : 9.67.38.65          : RALYESA,VMESA,NAMESERVER  ::::
HOST : 9.67.38.66          : VM14,RAL9390,NAMESERVER ::::
HOST : 9.67.38.98          : FRED     ::::
```

D.4 File "LPD CONFIG" on LPSERVE 191

```
; DEBUG
;
SERVICE LOCPRT PRINTER
   LOCAL
   FILTERS f l p
   LINESIZE 132
   PAGESIZE 66
;
SERVICE ITSC PRINTER
   RSCS DEST=RALYDPD IDENTIFIER=ITSC
        PRIORITY=44 OTHERS=R=WTCR11
   FILTERS f l p
   LINESIZE 132
   PAGESIZE 66
;
 SERVICE VM14 PRINTER
   REMOTE LOCPRT@VM14
;
 SERVICE OS25 PRINTER
   REMOTE LPT1@OS25
;
 SERVICE ROLF PRINTER
   REMOTE LPT1@ROLF
   FILTERS f l p
   LINESIZE 132
   PAGESIZE 66
;
  SERVICE JCLMVS20 PUNCH
   RSCS DEST=RALVSMV6 IDENTIFIER=JOB
   FILTERS f l p
   LINESIZE 80
;
  SERVICE JCLMVS18 PUNCH
   RSCS DEST=RAIANJE  IDENTIFIER=JOB
   FILTERS f l p
   LINESIZE 80
;
```

D.5 File "PW SRC" on SNMPD 191

```
RALEIGH  9.67.32.0  255.255.255.0
IBM      9.0.0.0    255.0.0.0
ROLF     9.67.32.28 255.255.255.255
OS25     9.67.32.21 255.255.255.255
```

D.6 File "SNMPTRAP DEST" on SNMPD 191

```
VM14 UDP
Rs60001 UDP
```

D.7 File "SMTP CONFIG" on SMTP 191

```
PORT 25                 ; port to accept incoming mail on
POSTMASTER  TCPMAINT    ; where mail addressed to postmaster is spooled
LOOPINGMAIL TCPMAINT    ; where looping mail is spooled to
LOG SPOOL               ; log all mail delivered in SMTP LOG file
INACTIVE 180            ; timeout for inactive connections
FINISHOPEN 120          ; timeout for opening up TCP connections
RETRYAGE 3              ; keep retrying mail delivery for 3 days
RETRYINT 20             ; retry mail delivery every 20 minutes
MAXMAILBYTES 524288     ; largest mail to accept over a TCP connection
RESOLVERRETRYINT 20     ; Retry pending name resolutions every 20 minutes
RCPTRESPONSEDELAY 60    ; How long to delay RCPT TO: response when waiting
                        ; for an address resolution.
TEMPERRORRETRIES 0      ; How many times to retry temporary deliver errors
; DEBUG                 ; Normally not used, causes debug information to
; GATEWAY               ; ACCEPT MAIL FROM AND DELIVER MAIL TO RSCS HOSTS
; RSCSDOMAIN RAL9360    ; PSEUDO DOMAIN NAME OF ASSOCIATED RSCS NETWORK
LOCALFORMAT PUNCH       ; local recipients receive mail in Punch format
RSCSFORMAT NETDATA      ; RSCS RECIPIENTS RECEIVE MAIL IN PUNCH FORMAT
REWRITE822HEADER YES    ; Only set to no if you do not want SMTP to
; ALTTCPHOSTNAME  hostname
; MAILER MUSER@MNODE OLD LOCAL RSCS UNKNOWN
RESTRICT RETURN         ; return mail from restricted users
charming@ourvm.our.edu ; Don¢t accept any mail from Prince Charming
charming@OURVMX         ; via RSCS or TCP network.
charming@ourvm*         ; This line takes place of previous two lines!
*@castle                ; Don¢t accept mail from anyone at node: castle
ENDRESTRICT
; SECURE
; KANJI   parameters
```

D.8 File "AD115MTC VTAMLST"

```
AD115MTC VBUILD TYPE=APPL
*********************************************************************
*     TCPIP  ACB                                                    *
*********************************************************************
AD115TC1 APPL ACBNAME=AD115TC1,AUTH=(ACQ,NVPACE),AUTHEXIT=NO,EAS=12,    +
               PARSESS=YES,SONSCIP=YES,VPACING=0
```

Appendix E. Configuration Listings for VM System RALYESA (VMESA)

RALYESA is an IBM 3090-200 running VM/ESA with IBM TCP/IP Version 2 Release 2 for VM.

E.1 File "PROFILE TCPIP" on TCPMAINT 591

```
ACBPOOLSIZE                 1000
ADDRESSTRANSLATIONPOOLSIZE  1500
CCBPOOLSIZE                 150
DATABUFFERPOOLSIZE          160
ENVELOPEPOOLSIZE            750
IPROUTEPOOLSIZE             300
LARGEENVELOPEPOOLSIZE       50
RCBPOOLSIZE                 50
SCBPOOLSIZE                 256
SKCBPOOLSIZE                256
SMALLDATABUFFERPOOLSIZE     0
TCBPOOLSIZE                 256
UCBPOOLSIZE                 100

NOTRACE

ARPAGE 5

SYSCONTACT
   Philippe Beaupied (919) 352-2365
ENDSYSCONTACT

SYSLOCATION
   COMPUTER ROOM
ENDSYSLOCATION

INFORM
    OPERATOR TCPMAINT
ENDINFORM

OBEY
    OPERATOR TCPMAINT SNMPD SNMPQE ROUTED REXECD
ENDOBEY

AUTOLOG
    FTPSERVE password
    LPSERVE  password
    NAMESRV  password
    PORTMAP  password
    PORTSRVS password
    NDBSERVE password
    NDBSERV1 password
    REXECD   password
    ROUTED   password
    SMTP     password
    SNMPD    password
    SNMPQE   password
    VMNFS    password
```

```
    SNALNKA  password
ENDAUTOLOG

PORT
    20 TCP FTPSERVE  NOAUTOLOG
    21 TCP FTPSERVE
    23 TCP INTCLIEN
    25 TCP SMTP
    53 TCP NAMESRV
    53 UDP NAMESRV
   111 TCP PORTMAP
   111 UDP PORTMAP
   135 UDP NCSLLBD
   161 UDP SNMPD
   162 UDP SNMPQE
   512 TCP REXECD
   514 TCP REXECD
   515 TCP LPSERVE
   520 UDP ROUTED
   750 TCP VMKERB
   750 UDP VMKERB
   751 TCP ADMSERV
   751 UDP ADMSERV
  2049 UDP VMNFS

DEVICE LCS1  LCS     EC2   NETMAN
LINK TR1     IBMTR 1 LCS1

DEVICE LCS2  LCS     EC4   NETMAN
LINK ETH1    ETHERNET 1 LCS2

DEVICE RS60001 CLAW E60 HOST PSCA NONE 20 20 4096 4096
LINK AIX32   IP 0 RS60001

DEVICE SNADMV18 SNAIUCV SNALINK RAIATC1  SNALNKA
LINK   SNALMV18 IUCV  2 SNADMV18

HOME
    9.67.32.65   ETH1
    9.67.38.65   TR1
    9.67.38.148  AIX32
    9.67.38.136  SNALMV18

GATEWAY

 9.67.38.149    =           AIX32     4096          HOST
DEFAULTNET      =           SNALMV18  DEFAULTSIZE   0

  BSDROUTINGPARMS FALSE
     ETH1     DEFAULTSIZE 0        255.255.255.192   0
     TR1      DEFAULTSIZE 0        255.255.255.192   0
     SNALMV18 DEFAULTSIZE 0        255.255.255.192   9.67.38.136
  ENDBSDROUTINGPARMS

TRANSLATE

START SNADMV18
START LCS1
```

```
START LCS2
START RS60001
```

E.2 File "TCPIP DATA"

```
TCPIPUSERID TCPIP
RALYESA:   HOSTNAME  VMESA
DOMAINORIGIN  ITSC.RALEIGH.IBM.COM
NSINTERADDR  14.0.0.0
NSINTERADDR  9.67.38.66
NSPORTADDR 53
RESOLVEVIA UDP
RESOLVERTIMEOUT 20
RESOLVERUDPRETRIES 2
; TRACE RESOLVER
```

E.3 File "MASTER IBM-COM" on NAMESRV 191

```
$ORIGIN ITSC.RALEIGH.IBM.COM.
;
@             IN SOA RALYESA.ITSC.RALEIGH.IBM.COM. TCPMAINT.RALYESA (
                                 901215
                                 3600
                                 600
                                 3600000
                                 86400 )
;
RALYESA                          IN A      9.67.38.65
VMESA                            IN CNAME  RALYESA
RAL9360                          IN A      9.67.38.67
VM15                             IN CNAME  RAL9360
VM14                             IN A      9.67.38.66
RAL9390                          IN CNAME  VM14
ALFRED                           IN A      9.67.38.92
PHILIPPE                         IN A      9.67.38.93
FRED                             IN A      9.67.38.98
RS60001                          IN A      9.67.38.71
RS60002                          IN A      9.67.38.72
RS60003                          IN A      9.67.38.73
MVS20                            IN A      9.67.38.133
MVS18                            IN A      9.67.38.135
RAIANJE                          IN CNAME  MVS18
;
```

E.4 File "MASTER IN-ADDR" on NAMESRV 191

```
$ORIGIN 38.67.9.IN-ADDR.ARPA.
@ IN SOA RALYESA.ITSC.RALEIGH.IBM.COM. TCPMAINT.RALYSEA.ITSC.RALEIGH.IBM.COM. (
                                 901215
                                 3600
                                 600
                                 3600000
                                 86400 )
;
65                               IN PTR    RALYESA.itsc.raleigh.ibm.com.
67                               IN PTR    VM15.itsc.raleigh.ibm.com.
```

```
66                              IN PTR    VM14.itsc.raleigh.ibm.com.
92                              IN PTR    ALFRED.itsc.raleigh.ibm.com.
93                              IN PTR    PHILIPPE.itsc.raleigh.ibm.com.
98                              IN PTR    FRED.itsc.raleigh.ibm.com.
71                              IN PTR    RS60001.itsc.raleigh.ibm.com.
72                              IN PTR    RS60002.itsc.raleigh.ibm.com.
73                              IN PTR    RS60003.itsc.raleigh.ibm.com.
133                             IN PTR    MVS20.itsc.raleigh.ibm.com.
135                             IN PTR    MVS18.itsc.raleigh.ibm.com.
```

E.5 File "NSMAIN DATA" on NAMESRV 191

```
PRIMARY ITSC.RALEIGH.IBM.COM ITSC
NEGATIVECACHING
STANDARDQUERYCACHE     100
INTERMEDIARYQUERYCACHE  50
INVERSEQUERYCACHE        5
DATABASEQUERYCACHE       5
LRUTIME                300
HOSTNAMECASE          LOWER
DOMAINNAMEPORT           53
UDPRETRYINTERVAL          5
MSGNOH
SMSGUSERFILE     VALIDUSR EXEC A
TRACE                 QUEUE
```

Appendix F. Configuration Listings for MVS System MVS20

MVS20 is an IBM 3090 running MVS/ESA with TCP/IP V2 for MVS.

F.1 Data Set "TCPIP.RALVSMV6.TCPIP"

```
; ---------------------------------------------------------------------
;
;     ITSC-Raleigh:  TCP/IP MVS V2 System  *RALVSMV6*
;     -------------------------------------------
;
;   - MVS/ESA 3.1 - SA20 / MVS20
;
;   - Name of TCPIP Configuration File = TCPIP.RALVSMV6.TCPIP
;
; ---------------------------------------------------------------------
;
ACBPOOLSIZE                 1000
ADDRESSTRANSLATIONPOOLSIZE  1500
CCBPOOLSIZE                 150
DATABUFFERPOOLSIZE          160
ENVELOPEPOOLSIZE            750
IPROUTEPOOLSIZE             300
LARGEENVELOPEPOOLSIZE       50
RCBPOOLSIZE                 50
SCBPOOLSIZE                 256
SKCBPOOLSIZE                256
SMALLDATABUFFERPOOLSIZE     0
TCBPOOLSIZE                 256
UCBPOOLSIZE                 100
;
NOTRACE SCREEN

; Inform the following users of serious errors
INFORM
    WTCR22
ENDINFORM

; Obey the following users for restricted commands
OBEY
    STCTCP1              ;  User-Id of SNMPD SNMPQE ROUTED
    WTCR22               ;
    Leisa                ;
ENDOBEY

; Flush the arp tables every 5 minutes
ARPAGE 5

; The SYSCONTACT and SYSLOCATION statements are used for SNMP.
;
; SYSCONTACT is the contact person for this managed node and how to
; contact this person.  Used for VM agent MIB variable sysContact
SYSCONTACT
   Lesia Cox     (850-2323)
ENDSYSCONTACT
```

```
; SYSLOCATION is the physical location of this node.  Used for VM
; agent MIB variable sysLocation
SYSLOCATION
   COMPUTER ROOM BLDG 657
ENDSYSLOCATION

; Set Telnet timeout to 10 minutes
INTERNALCLIENTPARMS TIMEMARK 600 ENDINTERNALCLIENTPARMS

;
; Hardware definitions:
;

DEVICE RAKSNA1  SNAIUCV SNALINK RAIASNAL SNALK20
LINK   SNALK1   IUCV    1       RAKSNA1

AUTOLOG
;    FTPSE20     ; FTP Server
;    FTPSE20A    ; FTP Server
;    FTPSE20B    ; FTP Server
;    NAMESE20    ; Domain Name Server
;    PORTMP20    ; Portmap server
;    ROUTED20    ; RouteD Server
;    SMTP20      ; SMTP Server
;    SNMPD20     ; SNMP Agent Server
;    SNMPQE20    ; SNMP Client Address space
;    TCPX2520    ; X25
;    NFS20       ; Network File System Server
ENDAUTOLOG

PORT
 ; Values from RFC 1010, †Assigned numbers†
    20 TCP FTPSE20   NOAUTOLOG ; FTP Server
    21 TCP FTPSE20             ; FTP Server
;   20 TCP FTPSE20A  NOAUTOLOG ; FTP Server
;   21 TCP FTPSE20A            ; FTP Server
;   20 TCP FTPSE20B  NOAUTOLOG ; FTP Server
;   21 TCP FTPSE20B            ; FTP Server
    23 TCP INTCLIEN            ; TELNET Server
    25 TCP SMTP20              ; SMTP Server
    53 TCP NAMESE20            ; Domain Name Server
    53 UDP NAMESE20            ; Domain Name Server
   111 TCP PORTMP20            ; Portmap Server
   111 UDP PORTMP20            ; Portmap Server
   161 UDP SNMPD20             ; SNMP Agent
   162 UDP SNMPQE20            ; SNMPQE Agent
   520 UDP ROUTED20            ; RouteD Server
; 2049 UDP NFS20               ; NFS Server

HOME
 ; Local host¢s Internet addresses
   9.67.38.133    SNALK1


GATEWAY
 ; Network  First hop   Driver  Packet size  Subnet mask  Subnet value
9           9.67.38.134  SNALK1    DEFAULTSIZE  0.255.255.192 0.67.38.64
9           9.67.38.134  SNALk1    DEFAULTSIZE  0.255.255.192 0.67.32.64
9.67.38.136   =          SNALK1    DEFAULTSIZE  HOST
```

```
DEFAULTNET  9.67.38.134  SNALK1    DEFAULTSIZE  0
; ; RouteD Routing information (if you are using the ROUTED server)
; ; If you are using RouteD, uncomment all the lines below for
; ; ¢BSDROUTINGPARMS¢, and comment out all the lines for the ¢GATEWAY¢
; ; statement.
;
; ;  link     maxmtu   metric   subnet mask     dest addr
; BSDROUTINGPARMS false
;    X25LA     1024       0     255.255.255.0   0
;    ETH1      1500       0     255.255.255.0   0
;    ETH2      1500       0     255.255.255.0   0
;    PCN1      2000       0     255.255.255.0   0
;    TR1       2000       0     255.255.255.0   0
;    TR2       2000       0     255.255.255.0   0
;    TR3       2000       0     255.255.255.0   0
;    HCH1      1018       0     255.255.255.0   0
;    X25NPL1 DEFAULTSIZE  0     255.255.255.0   0
;    TESTLINK  1500       0     255.255.0.0     129.34.12.6
;    YORKTOWN  1500       0     255.0.0.0       0
; ENDBSDROUTINGPARMS
;

TRANSLATE

 ; Define the VTAM parameters required for the TELNET server
BEGINVTAM
    ; Define logon mode tables to be the defaults shipped with the lates
    ; level of VTAM
  3278-3-E NSX32703 ; 32 line screen - default of NSX32702 is 24 line sc
  3279-3-E NSX32703 ; 32 line screen - default of NSX32702 is 24 line sc
  3278-4-E NSX32704 ; 48 line screen - default of NSX32702 is 24 line sc
  3279-4-E NSX32704 ; 48 line screen - default of NSX32702 is 24 line sc
  3278-5-E NSX32705 ; 132 column screen - default of NSX32702 is 80 colu
  3279-5-E NSX32705 ; 132 column screen - default of NSX32702 is 80 colu
    ; Define the LUs to be used for general users
  DEFAULTLUS
      RAKAE001 RAKAE002 RAKAE003 RAKAE004 RAKAE005
      RAKAE006 RAKAE007 RAKAE008 RAKAE009 RAKAE010
      RAKAE011 RAKAE012 RAKAE013 RAKAE014 RAKAE015
      RAKAE016 RAKAE017 RAKAE018 RAKAE019 RAKAE020
      RAKAE021 RAKAE022 RAKAE023 RAKAE024 RAKAE025
      RAKAE026 RAKAE027 RAKAE028 RAKAE029 RAKAE030

  DEFAULTAPPL NVAS20 ; 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 applicat
              ; TSO is multiple applications all beginning with TSO so u
              ; the * to get them all. If a session is closed, disconnec
              ; the user rather than log off the user.
  RESTRICTAPPL IMS ; Only three users may use IMS
    USER USER1 ; Allow user1 access
      LU TCPIMS01 ; Assign USER1 LU TCPIMS01
    USER USER2 ; Allow user2 access from the default LU pool
    USER USER3 ; Allow user3 access from three telnet sessions, each wit
                ; different reserved LU.
      LU TCPIMS31 LU TCPIMS32 LU TCPIMS33
  ALLOWAPPL * ; Allow all applications that have not been previously
```

```
                ; specified to be accessed
ENDVTAM

; START RAKSNA1              ;  SNALINK
```

F.2 Data Set "TCPIP.V2.TCPIP.DATA"

```
;*********************************************************************
;                                                                    *
;        TCP/IP MVS V2 at ITSC-Raleigh                               *
;        ----------------------------                                *
;                                                                    *
;   - This file is shared between both TCP/IP MVS V2 systems of:     *
;                                                                    *
;     + MVS/ESA 4.1    (MVS18 / SA18)      ** RAIANJE  **           *
;                                                                    *
;     + MVS/ESA 3.1    (MVS20 / SA20)      ** RALVSMV6 **           *
;                                                                    *
;************************************************** 91/06/11 *********
;                                                                    *
;   Name of File:     TCPIP.V2.TCPIP.DATA                            *
;                                                                    *
;   This data set, TCPIP.DATA, is used to specify configuration      *
;   information required by TCP/IP client programs.                  *
;                                                                    *
;   Syntax Rules for the TCPIP.DATA configuration data set:          *
;                                                                    *
;   (a) All characters to the right of and including a ¢;¢ will be   *
;       treated as a comment.                                        *
;                                                                    *
;   (b) Blanks and  end-of-line  are used to delimit tokens.         *
;                                                                    *
;   (c) The format for each configuration statement is:              *
;                                                                    *
;        SystemName||¢:¢   keyword  value                            *
;                                                                    *
;       where  SystemName||¢:¢  is an optional label which may be    *
;       specified before a keyword; if present, then the keyword-    *
;       value pair will only be recognized if the SystemName matches *
;       the node name of the system, as defined in the IEFSSNxx      *
;       PARMLIB member.  This optional label permits configuration   *
;       information for multiple systems to be specified in a single *
;       TCPIP.DATA data set.                                         *
;                                                                    *
;*********************************************************************
;
; TCPIPuserid specifies the Name of the TCPIP address space and *not*
; the name of the USERID of the TCPIP address space!!!!
; TCPIP is used as the default name of the started procedure name.
;
; RAIANJE:  TCPIPuserid IBM          ; For testing S TCPIP18.IBM
  RAIANJE:  TCPIPuserid TCPIP18      ; On MVS/ESA 4.1  (MVS18 / SA18)
;                                      Name of TCPIP started procedure
;
  RALVSMV6: TCPIPuserid TCPIP20      ; On MVS/ESA 3.1  (MVS20 / SA20)
;                                      Name of TCPIP started procedure
;
;
```

```
; HostName specifies the TCP host name of this system.  If not
; specified, the default HostName will be the node name specified
; in the IEFSSNxx PARMLIB member.
;
; For example, if this TCPIP.DATA data set is shared between two
; systems, MVSXA and MVSESA, then the following two lines will define
; the HostName correctly on each system.
;
RAIANJE:  HostName RAIANJE
RALVSMV6: HostName RALVSMV6
;
; DomainOrigin specifies the domain origin that will be appended
; to host names passed to the resolver.  If a host name contains
; any dots, then the DomainOrigin will not be appended to the
; host name.
;
 DomainOrigin ITSC.RALEIGH.IBM.COM
;
; NSinterAddr specifies the internet address of the Name Server.
; Multiple Name Server addresses may be specified.  The Name Servers
; will be tried in the given order.
; If a Name Server will not be used, then do not code an NSinterAddr
; statement.  This will cause all names to be resolved via host table
; lookup.
;
NSinterAddr  9.67.38.65
;
; NSportAddr specifies the Name Server port.
; 53 is the default value.
;
NSportAddr 53
;
; ResolveVia specifies how the Resolver is to communicate
; with the Name Server. TCP indicates use of TCP virtual circuits.
; UDP indicates use of UDP Datagrams.
; The default is UDP.
;
ResolveVia UDP
;
; ResolverTimeout specifies the time in seconds that the Resolver
; will wait while trying to open a TCP connection to the name server,
; or how long it will wait for a response when using UDP.
; Default is 20 seconds.
;
ResolverTimeout 20
;
; ResolverUdpRetries specifies the number of times the resolver should
; retry a query to the Nameserver when using UDP datagrams.
; Default is 1.
;
ResolverUdpRetries 1
;
;Trace Resolver
;
; End of file.
```

F.3 Data Set ″SMTP.RALVSMV6.SMTP.CONFIG″

```
;*********************************************************************
;                                                                    *
;     ITSC-Raleigh:  TCP/IP MVS V2 System  *RALVSMV6*                *
;     ------------------------------------------                     *
;                                                                    *
;   - MVS/ESA 3.1 - MVS20 / SA20                                     *
;                                                                    *
;   - Mail Data Set HLQ prefix name = SMTPM.RALVSMV6                 *
;                                                                    *
;*********************************************************************
;                                                                    *
;   Name of Data Set:     SMTP.RALVSMV6.SMTP.CONFIG                  *
;                                                                    *
;   This data set is pointed to by the CONFIG DD statement in the    *
;   SMTP Cataloged Procedure (SMTPPROC).                             *
;                                                                    *
;   This data set is used to specify runtime options and data        *
;   to the SMTP server address space.                                *
;                                                                    *
;   Syntax Rules for the SMTP configuration data set:                *
;                                                                    *
;   (a) All characters to the right of and including a ¢;¢ will be   *
;       treated as a comment.                                        *
;                                                                    *
;   (b) Blanks and  end-of-line  are used to delimit tokens.         *
;                                                                    *
;******************************************************* 91/07/11 *****
;
;  Defaults which normally aren¢t changed:
;
PORT 25                ; port to accept incoming mail on
POSTMASTER  WTCR22     ; where mail addressed to postmaster is spooled
LOOPINGMAIL WTCR22     ; where looping mail is spooled to
LOG                    ; log all SMTP mail delivered
INACTIVE 180           ; timeout for inactive connections
FINISHOPEN 120         ; timeout for opening up TCP connections
RETRYAGE 3             ; keep retrying mail delivery for 3 days
RETRYINT 20            ; retry mail delivery every 20 minutes
MAXMAILBYTES 524288    ; largest mail to accept over a TCP connection
RESOLVERRETRYINT 20    ; Retry pending name resolutions every 20 minutes
RCPTRESPONSEDELAY 60   ; How long to delay RCPT TO: response when waitin
                       ; for an address resolution.
TEMPERRORRETRIES 0     ; How many times to retry temporary deliver error
                       ; default, 0, means retry for RETRYAGE days;
                       ; otherwise the mail is returned after this numbe
                       ; of deliver attempts.
                       ;
                       ; Data set prefix name for queued mail files:
MAILFILEDSPREFIX SMTPM.RALVSMV6    ;  ==== MVS20 / SA20
                       ;
; MAILFILEDPREFIX SMTP   ; Data set prefix error in INSTALL(SMTPCONF)
MAILFILEUNIT SYSDA     ; MVS unit name for new file allocations
;MAILFILEVOLUME volume ; MVS volume name for new file allocations
SPOOLPOLLINTERVAL 30   ; Amount of time in seconds between spool polling
TIMEZONE EST           ; Specifies the printable three letter name of
                       ; the local time zone.  Remember to change this
```

```
                        ; for daylight savings time.
; DEBUG                 ; Normally not used, causes SMTP to record all
                        ; SMTP commands and reply transactions.
;
;  Configuration for a typical NJE to TCP/IP mail gateway.
;
  GATEWAY                 ; accept mail from and deliver mail to NJE host
NJEDOMAIN ITSCNET       ; pseudo domain name of associated NJE network
LOCALCLASS B            ; local spool class for local mail delivery
LOCALFORMAT NETDATA     ; local recipients receive mail in Punch format
NJEFORMAT NETDATA       ; NJE recipients receive mail in Punch format
NJECLASS B              ; spool class for mail delivery by SMTP to NJE
REWRITE822HEADER YES    ; Only set to no if you do not want SMTP to
                        ; rewrite the 822 headers on all mail passing
                        ; from NJE to TCP through gateway.
;
; ALTTCPHOSTNAME is used to specify an alternative fully qualified host
; name by which SMTP will know the local host.  Mail sent to users at
;  hostname  are treated as if they were local users.  The ALTTCPHOSTNAM
; statement can be repeated up to 16 times.
;
; ALTTCPHOSTNAME  hostname
;
;
; Use MAILER option if you run with Columbia Mailer.  Specify NEW
; if Columbia Mailer 2.03B or newer; OLD if prior to version 2.0.
; FOLDnoSOURCEroute if mailer is at level between OLD and NEW.
; LOCAL, NJE, and UNKNOWN specify conditions under which mail will
; be forwarded to the MAILER - see Installation and Maintenance Manual
; for further details.
;
; MAILER MUSER@MNODE OLD LOCAL NJE UNKNOWN
;
;
; Restriction Lists
;
RESTRICT RETURN         ; return mail from restricted users
; charming@ourmvs.our.edu; Don¢t accept any mail from Prince Charming
charming@ourmvs.our.edu ; Don¢t accept any mail from Prince Charming
charming@OURMVSX        ; via NJE or TCP networks.
charming@ourvm*         ; This line takes place of previous two lines!
*@castle                ; Don¢t accept mail from anyone at node: castle
ENDRESTRICT             ;
;
;
; Use the SECURE statement if this SMTP machine is to run as an SMTP
; to NJE Secure Gateway.  Only users in the data set pointed to by the
; SECTABLE DD statement will be allowed to send mail, all other mail
; will be returned or rejected.  Note that the data set pointed to by
; the SECMEMO DD statement will be sent to NJE users that are not
; authorized to use the gateway.
;
; SECURE
;
;
; Use the KANJI statement if this SMTP machine is to perform Kanji
; code conversion on the mail.  Consult the Installation and Maintenance
; manual for the necessary parameters.
;
```

```
; KANJI   parameters
;
```

Appendix G. Configuration Listings for MVS System MVS18

MVS18 is an IBM 3090 running MVS/ESA with TCP/IP V2 for MVS.

G.1 Data Set "TCPIP.V2.RAIANJE.TCPIP"

```
; ------------------------------------------------------------------*
;                                                                   *
;     ITSC-Raleigh:  TCP/IP MVS V2 System  *RAIANJE*                *
;     ---------------------------------------------                 *
;                                                                   *
;   - MVS/ESA 4.1 - MVS18 / SA18                                    *
;                                                                   *
;   - Name of TCPIP Configuration File = TCPIP.V2.RAIANJE.TCPIP     *
;                                                                   *
;   - Using the *RIP*  Routing Version of the Configuration File    *
;                                                                   *
;   - IBM 3172 Configuration File loaded in 3172  =  TCP3172        *
;                                                                   *
;   - Oct 28 added SNALVM33                                         *
;                                                                   *
; ------------------------------------------------------ 91/07/09 ----*
;
ACBPOOLSIZE  1000
ADDRESSTRANSLATIONPOOLSIZE  1500
CCBPOOLSIZE  150
DATABUFFERPOOLSIZE  160
ENVELOPEPOOLSIZE  750
IPROUTEPOOLSIZE  300
LARGEENVELOPEPOOLSIZE  50
RCBPOOLSIZE  50
SCBPOOLSIZE  256
SKCBPOOLSIZE  256
SMALLDATABUFFERPOOLSIZE  0
TCBPOOLSIZE  256
UCBPOOLSIZE  100
;
; NOTRACE                          ; SCREEN
; TRACE TELNET MORETRACE TELNET NOSCREEN
; MORETRACE ALL NOSCREEN
; MORETRACE PCCA NOSCREEN
INFORM
  PHILB                       ; Phillipe  Beaupied
  LESIA
ENDINFORM
;
OBEY
  STCTCP1                     ; Real RACF User Id¢s  of SNMPQE SNMPD
                              ; RouteD
;
  SNMPQE                      ; Dummy RACF ID for SNMPQE
; SNMPQE18                    ; Dummy RACF ID for SNMPQE
  SNMPD18 SNMPD20             ; Dummy RACF ID for SNMPD
  ROUTED18 ROUTED20           ; Dummy Racf ID for ROUTED
  PHILB                       ; Phillipe  Beaupied
```

```
  LESIA
ENDOBEY
;
SYSCONTACT
  RUY S. MEDEIROS
ENDSYSCONTACT
;
SYSLOCATION
  RALEIGH
ENDSYSLOCATION
;
; Set Telnet timeout to 10 minutes
INTERNALCLIENTPARMS
  TIMEMARK 600
ENDINTERNALCLIENTPARMS
;
; Hardware definitions:
;
  DEVICE SNADVM15 SNAIUCV SNALINK  AD115TC1 SNALK18
  LINK   SNALVM15 IUCV       40    SNADVM15
;
  DEVICE SNADMV20 SNAIUCV SNALINK  RAKASNAL SNALK18
  LINK   SNALMV20 IUCV       60    SNADMV20
;
  DEVICE SNADVM33 SNAIUCV SNALINK  RAXASNAL SNALK18
  LINK   SNALVM33 IUCV       70    SNADVM33
;
;
AUTOLOG
  SNALK18                       ; SNAlink
  FTPSE18                       ; FTP server - #1
  ROUTED18                      ; ROUTED server
; FTPSE18A                      ; FTP server - #2
  SNMPQE18                      ; SNMP Query Engine
  SNMPD18                       ; SNMP Agent
  PORTMP18                      ; Port Mapper
  SMTP18                        ; SMTP server
; NAMESE18                      ; Name Server
  NFS18                         ; NFS Server
ENDAUTOLOG
;
PORT
; Values from RFC 1010, †Assigned numbers†
   21 TCP FTPSE18               ; FTP server  -  #1
   20 TCP FTPSE18  NOAUTOLOG    ; FTP default data port
;  21 TCP FTPSE18A              ; FTP server  -  #2
;  20 TCP FTPSE18A NOAUTOLOG    ; FTP default data port
;  21 TCP FTPSE18B              ; FTP server  -  #3
;  20 TCP FTPSE18B NOAUTOLOG    ; FTP default data port
   23 TCP INTCLIEN              ; Telnet server
   25 TCP SMTP18                ; SMTP server
   53 TCP NAMESE18              ; Name Server
   53 UDP NAMESE18              ; Name Server
   111 UDP PORTMP18             ; Portmapper Server
   111 TCP PORTMP18             ; Portmapper Server
   161 UDP SNMPD18              ; Agent Port for SNMP Messages
   162 UDP SNMPQE               ; SNMP Client port for Trap¢s, *temp*
;  162 UDP SNMPQE18             ; SNMP Client port for Trap¢s
   520 UDP ROUTED18             ; RouteD Server
```

```
  2049 UDP NFS18                ; NFS Server
;
HOME
; Local host¢s Internet addresses
   9.67.38.137    SNALVM15
   9.67.38.134    SNALMV20
   9.67.38.135    SNALVM33
;
GATEWAY
9           9.67.38.136  SNALVM33  DEFAULTSIZE  0.255.255.192 0.67.38.64
9           9.67.38.136  SNALVM33  DEFAULTSIZE  0.255.255.192 0.67.32.64
9.67.38.136   =          SNALVM33  DEFAULTSIZE  HOST
DEFAULTNET  9.67.38.136  SNALVM33  DEFAULTSIZE  0
;
; ; RouteD Routing information (if you are using the ROUTED server)
; ; If you are using RouteD, uncomment all the lines below for
; ; ¢BSDROUTINGPARMS¢, and comment out all the lines for the ¢GATEWAY¢
; ; statement.
;
; ;  link     maxmtu   metric   subnet mask     dest addr
; BSDROUTINGPARMS false
;    SNALMV20  2000       0     255.255.255.240 9.67.32.97
;    SNALVM15  2000       0     255.255.255.240 9.67.32.113
;    SNALVM33  2000       0     255.255.255.240 9.67.32.129
; ENDBSDROUTINGPARMS
;
;
; TRANSLATE
; Define the VTAM parameters required for the TELNET server
BEGINVTAM
    ; Define logon mode tables to be the defaults shipped with the lates
    ; level of VTAM
  LINEMODE INTERACT ; LINEMODETERMINAL
  3277     D4B32702 ; 32 line screen - default of NSX32702 is 24 line sc
  3278-2-E NSX32702 ; 48 line screen - default of NSX32702 is 24 line sc
  3278-2   D4B32782 ; 48 line screen - default of NSX32702 is 24 line sc
  3278-3-E NSX32702 ; 48 line screen - default of NSX32702 is 24 line sc
  3278-3   D4B32783 ; 48 line screen - default of NSX32702 is 24 line sc
  3278-4-E NSX32702 ; 48 line screen - default of NSX32702 is 24 line sc
  3278-4   D4B32784 ; 48 line screen - default of NSX32702 is 24 line sc
  3278-5-E NSX32702 ; 48 line screen - default of NSX32702 is 24 line sc
  3278-5   D4B32784 ; 48 line screen - default of NSX32702 is 24 line sc
  3279-2-E NSX32702 ; 48 line screen - default of NSX32702 is 24 line sc
  3279-2   D4B32782 ; 48 line screen - default of NSX32702 is 24 line sc
  3279-3-E NSX32702 ; 32 line screen - default of NSX32702 is 24 line sc
  3279-3   D4B32783 ; 48 line screen - default of NSX32702 is 24 line sc
  3279-4-E NSX32702 ; 32 line screen - default of NSX32702 is 24 line sc
  3279-4   D4B32784 ; 48 line screen - default of NSX32702 is 24 line sc
  3279-5-E NSX32702 ; 32 line screen - default of NSX32702 is 24 line sc
  3279-5   D4B32785 ; 48 line screen - default of NSX32702 is 24 line sc
; 3278-4-E NSX32704 ; 48 line screen - default of NSX32702 is 24 line sc
; 3279-4-E NSX32704 ; 48 line screen - default of NSX32702 is 24 line sc
; 3278-5-E NSX32705 ; 132 column screen - default of NSX32702 is 80 colu
; 3279-5-E NSX32705 ; 132 column screen - default of NSX32702 is 80 colu
; Define the LUs to be used for general users
  DEFAULTLUS
      RAIAE001 RAIAE002 RAIAE003 RAIAE004 RAIAE005
      RAIAE006 RAIAE007 RAIAE008 RAIAE009 RAIAE010
      RAIAE011 RAIAE012 RAIAE013 RAIAE014 RAIAE015
```

```
      RAIAE016 RAIAE017 RAIAE018 RAIAE019 RAIAE020
      RAIAE021 RAIAE022 RAIAE023 RAIAE024 RAIAE025
      RAIAE026 RAIAE027 RAIAE028 RAIAE029 RAIAE030
  ENDDEFAULTLUS
;
; DEFAULTAPPL RAKAA     ; Set the default application for all
                        ; TELNET sessions, NetView Access
                        ;
  ALLOWAPPL RAKAT* DISCONNECTABLE ; Allow all users access to TSO
              ; TSO is multiple applications all beginning with TSO so u
              ; the * to get them all. If a session is closed, disconnec
              ; the user rather than log off the user.
              ;
; RESTRICTAPPL RAKAN    ; Only three users may use IMS
; USER SHOGREN ; Allow user1 access
;   LU RAIAE030 ; Assign USER1 LU TCPIMS01
; RESTRICTAPPL IMS ; Only three users may use IMS
;   USER USER1 ; Allow user1 access
;     LU TCPIMS01 ; Assign USER1 LU TCPIMS01
;   USER USER2 ; Allow user2 access from the default LU pool
;   USER USER3 ; Allow user3 access from three telnet sessions, each wit
;              ; different reserved LU.
;     LU TCPIMS31 LU TCPIMS32 LU TCPIMS33
  ALLOWAPPL * ; Allow all applications that have not been previously
              ; specified to be accessed
ENDVTAM
;
; START LCS1                ;  3172
; START SNADVM15            ;  SNALINK to  VM15 / SA15
; START SNADMV20            ;  SNALINK to MVS20 / SA20
  START SNADVM33            ;  SNALINK to RALYESA / SA33
;
```

G.2 Data Set ″TCPIP.V2.TCPIP.DATA″

```
;*********************************************************************
;                                                                    *
;        TCP/IP MVS V2 at ITSC-Raleigh                               *
;        ----------------------------                                *
;                                                                    *
;   - This file is shared between both TCP/IP MVS V2 systems of:     *
;                                                                    *
;     + MVS/ESA 4.1    (MVS18 / SA18)      ** RAIANJE  **           *
;                                                                    *
;     + MVS/ESA 3.1    (MVS20 / SA20)      ** RALVSMV6 **           *
;                                                                    *
;   Name of File:     TCPIP.V2.TCPIP.DATA                            *
;                                                                    *
;*********************************************************************
```

See Appendix F.2, “Data Set ″TCPIP.V2.TCPIP.DATA″” on page 294 for the content of this file.

G.3 Data Set ″SMTP.RAIANJE.SMTP.CONFIG″

```
;*********************************************************************
;                                                                     *
;     ITSC-Raleigh:  TCP/IP MVS V2 System  *RAIANJE*                  *
;     ------------------------------------------                      *
;                                                                     *
;   - MVS/ESA 4.1 - MVS18 / SA18                                      *
;                                                                     *
;   - Mail Data Set HLQ prefix name = SMTPM.RAIANJE                   *
;                                                                     *
;*********************************************************************
;                                                                     *
;   Name of Data Set:     SMTP.RAIANJE.SMTP.CONFIG                    *
;                                                                     *
;   This data set is pointed to by the CONFIG DD statement in the     *
;   SMTP Cataloged Procedure (SMTPPROC).                              *
;                                                                     *
;   This data set is used to specify runtime options and data         *
;   to the SMTP server address space.                                 *
;                                                                     *
;   Syntax Rules for the SMTP configuration data set:                 *
;                                                                     *
;   (a) All characters to the right of and including a ¢;¢ will be    *
;       treated as a comment.                                         *
;                                                                     *
;   (b) Blanks and  end-of-line  are used to delimit tokens.          *
;                                                                     *
;***************************************************** 91/07/11 *****
;
;  Defaults which normally aren¢t changed:
;
PORT 25                 ; port to accept incoming mail on
POSTMASTER  WTCR22      ; where mail addressed to postmaster is spooled
LOOPINGMAIL WTCR22      ; where looping mail is spooled to
LOG                     ; log all SMTP mail delivered
INACTIVE 180            ; timeout for inactive connections
FINISHOPEN 120          ; timeout for opening up TCP connections
RETRYAGE 3              ; keep retrying mail delivery for 3 days
RETRYINT 20             ; retry mail delivery every 20 minutes
MAXMAILBYTES 524288     ; largest mail to accept over a TCP connection
RESOLVERRETRYINT 20     ; Retry pending name resolutions every 20 minutes
RCPTRESPONSEDELAY 60    ; How long to delay RCPT TO: response when waitin
                        ; for an address resolution.
TEMPERRORRETRIES 0      ; How many times to retry temporary deliver error
                        ; default, 0, means retry for RETRYAGE days;
                        ; otherwise the mail is returned after this numbe
                        ; of deliver attempts.
                        ;
                        ; Data set prefix name for queued mail files:
MAILFILEDSPREFIX SMTPM.RAIANJE     ;  ==== MVS18 / SA18
                        ;
; MAILFILEDPREFIX SMTP   ; Data set prefix error in INSTALL(SMTPCONF)
MAILFILEUNIT SYSDA      ; MVS unit name for new file allocations
;MAILFILEVOLUME volume ; MVS volume name for new file allocations
SPOOLPOLLINTERVAL 30    ; Amount of time in seconds between spool polling
TIMEZONE EST            ; Specifies the printable three letter name of
                        ; the local time zone.  Remember to change this
```

```
                        ; for daylight savings time.
; DEBUG                 ; Normally not used, causes SMTP to record all
                        ; SMTP commands and reply transactions.
;
;  Configuration for a typical NJE to TCP/IP mail gateway.
;
; GATEWAY                 ; accept mail from and deliver mail to NJE host
NJEDOMAIN ITSCNET       ; pseudo domain name of associated NJE network
LOCALCLASS B            ; local spool class for local mail delivery
LOCALFORMAT NETDATA     ; local recipients receive mail in Punch format
NJEFORMAT NETDATA       ; NJE recipients receive mail in Punch format
NJECLASS B              ; spool class for mail delivery by SMTP to NJE
REWRITE822HEADER YES    ; Only set to no if you do not want SMTP to
                        ; rewrite the 822 headers on all mail passing
                        ; from NJE to TCP through gateway.
;
; ALTTCPHOSTNAME is used to specify an alternative fully qualified host
; name by which SMTP will know the local host.  Mail sent to users at
;  hostname  are treated as if they were local users.  The ALTTCPHOSTNAM
; statement can be repeated up to 16 times.
;
; ALTTCPHOSTNAME  hostname
;
;
; Use MAILER option if you run with Columbia Mailer.  Specify NEW
; if Columbia Mailer 2.03B or newer; OLD if prior to version 2.0.
; FOLDnoSOURCEroute if mailer is at level between OLD and NEW.
; LOCAL, NJE, and UNKNOWN specify conditions under which mail will
; be forwarded to the MAILER - see Installation and Maintenance Manual
; for further details.
;
; MAILER MUSER@MNODE OLD LOCAL NJE UNKNOWN
;
;
; Restriction Lists
;
RESTRICT RETURN         ; return mail from restricted users
; charming@ourmvs.our.edu; Don¢t accept any mail from Prince Charming
charming@ourmvs.our.edu ; Don¢t accept any mail from Prince Charming
charming@OURMVSX        ; via NJE or TCP networks.
charming@ourvm*         ; This line takes place of previous two lines!
*@castle                ; Don¢t accept mail from anyone at node: castle
ENDRESTRICT             ;
;
;
; Use the SECURE statement if this SMTP machine is to run as an SMTP
; to NJE Secure Gateway.  Only users in the data set pointed to by the
; SECTABLE DD statement will be allowed to send mail, all other mail
; will be returned or rejected.  Note that the data set pointed to by
; the SECMEMO DD statement will be sent to NJE users that are not
; authorized to use the gateway.
;
; SECURE
;
;
; Use the KANJI statement if this SMTP machine is to perform Kanji
; code conversion on the mail.  Consult the Installation and Maintenance
; manual for the necessary parameters.
;
```

```
; KANJI   parameters
;
```

Appendix H. Configuration Listings for OS/400 System RALYAS4B

RALYAS4B is an IBM AS/400 E20 running OS/400 V2R2 with TCP/IP. Please refer to *AS/400 TCP/IP Guide - SC41-9875* and to *IBM AS/400 TCP/IP Configuration and Operation - GG24-3442* for more information about TCP/IP on an OS/400 platform.

From the main panel, the command CFGTCP allows you to configure TCP/IP via different subpanels. The main panel looks like the following (the bottom part of the panel, that is, the function keys, has been omitted):

```
                              Configure TCP/IP
                                                        System: RALYAS4B
Select one of the following:

     1. Work with TCP/IP host table entries
     2. Work with TCP/IP links
     3. Work with TCP/IP route entries
     4. Change local domain name
     5. Work with names for SMTP
     6. Work with TCP/IP remote system information

    10. Change remote name server
    11. Change TCP/IP attributes
    12. Work with TCP/IP port entries
    13. Change SMTP distribution retries
    14. Change TCP/IP tuning values

    25. Convert host table

Selection or command
```

All the following panels have been obtained via the different options of the main panel.

H.1 Work with TCP/IP Host Table Entries

```
                    Work with TCP/IP Host Table Entries
                                                        System: RALYAS4B
Type options, press Enter.
  1=Add   2=Change   4=Remove   5=Display

        Internet            Host
Opt     Address             Name
 -      ---------------
 -      9.67.38.135         MVS18
 -      9.67.38.65          VMESA
 -      9.67.38.82          RALYAS4A
 -      9.67.38.83          RALYAS4B
```

H.2 Work with TCP/IP Links

```
                          Work with TCP/IP Links
                                                          System: RALYAS4B
Type options, press Enter.
  1=Add   2=Change   4=Remove   5=Display   9=Start   10=End

           Line          Internet            Link
Opt     Description      Address             Type
 -      ----------
 5      L31TR            9.67.38.83          *TRLAN
```

```
                          Display TCP/IP Link
                                                          System: RALYAS4B
Line description . . . . . :   L31TR
Internet address . . . . . :   9.67.38.83
Link type  . . . . . . . . :   *TRLAN
Auto start link  . . . . . :   *YES
TRLAN bit sequencing . . . . . :   *MSB
```

H.3 Work with TCP/IP Route Entries

```
                      Work with TCP/IP Route Entries
                                                          System: RALYAS4B
Type options, press Enter.
  1=Add   2=Change   4=Remove   5=Display

                         Line                          Maximum
Opt  Network          Description  First Hop        Datagram Size
 -   --------------
 -   *DEFAULT         L31TR        9.67.38.65          *CALC
 -   9                L31TR        *HOME               *CALC
```

The following screen is the display of the "*DEFAULT" entry:

```
                        Display TCP/IP Route Entry
                                                          System: RALYAS4B
Network  . . . . . . . . . :   *DEFAULT
Line description . . . . . :   L31TR
First hop  . . . . . . . . :   9.67.38.65
Maximum datagram size  . . . :   *CALC
Subnet mask  . . . . . . . :   *NONE
Subnet value . . . . . . . :   *NONE
```

The following screen is the display of the "9" entry:

```
                        Display TCP/IP Route Entry
                                                          System: RALYAS4B
Network  . . . . . . . . . :   9
Line description . . . . . :   L31TR
First hop  . . . . . . . . :   *HOME
Maximum datagram size  . . . :   *CALC
Subnet mask  . . . . . . . :   0.255.255.192
Subnet value . . . . . . . :   0.67.38.64
```

H.4 Change the Local Domain Name

```
                        Change Local Domain Name
                                                        System: RALYAS4B
Type choices, press Enter.

  Local domain name . . . .   ITSC.RALEIGH.IBM.COM

  Local host name . . . . .   RALYAS4B
```

H.5 Work with Names for SMTP

Please refer to *IBM AS/400 TCP/IP Configuration and Operation - GG24-3442* for more information about SMTP.

H.6 Work with TCP/IP Remote System Information

Please refer to *AS/400 TCP/IP Guide - SC41-9875* for more information about how to configure an X.25 link.

H.7 Change the Remote Name Server

```
                        Change Remote Name Server
                                                        System: RALYAS4B
Type choices, press Enter.

  Server address . . . . . .   9.67.38.65         Internet address
  Server port  . . . . . . .   53                 0-65534
  Server protocol  . . . . .   *UDP               *UDP, *TCP
  Retries  . . . . . . . . .   3                  1-99
  Time interval  . . . . . .   30                 1-99 (seconds)
```

H.8 Change the TCP/IP Attributes

```
                    Change TCP/IP Attributes (CHGTCPA)

Type choices, press Enter.

Checksum on incoming messages  .   *NO           *SAME, *YES, *NO
IP datagram forwarding . . . . .   *NO           *SAME, *YES, *NO
TELNET inactivity timeout  . . .   0            0-2147483647, *SAME
TELNET timemark timeout  . . . .   600          0-2147483647, *SAME
TELNET default NVT type  . . . .   *VT100        *SAME, *VT100, *NVT
SMTP - outgoing mapping table  .   *DFT         Name, *SAME, *DFT
                                                Name, *LIBL, *CURLIB
SMTP - incoming mapping table  .   *DFT         Name, *SAME, *DFT
                                                Name, *LIBL, *CURLIB
FTP - outgoing mapping table . .   *DFT         Name, *SAME, *DFT
                                                Name, *LIBL, *CURLIB
FTP - incoming mapping table . .   *DFT         Name, *SAME, *DFT
                                                Name, *LIBL, *CURLIB
VT100 - outgoing mapping table     *DFT         Name, *SAME, *DFT
                                                Name, *LIBL, *CURLIB

                                                                More...
```

```
                    Change TCP/IP Attributes (CHGTCPA)

Type choices, press Enter.

VT100 - incoming mapping table     *DFT         Name, *SAME, *DFT
                                                Name, *LIBL, *CURLIB

                                                                Bottom
```

H.9 Work with TCP/IP Port Entries

```
                        Work with TCP/IP Port Entries
                                                      System: RALYAS4B
Type options, press Enter.
  1=Add   2=Change   4=Remove

                   Application     User
Opt      Port        Type         Profile


  (No port entries)
```

H.10 Change SMTP Distribution Retries

```
                      Change SMTP Distribution Retries
                                                        System: RALYAS4B
Type choices, press Enter.

  First level:
    Retries . . . . . . . . . . . . .   3          0-99
    Time interval . . . . . . . . . .   30         0-99 (minutes)
  Second level:
    Retries . . . . . . . . . . . . .   3          0-9
    Time interval . . . . . . . . . .   1          0-9 (days)
```

H.11 Change TCP/IP Tuning Values

```
                    Change TCP/IP Tuning Values
                                                    System: RALYAS4B
Type choices, press Enter.

  Activity control blocks . . . . . .   *CALC    20-2000, *CALC
  Client control blocks . . . . . . .   *CALC    10-300, *CALC
  Socket control blocks . . . . . . .   *CALC    10-256, *CALC
  Transmission control blocks . . . .   *CALC    10-256, *CALC
  Data buffers  . . . . . . . . . . .   *CALC    10-160, *CALC
  UDP control blocks  . . . . . . . .   *CALC    10-200, *CALC
  Data envelopes  . . . . . . . . . .   *CALC    40-1500, *CALC
```

H.12 Convert the Host Table

```
                    Convert TCP/IP Host Table
                                                System:   RALYAS4B
Type choices, press Enter.

  File  . . . . . . . . . . . . . .   QATMTCP          Name
    Library . . . . . . . . . . . .     QUSRSYS        *LIBL, name
  Member  . . . . . . . . . . . . .   HOSTS            *FIRST, name
  File format . . . . . . . . . . .   *AIX             *AIX, *NIC
  Merge with host table . . . . . .   N                Y=Yes, N=No
```

Appendix I. Configuration Listings for OS/2 System FRED

FRED is an IBM PS/2 80-311 running OS/2 V2.0, E/S V1.0 with TCP/IP V1.2.1 for OS/2.

I.1 File ″c:\config.sys″

```
IFS=C:\OS2\HPFS.IFS /CACHE:64 /CRECL:4
PROTSHELL=C:\OS2\PMSHELL.EXE
SET USER_INI=C:\OS2\OS2.INI
SET SYSTEM_INI=C:\OS2\OS2SYS.INI
SET OS2_SHELL=C:\OS2\CMD.EXE
SET AUTOSTART=PROGRAMS,TASKLIST,FOLDERS
SET RUNWORKPLACE=C:\OS2\PMSHELL.EXE
SET COMSPEC=C:\OS2\CMD.EXE
LIBPATH=.;C:\OS2\DLL;C:\MUGLIB\DLL;C:\OS2\MDOS;C:\CMLIB\DLL;C:\;C:\OS2\APPS\DLL;C:\TCPIP\DLL;C:\IBMCOM\DLL;
SET
PATH=C:\OS2;C:\MUGLIB;C:\OS2\SYSTEM;C:\OS2\MDOS\WINOS2;C:\CMLIB;C:\CMLIB\APPN;C:\OS2\INSTALL;C:\;C:\OS2\MDOS;C:\OS2\APPS;
C:\TCPIP\BIN;;C:\IBMCOM;
SET DPATH=C:\OS2;C:\MUGLIB\DLL;C:\CMLIB;C:\CMLIB\APPN;C:\OS2\SYSTEM;C:\OS2\MDOS\
WINOS2;C:\OS2\INSTALL;C:\;C:\OS2\BITMAP;C:\OS2\MDOS;C:\OS2\APPS;C:\IBMCOM;
SET PROMPT=$i[$p
SET HELP=C:\CMLIB\APPN;C:\OS2\HELP;C:\OS2\HELP\TUTORIAL;C:\TCPIP\HELP;
SET GLOSSARY=C:\OS2\HELP\GLOSS;
PRIORITY_DISK_IO=YES
FILES=20
DEVICE=C:\OS2\R0CSDD.SYS
DEVICE=C:\IBMCOM\LANMSGDD.OS2 /I:C:\IBMCOM
DEVICE=C:\IBMCOM\PROTMAN.OS2 /I:C:\IBMCOM
DEVICE=C:\ibmcom\protocol\LANDD.OS2
DEVICE=C:\ibmcom\protocol\LANDLLDD.OS2
DEVICE=C:\OS2\TESTCFG.SYS
DEVICE=C:\OS2\DOS.SYS
DEVICE=C:\OS2\PMDD.SYS
BUFFERS=30
IOPL=YES
DISKCACHE=512,LW
MAXWAIT=3
MEMMAN=SWAP,PROTECT
SWAPPATH=C:\OS2\SYSTEM 2048 2048
BREAK=OFF
THREADS=256
PRINTMONBUFSIZE=134,134,134
COUNTRY=001,C:\OS2\SYSTEM\COUNTRY.SYS
SET KEYS=ON
REM SET DELDIR=C:\DELETE,512;D:\DELETE,512;
BASEDEV=PRINT02.SYS
BASEDEV=IBM2FLPY.ADD
BASEDEV=IBM2ADSK.ADD
BASEDEV=OS2DASD.DMD
SET BOOKSHELF=C:\OS2\BOOK;
SET EPATH=C:\OS2\APPS
REM DEVICE=C:\OS2\APPS\SASYNCDB.SYS
PROTECTONLY=NO
SHELL=C:\OS2\MDOS\COMMAND.COM C:\OS2\MDOS /P
FCBS=16,8
RMSIZE=640
DEVICE=C:\OS2\MDOS\VEMM.SYS
DEVICE=C:\OS2\MDOS\VMOUSE.SYS
DOS=LOW,NOUMB
DEVICE=C:\OS2\MDOS\VDPX.SYS
DEVICE=C:\OS2\MDOS\VXMS.SYS /UMB
DEVICE=C:\OS2\MDOS\VDPMI.SYS
DEVICE=C:\OS2\MDOS\VWIN.SYS
DEVICE=C:\OS2\MDOS\VCDROM.SYS
DEVINFO=SCR,VGA,C:\OS2\VIOTBL.DCP
SET VIDEO_DEVICES=VIO_VGA
SET VIO_VGA=DEVICE(BVHVGA)
DEVICE=C:\OS2\MDOS\VVGA.SYS
DEVICE=C:\OS2\MDOS\V8514A.SYS
DEVICE=C:\OS2\POINTDD.SYS
DEVICE=C:\OS2\MOUSE.SYS
DEVICE=C:\OS2\COM.SYS
DEVICE=C:\OS2\MDOS\VCOM.SYS
CODEPAGE=437,850
DEVINFO=KBD,US,C:\OS2\KEYBOARD.DCP
DEVICE=C:\CMLIB\ACSLDLAN.SYS
RUN=C:\OS2\EPW.EXE
RUN=C:\IBMCOM\PROTOCOL\NETBIND.EXE
RUN=C:\IBMCOM\LANMSGEX.EXE
DEVICE=C:\IBMCOM\MACS\IBMTOK.OS2
DEVICE=C:\CMLIB\APPN\CMKFMDE.SYS
DEVICE=C:\IBMCOM\PROTOCOL\INET.SYS
DEVICE=C:\IBMCOM\PROTOCOL\IFNDIS.SYS
SET ETC=C:\TCPIP\ETC
SET TMP=C:\TCPIP\TMP
RUN=C:\TCPIP\BIN\CNTRL.EXE
IFS=C:\TCPIP\BIN\NFS200.IFS
SET XFILES=C:\TCPIP\X11
```

```
SET DISPLAY=fred:0
SET TELNET.PASSWORD.ID=pass
SET HOSTNAME=fred
SET USER=fred
SET PASSWD=pass
```

I.2 File "c:\tcpip\bin\tcpstart.cmd"

```
@echo off
echo CONFIGURING TCP/IP .....
CALL C:\TCPIP\BIN\SETUP.CMD
echo ..... FINISHED CONFIGURING TCP/IP

echo STARTING THE TCP/IP PROCESSES .....
start inetd
echo     ..... INET Daemon Started
REM start telnetd
REM echo     ..... TELNET Daemon Started
REM start ftpd
REM echo     ..... FTP Daemon Started
REM start tftpd
REM echo     ..... TFTP Daemon Started
REM start rexecd
REM echo     ..... REXEC Daemon Started
REM start rshd
REM echo     ..... RSH Daemon Started
REM start lpd
REM echo     ..... LP Daemon Started
start pmx -nocopyright
echo     ..... X System Server Started
REM start talkd
REM echo     ..... TALK Daemon Started
REM start portmap
REM echo     ..... Portmapper Started
REM start nfsd
REM echo     ..... Network File System Server Started
REM call nfsstart
REM echo     ..... Network File System Client Started
REM start routed
REM echo     ..... ROUTED Started
REM start sendmail -bd -q30m
REM echo     ..... SENDMAIL Started
REM start lamail
REM echo     ..... LAMAIL Started
REM echo     ..... INITIATING call to SLIP.CMD
REM call slip
REM echo     ..... FINISHED call of SLIP.CMD
echo ..... FINISHED STARTING THE TCP/IP PROCESSES
echo .....  EXITING TCPSTART.CMD  .....
```

I.3 File "c:\tcpip\bin\setup.cmd"

```
route -fh
arp -f
ifconfig lan0 9.67.38.98 netmask 255.255.255.192
REM ifconfig lan1
REM ifconfig lan2
REM ifconfig lan3
```

```
REM ifconfig sl
route add default 9.67.38.65 1
```

I.4 File "c:\tcpip\etc\hosts"

```
# Just in case the Name Server is not working
9.67.38.66 ral9390 vm14
9.67.38.67 vm15
9.67.38.71 rs60001
9.67.38.98 fred.itsc.raleigh.ibm.com fred
9.67.38.65 vmesa
```

I.5 File "c:\tcpip\etc\gateways"

Nothing defined

I.6 File "c:\tcpip\etc\resolv"

```
domain itsc.raleigh.ibm.com
nameserver 9.67.38.65
nameserver 9.67.38.66
```

I.7 File "c:\tcpip\etc\inetd.lst"

```
telnet tcp telnetd
```

I.8 File "d:\tcpip\etc\pw.src"

```
tcp 9.67.38.0 255.255.255.0
ITSC 9.67.38.0 255.255.255.0
```

Appendix J. Configuration Listings for RS/6000 System RS60001

J.1 SMIT: Minimum Configuration and Startup

```
                                    TCP/IP

Move cursor to desired item and press Enter.

  Minimum Configuration & Startup
  Further Configuration




  ″
  ]                        Available Network Interfaces                    ]
  ]                                                                        ]
  ] Move cursor to desired item and press Enter.                           ]
  ]                                                                        ]
  ]   en0                                                                  ]
  ]   et0                                                                  ]
  ]   tr0                                                                  ]
  ]                                                                        ]
  ] F1=Help                F2=Refresh              F3=Cancel               ]
  ] Esc+8=Image            Esc+0=Back              Enter=Do                ]
  î
```

```
                        Minimum Configuration & Startup

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                      [Entry Fields
* HOSTNAME                                           [rs60001
* Internet ADDRESS (dotted decimal)                  [9.67.32.85
  Network MASK (dotted decimal)                      [255.255.255.192
* Network INTERFACE                                   en0
  NAMESERVER
           Internet ADDRESS (dotted decimal)         [9.67.38.65
           DOMAIN Name                               [itsc.raleigh.ibm.com
  ROUTE
  GATEWAY Address                                    [
  (dotted decimal or symbolic name)
  START Now                                           no                    +




F1=Help           F2=Refresh        F3=Cancel         F4=List
Esc+5=Undo        Esc+6=Command     Esc+7=Edit        Esc+8=Image
Esc+9=Shell       Esc+0=Back        Enter=Do
```

```
                        Minimum Configuration & Startup

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                        [Entry Fields
* HOSTNAME                                             [rs60001
* Internet ADDRESS (dotted decimal)                    [9.67.32.85
  Network MASK (dotted decimal)                        [255.255.255.192
* Network INTERFACE                                     et0
  NAMESERVER
           Internet ADDRESS (dotted decimal)           [9.67.38.65
           DOMAIN Name                                 [itsc.raleigh.ibm.com
  ROUTE
  GATEWAY Address                                      [
  (dotted decimal or symbolic name)
  START Now                                             no                    +



F1=Help             F2=Refresh          F3=Cancel           F4=List
Esc+5=Undo          Esc+6=Command       Esc+7=Edit          Esc+8=Image
Esc+9=Shell         Esc+0=Back          Enter=Do
```

```
                        Minimum Configuration & Startup

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                        [Entry Fields
* HOSTNAME                                             [rs60001
* Internet ADDRESS (dotted decimal)                    [9.67.38.71
  Network MASK (dotted decimal)                        [255.255.255.192
* Network INTERFACE                                     tr0
  NAMESERVER
           Internet ADDRESS (dotted decimal)           [9.67.38.65
           DOMAIN Name                                 [itsc.raleigh.ibm.com
  ROUTE
  GATEWAY Address                                      [
  (dotted decimal or symbolic name)
  START Now                                             no                    +



F1=Help             F2=Refresh          F3=Cancel           F4=List
Esc+5=Undo          Esc+6=Command       Esc+7=Edit          Esc+8=Image
Esc+9=Shell         Esc+0=Back          Enter=Do
```

J.2 SMIT: Further Configuration

```
          Change / Show Restart Characteristics of routed Subsystem


Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                      [Entry Fields
* LOG DEBUGGING information                          no                    +
* This host is acting as a GATEWAY                   yes                   +
* SUPPRESS sending routing information               no                    +
* DO supply routing information                      yes                   +
* Write all packets sent and received to STDOUT      no                    +
  Write all PACKETS to LOGFILE                      [

F1=Help             F2=Refresh          F3=Cancel           F4=List
Esc+5=Undo          Esc+6=Command       Esc+7=Edit          Esc+8=Image
Esc+9=Shell         Esc+0=Back          Enter=Do
```

```
                              COMMAND STATUS

Command: OK            stdout: yes           stderr: no

Before command completion, additional instructions may appear below.

[TOP
Service Socket  Protocol Wait/  User    Server Program  Server Program
Name    Type             Nowait                         Arguments
-------------------------------------------------------------------------------
echo    stream  tcp      nowait  root    internal
echo    dgram   udp      wait    root    internal
discard stream  tcp      nowait  root    internal
discard dgram   udp      wait    root    internal
daytime stream  tcp      nowait  root    internal
daytime dgram   udp      wait    root    internal
chargen stream  tcp      nowait  root    internal
chargen dgram   udp      wait    root    internal
ftp     stream  tcp      nowait  root    /etc/ftpd ftpd
telnet  stream  tcp      nowait  root    /etc/telnetd telnetd
time    stream  tcp      nowait  root    internal
time    dgram   udp      wait    root    internal
exec    stream  tcp      nowait  root    /etc/rexecd rexecd
login   stream  tcp      nowait  root    /etc/rlogind rlogind
shell   stream  tcp      nowait  root    /etc/rshd rshd
ntalk   dgram   udp      wait    root    /etc/talkd talkd
uucp    stream  tcp      nowait  root    /etc/uucpd uucpd

[BOTTOM

F1=Help             F2=Refresh          F3=Cancel           Esc+6=Command
Esc+8=Image         Esc+9=Shell         Esc+0=Back
```

J.3 SMIT: Block Multiplexer Configuration

```
     Change/Show Characteristics of a 370 Parallel Channel Adapter


Type or select values in entry fields.
Press Enter AFTER making all desired changes.

[TOP                                                  [Entry Fields
  370 Parallel Channel                               cat0
  Description                                        370 Parallel Channel A
  Status                                             Available
  Location                                           00-05
  Receive data transfer OFFSET                       [0                    +#
  Channel SPEED                                      [0                    +#
  NUMBER of transmit buffers                         [26                   +#
  NUMBER of receive buffers                          [26                   +#
  Transmit buffer SIZE                               [4096                 +#
  Receive buffer SIZE                                [4096                 +#
  STARTING subchannel address                        [0x60                 +
  NUMBER of subchannel addresses                     [2                    +#
  CLAW Mode
          Host NAME                                  [HOST
          Adapter NAME                               [PSCA                 +
          subchannel set                             [0x60                 +
  Online/Offline SWITCH                              [online               +
  Online/Offline INDICATOR                            offline              +
  Apply change to DATABASE only                       no
[BOTTOM                                                                     +


F1=Help           F2=Refresh        F3=Cancel         F4=List
Esc+5=Undo        F6=Command        F7=Edit           F8=Image
F9=Shell          F10=Exit          Enter=Do
```

Appendix K. Configuration Listings for PS/2 AIX System PSAIX

K.1 File "/etc/rc.tcpip"

```
# SCCSID(@(#)rc.tcpip      1.6  AIX) /* Modified 2/17/89 20:53:31 */
#
#5713-AEQ COPYRIGHT IBM 1988
#LICENSED MATERIAL - PROGRAM PROPERTY OF IBM
#REFER TO COPYRIGHT INSTRUCTIONS FORM NO. G120-2083
# This file initializes the site independant portions of the IP interface.
#
# $SITE && $LOCAL are exported in Singl2multi.
#
if [ -x $LOCAL/rc.tcpip.local             # Needs site specific initialization
then
     /bin/sh $LOCAL/rc.tcpip.local   /dev/syscon 2 &1
else
     /etc/ifconfig net0 psaix netmask 255.255.255.192     # Default to Ethernet board
fi

#
# Setup local routing info.
#
# example:
#
# /etc/route add prod-net $SITE 0    # Production net
#

/etc/route add default 9.67.32.85 1
#
# Start portmap.
#
if [ -x /etc/portmap
then
     /etc/portmap
     sleep 3
fi

#
# Start master internet daemon.
#
echo  Starting master internet daemon...
/etc/inetd

#
# Start rwho daemon.
#
# uncomment the following code to start the rwhod.
# if u370 || i386
# then
#    if [ -x /etc/rwhod
#    then
#         echo  Starting rwho daemon...
#         /etc/rwhod
#    fi
# fi
```

```
#
# Start uucpd daemon.
#
# uncomment the following code to start the uucpd.
#
# if [ -x /etc/uucpd
# then
#    echo  Starting uucp daemon...
#    /etc/uucpd
# fi

# Start routed.
#
if [ -x /etc/routed
then
     /etc/routed
     sleep 3
fi
```

K.2 File "/etc/inetd"

```
# SCCSID(@(#)inetd.conf   1.16 AIX) /* Modified: 20:50:16 2/17/89 */
#
# Internet server configuration database
#5713-AEQ COPYRIGHT IBM 1988
#LICENSED MATERIAL - PROGRAM PROPERTY OF IBM
#REFER TO COPYRIGHT INSTRUCTIONS FORM NO. G120-2083
#
ftp  stream    tcp  nowait    root /etc/ftpd  ftpd
telnet    stream    tcp  nowait    root /etc/telnetd    telnetd
shell     stream    tcp  nowait    root /etc/rshd  rshd
login     stream    tcp  nowait    root /etc/rlogind    rlogind
exec stream    tcp  nowait    root /etc/rexecd     rexecd
# Run as user †uucp† if you don¢t want uucpd¢s wtmp entries.
#uucp      stream    tcp  nowait    root /etc/uucpd      uucpd
finger    stream    tcp  nowait    nobody    /etc/fingerd    fingerd
printer   stream    tcp  nowait    root /etc/lpd   lpd
#smtp      stream    tcp  nowait    root /usr/lib/sendmail    sendmail -bn
tftp dgram     udp  wait nobody    /etc/tftpd      tftpd
#comsat    dgram     udp  wait root /etc/comsat     comsat
ntalk     dgram     udp  wait root /etc/talkd      talkd
#talk      dgram     udp  wait root /etc/talkd      talkd
#ntalk     dgram     udp  wait root /etc/ntalkd     ntalkd
#echo      stream    tcp  nowait    root internal
#discard   stream    tcp  nowait    root internal
#chargen   stream    tcp  nowait    root internal
#daytime   stream    tcp  nowait    root internal
#time      stream    tcp  nowait    root internal
#echo      dgram     udp  wait root internal
#discard   dgram     udp  wait root internal
#chargen   dgram     udp  wait root internal
#daytime   dgram     udp  wait root internal
#time      dgram     udp  wait root internal
# rpc entries have following format :-
```

```
# rpc sockettype protocol wait/nowait user program prog-number version-number
#rpc dgram   udp     wait    root    /etc/mountd     100005 1
```

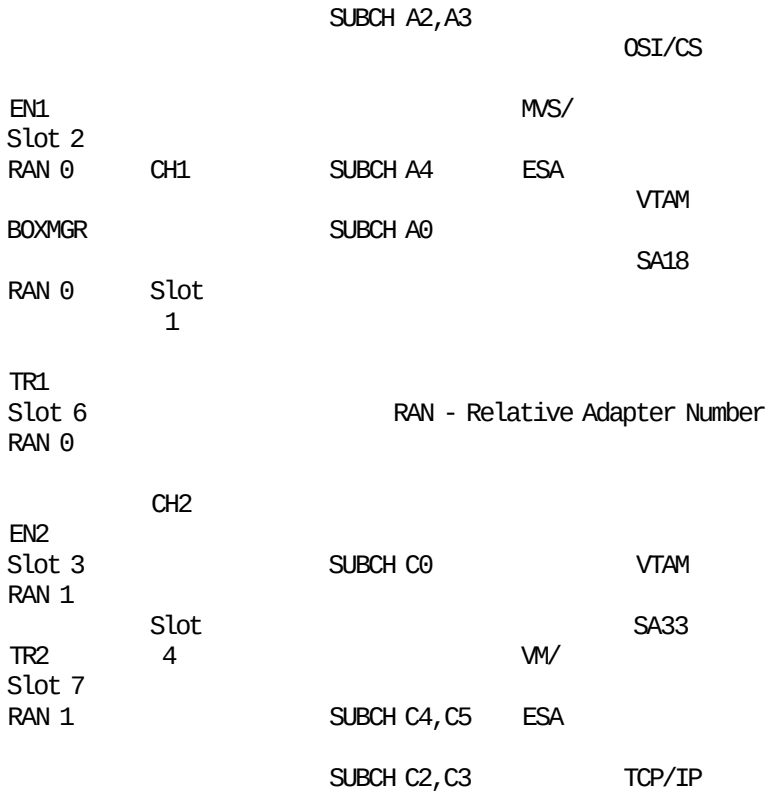
K.3 File "/etc/hosts"

```
# SCCSID(@(#)hosts   1.6  LCC) /* Modified: 20:22:44 9/25/89 */
# Berkeley 4.3 Distribution Host Database
#
9.67.32.67      psaix

#
# loopback network (for testing on a local site)
#
127.0.0.1  localhost
9.67.32.85      rs60001
9.67.38.72      rs60002
9.67.32.73      rs60003
9.67.32.64;vmesa
9.67.38.66      vm14
9.67.38.67      vm15
```

Appendix L. 3172 Configuration

L.1 ITSC 3172 Configuration



L.1.1.1 Configuring the Token-Ring Adapters

Note: If you configure a TRA you MUST have it connected to the network for an IPL to complete. Failure to have a cable connected to the adapter will cause ″180F″ error. Failure to have the adapter connected to the ring will cause a ″1803″ error.

L.1.1.2 Configuring the Ethernet Adapters

We chose BNC for the Ethernet adapters. This indicates that we are using the thin-cable Ethernet adapter and using the transceiver on the adapter. The multicast address is chosen at random, with the stipulation that the second digit must be odd.

Note: Ethernet attachment is not checked at IPL time.

```
                                  Channel Adapter Configuration

System Name : ITS3172          Channel Adapter 1    Channel Adapter 2


Data Transfer Mode.............:     S                    S
Channel Speed..................:     1                    1
Slot Number....................:     1                    4
Subchannel Pair................:     A0                   C0
  Block Delay Time.............:     10                   10
  Maximum Response Length......:     500                  500
Subchannel Pair................:     A2,A3                C2,C3
  Block Delay Time.............:     10                   10
  Maximum Response Length......:     500                  500
Subchannel Pair................:     A4                   C4,C5
  Block Delay Time.............:     10                   10
  Maximum Response Length......:     500                  500
Subchannel Pair................:
  Block Delay Time.............:
  Maximum Response Length......:
```

Note: Channel Adapter 1 Subchannel A0 and Channel Adapter 2 Subchannel C0 are used for VTAM traffic (VTAM only requires 1 subchannel address). A4 is used as the Netview Reporting Subchannel.

```
                                  LAN Adapter Configuration

                                LAN            LAN            LAN            LAN
System Name : its3172        Adapter 1      Adapter 2      Adapter 3      Adapter 4


LAN Adapter Type......:        TR4            TR4            EN             EN
Slot Number...........:        6              7              2              3
Uses Channel Adapter..:        1,2            2              1              2
Uses Subchannel Pair..:        A0,C0          C2,C3          A2,A3          C4,C5
Max Outbound Rec Len..:        2048           2048           n/a            n/a
Max Inbound Rec Len...:        2048           2048           n/a            n/a
Node Address..........:  400050013172   400050023172   400060013172   400060023172
Adapter ID............:        2              2              1              1
Transceiver Type......:        n/a            n/a            BNC            BNC
Receive Mode..........:        n/a            n/a            S              S
Multi-cast Address....:        n/a            n/a            410060013172   None
Multi-cast Address....:        n/a            n/a            None           None
Relative Adapter Num..:        0              1              0              1
Host Program..........:        VTAM           OTHER          OTHER          OTHER
Operator Facility.....:        Yes            No             No             No
```

L.2 ITSC 3172 Definition

The following definitions have been made in the ″PROFILE TCPIP″ of the RALYESA system. The TCP/IP server uses channel adapter 2 in the 3172 and subchannel pair (C2, C3) for the token-ring adapter and (C4, C5) for the Ethernet adapter.

```
; Sample device statements

DEVICE LCS1       LCS         EC2
LINK TR1     IBMTR    1 LCS1

DEVICE LCS2       LCS         EC4
LINK ETH1    ETHERNET 1 LCS2

; the local host¢s internet addresses
HOME
    9.67.32.81   ETH1
    9.67.32.17   TR1

; ; RouteD Routing information (if you are using the ROUTED server)
; ; If you are using RouteD, uncomment all the lines below for
; ; ¢BSDROUTINGPARMS¢, and comment out all the lines for the ¢GATEWAY¢
; ; statement.
;
; ;  LINK     MAXMTU      METRIC   SUBNET MASK       DEST ADDR
  BSDROUTINGPARMS FALSE
     ETH1     DEFAULTSIZE 0        255.255.255.240   0
     TR1      DEFAULTSIZE 0        255.255.255.240   0
  ENDBSDROUTINGPARMS
;

; Start all the interfaces
START LCS1
START LCS2
```

L.3 3172 Hints and Tips

The following problem appears when the *link number* of the LINK statement in your PROFILE TCPIP does not correspond to the channel adapter number defined in your 3172 configuration.

The example below shows the TCPIP startup console log if we assign 0 as the *link number* for the token-ring and Ethernet links.

```
Devices and Links:
Device:LCS3172, Type: LCS
   Link: TR1, Type: IBMTR, Network number: 0
   Link: ETH1, Type: ETHERNET, Network number: 0
Device:SNADMV18, Type: SNA IUCV
   Link: SNALMV18, Type: IUCV, Network number: 2

Default home address and link: 9.67.32.81; ETH1
Other home addresses and links:
    9.67.32.17;  TR1
    9.67.32.129;  SNALMV18
The only known network is:    Default     direct     Link Name: SNALMV18,
Link Type: IUCV, Dev Name: SNADMV18, Dev Type: SNA IUCV   max 0  not subnetted
I can translate 0 internet addresses:
Telnet server: Global connection to *CCS CP System Service established
Telnet server: First line of *CCS logo is: VM/ESA ONLINE-PRESS ENTER KEY TO
BEGIN SESSION
Device LCS3172: Received startup packet
Device LCS3172: Error 1 in getting home hardware address for IBMTR number 0
Device LCS3172: Error 1 in getting home hardware address for ETHERNET number 0
PCCA3 device LCS3172: FindLinkNumber fails for NetType 48758784 AdapterNumber 28
630772
PCCA3 device LCS3172: FindLinkNumber fails for NetType 48758784 AdapterNumber 28
630772
```

Appendix M. Abbreviations

Abbreviation	Meaning
API	Application Programming Interface
ARP	Address Resolution Protocol
ASN.1	Abstract Syntax Notation One
BSD	Berkeley Software Distribution
CCS	Console Communication Facility
DAC	Dual Access Concentrator
DAS	Dual Access Station
DNS	Domain Name System
DPI	Distributed Programming Interface
EGP	Exterior Gateway Protocol
FDDI	Fiber Distributed Data Interface
FTP	File Transfer Protocol
GCS	Group Control System
ICMP	Internet Control Message Protocol
IGP	Interior Gateway Protocol
IP	Internet Protocol
ITSC	International Technical Support Center
IUCV	Inter-User Communication Vehicle
LDSF	Logical Device Support Facility
MIB	Management Information Base
MX	Mail Exchange
NCS	Network Computing System
NCST	NCP Connectionless SNA Transport
NDB	Network Database System
NFS	Network File System
NIDL	Network Interface Definition Language
NLS	National Language Support
PDU	Protocol Data Unit
RARP	Reverse Address Resolution Protocol
REXEC	Remote Execution Protocol
RFC	Request for Comments
RIP	Routing Information Protocol
RPC	Remote Procedure Call

RSCS	Remote Spooling Communication Subsystem
SAC	Single Access Concentrator
SAS	Single Access Station
SMI	Structure of Management Information
SMSG	Special Message
SMTP	Simple Mail Transfer Protocol
SNMP	Simple Network Management Protocol
SNA	System Network Architecture
TCP	Transmission Control Protocol
TFTP	Trivial File Transfer Protocol
UDP	User Data Protocol
VMCF	VM Communication Facility
VTAM	Virtual Telecomunication Access Method
XDR	External Data Representation Standard

Index

Special Characters

Numerics

A

B

C

D

E

F

H

I

J

L

M

N

O

P

R

S

ITSO Technical Bulletin Evaluation RED000

IBM TCP/IP Version 2 Release 2 for VM
Installation and Interoperability

Publication No. GG24-3624-02

Your feedback is very important to help us maintain the quality of ITSO Bulletins. **Please fill out this questionnaire and return it using one of the following methods:**

- Mail it to the address on the back (postage paid in U.S. only)
- Give it to an IBM marketing representative for mailing
- Fax it to: Your International Access Code + 1 914 432 8246
- Send a note to REDBOOK@VNET.IBM.COM

Please rate on a scale of 1 to 5 the subjects below.
(1 = very good, 2 = good, 3 = average, 4 = poor, 5 = very poor)

Overall Satisfaction	____		
Organization of the book	____	Grammar/punctuation/spelling	____
Accuracy of the information	____	Ease of reading and understanding	____
Relevance of the information	____	Ease of finding information	____
Completeness of the information	____	Level of technical detail	____
Value of illustrations	____	Print quality	____

Please answer the following questions:

a) If you are an employee of IBM or its subsidiaries:

Do you provide billable services for 20% or more of your time? Yes____ No____

Are you in a Services Organization? Yes____ No____

b) Are you working in the USA? Yes____ No____

c) Was the Bulletin published in time for your needs? Yes____ No____

d) Did this Bulletin meet your needs? Yes____ No____

If no, please explain:

_____

_____

What other topics would you like to see in this Bulletin?

_____

_____

What other Technical Bulletins would you like to see published?

_____

Comments/Suggestions: (THANK YOU FOR YOUR FEEDBACK!)

_____ _____
Name Address

_____ _____
Company or Organization

_____ _____
Phone No.

Fold and Tape **Please do not staple** Fold and Tape

NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

IBM International Technical Support Organization
Department 985, Building 657
P.O. BOX 12195
RESEARCH TRIANGLE PARK NC
USA 27709-2195

Fold and Tape **Please do not staple** Fold and Tape

Printed in U.S.A.

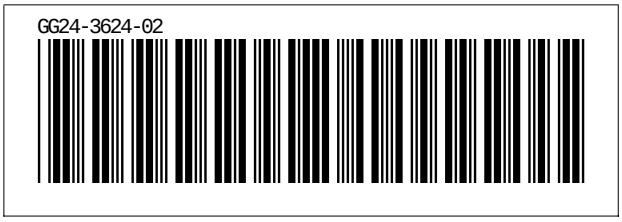


Table Definitions

Figures

3745CNF	GG243624 SCRIPT	93	25 92, 92, 92
SMTPGW	GG243624 SCRIPT	103	26
HDR822	GG243624 SCRIPT	107	27 107
SNALNK	GG243624 SCRIPT	118	28 118
SNAXRE	GG243624 SCRIPT	119	29
XWSSD13	GG243624 SCRIPT	130	30 130
XWSSD12	GG243624 SCRIPT	134	31 133
XWSSD2D	GG243624 SCRIPT	139	32 139, 139
XWSSD2E	GG243624 SCRIPT	140	33 140
NLSTELO	GG243624 SCRIPT	264	38
ITSCENV	GG243624 SCRIPT	267	39 xx, 233, 235

Headings

id	File	Page	References
NOTICES	GG243624 SCRIPT	xvii	Special Notices ii
BIBL	GG243624 SCRIPT	xxi	Related Publications
INTRO	GG243624 SCRIPT	1	Chapter 1, Introduction xix
FOVER	GG243624 SCRIPT	1	1.1, Functional Overview
FCOGW	GG243624 SCRIPT	1	1.1.1, Connectivity and Gateway Functions
FSRV	GG243624 SCRIPT	1	1.1.2, Server Functions
FCLI	GG243624 SCRIPT	2	1.1.3, Client Functions
FNWS	GG243624 SCRIPT	2	1.1.4, Network Status and Management Functions
FAPI	GG243624 SCRIPT	3	1.1.5, Application Programming Interfaces (APIs)
NEWFNS	GG243624 SCRIPT	3	1.2, Functions and Enhancements Included in IBM TCP/IP Version 2 Release 1 for VM
ISMTPE	GG243624 SCRIPT	3	1.2.1, Simple Mail Transfer Protocol (SMTP) Enhancements
IKERB	GG243624 SCRIPT	4	1.2.2, Kerberos Services and API
INCS	GG243624 SCRIPT	6	1.2.3, Network Computing System (NCS) API
ISNMP	GG243624 SCRIPT	7	1.2.4, Simple Network Management Protocol (SNMP) 86
CSNMO	GG243624 SCRIPT	7	1.2.4.1, Overview
CSNMVM	GG243624 SCRIPT	10	1.2.4.2, SNMP Implementation in IBM TCP/IP V2 for VM
C1SNMM	GG243624 SCRIPT	11	1.2.5, SNMP Command Processing
CDPIVM	GG243624 SCRIPT	11	1.2.5.1, SNMP Agent Distributed Program Interface (DPI)
IRIP	GG243624 SCRIPT	12	1.2.6, Routing Information Protocol (RIP)
GWPO	GG243624 SCRIPT	13	1.2.6.1, Routing Protocol Overview
RIPO	GG243624 SCRIPT	13	1.2.6.2, RIP Overview
LIMRIP	GG243624 SCRIPT	14	RIP Limitations
ILPD	GG243624 SCRIPT		

Index Entries

Footnotes

id	File	Page	References
SMTDOS	GG243624 SCRIPT	56	3
			56
VT92	GG243624 SCRIPT	56	4
			56, 56
XWINDG	GG243624 SCRIPT	56	5
			56, 56
XWINDOS	GG243624 SCRIPT	56	6
			56
X400	GG243624 SCRIPT	56	7
			56
REXFTP	GG243624 SCRIPT	56	8
			56

Revisions

id	File	Page	References
DASH1	GG243624 SCRIPT	xviii	
			iii, iii, xxi, xxi, xxi, xxi, 1, 1, 2, 2, 2, 2, 4, 6, 11, 11, 15, 15, 16, 54, 58, 58, 58, 58, 59, 59, 59, 59, 59, 59, 59, 59, 59, 60, 60, 178, 179, 184, 186, 250, 268, 268, 287, 289, 290, 290, 323, 328

Spots

id	File	Page	References
RUNEX	GG243624 SCRIPT	75	(no text)
			158
VMCLAW	GG243624 SCRIPT	90	(no text)
			89
MASFIL1	GG243624 SCRIPT	152	(no text)
			148, 155, 157
MASFIL2	GG243624 SCRIPT	153	(no text)
			148, 150, 155, 157
LPDCO	GG243624 SCRIPT	187	(no text)
			221

3174	GG243624 SCRIPT		
		218	(no text)
			55, 265
PAGENO	GG243624 SCRIPT		
		331	(no text)

Tables

id	File	Page	References
TSUM	GG243624 SCRIPT		
		53	2
			53
TFILE	GG243624 SCRIPT		
		70	4

Processing Options

Runtime values:

- Document fileid ..... GG243624 SCRIPT
- Document type ..... USERDOC
- Document style ..... IBMXAGD
- Profile EDFPRF30
- Service Level 0029
- SCRIPT/VS Release 4.0.0
- Date 93.09.16
- Time 17:06:33
- Device 3820A
- Number of Passes 3
- Index ..... YES
- SYSVAR D ..... YES
- SYSVAR G ..... INLINE
- SYSVAR V ..... ITSCEVAL

Formatting values used:

- Annotation ..... NO
- Cross reference listing ..... YES
- Cross reference head prefix only ..... NO
- Dialog ..... LABEL
- Duplex ..... YES
- DVCF conditions file ..... (none)
- DVCF value 1 ..... (none)
- DVCF value 2 ..... (none)
- DVCF value 3 ..... (none)
- DVCF value 4 ..... (none)
- DVCF value 5 ..... (none)
- DVCF value 6 ..... (none)
- DVCF value 7 ..... (none)
- DVCF value 8 ..... (none)
- DVCF value 9 ..... (none)
- Explode ..... NO
- Figure list on new page ..... YES
- Figure/table number separation ..... YES
- Folio-by-chapter ..... NO
- Head 0 body text ..... Part
- Head 1 body text ..... Chapter
- Head 1 appendix text ..... Appendix
- Hyphenation ..... NO
- Justification ..... NO
- Language ..... ENGL
- Layout ..... OFF
- Leader dots ..... YES
- Master index ..... (none)
- Partial TOC (maximum level) 4
- Partial TOC (new page after) ..... INLINE
- Print example id's ..... NO
- Print cross reference page numbers ..... YES
- Process value ..... (none)
- Punctuation move characters ..... .,
- Read cross-reference file ..... (none)
- Running heading/footing rule ..... NONE
- Show index entries ..... NO
- Table of Contents (maximum level) 3
- Table list on new page ..... YES
- Title page (draft) alignment ..... RIGHT
- Write cross-reference file ..... (none)

Imbed Trace

Page 336	3624EVAL
Page 336	RCFADDR
Page 336	ITSCADDR FILE
Page 337	RCFADDR
Page 337	ITSCADDR FILE