IBM AnyNet

IBM

Guide to SNA over TCP/IP for Windows Version 1.0

GV40-0113-00

IBM AnyNet

IBM

Guide to SNA over TCP/IP for Windows Version 1.0

GV40-0113-00

Note

Before using this document, read the general information under “Notices” on page ix.

First Edition (March 1996)

This edition applies to IBM AnyNet SNA over TCP/IP for Windows, Version 1.0, an IBM licensed program.

This publication is not available through the address given below. If you want a copy of this IBM publication, ask your IBM representative or write to the IBM branch office serving your locality.

In addition, procedures for printing the softcopy manual are provided in the installation information package that ships on the product CD or the product diskettes. A formatted PostScript file is also available.

If you have comments, please send them to the following address:

IBM Corporation
Department CGM
P.O. Box 12195
Research Triangle Park, North Carolina 27709-2195
U.S.A.

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

© **Copyright International Business Machines Corporation 1996. All rights reserved.**
Note to U.S. Government Users — Documentation related to restricted rights — Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract with IBM Corp.

Contents

Notices ix
Programming Interface Information x
Trademarks x

About This Book xi
AnyNet Product Family xi
APPC over TCP/IP xi
IPX over SNA Gateway xi
NetBEUI over SNA xi
SNA over TCP/IP xii
SNA over TCP/IP Gateway xii
Sockets over IPX xii
Sockets over NetBIOS xii
Sockets over SNA xii
Sockets over SNA Gateway xii
Online and Hardcopy Documentation xiii
Printing the Online Manual xiii
Printing the Formatted PostScript File xiii

Chapter 1. Introducing SNA over TCP/IP for Windows 1
What Does SNA over TCP/IP Do? 1
Connecting Two Windows Workstations Running APPC Networking Services . . 1
Connecting a Windows Workstation Running APPC Networking Services with Other Workstations on Different Platforms 2
Connecting a Windows Workstation Running Personal Communications to a Subarea Network 4
Connecting a Windows Workstation Running Personal Communications to an APPN Network 5
Connecting a Windows Workstation Running Personal Communications to an AS/400 Host 6
How Does SNA over TCP/IP Work? 7
What Do You Need to Make SNA over TCP/IP Work? 9
Hardware Requirements 9
Software Requirements 9
Additional Software Required for APPC Networking Services 9
Additional Software Required for Personal Communications 10
Application Support 10
Network Interface Protocols 11
TCP/IP Support Services 11
Restriction on Using One SNA Transport User 11

Chapter 2. Planning for SNA over TCP/IP for Windows 13
Address Mapping Overview 13
How Does a Destination LU Get an IP Address? 13
Defining Domain Names and IP Addresses 14

© Copyright IBM Corp. 1996

Planning Considerations for Using the Personal Communications Emulator Package 15
Getting Started 15
How to Configure a Session 16

Chapter 3. Installing, Updating, and Removing SNA over TCP/IP for Windows 19
Getting Started 19
Installing SNA over TCP/IP on a Workstation 21
Installing SNA over TCP/IP on a Code Server 23
Setting Up a Code Server 23
Using the CID Setup Tool to Automate Installation 24
Creating a Response File 25
Creating a Change File Profile for NetView Distribution Manager/2 26
Verifying Additions and Changes Made during Installation 27
File Created in the Windows Directory 27
Files Created in the AnyNet Directory 27
APIPAW Subdirectory 27
BIN Subdirectory 27
DOC Subdirectory 28
MISC Subdirectory 29
Program Group and Programs Created 29
System File Changes 29
APPC Networking Services Changes 30
Personal Communications Changes 30
Updating SNA over TCP/IP on a Workstation or Code Server 30
Removing SNA over TCP/IP from a Workstation 31
Removing SNA over TCP/IP from a Code Server 32

Chapter 4. Configuring SNA over TCP/IP for Windows 35
Using the AnyNet Configuration Tool 35
Selecting the SNA Transport User 35
Configuring SNA-over-TCP/IP Environment Settings 36
SNA Domain Name Suffix 36
Connection Retry Duration 37
Connection Wait Time Limit 37
Remote Node Inactivity Poll Interval 37
Unacknowledged Datagram Retry Interval 38
Unsent Datagram Retry Interval 38
Configuring TCP/IP 38
Configuration Examples 39
Establishing Communications between Two Windows Workstations Running APPC Networking Services 39
Definitions 40
Steps 40
Establishing Communications between a Windows Workstation Running Personal Communications and a VTAM Host Processor 42
Definitions 42
Steps 43

Establishing Communications between a Windows Workstation Running Personal Communications and an AS/400 Host . . . 44
Definitions . . . 45
Steps . . . 46

Chapter 5. Operating and Managing SNA over TCP/IP for Windows . . . 49
Starting SNA over TCP/IP . . . 49
Using APPC Networking Services to Start SNA over TCP/IP . . . 49
Using Personal Communications to Start SNA over TCP/IP . . . 50
Stopping SNA over TCP/IP . . . 51
Using APPC Networking Services to Stop SNA over TCP/IP . . . 51
Using Personal Communications to Stop SNA over TCP/IP . . . 51
Tools for Managing SNA over TCP/IP . . . 51
Managing the IP Network . . . 51
Managing the SNA Network . . . 52

Chapter 6. Troubleshooting . . . 53
Troubleshooting Steps . . . 54
Checking TCP/IP . . . 55
Is TCP/IP Operational on the Local System? . . . 56
Is TCP/IP Operational on the Remote System? . . . 57
Checking APPC Networking Services for Windows . . . 59
Is DLCTYPE Set to MPTN? . . . 59
Is APPC Networking Services Active? . . . 60
Checking Personal Communications for Windows . . . 61
Checking SNA over TCP/IP for Windows . . . 62
Verifying Connectivity Using the APING Program . . . 63
Troubleshooting Aids for APPC Networking Services . . . 64
Networking Services Message Log . . . 64
Networking Services Trace Utility . . . 65
Starting the Trace Utility . . . 65
Formatting and Viewing Trace Output . . . 67
Reading Trace Data . . . 67
Session Initiation Trace Examples . . . 68
Abinfo Mapping Utility . . . 68
Troubleshooting Aids for Personal Communications . . . 69
Personal Communications Message and Trace Log . . . 69
Personal Communications Trace Utility . . . 70
Starting the Trace Utility . . . 70
Reading Trace Data . . . 71
Stopping the Trace Utility . . . 71
Session Initiation Trace Example . . . 72
Using IBM Support Services . . . 72
Contract Support . . . 72
Informal Technical Assistance . . . 72

Appendix A. AnyNet Messages and Error Codes . . . 73
Message and Error Code Descriptions . . . 73

Installation and Configuration Message Descriptions 86

Appendix B. Automatic Installation Keywords, Parameters, and Return Codes 91
Response File Keywords 91
Install, Update, Restore, and Delete Keywords 91
Configuration Keywords 93
Parameters for the AnyNet INSTALLW.EXE and ANYNVDMD.EXE Programs . . 93
Return Codes from the AnyNet ANYNVDMD.EXE Program 94

Appendix C. Syntax Conventions 97

Appendix D. Trace Examples for APPC Networking Services 99
Component Trace Examples 100
MPTNASNA Trace 100
MPTNCMM Trace 101
MPTNBTCP Trace 101
Session Initiation Trace Examples 104
Successful Session Initiation 104
Unsuccessful Session Initiation 112

Appendix E. Trace Example for Personal Communications 117

Glossary 131

Bibliography 147
AnyNet Product Family Publications 147
VTAM AnyNet Feature for V4R3 Publications 147
Softcopy Information 147
APPC Publications 147
Communications Server Publications 147
CPI-C Publications 147
MPTN Publications 147
NetView Publications 148
Networking Services Publications 148
Personal Communications Publications 148
TCP/IP for DOS Publications 148
Requests for Comments (RFCs) 148
Obtaining Electronic Copies through FTP 148
Obtaining Electronic Copies through Electronic Mail 149
Obtaining Printed Copies 149

Index 151

Figures

1. Configuration to Connect Two Windows Workstations 2
2. Configuration to Connect Windows, OS/2, OS/400, and AIX Workstations . . 3
3. Configuration to Connect a Windows Downstream Workstation to a Subarea Network 4
4. Configuration to Connect a Windows Downstream Workstation to an APPN Network 5
5. Configuration to Connect a Windows Workstation to an AS/400 Host 6
6. Structure of a Windows SNA-over-TCP/IP Node Running APPC Networking Services 7
7. Structure of a Windows SNA-over-TCP/IP Node Running Personal Communications 8
8. Format of the Domain Name That SNA over TCP/IP Builds 13
9. Address Assignments for Establishing Communication between Two Windows Workstations Running APPC or CPI-C Applications 40
10. Address Assignments for Establishing Communication between a Windows Downstream Workstation and a VTAM Host Processor 42
11. Address Assignments for Establishing Communication between a Windows Workstation and an AS/400 Host 45
12. Troubleshooting Steps 54
13. Using the Netstat Command to Obtain an IP Address 56
14. Using the Ping Command to Check TCP/IP on a Local System 57
15. Using the Ping Command to Check TCP/IP on a Remote System 58
16. Using the Netstat Command to Check SNA over TCP/IP 62
17. Example of an APPC Networking Services Message Log 64
18. Example of a Personal Communications Message and Trace Log 69
19. Example of Error Codes for Message ANY0005E 73
20. MPTNASNA Component Trace Example for APPC Networking Services . 100
21. MPTNCMM Component Trace Example for APPC Networking Services . . 101
22. MPTNBTCP Component Trace Example for APPC Networking Services . 102

Tables

1. Ping Command Responses and Actions for the Local System 57
2. Host Command Responses and Actions for the Remote System 58
3. Ping Command Responses and Actions for the Remote System 59
4. WINNSD.EXE Responses and Actions for APPC Networking Services . . 61
5. Responses and Actions for Personal Communications 62
6. Netstat Command Responses and Actions for SNA over TCP/IP 63
7. APING Responses and Actions 64
8. Message Types 73

© Copyright IBM Corp. 1996

Notices

References in this publication to IBM products, programs, or services do not imply that IBM intends to make them available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any of the intellectual property rights of IBM may be used instead of the IBM product, program, or service. The evaluation and verification of operation in conjunction with other products, except those expressly designated by IBM, are the responsibility of the user.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
500 Columbus Avenue
Thornwood, New York 10594 U.S.A.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

Site Counsel
IBM Corporation
P.O. Box 12195
3039 Cornwallis
Research Triangle Park, NC 27709-2195
USA

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Program License Agreement.

This document is not intended for production use and is furnished as is without any warranty of any kind, and all warranties are hereby disclaimed including the warranties of merchantability and fitness for a particular purpose.

© Copyright IBM Corp. 1996

Programming Interface Information

This book is intended to help the customer install, configure, use, and diagnose SNA over TCP/IP. This book also documents General-Use Programming Interface and Associated Guidance Information provided by SNA over TCP/IP. General-Use Programming Interfaces allow the customer to write programs that obtain the services of SNA over TCP/IP. General-Use Programming Interfaces and Associated Guidance Information is identified where it occurs by an introductory statement to the section.

Trademarks

The following terms are trademarks of the IBM Corporation in the United States, other countries, or both:

AIX	AnyNet
APPN	AS/400
BookManager	IBM
Library Reader	MVS/ESA
NetView	OS/2
OS/400	S/390
SystemView	Trouble Ticket
VTAM	

The following terms are trademarks of other companies:

386SX	Intel Corporation
CompuServe	CompuServe, Incorporated and H&R Block, Incorporated
IPX	Novell, Incorporated
NetWare	Novell, Incorporated
Novell	Novell, Incorporated
Novell LAN WorkPlace	Novell, Incorporated
PostScript	Adobe Systems Incorporated

Windows is a trademark of Microsoft Corporation.

About This Book

This book describes the primary tasks involved with installing, configuring, using, and troubleshooting SNA over TCP/IP for Windows.

AnyNet Product Family

SNA over TCP/IP is a Windows-based AnyNet software offering that implements the multiprotocol transport networking (MPTN) architecture. MPTN architecture supports:

Mixed protocol networking: running application programs that were designed to operate over one transport protocol (such as SNA or TCP/IP) over additional transport protocols.

Network interconnection: connecting matching application program partners across networks of different types, for example, allowing a socket application running on an IP network to communicate with a socket application running on an SNA network.

For more information on MPTN architecture, refer to “MPTN Publications” on page 147.

IBM's AnyNet software enables application programs to communicate over different transport networks and across interconnected networks. Using AnyNet software, you can reduce the number of transport networks and reduce operational complexity. These benefits are gained without modification to your existing application programs or hardware.

The following sections provide an overview of the functions that are part of IBM's AnyNet software offerings. For more information about documentation for these products, refer to “AnyNet Product Family Publications” on page 147.

APPC over TCP/IP

This function, available in the MVS, AIX, Windows, OS/400, and OS/2 environments, enables APPC or CPI-C application programs to communicate with other APPC or CPI-C application programs over an IP network. LU 6.2 protocols for independent LUs are supported.

IPX over SNA Gateway

This function, available in the OS/2 environment, supports Novell NetWare Internet Packet Exchange (IPX) communication from one LAN to another over an existing wide area network (WAN). The gateway communicates with gateways on other LANs that are connected to the same WAN.

NetBEUI over SNA

This function, available in the OS/2 environment, enables NetBIOS applications to communicate over an SNA network with other NetBIOS applications. NetBEUI over SNA uses full-duplex conversations, non-blocking APPC operation, data compression, LU-LU session-level security, and modem support.

© Copyright IBM Corp. 1996

About This Book

SNA over TCP/IP

This function, available in the MVS, Windows, and OS/2 environments, enables SNA application programs to communicate over an IP network. In addition to supporting independent LU 6.2 communication, SNA over TCP/IP provides support for dependent logical unit communications such as emulators and printers. For configuration examples, see “What Does SNA over TCP/IP Do?” on page 1.

SNA over TCP/IP Gateway

This function, available in the MVS and OS/2 environments, connects an IP and an SNA network to enable communication over these networks. Using the SNA over TCP/IP gateway function, SNA applications running nonnatively on the IP network can communicate, without change, with SNA applications on the SNA network.

For independent LU communications, two or more gateways can be used to connect multiple IP networks and SNA networks, thereby allowing SNA applications to communicate with each other across multiple networks.

Note: For MVS environments, this function is provided as part of AnyNet SNA over TCP/IP for MVS.

Sockets over IPX

This function, available in the OS/2 environment, enables C application programs using the IBM TCP/IP AF_INET socket interface to communicate with each other over IPX networks.

Sockets over NetBIOS

This function, available in the OS/2 environment, enables C application programs using the IBM TCP/IP AF_INET socket interface to communicate with each other over NetBIOS networks.

Sockets over SNA

This function, available in the MVS, AIX, OS/2, and OS/400 environments, enables MVS, AIX, OS/2, and OS/400 application programs that use the C socket interface to communicate over SNA networks. Sockets over SNA uses LU 6.2 conversations.

Sockets over SNA Gateway

This function, available in the OS/2 environment, connects IP and SNA networks to enable communication between socket applications. Socket applications on existing IP systems can communicate, without change, with socket applications on SNA systems that have AnyNet Sockets-over-SNA access nodes installed. AnyNet sockets gateways for OS/2 can also be used to connect IP networks across an SNA network, thereby allowing communication between socket applications running on different IP networks.

Online and Hardcopy Documentation

You can view this guide online using the IBM Library Reader or BookManager product. To view the online documentation, click on the "AnyNet Books" icon that is added during the installation of SNA over TCP/IP.

You can order this guide through your IBM Marketing Center by requesting a copy of publication number GV40-0113-00.

You can also obtain a hardcopy of this guide by installing and printing the online manual or the formatted PostScript file. For information on installing the product documentation, see "Installing SNA over TCP/IP on a Workstation" on page 21.

Printing the Online Manual

You can print the online manual using the IBM Library Reader or BookManager by using the following procedures:

1. Open the book to the Edition Notice.
2. Select the entire book as a marked area.
   a. Place the cursor on the first character of the first word.
   b. Press mouse button 1.
   c. Go to the end of the last topic in the bibliography.
   d. Press Shift+mouse button 1.
3. Press Ctrl+P to print the book.

The table of contents and index are not printed.

Printing the Formatted PostScript File

You can print the formatted PostScript file on most PostScript printers. When printing a PostScript file from Windows, copy the file using one of the following methods:

To use DOS commands:

1. Open a DOS window.
2. Enter **either**:

   x:> PRINT *filename.filetype*

   or

   x:> COPY *filename.filetype* LPT*x*:

   where:

   x is the drive where the file is located.
   filename is the name of the file.
   filetype is the three-letter extension.
   LPT*x* is the printer port attached to a PostScript printer.

To use the Windows file manager:

1. Select the PostScript file in the file manager.
2. Click on the “File” menu bar option.
3. Select the “Copy” option from the selection list.
4. When the “Copy” window is displayed, specify the printer port in the “To” field.
5. Select “Yes” in the “Confirm File Replace” window.

Note: The copy function can affect workstation performance until the entire file is copied.

Chapter 1. Introducing SNA over TCP/IP for Windows

This chapter introduces IBM AnyNet SNA over TCP/IP for Windows and contains the following major sections:

"What Does SNA over TCP/IP Do?"
"How Does SNA over TCP/IP Work?" on page 7
"What Do You Need to Make SNA over TCP/IP Work?" on page 9

What Does SNA over TCP/IP Do?

The support provided by SNA over TCP/IP is determined by the SNA transport user you select. You can select either IBM APPC Networking Services for Windows or IBM Personal Communications for Windows as your SNA transport user.

The following outlines the support that is provided by each SNA transport user:

APPC Networking Services for Windows supports CPI-C application programs and APPC application programs that use the Client Access/400 for Windows 3.1 Communications Router application program interface (API). For examples, see:

- "Connecting Two Windows Workstations Running APPC Networking Services"
- "Connecting a Windows Workstation Running APPC Networking Services with Other Workstations on Different Platforms" on page 2

Personal Communications for Windows is an emulator package that provides 3270 and 5250 terminal display and printer support.

- For 3270 examples, see:
  - "Connecting a Windows Workstation Running Personal Communications to a Subarea Network" on page 4
  - "Connecting a Windows Workstation Running Personal Communications to an APPN Network" on page 5
- For a 5250 example, see "Connecting a Windows Workstation Running Personal Communications to an AS/400 Host" on page 6.

Connecting Two Windows Workstations Running APPC Networking Services

Figure 1 on page 2 shows an IP network in which application A communicates with application B. The applications are either APPC or CPI-C applications. The two Windows nodes are running:

IBM APPC Networking Services for Windows
IBM AnyNet SNA over TCP/IP for Windows
IBM TCP/IP for DOS

© Copyright IBM Corp. 1996

Introduction

For an example showing the definitions required to implement the configuration in Figure 1, see “Establishing Communications between Two Windows Workstations Running APPC Networking Services” on page 39.

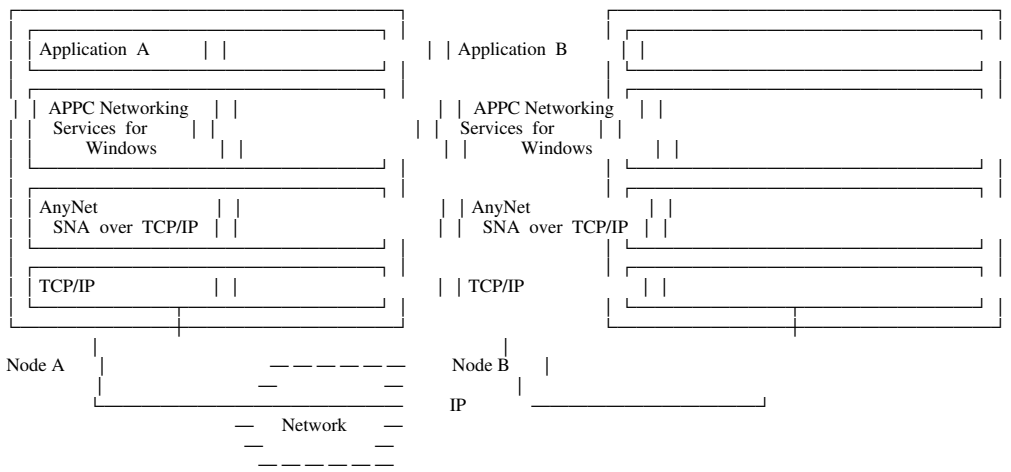


Figure 1. Configuration to Connect Two Windows Workstations

Connecting a Windows Workstation Running APPC Networking Services with Other Workstations on Different Platforms

Figure 2 on page 3 shows how application A, running on a Windows node, communicates with applications running on nodes B, C, D, and E. In this configuration, all applications are either APPC or CPI-C applications.

Node A is running:

- IBM APPC Networking Services for Windows
- IBM AnyNet SNA over TCP/IP for Windows
- IBM TCP/IP for DOS

Node B is running:

- Communications Manager/2
- IBM AnyNet SNA over TCP/IP for OS/2
- IBM TCP/IP for OS/2

Node C is running OS/400, which includes the SNA, TCP/IP, and AnyNet components.

Node D is running:

- IBM SNA Server/6000
- IBM AnyNet APPC over TCP/IP for AIX

Node E has independent LU 6.2 applications running on any operating system that supports SNA (AIX, MVS, VM, OS/2, DOS, OS/400, for example).

Node F is running SNA over TCP/IP Gateway for OS/2.

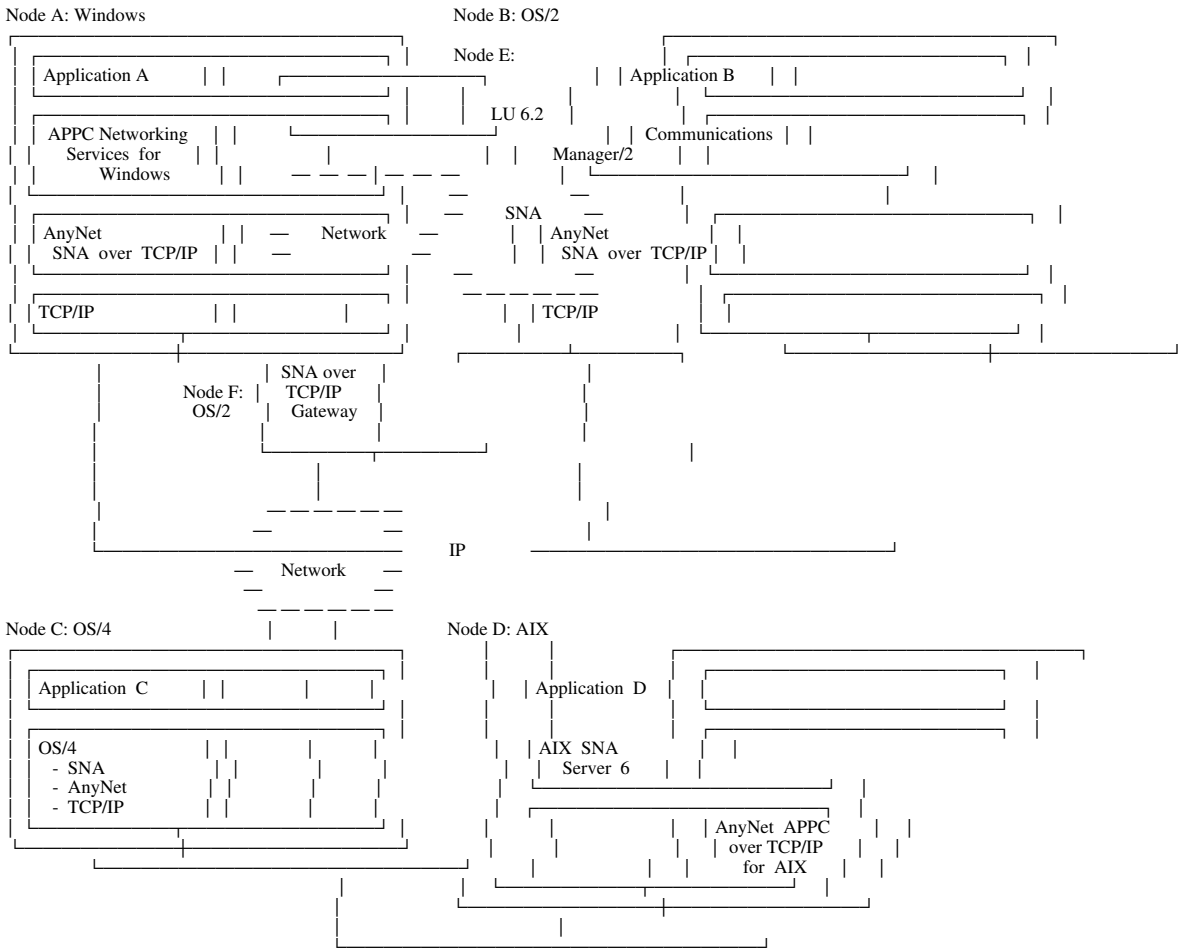


Figure 2. Configuration to Connect Windows, OS/2, OS/400, and AIX Workstations

Introduction

Connecting a Windows Workstation Running Personal Communications to a Subarea Network

Figure 3 shows a configuration that supports dependent LU communication for a downstream workstation using a subarea host:

Node A is running VTAM.

Node B is running Communications Server with the SNA over TCP/IP gateway function enabled. Node B is also configured as an SNA gateway.

Node C, a 3270 emulator, is running:

- IBM AnyNet SNA over TCP/IP for Windows
- IBM Personal Communications for Windows
- IBM TCP/IP for DOS

For an example showing the definitions required to implement the configuration in Figure 3, see “Establishing Communications between a Windows Workstation Running Personal Communications and a VTAM Host Processor” on page 42.

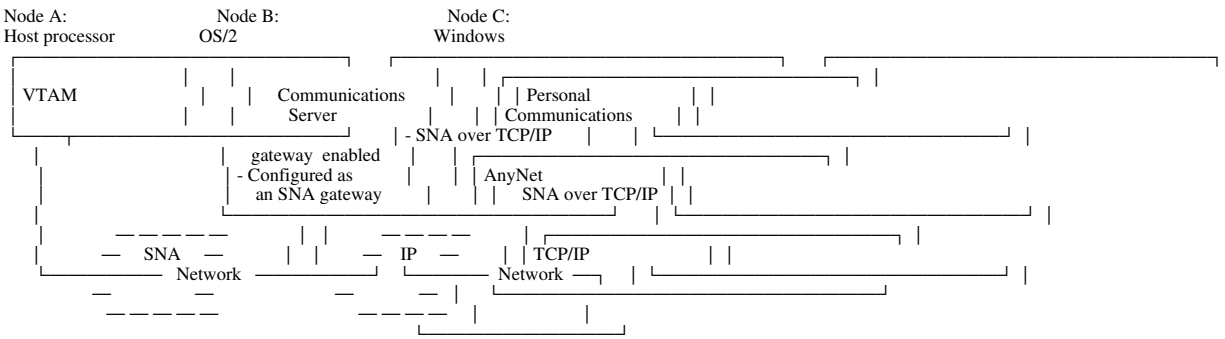


Figure 3. Configuration to Connect a Windows Downstream Workstation to a Subarea Network

Connecting a Windows Workstation Running Personal Communications to an APPN Network

Figure 4 shows a configuration that supports dependent LU communication for a downstream workstation using an APPN host:

Node A is running VTAM and is defined as a dependent LU server.

Node B, is running Communications Server with the SNA over TCP/IP gateway function enabled. Node B is also configured as both a dependent LU requester and an SNA gateway.

Node C, a 3270 emulator, is running the following:

- IBM AnyNet SNA over TCP/IP for Windows
- IBM Personal Communications for Windows
- IBM TCP/IP for DOS

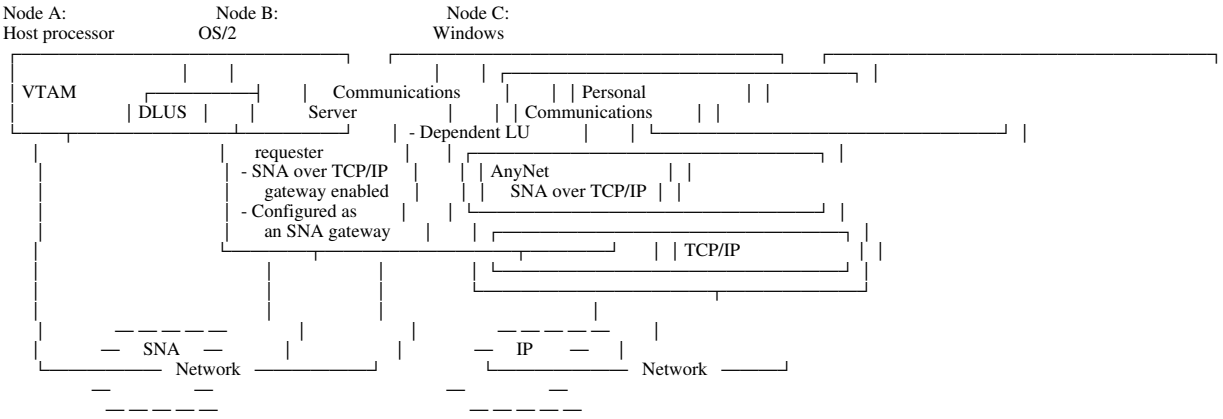


Figure 4. Configuration to Connect a Windows Downstream Workstation to an APPN Network

Introduction

Connecting a Windows Workstation Running Personal Communications to an AS/400 Host

Figure 5 shows a configuration that provides terminal display and printer support for a Windows workstation using an AS/400 host:

Node A, a 5250 emulator, is running:

- IBM Personal Communications for Windows
- IBM AnyNet SNA over TCP/IP for Windows
- IBM TCP/IP for DOS

Node B is an AS/400 host running OS/400. OS/400 includes the SNA, AnyNet, and TCP/IP components.

For an example showing the definitions required to implement the configuration in Figure 5, see “Establishing Communications between a Windows Workstation Running Personal Communications and an AS/400 Host” on page 44.

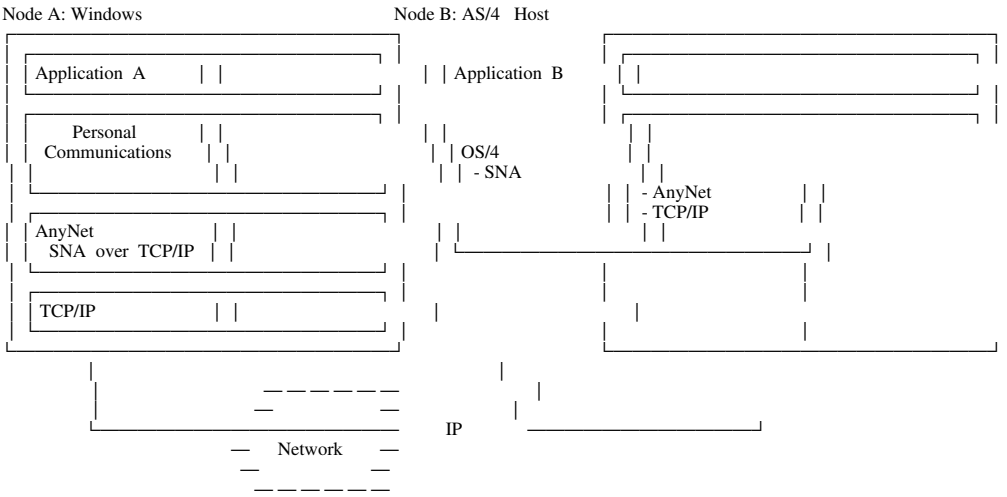


Figure 5. Configuration to Connect a Windows Workstation to an AS/400 Host

How Does SNA over TCP/IP Work?

Figure 6 and Figure 7 show the structure of a Windows node running SNA over TCP/IP for Windows.

As illustrated in Figure 6, CPI-C and APPC applications use APPC Networking Services for Windows to provide SNA transport user services.

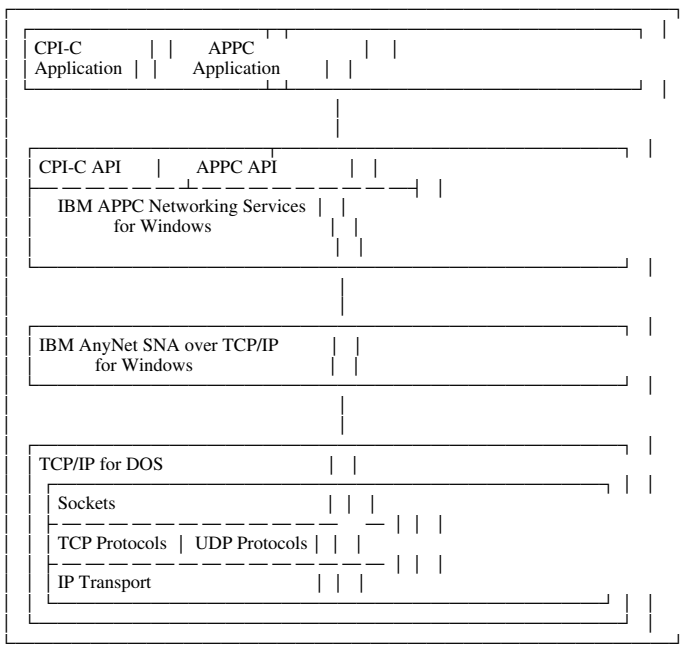


Figure 6. Structure of a Windows SNA-over-TCP/IP Node Running APPC Networking Services

As illustrated in Figure 7, dependent logical units (LUs) use Personal Communications to provide SNA transport user services.

Introduction

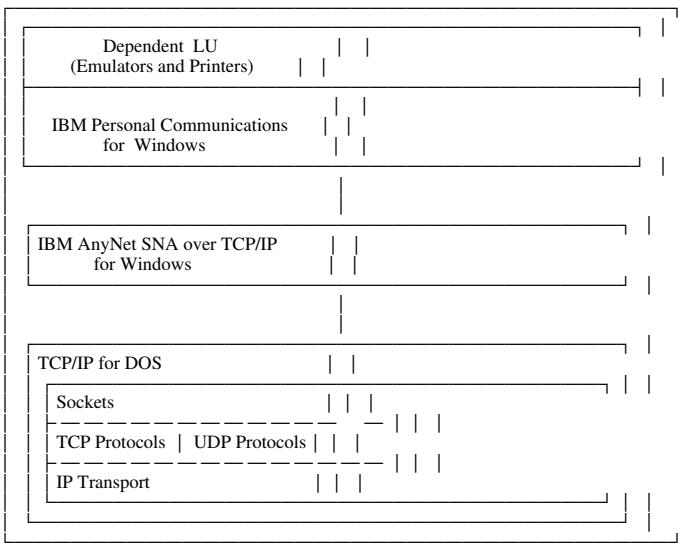


Figure 7. Structure of a Windows SNA-over-TCP/IP Node Running Personal Communications

Note: An SNA over TCP/IP node can run only one SNA transport user at a time. For configuration information, see "Selecting the SNA Transport User" on page 35.

To initiate a session:

CPI-C or APPC applications use the APPC Networking Services application program interface (API).

Dependent LUs, such as emulators and printers, use Personal Communications.

APPC Networking Services and Personal Communications use AnyNet SNA over TCP/IP for SNA path control.

When a SNA session is initiated, AnyNet translates the SNA routing information (network-qualified LU name) into IP routing information (IP address) and uses the IP address to create a TCP connection to the appropriate system. The configuration information that you define enables AnyNet to determine the IP address associated with the network-qualified LU name. See "How Does a Destination LU Get an IP Address?" on page 13 for additional information.

AnyNet uses both stream (TCP) and datagram (UDP) sockets that are bound to the well-known port (397). Any information received over this port is routed to AnyNet.

What Do You Need to Make SNA over TCP/IP Work?

This section describes the hardware and software needed to run SNA over TCP/IP for Windows.

Hardware Requirements

SNA over TCP/IP for Windows requires the following system hardware at a minimum:

386SX, or compatible, processor

6MB RAM (8MB recommended)

2MB of fixed-disk drive space

The product documentation and the Library Reader require additional disk space. Use the "Disk Space" push button during installation to determine whether there is adequate space on your hard disk to install the product documentation and the Library Reader.

If installed from diskettes, a 3.5-inch diskette drive (high-density)

If installed from a CD, a CD-ROM reader

Any network adapter supported by TCP/IP

Software Requirements

SNA over TCP/IP for Windows requires the following system software at a minimum:

Either of the following:

- Microsoft Windows, Version 3.1, running in Enhanced Mode
- Microsoft Windows for Workgroups, Version 3.1, running in Enhanced Mode

One of the following TCP/IP communications products:

- IBM TCP/IP for DOS, Version 2.1.1, with CSD UB10718
- FTP PC/TCP Network Software for DOS/Windows, Version 3.0
- FTP PC/TCP OnNet 1.1 for DOS/Windows (Windows VxD Kernel)
- Microsoft TCP/IP 3.11a for Windows for Workgroups, Version 3.11
- NetManage Chameleon TCP/IP for Windows, Version 4.5.1
- Novell LAN WorkPlace for DOS, Version 5.0

Additional software required by SNA over TCP/IP is determined by your SNA transport user:

For IBM APPC Networking Services, see "Additional Software Required for APPC Networking Services."

For IBM Personal Communications, see "Additional Software Required for Personal Communications" on page 10.

Additional Software Required for APPC Networking Services

If APPC Networking Services is selected as the SNA transport user, SNA over TCP/IP for Windows requires the following additional system software:

Introduction

IBM APPC Networking Services for Windows, Version 1.0, with CSD IP20310

If you connect to an AIX system running the APPC over TCP/IP AIX SNA Server/6000 AnyNet feature, Version 2.1.1, you must apply PTF U435033 to the AIX system.

If you connect to an AS/400 system running OS/400, Version 3.1, you must apply APAR MA10996 to the AS/400 system.

If you connect to an MVS system running the VTAM 3.4.2 AnyNet/MVS APPC over TCP/IP feature, you must apply PTF UW02865 to VTAM.

If you connect to an OS/2 system running AnyNet/2, Version 2.0, you must apply APAR IC08464 to the OS/2 system. This APAR is not needed if you are running AnyNet/2, Version 2.0.2.

For information about obtaining CSDs, PTFs, and APARs, contact IBM at 1-800-237-5511.

Additional Software Required for Personal Communications

If Personal Communications is selected as the SNA transport user, SNA over TCP/IP for Windows requires the following additional system software:

IBM Personal Communications, Version 4.1 for Windows

For 3270 support, Communications Server, Version 4.0

To enable 3270 support, you must have a host connection through a node that is running Communications Server with the following:

- The SNA-over-TCP/IP gateway function enabled.
- The node configured as an SNA gateway.

If the host connection is over an APPN network, the Communications Server node can also be defined as a dependent LU requester.

For more information on configuring the Communications Server node, see the documentation listed in “Communications Server Publications” on page 147.

For 5250 support, you must apply APAR MA10996 to the AS/400 system if the AS/400 system is running OS/400, Version 3.1.

For information about obtaining APARs, contact IBM at 1-800-237-5511.

Application Support

SNA over TCP/IP supports the following:

SNA over TCP/IP running on IBM APPC Networking Services for Windows supports:

- CPI-C application programs
- APPC application programs that use the Client Access/400 for Windows 3.1 Communications Router application program interface (API)

For information on CPI-C APIs, see the *CPI-C 2.0 Reference*. For information on APPC APIs, see the *API and Technical Reference*.

SNA over TCP/IP for Windows running on IBM Personal Communications for Windows provides 3270 and 5250 terminal display and printer support.

For information on Personal Communications, see *IBM Personal Communications: Up and Running for Windows*. See "Personal Communications Publications" on page 148 for more reference information.

Network Interface Protocols

All network interface protocols supported by TCP/IP for DOS (IEEE 802.3 LAN, for example) can be used with SNA over TCP/IP. Refer to your TCP/IP documentation to determine which network interface protocols are supported. If you are using IBM TCP/IP, see *IBM TCP/IP Version 2.1.1 for DOS: Installation and Administration*.

TCP/IP Support Services

SNA over TCP/IP uses the following support services provided by TCP/IP:

Windows socket application program interface (API), Version 1.1
Transmission Control Protocol (TCP) support
User Datagram Protocol (UDP) support
Internet Protocol (IP) support

Restriction on Using One SNA Transport User

Select either APPC Networking Services or Personal Communications as your SNA transport user. AnyNet can use only one SNA transport user at a time. For information on selecting and changing your SNA transport user, see "Selecting the SNA Transport User" on page 35.

Chapter 2. Planning for SNA over TCP/IP for Windows

This chapter describes planning considerations for SNA over TCP/IP, and contains the following sections:

"Address Mapping Overview"
"Planning Considerations for Using the Personal Communications Emulator Package" on page 15

Address Mapping Overview

This section describes the concepts and methods of address mapping, and includes the following sections:

"How Does a Destination LU Get an IP Address?"
"Defining Domain Names and IP Addresses" on page 14

How Does a Destination LU Get an IP Address?

The following steps outline how a destination LU gets an IP address:

1. SNA over TCP/IP receives the SNA LU name in the format netid.luname.
2. SNA over TCP/IP takes the LU name (netid.luname) and generates a domain name in the format luname.netid.snasuffix.

   Note: The domain name must be added to the HOSTS file or the domain name server database. See "Defining Domain Names and IP Addresses" on page 14 for additional information.

   Figure 8 shows an example of a domain name. For additional information on the SNA domain name suffix, see "SNA Domain Name Suffix" on page 36.

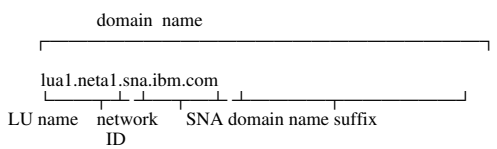


Figure 8. Format of the Domain Name That SNA over TCP/IP Builds

3. SNA over TCP/IP requests that the domain name be translated into an IP address.
4. TCP/IP uses the HOSTS file or the domain name server to translate the domain name into an IP address (for example, 9.67.192.28).

© Copyright IBM Corp. 1996

Planning

Defining Domain Names and IP Addresses

This section discusses the different methods that can be used to provide address mapping.

Host Name

The first step in the definition process is to add the host name to the TCP/IP HOSTS file or domain name server database. SNA over TCP/IP uses the host name to locate the IP address of the local node.

During your TCP/IP installation, the host name is added to one of the following places:

On the HOSTNAME environment variable which is located in the AUTOEXEC.BAT file

In the TCPDOS.INI file which is located in default directory \TCPDOS\ETC

HOSTS File

You can use the TCP/IP HOSTS file to define domain names and IP addresses for your network. However, as your network becomes larger and maintaining the HOSTS file on each end user workstation becomes too time consuming, it is recommended that you use the domain name server.

The location of the HOSTS file is defined by the ETC environment setting in your AUTOEXEC.BAT file. The default is \TCPDOS\ETC.

The HOSTS file in the ETC directory lists:

IP address
Domain name
Host name
Other aliases for the domain name

When you enter the information in the HOSTS file, include the IP address and the domain name in the format *luname.netid.snasuffix*.

For example, if your IP address is 9.67.192.28, network ID is NETA1, LU name is LUA1, and SNA domain name suffix is SNA.IBM.COM, define the following:

```
9.67.192.28     lua1.neta1.sna.ibm.com
```

Domain Name Server

Domain names and IP addresses can also be defined in the domain name server database.

Each destination LU name is mapped to a corresponding IP address by a domain name server. On your local system, you should have a RESOLV file in the ETC subdirectory that points to the domain name servers that handle this mapping for you.

SNA over TCP/IP uses the TCP/IP name resolution function. This function uses a RESOLV file to locate the domain name server. If the domain name server is not accessible or the name cannot be found, TCP/IP searches the HOSTS file.

Note: When running SNA applications over TCP/IP, be aware that there are restrictions on the characters used in LU names. For an explanation of the restrictions, see "SNA Domain Name Suffix" on page 36.

For information on the HOSTS file, the domain name server, and the Domain Name System (DNS), refer to your TCP/IP documentation. If you are running IBM TCP/IP for DOS, refer to the following:

IBM TCP/IP Version 2.1.1 for DOS Installation and Administration
IBM TCP/IP Version 2.1.1 for DOS User's Guide

Planning Considerations for Using the Personal Communications Emulator Package

This section provides a brief overview of planning considerations and requirements for using the Personal Communications emulator package.

Getting Started

Before configuring a session on your Windows workstation, do the following:

Determine the address mapping method.

Use a HOSTS file or domain name server database to map LU names to IP addresses. For more information, see "Address Mapping Overview" on page 13.

Check your TCP/IP documentation to ensure that the value of the FILES parameter in your CONFIG.SYS file is sufficient.

If the FILES parameter value is too low, AnyNet error message ANY0012E is issued with error code 0000000C-00090001. This message indicates that the local resource limit has been exceeded because no more sockets are available for creating connections or sending datagrams.

For a 3270 session, obtain the following values:

- CP name of the Communications Server gateway node

  The CP name of the gateway node is assigned by your network administrator and must be defined in Communications Manager. For more information, see the Personal Communications configuration tool online help or the documentation listed in "Communications Server Publications" on page 147.

- Block ID (IDBLK) and Physical Unit ID (IDNUM) values for the Windows workstation

  IDBLK and IDNUM values are assigned by your network administrator. For more information, see the Personal Communications configuration tool online help.

- SNA domain name suffix (SNASUFFIX)

  The SNA domain name suffix for your workstation is determined by your network administrator. The default SNASUFFIX is sna.ibm.com. For more information, see "SNA Domain Name Suffix" on page 36.

Planning

For a 5250 session, obtain the following values:

- System location name

  The system location name is assigned by your network administrator, and is the network ID and CP name of the AS/400 host to which this workstation will be connected.

- PC location name

  The PC location name is assigned by your network administrator, and is the network ID and LU name of this workstation.

For more information, see the Personal Communications configuration tool online help or the documentation listed in “Personal Communications Publications” on page 148.

How to Configure a Session

To configure a session on your Windows workstation, use the Personal Communications configuration tool to create a workstation profile:

1. Double-click on the “Start/Configure Session” icon in the Personal Communications folder.

2. Select your configuration options. Possible options for AnyNet are:

   PC
   LAN
   TCP/IP Anynet
   S/390 (3270 session) **or** AS/400 (5250 session)

3. Click on the “Configure...” push button. The “Customize Communication” window is displayed.

   To configure a 3270 session:

   a. In the “Link Parameters” section, click on the “Configure Link...” push button. The “AnyNet” window is displayed.

   b. Enter the following values:

Block ID	IDBLK value
Physical Unit ID	IDNUM value
OS/2 Gateway Name	CP name of the gateway

   For more information, see “Getting Started” on page 15.

   c. Click on the “OK” push button to save your workstation profile.

For an example, see “Establishing Communications between a Windows Workstation Running Personal Communications and a VTAM Host Processor” on page 42.

To configure a 5250 session:

a In the “Session Parameters” section, enter the following names:

System Location Name	Network ID and CP name of the AS/400 host in the format *netid.cp_name*
PC Location Name	Network ID and LU name of the local workstation in the format *netid.lu_name*

For more information, see “Getting Started” on page 15.

b Click on the “OK” push button to save your workstation profile.

For an example, see “Establishing Communications between a Windows Workstation Running Personal Communications and an AS/400 Host” on page 44.

Chapter 3. Installing, Updating, and Removing SNA over TCP/IP for Windows

This chapter describes how to install, update, and remove AnyNet SNA over TCP/IP from a workstation or code server, and contains the following sections:

"Getting Started"
"Installing SNA over TCP/IP on a Workstation" on page 21
"Installing SNA over TCP/IP on a Code Server" on page 23
"Verifying Additions and Changes Made during Installation" on page 27
"Updating SNA over TCP/IP on a Workstation or Code Server" on page 30
"Removing SNA over TCP/IP from a Workstation" on page 31
"Removing SNA over TCP/IP from a Code Server" on page 32

Getting Started

Before you install SNA over TCP/IP:

1. Ensure that all required software is installed.

   See "Software Requirements" on page 9 for a list of required software.

2. Move as many programs and device drivers into upper memory (640K-1MB) as you can.

   Your level of DOS determines what you can do to maximize the use of the high memory area (HMA) and enable as much upper memory block (UMB) space as possible. Refer to your DOS documentation for information on memory management.

3. Enable virtual memory.

   AnyNet performance may be slow if you are not running with virtual memory enabled. To enable virtual memory:

   a. Double-click on the "Control Panel" icon in the "Main" Windows program group.

   b. Double-click on the "386 Enhanced" icon in the "Control Panel" window.

   c. In the "386 Enhanced" window, click on "Virtual Memory" to view your current settings.

   d. In the "Virtual Memory" window:

      If "Type" is "None," go to Step 3e .
      If "Type" is "Permanent" or "Temporary," you do not need to change the setting.

   e. To change the "Type" setting, click on "Change."

© Copyright IBM Corp. 1996

f. From the “Type” pull-down menu, select either “Permanent” or “Temporary.” When a selection is made, a recommended “Size” is displayed based on your system.

4 Check the README.ANY file that is shipped on the first product diskette, product CD, or code server for any special instructions.

On a code server, this file is located in the APIPAW10 subdirectory of the directory provided by your server administrator.

5 Gather the following information:

Identify the directory in which SNA over TCP/IP is to be installed. The default directory is C:\ANYNET.

Identify your SNA transport user. For more information, see “Selecting the SNA Transport User” on page 35.

Identify the SNA domain name suffix for your workstation. Contact your network administrator to obtain an SNA domain name suffix. See “SNA Domain Name Suffix” on page 36 for a description of the SNA domain name suffix environment setting.

If you are planning to install the online documentation and either BookManager READ or the Library Reader is already installed, identify the directory in which the viewer is installed.

6 If you install SNA over TCP/IP Version 1.0 over a prior version of APPC over TCP/IP for Windows, it is recommended that you delete the prior version of APPC over TCP/IP for Windows before installing SNA over TCP/IP Version 1.0.

Attention

If you do not delete the prior version of APPC over TCP/IP for Windows, both APPC over TCP/IP and SNA over TCP/IP will appear in the list of “Installed Products” in the “View” menu bar option of the AnyNet installation program.

If either product is later deleted using the AnyNet installation program, the product files for both products are deleted. Therefore, it is recommended that you delete the prior version of APPC over TCP/IP before installing SNA over TCP/IP, Version 1.0.

To delete your current APPC over TCP/IP functions using the SNA over TCP/IP Version 1.0 diskettes:

Note: If you are using a CD, refer to your base package documentation for instructions. If you are using a code server, enter the commands provided by the server administrator.

a Insert the diskette labeled AnyNet Diskette 1 in your A: drive, and run the install command from either the DOS prompt or in the Windows “Program Manager” window:

From the DOS prompt, type:

```
win  a:\install
```

In the Windows "Program Manager" window:

1) Select "File" on the menu bar and click on the "Run" option.
2) In the "Run" window, type a:\install on the command line.
3) Click on the "OK" push button.

b In the "Installation and Maintenance" window, select the "View" menu bar option. Click on "Installed Products" to view the AnyNet products that are installed on your system.

c Select the APPC over TCP/IP product.

d Select the "Action" menu bar option. Click on "Delete" to remove the product files from your system.

e When the "Delete" action is complete, shut down and restart your system.

Installing SNA over TCP/IP on a Workstation

Use the following procedures to install the SNA-over-TCP/IP product, online and printable (PostScript) documentation, and the Library Reader.

To use the online help during installation, click on the "Help" push button.

1 **To install from diskettes:** Insert the diskette labeled AnyNet Diskette 1 in your A: drive, and run the install command from either the DOS prompt or in the Windows "Program Manager" window:

From the DOS prompt, type:

```
win  a:\install
```

In the Windows "Program Manager" window:

a. Select "File" on the menu bar and click on the "Run" option.
b. In the "Run" window, type a:\install on the command line.
c. Click on the "OK" push button.

The AnyNet README file is displayed in the "Information" window. To continue the installation, click on the "Continue" push button.

The "AnyNet Installation" and "Installation and Maintenance" windows are displayed.

To install from a CD: Refer to your base package documentation for installation instructions.

To install from a code server: Run the commands provided by your server administrator. For information on setting up a code server, see "Setting Up a Code Server" on page 23.

Installing

2 Select “AnyNet Workstation Setup” in the “Installation and Maintenance” window.

3 Select “Action” on the menu bar and click on the “Install...” option. The “Install” window is displayed.

4 Optionally, you can deselect the “Update CONFIG.SYS/AUTOEXEC.BAT” option. If you deselect this option, any changes that should be made to the AUTOEXEC.BAT file are placed in the AUTOEXEC.ADD file.

> **Attention**
>
> When the installation is complete, make the changes manually **before** you shut down and restart your system.

5 Optionally, you can select the “Overwrite files” option. If you do not select this option, the installation utility prompts you if it needs to overwrite an existing file on your system.

6 Click on the “OK” push button to continue the installation. The “Install - directories” window is displayed.

7 Select the component(s) that you want to install. Choices include:

The SNA-over-TCP/IP product
SNA-over-TCP/IP online and printable (PostScript) documentation
Library Reader for viewing the online documentation

Components that are not selected can be installed later using this same procedure.

8 Either accept the default directory, C:\ANYNET, for the specified component(s) or type another directory in the space provided.

When you install an AnyNet product, the ANYNETPATH environment variable is set in your AUTOEXEC.BAT file. This variable is required and is used by AnyNet to point to the directory where the product is installed.

9 Use the “Disk Space” push button to look at the amount of unused space on each workstation hard disk. To change the default drive C:, return to the “Install - directories” window and use the default directory entry field.

10 Click on the “Install...” push button at the bottom of the “Install - directories” window to start the installation. The “Install - progress” window is displayed and shows the installation status. To cancel the installation, click on the “Stop” push button.

During the installation of the SNA-over-TCP/IP product, the “AnyNet Configuration” window is displayed.

a. Click on the “SNA Transport User” option to select either APPC Networking Services or Personal Communications as your SNA transport user. For more information, see “Selecting the SNA Transport User” on page 35.

b. To change any of the environment settings, see “Configuring SNA-over-TCP/IP Environment Settings” on page 36.

When you are finished using the configuration tool, click on the “OK” push button to continue the installation.

11 When the installation is complete, exit the installation utility by pressing F3 twice or by closing the installation program windows.

12 If you deselected the “Update CONFIG.SYS/AUTOEXEC.BAT” option in step 4 on page 22, update your AUTOEXEC.BAT file manually before you shut down and restart your system. Use the AUTOEXEC.ADD file as a guide.

13 After installing the SNA over TCP/IP product, shut down and restart your system.

If you need to remove the SNA-over-TCP/IP product, documentation, or Library Reader from your system, see “Removing SNA over TCP/IP from a Workstation” on page 31.

Installing SNA over TCP/IP on a Code Server

If you are installing SNA over TCP/IP on multiple workstations, you can set up a code server on your local area network (LAN) to simplify the installation process. You can use NetView Distribution Manager/2 (NVDM/2) to automatically install AnyNet on multiple workstations, or users can install the product on their own workstations from the code server.

Setting Up a Code Server

This procedure copies installation utilities and the product images of SNA over TCP/IP from the product diskettes or product CD to the code server. To set up a local or remote code server:

1 Insert the diskette labeled AnyNet Diskette 1 in your A: drive, and run the install command from either the DOS prompt or in the Windows “Program Manager” window:

From the DOS prompt, type:

```
win a:\install
```

In the Windows “Program Manager” window:

a. Select “File” on the menu bar and click on the “Run” option.
b. In the “Run” window, type a:\install on the command line.
c. Click on the “OK” push button.

The “AnyNet Installation” and “Installation and Maintenance” windows are displayed.

Installing

Note: If the AnyNet product is shipped on a CD as part of a package, refer to your base package documentation for instructions.

2 Select “AnyNet Code Server Setup” in the “Installation and Maintenance” window.

3 Select “Action” on the menu bar and click on the “Install...” option. The “Install” window is displayed.

4 Optionally, you can select the “Overwrite files” option. If you do not select this option, the installation utility prompts you if it needs to overwrite an existing file on your system.

5 Click on the “OK” push button to continue the installation. The “Install - directories” window is displayed.

6 Select the component(s) that you want to install. Choices include:

The SNA-over-TCP/IP product
SNA-over-TCP/IP online and printable (PostScript) documentation
Library Reader for viewing the online documentation

Components that are not selected can be installed later using this same procedure.

7 Specify the directory in which you want to put your selections. The directory can be either a local or a remote directory.

Note: If Windows is not installed on your code server, run the AnyNet installation program on a Windows client machine and specify a remote directory which resides on your code server.

8 Use the “Disk Space” push button at the bottom of the window to determine whether the directory contains adequate space for the items that you have selected.

9 Click on the “Install” push button at the bottom of the window to start the installation.

10 When the installation is complete, exit the installation utility by pressing F3 twice or by closing the installation program windows.

Using the CID Setup Tool to Automate Installation

The AnyNet configuration, installation, and distribution (CID) setup tool allows you to create a response file that can be used for automating an install, restore, or delete of an AnyNet product from a code server. You can also use this tool to revise an existing response file.

To automate installation, you must first create a response file. A response file contains information used by the installation program to automatically install and configure an individual workstation.

Creating a Response File

To create a response file:

1. Start the AnyNet CID setup tool using one of the following methods:

   Double-click on the "AnyNet CID Setup" icon in the "AnyNet" program group on your code server.

   Type the following command at the DOS prompt:

   win *X:\IMG\ANYNET*\anycfgw cid

   For *X:\IMG\ANYNET* enter the AnyNet code server directory.

2. Select the "Product" to be installed using this response file.

3. Select the "Action" to be performed:

   If this is the first installation of the product on the workstations, select the "Install" action.

   a. Either accept the default AnyNet directory or type another directory in the space provided.

   b. For the SNA-over-TCP/IP product, click on the "Configure..." push button to specify the environment settings for the workstations.

   To install an APAR or ServicePak for AnyNet, select the "Update" action. Choose whether or not to save a backup copy of the current level of the product on the workstations before it is updated.

   To undo an update of the product, select the "Restore" action. This can only be done if a backup copy was saved on the workstations during the update.

   To automatically delete the entire product or just the backup saved on the workstations during an update, select the "Delete" action. Specify the object you would like to delete.

4. When you are finished and ready to save the file:

   a. Select "File" on the menu bar.

   b. Click on the "Save as..." option.

   c. Complete the fields for file name, file type, and drive.

   d. When you are finished, click on the "OK" push button to save and exit the window.

   For a detailed description of the keywords generated in the response file, see "Response File Keywords" on page 91.

5. After creating your response file:

Installing

If you are using NVDM/2, go to “Creating a Change File Profile for NetView Distribution Manager/2” on page 26.

If you are not using NVDM/2, users on the LAN can now install SNA over TCP/IP on their workstations by running the “Automatic AnyNet Install” program, AUTOINST.EXE:

a. Provide users with the name of the AnyNet code server directory where the AUTOINST.EXE program is located. This is the directory you specified using the procedure described in “Setting Up a Code Server” on page 23.

b. Provide users with the name of the response file and its location on the code server. The AUTOINST.EXE program will allow users to select this response file.

c. Inform users to run the AUTOINST.EXE program either from the DOS prompt or in the Windows “Program Manager” window:

   – From the DOS prompt, users type:

     win *X:\IMG\ANYNET*\AUTOINST.EXE

     For *X:\IMG\ANYNET*, the user enters the LAN-accessed directory where the product images are located.

   – In the Windows “Program Manager” window, users do the following:

     1) Select “File” on the menu bar and click on the “Run” option.
     2) In the “Run” window, type:

        X:\IMG\ANYNET\AUTOINST.EXE

        For *X:\IMG\ANYNET*, the user enters the LAN-accessed directory where the product images are located.
     3) Click on the “OK” push button.

Creating a Change File Profile for NetView Distribution Manager/2

If you are using NVDM/2 to automate the installation of SNA over TCP/IP, you must create a change file.

A sample change file profile is provided in *X:\IMG\ANYNET*\anynetw.pro, where *X:\IMG\ANYNET* is the directory you specified using the procedure described in “Setting Up a Code Server” on page 23. Make a copy of the sample profile and modify the parameters under “Section Install.” The sample profile includes commented instructions.

For more information on change file profile parameters, see “Parameters for the AnyNet INSTALLW.EXE and ANYNVDMD.EXE Programs” on page 93. For more information on the format of the change file profile, see the *IBM NetView Distribution Manager/2: Change Distribution Manager User's Guide*. This document also describes how to use a profile to create a change file.

Verifying Additions and Changes Made during Installation

This section describes the files, program group, and programs that are created when SNA over TCP/IP is installed.

File Created in the Windows Directory

The ANYNET.INI file is the only AnyNet file placed in the Windows directory. This file contains version information.

Files Created in the AnyNet Directory

To see the files that are created when you finish installation, go to the directory where the files are installed. If you chose the default directory, go to C:\ANYNET.

The following files should be in that directory:

ANYNET.CFN
ANYNET.NSW
ANYNET.PCS
WORKSETW.OLD

Note: Files in this directory with a file extension of .PKG are used by the installation program if you update or delete the product. These files should not be deleted or changed.

APIPAW Subdirectory

The following files should be in the \APIPAW subdirectory of the directory where SNA over TCP/IP is installed:

CHKTCPD.EXE
CHKWIN.EXE

BIN Subdirectory

The following files should be in the \BIN subdirectory of the directory where SNA over TCP/IP is installed:

ABINFO.EXE
ABINFO.HLP
ABINFO.MR0
ANYCFGW.EXE
ANYCFGW.INI
ANUTIL.DLL
ANY.MR0
ANYTRACE.DLL
ANYTRC.MR0
ANYVXD.386
APPENDIT.EXE
APIPAW.ADD
APIPAW.HLP
APIPAW.INI
CHECKSNA.EXE

EHNCL02.DLL
MAKESETW.EXE
MPTNCMM.DLL
NSDMPTN.DLL
RENAMEIT.EXE
TCPPMM.DLL
TRACE.DLL
WINTHRD.EXE

DOC Subdirectory

If you install the documentation and Library Reader, the following files should be in the \DOC subdirectory of the directory where SNA over TCP/IP is installed:

ADMDVSS.SYM
ADMUUARP.SYM
ADMUUCIP.SYM
ADMUUCRP.SYM
ADMUUCSP.SYM
ADMUUDRP.SYM
ADMUUFSS.SYM
ADMUUGEP.SYM
ADMUUGGP.SYM
ADMUUGIP.SYM
ADMUUKRF.SYM
ADMUUKRO.SYM
ADMUUKSF.SYM
ADMUUKSO.SYM
ADMUUMOD.SYM
ADMUUNSF.SYM
ADMUUNSO.SYM
ADMUUORP.SYM
ADMUUSHD.SYM
ADMUUSRP.SYM
ADMUUTIP.SYM
ADMUUTRP.SYM
ADMUUTSS.SYM
ANYBOOKW.ICO
ANYNETW.BKS
ANYSHLFW.EXE
APIPAW10.BOO
APIPAW10.PS
BOOKMGR.PIF
CHKREADW.EXE
EOZ2LM03.BOO
EOZGDF.EXE
EOZGHELP.NLS
EOZGRIP.COM
EOZGX06.EXE
EOZGX10.EXE

EOZHELP.HLP
EOZSHOW.EXE
EOZSHOW.HLP
RDPCSI.EXE
READIBM.EXE
READIBM.PIF
READIBM.PRO
SHLFSI.EXE

MISC Subdirectory

The following file should be in the \MISC subdirectory of the directory where SNA over TCP/IP is installed:

ANYTRACE.CFG

Program Group and Programs Created

When you finish installing the SNA-over-TCP/IP product, an "AnyNet" program group is added under the Windows Program Manager. This program group includes:

"AnyNet Configuration Tool" program

This program runs the AnyNet configuration tool. For more information, see "Using the AnyNet Configuration Tool" on page 35.

"AnyNet Books" program

This program runs the Library Reader or BookManager READ which enables you to view the online documentation.

"Read Me" icon

Double-click on this icon to view the README file for SNA over TCP/IP.

System File Changes

The following files are changed during the installation of the SNA-over-TCP/IP product:

AUTOEXEC.BAT

ANYNETPATH=*c:\anynet* is added.

c:\anynet\BIN is appended to the PATH statement.

New statements are added to set the SNA-over-TCP/IP configuration environment variables. See "Configuring SNA-over-TCP/IP Environment Settings" on page 36 for more information on environment variables.

c:\anynet is the directory where SNA over TCP/IP is installed. The default is C:\ANYNET.

SYSTEM.INI

In the "386Enh" section, the following statement is added:

DEVICE=*c:\anynet*\BIN\ANYVXD.386.

c:\anynet is the directory where SNA over TCP/IP is installed. C:\ANYNET is the default.

Updating

APPC Networking Services Changes

If you select APPC Networking Services as your SNA transport user, the AnyNet configuration file (ANYNET.CFN) is modified with the appropriate AnyNet settings for use with APPC Networking Services. The following files are also changed:

NSD.INI This file is located in your base Windows directory. Changes are:

In the "Configuration" section, the data link control type (DLCTYPE) is set to MPTN. For more information on restrictions related to DLCTYPE, see "Is DLCTYPE Set to MPTN?" on page 59.

In the "Configuration" section, EHNAPPCNonblocking=TRUE is added so that the APPC interface can be used with APPC Networking Services for Windows.

An "MPTN" section is added with the variable MPLD=ANYLINK. For more information on the MPLD variable, see the Networking Services *Configuration and Parameters Reference*.

NSTRACE.CFG This file is located in your base APPC Networking Services for Windows directory.

A line is added pointing to the ANYTRACE.CFG file in the directory where SNA over TCP/IP is installed.

Personal Communications Changes

If you select Personal Communications as your SNA transport user, the AnyNet configuration file (ANYNET.CFN) is modified with the appropriate AnyNet settings for use with Personal Communications.

Updating SNA over TCP/IP on a Workstation or Code Server

Use this option to update the SNA-over-TCP/IP product, online and printable (PostScript) documentation, or Library Reader, and to install an APAR or ServicePak.

1 **To update the SNA-over-TCP/IP product, documentation, or Library Reader from diskettes:** Insert the diskette labeled AnyNet Diskette 1 in your A: drive, and run the install command from either the DOS prompt or in the Windows "Program Manager" window:

From the DOS prompt, type:

```
win a:\install
```

In the Windows "Program Manager" window:

a. Select "File" on the menu bar and click on the "Run" option.
b. In the "Run" window, type a:\install on the command line.
c. Click on the "OK" push button.

To update the SNA-over-TCP/IP product, documentation, or Library Reader from a CD: Refer to your base package documentation for instructions.

To update the SNA-over-TCP/IP product, documentation, or Library Reader from a code server: Run the commands provided by your server administrator. The "AnyNet Installation" main window is displayed. For information on setting up a code server, see "Setting Up a Code Server" on page 23.

To install an APAR or ServicePak: Follow the instructions provided by IBM support services.

2 Select "AnyNet Workstation Setup" or "AnyNet Code Server Setup" in the "Installation and Maintenance" window.

3 Select "Action" on the menu bar and click on the Update option. The "Update" window is displayed.

4 If you do not want the installation utility to automatically update your AUTOEXEC.BAT file, remove the check mark from the "Update CONFIG.SYS/AUTOEXEC.BAT" option.

If you deselect this option, changes are placed in the AUTOEXEC.ADD file, and you must make the changes manually.

5 You can also mark the "Save a backup version?" option. This allows you to restore the previous version at a later date.

6 Click on the "Update" option.

7 When the update is complete, exit by pressing F3 twice or by closing the installation program windows.

8 Shut down and restart your system.

Removing SNA over TCP/IP from a Workstation

To remove SNA over TCP/IP from your workstation, do the following:

1 Stop SNA over TCP/IP if it is running. See "Stopping SNA over TCP/IP" on page 51 for additional information.

2 Insert the diskette labeled AnyNet Diskette 1 in your A: drive, and run the install command from either the DOS prompt or in the Windows "Program Manager" window:

From the DOS prompt, type:

```
win a:\install
```

In the Windows "Program Manager" window:

a. Select "File" on the menu bar and click on the "Run" option.

b. In the “Run” window, type a:\install on the command line.
c. Click on the “OK” push button.

Notes:

a. If you are using a CD, refer to your base package documentation for instructions.

b. If you are using a code server, enter the commands provided by your server administrator.

3 Select “AnyNet Workstation Setup” in the “Installation and Maintenance” window.

4 Select “Action” on the menu bar and click on the “Delete” option. The “Delete - progress” window is displayed and shows the status of the delete operation.

The product files are deleted from your fixed disk, and any related AnyNet program groups are removed from the Windows Program Manager.

All SNA-over-TCP/IP entries are also removed or disabled in your AUTOEXEC.BAT and SYSTEM.INI files.

If you selected APPC Networking Services as your SNA transport user, all SNA-over-TCP/IP entries are removed or disabled in your NSD.INI and trace configuration files.

5 Shut down and restart your system.

Removing SNA over TCP/IP from a Code Server

1 Insert the diskette labeled AnyNet Diskette 1 in your A: drive, and run the install command from either the DOS prompt or in the Windows “Program Manager” window:

From the DOS prompt, type:

```
win a:\install
```

In the Windows “Program Manager” window:

a. Select “File” on the menu bar and click on the “Run” option.
b. In the “Run” window, type a:\install on the command line.
c. Click on the “OK” push button.

Note: If the AnyNet product is shipped on a CD as part of a package, refer to your base package documentation for instructions.

2 Select “AnyNet Code Server Setup” in the “Installation and Maintenance” window.

3 Select “Action” on the menu bar and click on the “Delete...” option. The “Delete” window is displayed.

4 Select the components you want to delete and click on the “Delete” push button.

The product files that you chose for removal are deleted from the install directory on your fixed disk, and any related AnyNet program groups are removed from Program Manager.

Chapter 4. Configuring SNA over TCP/IP for Windows

This chapter describes how to configure SNA over TCP/IP using the AnyNet configuration tool. See Chapter 2, “Planning for SNA over TCP/IP for Windows” on page 13 before you configure SNA over TCP/IP.

This chapter includes the following sections:

“Using the AnyNet Configuration Tool”
“Selecting the SNA Transport User”
“Configuring SNA-over-TCP/IP Environment Settings” on page 36
“Configuring TCP/IP” on page 38
“Configuration Examples” on page 39

Using the AnyNet Configuration Tool

The AnyNet configuration tool is brought up during installation to allow you to select your SNA transport user and to change the default settings for environment variables.

During installation, a copy of the configuration tool is installed on your system. To reconfigure AnyNet after installation has completed, use **one** of the following methods:

Double-click on the “AnyNet” program group, then double-click on the “AnyNet Configuration Tool” icon.

Type the following command at the DOS prompt:

win *X:\ANYNET*anycfgw

For *X:\ANYNET*, enter the directory where SNA over TCP/IP for Windows is installed.

To use the online help during configuration:

1. Either click on the field or use the Tab key to move to the field.
2. Press the F1 key on your keyboard or select “Help” on the menu bar.

Selecting the SNA Transport User

Use the AnyNet configuration tool to select either APPC Networking Services or Personal Communications as your SNA transport user. AnyNet can use only one SNA transport user at a time. For an overview of support provided by each SNA transport user, see “What Does SNA over TCP/IP Do?” on page 1.

You can select your SNA transport user during installation or after installation has completed. By default, no SNA transport user is configured.

Note: SNA over TCP/IP will not function correctly until a SNA transport user is selected.

© Copyright IBM Corp. 1996

Configuring

During installation, the AnyNet configuration tool is automatically opened. After installation is completed, you can use the AnyNet configuration tool at any time by following the steps in "Using the AnyNet Configuration Tool."

To select or change the SNA transport user:

1. In the "AnyNet Configuration" window, click on the "SNA Transport User" option. The "SNA Transport User" window is displayed.
2. Select "APPC Networking Services" or "Personal Communications" in the pull-down menu.
3. Click on the "OK" push button to save your selection.

Configuring SNA-over-TCP/IP Environment Settings

Environment settings are the values for the environment variables in your AUTOEXEC.BAT file. It is recommended that the only variable you change is the SNA Domain Name Suffix. The default settings for the other environment variables are adequate for most installations.

This section describes the following environment variables:

"SNA Domain Name Suffix"
"Connection Retry Duration" on page 37
"Connection Wait Time Limit" on page 37
"Remote Node Inactivity Poll Interval" on page 37
"Unacknowledged Datagram Retry Interval" on page 38
"Unsent Datagram Retry Interval" on page 38

SNA Domain Name Suffix

environment variable SNASUFFIX
default SNA.IBM.COM

This field defines the SNA domain name suffix to be used when a domain name is created using the LU name and the network ID. For more information on address mapping, see "How Does a Destination LU Get an IP Address?" on page 13.

Consult your network administrator to obtain an SNA suffix for your MPTN network. If you change the SNA domain name suffix, use the following guidelines:

Specify this value in the format *ccc ... .ccc ... .ccc ...*, where *ccc ... .ccc ... .ccc ...*, is a user-defined domain name suffix in the hierarchical-naming format recognized by TCP/IP (for example, SNA.IBM.COM).

Because the generated domain name must be a valid IP domain name, follow these rules when specifying this value:

- The SNA domain name suffix must be less than or equal to 237 characters in length, derived as follows:

- An IP domain name can contain up to 255 characters.
- Subtract 18, which represents 8 characters for the LU name, 8 characters for the network ID, and 1 character for each period (.) delimiter.
- The difference is the 237-character limitation.

– Each label (the characters between the periods) in the SNA domain name suffix must:

- Start with a letter
- End with a letter or digit
- Contain only letters, digits, or the hyphen (-) character
- Be less than or equal to 63 characters in length

Note: Existing SNA LU names that do not follow this naming convention need to be changed before they are used in a domain name.

Connection Retry Duration

environment variable CONN_RETRY_SECS
default 300 seconds
range 1–65535

This field sets the maximum time, in seconds, that SNA over TCP/IP tries to set up an MPTN connection over TCP/IP. When an MPTN connection setup fails, AnyNet tries every IP address associated with an LU name in the domain name server until all the addresses are exhausted or until the time set in this field is reached.

Connection Wait Time Limit

environment variable CONNWAIT_SECS
default 30 seconds
range 1–65535

This field sets the maximum time, in seconds, that SNA over TCP/IP waits to receive an MPTN connection or connection response packet after the TCP connection is established. This interval limits the time the connecting node waits for a session partner to send a packet, if the partner node cannot be reached after the TCP connection is setup.

Remote Node Inactivity Poll Interval

environment variable INACTIVITY_TIMER_SECS
default 60 seconds
range 1–65535

This field sets the number of seconds of inactivity allowed between two partner nodes before SNA over TCP/IP tries to determine whether the partner node is still active. Setting this value below 10 seconds may seriously affect system performance.

To calculate how long it takes before an inactive partner is detected, do the following:

1. Take the value set for the maximum number of seconds to wait before retrying an unacknowledged datagram (UNACKED_DG_RETRY_SECS environment variable).

2. Multiply that value by 5.
3. Add this value (the number of seconds of inactivity allowed).

The resulting value is the approximate number of seconds it takes to detect an inactive partner.

Unacknowledged Datagram Retry Interval

environment variable UNACKED_DG_RETRY_SECS
default 30 seconds
range 1–65535

This field sets the time, in seconds, that SNA over TCP/IP waits to resend an unacknowledged MPTN out-of-band (OOB) and an MPTN keepalive datagram.

When expedited data is sent over TCP/IP, this value can be used to set timer intervals that help improve the delivery of expedited data in congested situations. In SNA, some control messages are sent as expedited data (for example, messages requesting the right to send data or messages taking down a session). Expedited data is not subject to congestion control and can move ahead of normal, non-expedited data. To assure delivery, SNA over TCP/IP might send expedited data as normal data and as an OOB datagram.

Unsent Datagram Retry Interval

environment variable UNSENT_DG_RETRY_SECS
default 3 seconds
range 1–65535

This field sets the time, in seconds, that AnyNet waits for an acknowledgment after sending expedited data on the TCP connection, before sending the data as an out-of-band (OOB) datagram.

When expedited data is sent, this value can be used to set timer intervals that help improve the delivery of expedited data in congested situations.

Configuring TCP/IP

Determine the address mapping method you will use for mapping LU names to IP addresses. This mapping is enabled using the TCP/IP HOSTS file, domain name server, or both. For additional information, see “Defining Domain Names and IP Addresses” on page 14.

IBM TCP/IP interfaces are configured using either the CUSTOM or IFCONFIG functions. For information on configuring IBM TCP/IP interfaces, refer to the following:

IBM TCP/IP Version 2.1.1 for DOS Installation and Administration
IBM TCP/IP Version 2.1.1 for DOS User's Guide

If you are using a non-IBM TCP/IP product, see your product documentation for configuration information.

Configuration Examples

This section includes the following examples:

For APPC Networking Services, see "Establishing Communications between Two Windows Workstations Running APPC Networking Services."

For Personal Communications, see:

- "Establishing Communications between a Windows Workstation Running Personal Communications and a VTAM Host Processor" on page 42
- "Establishing Communications between a Windows Workstation Running Personal Communications and an AS/400 Host" on page 44

Establishing Communications between Two Windows Workstations Running APPC Networking Services

Figure 9 on page 40 illustrates how to establish communications between two Windows nodes using APPC Networking Services as the SNA transport user:

Application A and B are APPC or CPI-C application programs.

Both Windows nodes are running:

- IBM APPC Networking Services for Windows
- IBM AnyNet SNA over TCP/IP for Windows
- IBM TCP/IP for DOS

Note: In this example, a TCP/IP HOSTS file is used for mapping LU names to IP addresses. For information on address mapping methods, see "Address Mapping Overview" on page 13.

Configuring

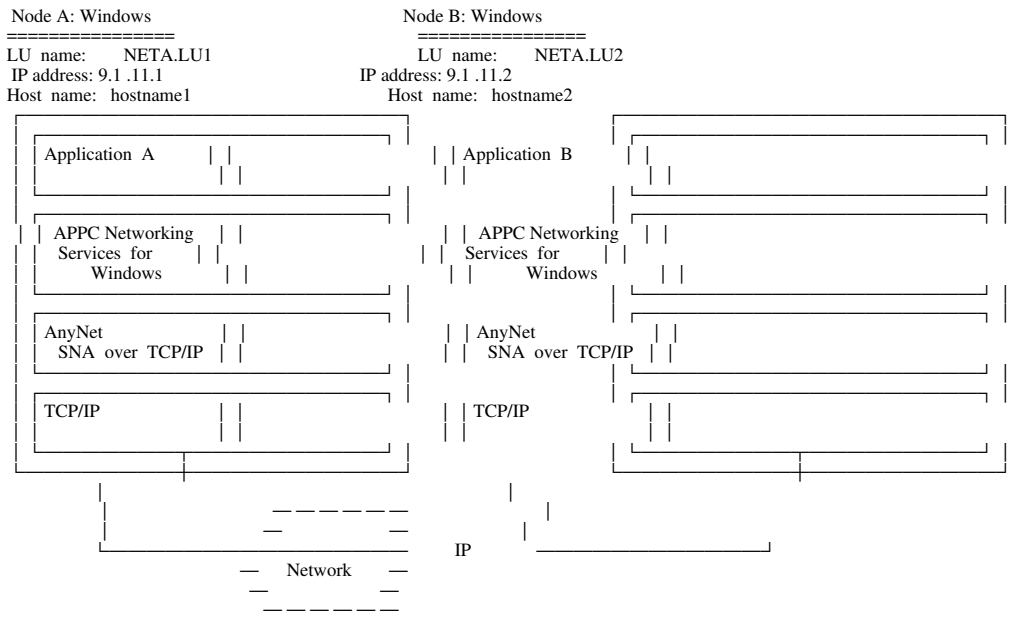


Figure 9. Address Assignments for Establishing Communication between Two Windows Workstations Running APPC or CPI-C Applications

Definitions

In this example, the following definitions are used to establish communications between the two Windows nodes:

Windows node A:

Local LU name	NETA.LU1 (defined in the APPC Networking Services NSD.INI file)
Host name	hostname1 (defined in TCP/IP)
IP address	9.10.11.1 (defined in TCP/IP)

Windows node B

Local LU name	NETA.LU2 (defined in the APPC Networking Services NSD.INI file)
Host name	hostname2 (defined in TCP/IP)
IP address	9.10.11.2 (defined in TCP/IP)

Steps

For each workstation, do the following:

1. Ensure that APPC Networking Services for Windows and TCP/IP for DOS are installed.

   For more information on software, see “Software Requirements” on page 9.

2. Install SNA over TCP/IP for Windows on both workstations.

3 Edit the TCP/IP HOSTS file on each machine.

For Node A, add the following entries in the local HOSTS file to enable communication with Node B:

```
9.1 .11.1  hostname1
9.1 .11.2  lu2.neta.sna.ibm.com
```

For Node B, add the following entries in the local HOSTS file to enable communication with Node A:

```
9.1 .11.2  hostname2
9.1 .11.1  lu1.neta.sna.ibm.com
```

4 On both workstations, configure Networking Services to use AnyNet by specifying MPTN as the data link control type (DLCTYPE) in the NSD.INI file.

For more information on DLCTYPE, see "Is DLCTYPE Set to MPTN?" on page 59.

5 Shut down and restart both workstations.

6 Start TCP/IP for DOS.

7 Start Windows.

8 Start APPC Networking Services for Windows. SNA over TCP/IP becomes active when the Networking Services LU is initialized.

See "Starting SNA over TCP/IP" on page 49 for information on starting Networking Services.

9 Use the ping command to verify that TCP/IP is operational. For more information, see "Checking TCP/IP" on page 55.

10 Use the netstat -an command to verify that SNA over TCP/IP is active. For more information, see "Checking SNA over TCP/IP for Windows" on page 62.

11 Start your APPC or CPI-C applications.

If problems occur, see Chapter 6, "Troubleshooting" on page 53 for a description of common problems and recommended actions. If you need additional assistance, see "Using IBM Support Services" on page 72.

Configuring

Establishing Communications between a Windows Workstation Running Personal Communications and a VTAM Host Processor

Figure 10 illustrates how to establish communications between a Windows workstation running the Personal Communications emulator package and a VTAM host processor:

Node A is running VTAM.

Node B is running Communications Server with the SNA over TCP/IP gateway function enabled. Node B is also configured as an SNA gateway.

Node C, a 3270 emulator, is running the following:

- IBM AnyNet SNA over TCP/IP for Windows
- IBM Personal Communications for Windows
- IBM TCP/IP for DOS

Notes:

1. In this example, a TCP/IP HOSTS file is used for mapping CP and LU names to IP addresses. For information on address mapping methods, see “Address Mapping Overview” on page 13.
2. For more information on planning considerations and configuration parameters for Personal Communications, see “Getting Started” on page 15.

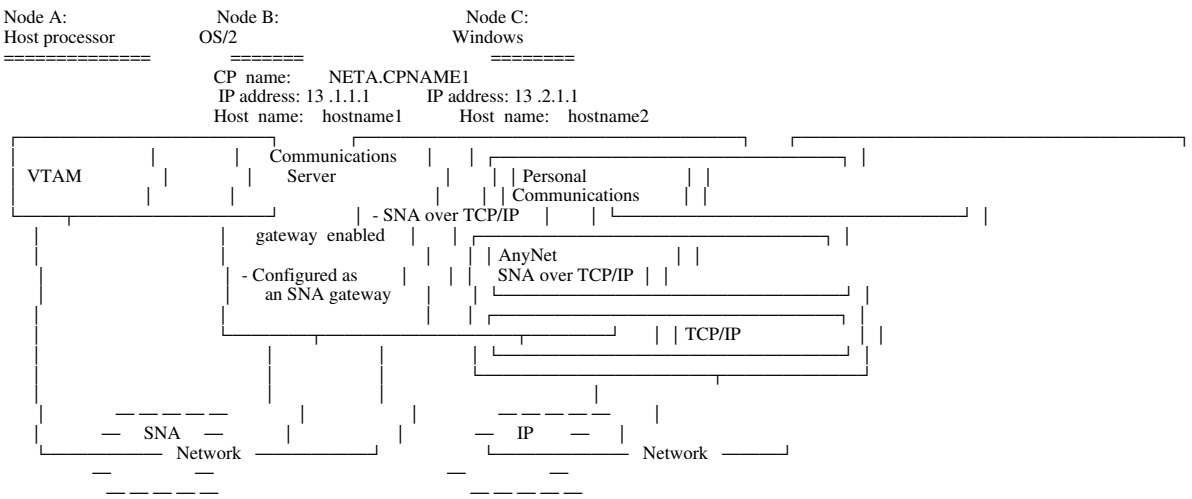


Figure 10. Address Assignments for Establishing Communication between a Windows Downstream Workstation and a VTAM Host Processor

Definitions

In this example, the following definitions are used to establish communications between the Windows workstation (Node C) and the Communications Server gateway (Node B). To enable communication between the VTAM host processor (Node A) and gateway

Node B, see the documentation listed in “Communications Server Publications” on page 147.

Windows Node C

Host name	hostname2 (defined in TCP/IP)
IP address	130.2.1.1 (defined in TCP/IP)
Block ID	123 (defined in Personal Communications)
Physical Unit ID	01020 (defined in Personal Communications)

OS/2 Node B (Communications Server)

Local CP name	NETA.CPNAME1 (defined in Communications Manager)
Host name	hostname1 (defined in TCP/IP)
IP address	130.1.1.1 (defined in TCP/IP)

Steps

1 For Windows Node C, do the following:

a Ensure that Personal Communications for Windows and TCP/IP for DOS are installed.

For more information on software, see “Software Requirements” on page 9.

b Install SNA over TCP/IP for Windows using the default values. Shut down and restart the workstation.

c To enable communication with OS/2 Node B, add the following entries in the local HOSTS file:

```
13 .2.1.1  hostname2
13 .1.1.1  cpname1.neta.sna.ibm.com
```

d Start TCP/IP for DOS.

e Start Windows.

f Create a workstation profile using the Personal Communications configuration tool:

1) In the Windows Program Manager window, select the “Personal Communications” program group.

2) Click on the “Start/Configure Session” icon. The “Customize Communication” window is displayed.

3) Select the following options:

PC
TCP/IP AnyNet
S/390

4) Click on the “Configure” push button.
5) Click on the “Configure Link” push button. The “AnyNet” window is displayed.
6) Enter the following values:

Block ID	123
Physical Unit ID	01020
OS/2 Gateway Name	NETA.CPNAME1

7) Click on “OK” to save the workstation profile.

2 For OS/2 Node B, add the following entry in the local HOSTS file to enable communication between Node B and Windows Node C:

```
13 .2.1.1  123 1 2 .sna.ibm.com
```

Note: IP address 130.2.1.1 is followed by the IDBLK (Block ID), IDNUM (Physical Unit ID), and SNASUFFIX (SNA domain name suffix) values. The default SNASUFFIX is sna.ibm.com.

3 Use the ping command to verify that TCP/IP is operational. For more information, see “Checking TCP/IP” on page 55.

4 Use the netstat -an command to verify that SNA over TCP/IP is active. For more information, see “Checking SNA over TCP/IP for Windows” on page 62.

5 Establish a connection between VTAM Node A and OS/2 Node B. A connection must exist between Nodes A and B before an emulator session can be started on Windows Node C.

6 Start Personal Communications on Windows Node C.

If problems occur, see Chapter 6, “Troubleshooting” on page 53 for a description of common problems and recommended actions. If you need additional assistance, see “Using IBM Support Services” on page 72.

Establishing Communications between a Windows Workstation Running Personal Communications and an AS/400 Host

Figure 11 on page 45 illustrates how to establish communications between a Windows workstation running the Personal Communications emulator package and an AS/400 host:

Node A, a 5250 emulator, is running:

- IBM Personal Communications for Windows
- IBM AnyNet SNA over TCP/IP for Windows
- IBM TCP/IP for DOS

Node B is an AS/400 host running OS/400. OS/400 includes the SNA, AnyNet, and TCP/IP components.

Notes:

1. In this example, a TCP/IP HOSTS file is used for mapping CP and LU names to IP addresses. For information on address mapping methods, see “Address Mapping Overview” on page 13.
2. For more information on planning considerations and configuration parameters for Personal Communications, see “Getting Started” on page 15.

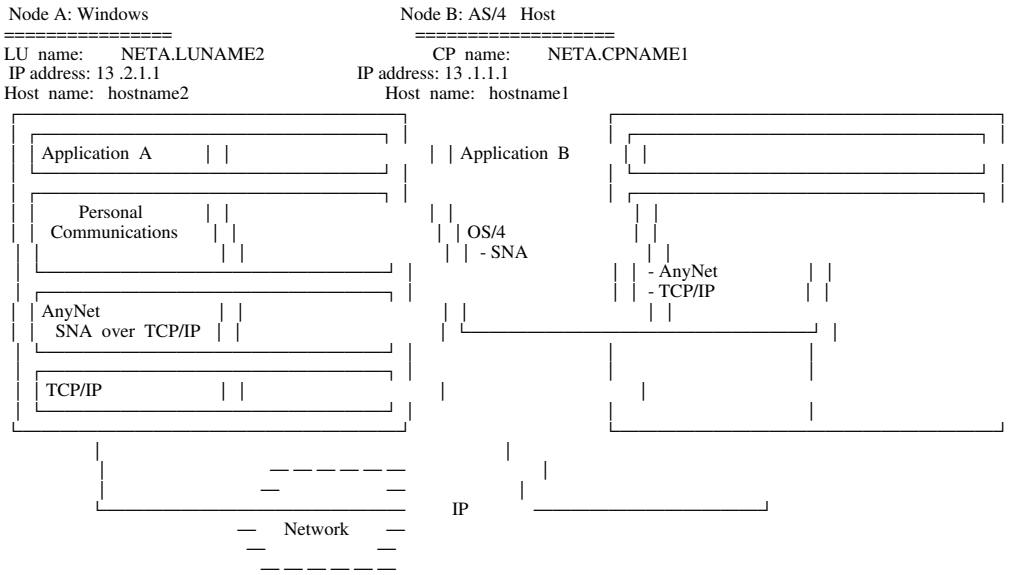


Figure 11. Address Assignments for Establishing Communication between a Windows Workstation and an AS/400 Host

Definitions

In this example, the following definitions are used to establish communications between the Windows workstation (Node A) and the AS/400 host (Node B).

Windows Node A

Local LU name	NETA.LUNAME2 (defined in Personal Communications)
IP address	130.2.1.1 (defined in TCP/IP)
Host name	hostname2 (defined in TCP/IP)

OS/400 Node B (AS/400 Host)

Local CP name	NETA.CPNAME1 (defined in AS/400)
IP address	130.1.1.1 (defined in TCP/IP)
Host name	hostname1 (defined in TCP/IP)

Configuring

Steps

1. For Windows Node A, do the following:

   a. Ensure that Personal Communications for Windows and TCP/IP for DOS are installed.

      For more information on software, see “Software Requirements” on page 9.

   b. Install SNA over TCP/IP for Windows using the default values. Shut down and restart the workstation.

   c. To enable communication with OS/400 Node B, add the following entries in the local HOSTS file:

      ```
      13 .2.1.1  hostname2
      13 .1.1.1  cpname1.neta.sna.ibm.com
      ```

   d. Start TCP/IP for DOS.

   e. Start Windows.

   f. Create a workstation profile using the Personal Communications configuration tool:

      1) In the Windows Program Manager window, select the “Personal Communications” program group.
      2) Click on the “Start/Configure Session” icon. The “Customize Communication” window is displayed.
      3) Select the following options:

         PC
         TCP/IP AnyNet
         AS/400

      4) Click on the “Configure” push button.
      5) In the “Session Parameters” section, enter the following names:

System Location Name	NETA.CPNAME1
PC Location Name	NETA.LUNAME2

      6) Click on “OK” to save the workstation profile.

2. For OS/400 Node B, add the following entries in the local HOSTS file to enable communication between Node B and Windows Node A:

   ```
   13 .1.1.1  cpname1.neta.sna.ibm.com
   13 .2.1.1  luname2.neta.sna.ibm.com
   ```

3. Use the ping command to verify that TCP/IP is operational. For more information, see “Checking TCP/IP” on page 55.

4 Use the netstat -an command to verify that SNA over TCP/IP is active. For more information, see “Checking SNA over TCP/IP for Windows” on page 62.

5 Start Personal Communications on Windows Node A.

If problems occur, see Chapter 6, “Troubleshooting” on page 53 for a description of common problems and recommended actions. If you need additional assistance, see “Using IBM Support Services” on page 72.

Chapter 5. Operating and Managing SNA over TCP/IP for Windows

This chapter includes the following sections:

"Starting SNA over TCP/IP"
"Stopping SNA over TCP/IP" on page 51
"Tools for Managing SNA over TCP/IP" on page 51

Starting SNA over TCP/IP

The SNA transport user you select determines how you start SNA over TCP/IP:

If APPC Networking Services is your SNA transport user, see "Using APPC Networking Services to Start SNA over TCP/IP."

If Personal Communications is your SNA transport user, see "Using Personal Communications to Start SNA over TCP/IP" on page 50.

Using APPC Networking Services to Start SNA over TCP/IP

To start SNA over TCP/IP, do the following:

1. If you have just installed SNA over TCP/IP, reboot your system. Otherwise, begin with step 2.
2. If you are using IBM TCP/IP, start TCP/IP for DOS, then start Windows.

   If you are using a non-IBM TCP/IP product, you might need to start Windows before TCP/IP. Refer to your TCP/IP documentation for instructions.
3. Start APPC Networking Services for Windows by using either of the following methods:

   In the "IBM APPC Networking Services" window, click on the "Autostart Networking Services" icon.

   For information on the autostart function, see the Networking Services publication, *Getting Started*.

   In the "IBM APPC Networking Services" window:

   a. Double-click on the "Advanced Operations" icon. The "Networking Services Advanced Operations" window is displayed.

   b. Select "Operation" on the menu bar and click on the "LU Control" option. The "Local LU Control" window is displayed.

   c. Click on the "Initialize LU" push button.

   SNA over TCP/IP is started and becomes active when the Networking Services LU is initialized.

© Copyright IBM Corp. 1996

Operating and Managing

Notes:

1. If there are initialization problems, APPC Networking Services for Windows displays a pop-up window and issues message 5239. Check the message log to determine the cause of the problem. See "Networking Services Message Log" on page 64 for additional information.
2. Under certain conditions during initialization, a pop-up window may be displayed by AnyNet or TCP/IP indicating that initialization was not successful. The most common message displayed in this situation is INET not running.

   You can click on the "OK" push button or press the enter key to clear the message even though the hourglass has not changed to an arrow pointer.

   To correct the error condition, exit Windows. This may require rebooting your workstation. Then restart SNA over TCP/IP using the steps in this section. If problems persist, contact IBM for service. See "Using IBM Support Services" on page 72 for information.

Use the APING program, which is supplied with Networking Services, to verify that your system can communicate with other APPC or CPI-C application programs over an IP network. See "Checking APPC Networking Services for Windows" on page 59 for additional information.

Using Personal Communications to Start SNA over TCP/IP

To start SNA over TCP/IP:

1 Double-click on a Personal Communications workstation profile icon in the "Personal Communications" program group.

   For information on how to create a workstation profile, see "How to Configure a Session" on page 16.

2 If you have selected the "Autostart" option, SNA over TCP/IP is started automatically.

   Otherwise, select the "Communication" option on the menu bar, and click on "Connect" to start SNA over TCP/IP.

Stopping SNA over TCP/IP

The SNA transport user you select determines how you stop SNA over TCP/IP:

If APPC Networking Services is your SNA transport user, see "Using APPC Networking Services to Stop SNA over TCP/IP."

If Personal Communications is your SNA transport user, see "Using Personal Communications to Stop SNA over TCP/IP."

Using APPC Networking Services to Stop SNA over TCP/IP

To stop SNA over TCP/IP:

1. Double-click on the Networking Services "Advanced Operations" icon. The "IBM APPC Networking Services" window is displayed.
2. Select "Operation" on the menu bar and click on the "LU Control" option.
3. Click on the "Terminate LU" button to stop SNA over TCP/IP.

Using Personal Communications to Stop SNA over TCP/IP

To stop SNA over TCP/IP:

1. Click on the "File" option on the menu bar.
2. Click on "Exit" to stop only this session. To stop all sessions, click on "Exit all."

Tools for Managing SNA over TCP/IP

This section describes tools that you can use to manage network communications among workstations and hosts, or among workstations only.

Managing the IP Network

Because SNA over TCP/IP operates as a TCP/IP application program, you can use Simple Network Management Protocol (SNMP) to provide network management of TCP/IP traffic that is routed through SNA over TCP/IP.

Because the NetView program provides SNMP support, you can use any of the following products to manage the IP network activity:

NetView Version 2 Release 1 (or later)
NetView for AIX Version 3 (or later), a feature of SystemView
NetView family of products

The underlying IP network is managed by NetView for AIX, a service point to the NetView program.

NetView for AIX provides the following TCP/IP management:

Generic support for Management Information Base 2 (MIB II) devices

Support for fault and performance applications
Support for Trouble Ticket/6000 applications
Support for TCP/IP routers and SNMP devices

For more information, see “NetView Publications” on page 148.

Managing the SNA Network

Because SNA over TCP/IP is a node in the SNA network, the NetView program can be used to provide network management of the SNA traffic that is routed through SNA over TCP/IP. For more information, see “NetView Publications” on page 148.

Chapter 6. Troubleshooting

This chapter describes what to do if you encounter problems when you are installing, configuring, or running SNA over TCP/IP for Windows.

If you are having network problems, check the status of your key products in this order:

1 TCP/IP

Note: Suggested actions, commands, and responses in this section apply to IBM TCP/IP for DOS. If you are using a non-IBM TCP/IP product, all of the information in this section may not apply. See your TCP/IP product documentation for additional information.

2 SNA transport user:

APPC Networking Services for Windows
Personal Communications for Windows

3 SNA over TCP/IP for Windows

"Troubleshooting Steps" on page 54 summarizes the troubleshooting procedure.

© Copyright IBM Corp. 1996

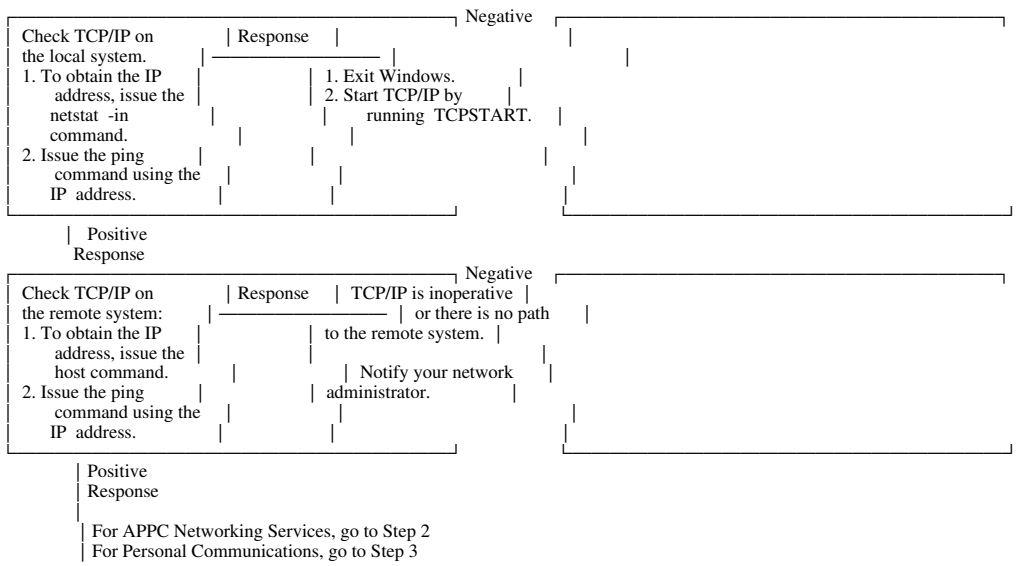
Troubleshooting Steps

The following flow chart illustrates one example of a step-by-step approach to isolating and solving network problems. Before using the steps in this flow chart, check for messages and error codes at both the local and remote nodes.

Note: Suggested actions, commands, and responses in this section apply to IBM TCP/IP for DOS. If you are using a non-IBM TCP/IP product, all of the information in this section may not apply. See your TCP/IP product documentation for additional information.

STEP 1: Checking TCP/IP

For more information, see "Is TCP/IP Operational on the Local System?" on page 56 and "Is TCP/IP Operational on the Remote System?" on page 57.



STEP 2: Checking APPC Networking Services for Windows

For more information, see "Checking APPC Networking Services for Windows" on page 59.

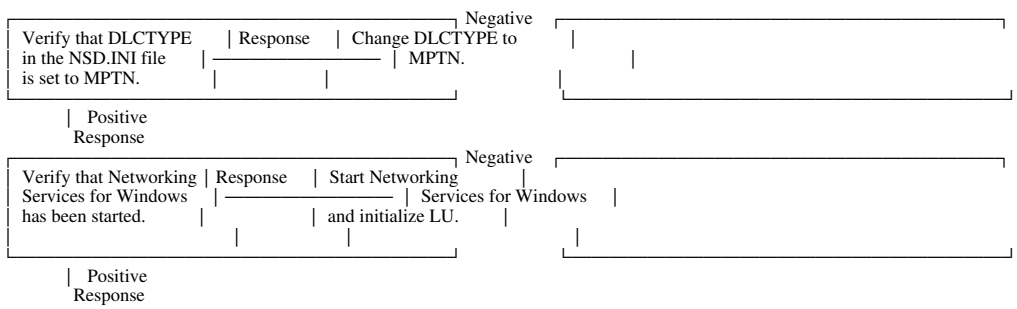
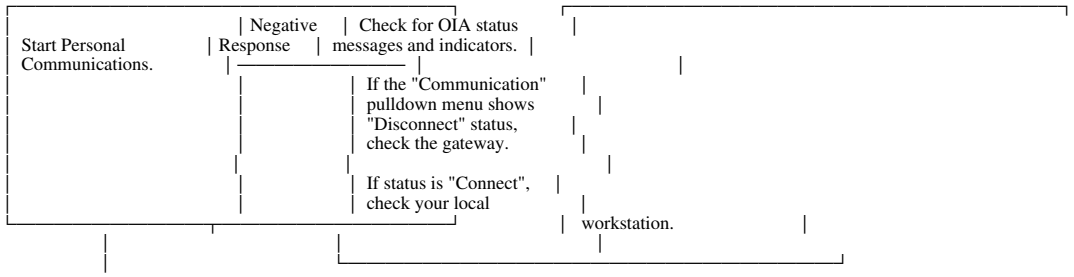


Figure 12 (Part 1 of 2). Troubleshooting Steps

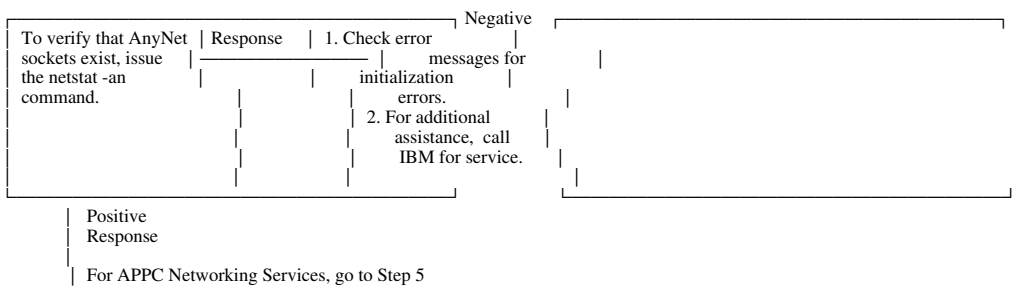
STEP 3: Checking Personal Communications for Windows

For more information, see “Checking Personal Communications for Windows” on page 61.



STEP 4: Checking SNA over TCP/IP for Windows

For more information, see “Checking SNA over TCP/IP for Windows” on page 62.



STEP 5: Verifying Connectivity

For more information, see “Verifying Connectivity Using the APING Program” on page 63.

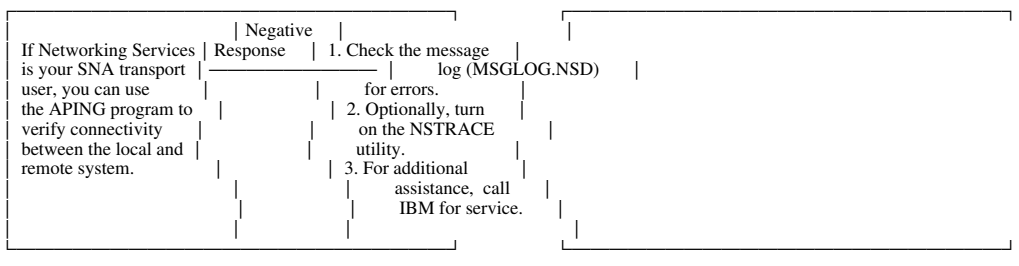


Figure 12 (Part 2 of 2). Troubleshooting Steps

Checking TCP/IP

This section explains how to verify that TCP/IP is fully operational on both the local and remote systems.

Troubleshooting

Notes:

1. Suggested actions, commands, and responses in this section apply to IBM TCP/IP for DOS. If you are using a non-IBM TCP/IP product, all of the information in this section may not apply.
2. If you are using a non-IBM TCP/IP product and are having problems with TCP/IP connections:

   Ensure that the value of the FILES parameter in your CONFIG.SYS file is sufficient. This value affects the number of available sockets for connections.

   Check to see if your TCP/IP product allows you to increase the number of available sockets or decrease the time you must wait before reusing sockets.

   Refer to your TCP/IP product documentation for additional information.

Is TCP/IP Operational on the Local System?

To check TCP/IP on a local system, do the following:

1 Obtain the IP address of the local system by entering `netstat -in` on the DOS command line. Information similar to Figure 13 should appear. In this example, the IP address is 9.67.98.162.

```
Name   Mtu   Network      Address            Ipkts Ierrs    Opkts Oerrs  Coll
lo    1536  <Link>
nd     2 8   <Link>                          294 1            69
nd     2 8   9.67.96     9.67.98.162         294 1            69
```

Figure 13. Using the Netstat Command to Obtain an IP Address

2 Issue the `ping` command using the IP address of the local system. For example, if the IP address is 9.67.98.162, enter the following command:

```
ping 9.67.98.162
```

Use Ctrl-C to stop the execution of the `ping` command.

A positive response looks similar to the example in Figure 14 on page 57 and indicates that TCP/IP is operational on the local system.

```
PING 9.67.98.162 (9.67.98.162): 56 data bytes
64 bytes from 9.67.98.162: icmp_seq=  ttl=255 time=  ms
64 bytes from 9.67.98.162: icmp_seq=1 ttl=255 time=  ms
64 bytes from 9.67.98.162: icmp_seq=2 ttl=255 time=  ms
64 bytes from 9.67.98.162: icmp_seq=3 ttl=255 time=  ms
64 bytes from 9.67.98.162: icmp_seq=4 ttl=255 time=  ms
64 bytes from 9.67.98.162: icmp_seq=5 ttl=255 time=  ms
64 bytes from 9.67.98.162: icmp_seq=6 ttl=255 time=  ms
64 bytes from 9.67.98.162: icmp_seq=7 ttl=255 time=  ms
64 bytes from 9.67.98.162: icmp_seq=8 ttl=255 time=  ms
64 bytes from 9.67.98.162: icmp_seq=9 ttl=255 time=  ms
^C

--- 9.67.98.162 ping statistics ---
1  packets transmitted, 1  packets received,  % packet loss
round-trip min/avg/max =  //  ms
```

Figure 14. Using the Ping Command to Check TCP/IP on a Local System

3 See Table 1 for recommended actions.

Table 1. Ping Command Responses and Actions for the Local System

Response	Action
Positive	Go to “Is TCP/IP Operational on the Remote System?.”
Negative	If TCP/IP is not operational, exit Windows and start TCP/IP by running TCPSTART at the DOS command line or from the AUTOEXEC.BAT file of the local system.

Is TCP/IP Operational on the Remote System?

To check TCP/IP on a remote system, do the following:

1 Issue the host command to obtain the IP address of the domain name which is derived from the destination LU name.

   a. Enter host *domain_name*, where *domain_name* is in the format *luname.netid.snasuffix*.

   b. In this example, the response is that *domain_name* resolves to 9.67.98.198.

2 See Table 2 on page 58 for recommended actions.

Troubleshooting

Table 2. Host Command Responses and Actions for the Remote System

Response	Action
Positive	Go to Step 3 on page 58.
Negative	Notify your network administrator. If the host command fails, possible causes are: The specified *domain_name* is not found. The domain name server is inoperative, or there is no path in the network to the remote system. For information on domain name definitions, see "Defining Domain Names and IP Addresses" on page 14.

3 Issue the ping command using the IP address of the destination LU. For example, if the IP address is 9.67.98.198, enter the following command:

```
ping 9.67.98.198
```

Use Ctrl-C to stop the execution of the ping command.

A positive response looks similar to the example in Figure 15. A positive response indicates that TCP/IP is operational on the remote system and a TCP/IP connection can be made.

```
PING 9.67.98.198 (9.67.98.198): 56 data bytes
64 bytes from 9.67.98.198: icmp_seq=  ttl=255 time=5  ms
64 bytes from 9.67.98.198: icmp_seq=1 ttl=255 time=  ms
64 bytes from 9.67.98.198: icmp_seq=2 ttl=255 time=  ms
64 bytes from 9.67.98.198: icmp_seq=3 ttl=255 time=  ms
^C

--- 9.67.98.198 ping statistics ---
4 packets transmitted, 4 packets received,  % packet loss
round-trip min/avg/max =  /12/5  ms
```

Figure 15. Using the Ping Command to Check TCP/IP on a Remote System

4 See Table 3 on page 59 for recommended actions.

Table 3. Ping Command Responses and Actions for the Remote System

Response	Action
Positive	If APPC Networking Services is your SNA transport user, go to "Checking APPC Networking Services for Windows" on page 59. If Personal Communications is your SNA transport user, go to "Checking Personal Communications for Windows" on page 61.
Negative	Notify your network administrator. If there is no response to the ping command or the IP address is not what you expected, possible causes are: The remote system's TCP/IP is inoperative. There is no path in the network to the remote system. For information on address mapping, see "Defining Domain Names and IP Addresses" on page 14.

For more information on using the host and ping commands, see the following publications:

IBM TCP/IP Version 2.1.1 for DOS: User's Guide
IBM TCP/IP Version 2.1.1 for DOS: Command Reference

Checking APPC Networking Services for Windows

If you are using APPC Networking Services as your SNA transport, it must be active for SNA over TCP/IP to operate. Networking Services provides a message log and trace utility to help you diagnose network problems. For more information, see:

"Networking Services Message Log" on page 64
"Networking Services Trace Utility" on page 65

Is DLCTYPE Set to MPTN?

Before checking that APPC Networking Services is active, verify that the data link control type (DLCTYPE) in the NSD.INI file is set to MPTN.

Restrictions: Networking Services supports only one DLCTYPE at a time. Therefore, on your Windows workstation, you cannot use AnyNet and an SNA network simultaneously. If you change from AnyNet to an SNA network, or vice versa, you must do the following:

1. Stop Networking Services.
2. Change DLCTYPE in the NSD.INI file to indicate your choice.
3. Restart Networking Services.

Troubleshooting

Is APPC Networking Services Active?

After checking the DLCTYPE setting, verify that APPC Networking Services is active by using the WINNSD.EXE program.

1. Run the "Advanced Operations" WINNSD.EXE program by using one of the following methods:

   Double-click on the "Advanced Operations" icon in the "IBM APPC Networking Services" program group.

   In the Windows "Program Manager" window:

   a. Select "File" on the menu bar and click on the "Run" option.

   b. In the "Run" window, type:

      X:\PATH\winnsd.exe

      For *X:\PATH*, enter the drive and directory where WINNSD.EXE is located.

   c. Press the "OK" push button.

   The "Networking Services Advanced Operations" window is displayed.

2. Select "Operation" on the pulldown menu.

3. Click on the "LU Control" option.

   Networking Services is active if all fields in the "LU Control" window contain data. Possible fields are:

   Local LU Name
   Node Type
   DLC Type
   Adapter Port
   Link Stations
   Partner LUs
   Sessions
   Conversations

   If there are blank data fields, Networking Services is not active.

4. See Table 4 on page 61 for recommended actions.

Table 4. WINNSD.EXE Responses and Actions for APPC Networking Services

Response	Action
Positive	Go to "Checking SNA over TCP/IP for Windows" on page 62.
Negative	Start Networking Services by clicking on the "Initialize LU" option. **Note:** Under certain conditions during initialization, a pop-up window may be displayed by AnyNet or TCP/IP indicating that initialization was not successful. The most common message displayed in this situation is INET not running. You can click on the "OK" push button or press the enter key to clear the message even though the hourglass has not changed to an arrow pointer. To correct the error condition, exit Windows. This may require rebooting your workstation. Then restart SNA over TCP/IP using the steps in this section. If problems persist, contact IBM for service. See "Using IBM Support Services" on page 72 for information.

Checking Personal Communications for Windows

If you are using Personal Communications as your SNA transport user, it must be active for SNA over TCP/IP to operate. If the trace function has been started, Personal Communications provides a message and trace log to help you diagnose network problems. For more information, see:

"Personal Communications Message and Trace Log" on page 69
"Personal Communications Trace Utility" on page 70

Verify that Personal Communications is active by using the following steps:

1. Attempt to start Personal Communications using the procedure in "Using Personal Communications to Start SNA over TCP/IP" on page 50.

2. If you cannot start Personal Communications:

   a. Check your session window and the OS/2 window for operator information area (OIA) status messages and indicators. Refer to Personal Communications *Up and Running* for an explanation of OIA messages and indicators.

   b. Select the "Communication" option on the menu bar to check emulator session status:

      If status is "Connect," this indicates that there is a problem with the local workstation. Go to Step 3 on page 62.

      If status is "Disconnect," this indicates that there is a problem with the gateway.

      Before checking the gateway, try to start the session by clicking on "Disconnect" and then clicking on "Connect." If this does not correct the problem, go to Step 3 on page 62.

3 See Table 5 on page 62 for recommended actions.

Table 5. Responses and Actions for Personal Communications

Response	Action
Positive	Go to "Checking SNA over TCP/IP for Windows."
Negative	1. Use the OIA messages and indicators to help determine the cause of the problem. 2. If the trace function is turned on, check the message and trace log for initialization errors. See "Troubleshooting Aids for Personal Communications" on page 69 for information. 3. If problems persist, contact IBM for service. See "Using IBM Support Services" on page 72 for information.

Checking SNA over TCP/IP for Windows

To verify that SNA over TCP/IP is initialized correctly, use the TCP/IP netstat command and specify the -an option:

1 Enter netstat -an.

If SNA over TCP/IP is initialized correctly, the netstat -an command displays information similar to the example in Figure 16.

```
Active Internet connections (including servers)
PCB        Proto Recv-Q Send-Q  Local Address          Foreign Address        (state)
5787a12 tcp                     .397                   .                      LISTEN
5787b12 udp                     .397                   .
```

Figure 16. Using the Netstat Command to Check SNA over TCP/IP

Note: This is displayed *before* a connection to a remote LU is set up.

The netstat command output shows that SNA over TCP/IP is enabled for the following reasons:

A stream socket (TCP) bound to the well-known port (397) is in a LISTEN state.

A datagram socket (UDP) is bound to the well-known port (397).

If SNA over TCP/IP is not initialized correctly, no information appears for the well-known port.

2 See Table 6 on page 63 for recommended actions.

Table 6. Netstat Command Responses and Actions for SNA over TCP/IP

Response	Action
Positive	If APPC Networking Services is your SNA transport user, go to "Verifying Connectivity Using the APING Program" on page 63. If Personal Communications is your SNA transport user, see "Troubleshooting Aids for Personal Communications" on page 69 and "Using IBM Support Services" on page 72 for other possible actions.
Negative	1. Check for error messages: If APPC Networking Services is your SNA transport user, check the message log for initialization errors. See "Networking Services Message Log" on page 64 for more information. If Personal Communications is your SNA transport user and the tracing function is turned on, check the message and trace log for initialization errors. See "Personal Communications Message and Trace Log" on page 69 for more information. 2. For additional assistance, contact IBM for service. See "Using IBM Support Services" on page 72 for more information.

Verifying Connectivity Using the APING Program

If APPC Networking Services is your SNA transport user, you can use the APING program to test connectivity between the local and remote system. This program is supplied with the Networking Services product. APING is an APPC program that exchanges data between two workstations over APPC.

To use APING:

1. Double-click on the "APING" icon.
2. Enter the destination LU name in the appropriate field.

   Note: The APINGD program must be installed on the remote machine.
3. Click on the "OK" push button.
4. Either click on the green "Go" push button, or click on the "Go" option on the "Actions" pulldown.
5. Check the results.

   If communication is successful, data transfer is possible between your system and the remote system over APPC.

   If a session cannot be allocated, extensive error information is displayed.
6. See Table 7 on page 64 for recommended actions.

Table 7. APING Responses and Actions

Response	Action
Positive	For other possible actions, see “Troubleshooting Aids for APPC Networking Services” on page 64 and “Using IBM Support Services” on page 72.
Negative	1. Check the APPC Networking Services for Windows message log for initialization errors. See “Networking Services Message Log” on page 64 for more information. For additional troubleshooting information, see *IBM APPC Networking Services for Windows: Getting Started*. 2. Optionally, turn on the NSTRACE utility. See “Networking Services Trace Utility” on page 65 for more information. 3. For additional assistance, contact IBM for service. See “Using IBM Support Services” on page 72 for more information.

Troubleshooting Aids for APPC Networking Services

This section includes the following information:

“Networking Services Message Log”
“Networking Services Trace Utility” on page 65
“Abinfo Mapping Utility” on page 68

Networking Services Message Log

Check the information that is recorded in message log MSGLOG.NSD. For an example, see Figure 17. To view messages, open the APPC Networking Services window and click on the “View Message Log” icon. For a description of AnyNet messages and error codes, see “Message and Error Code Descriptions” on page 73.

```
1 :43:52    1 /27/95    MPTNCMM
ANY  7E: AnyNet connection setup failed. Partner transport provider address = 9.67.98.25  397 .
Partner transport user address = USIBMNR.BADNAME.
     7-  4   1, 83142 1

1 :43:52    1 /27/95    MPTNCMM
   55534942 4D4E522E 4241444E 414D452    <USIBMNR.BADNAME >   <....(+.....+.(..>
   2 392E36 372E3938 2E32353  2 333937    < 9.67.98.25  397>   <.............>
   2 555349 424D4E52 2E434152 4F4C594E   < USIBMNR.CAROLYN>   <.....(+......<.+>
```

Figure 17. Example of an APPC Networking Services Message Log

Note: If message ERROR 100 is recorded repeatedly in the log file, this indicates that there is a problem with the AnyNet message file. SNA over TCP/IP must be reinstalled to correct the problem. For additional information, see the description of message ANY0999E in “Message and Error Code Descriptions” on page 73. This message is issued during the initialization of SNA over TCP/IP.

Networking Services Trace Utility

To use the Networking Services trace utility, follow the procedures outlined in the following sections:

"Starting the Trace Utility" explains how to start the NSTRACE trace utility.

"Formatting and Viewing Trace Output" on page 67 explains how to format raw trace data and create the trace log file (FMTNS.LOG).

Note: If message ERROR 100 is recorded repeatedly in the log file, this indicates that there is a problem with the AnyNet message file. SNA over TCP/IP must be reinstalled to correct the problem. For additional information, see the description of message ANY0999E in "Message and Error Code Descriptions" on page 73. This message is issued during the initialization of SNA over TCP/IP.

Starting the Trace Utility

To trace data for AnyNet, you can use the NSTRACE utility provided by APPC Networking Services. For details on using NSTRACE, see the Networking Services *Administrator's Guide*.

To start the trace utility:

1. Double-click on the "Trace" icon in the "APPC Networking Services" program group. The "Networking Services Trace Facility" window is displayed.
2. Click on the "Initialize" push button. The "Trace Initialization" window is displayed.
3. Change the following entry fields:

Entry Field	New Setting
Trace Log Size	1,000 kilobytes
Buffer Size	128 kilobytes
Dump Frequency	1

   If you do not use the recommended settings for Buffer Size and Dump Frequency, an incomplete trace may be produced.
4. Click on "OK" to initialize the tracing function.
5. Click on the "Configure" push button in the "Networking Services Trace Facility" window. The "Trace Configuration" window is displayed.
6. In the "Trace Configuration" window:
   a. Select "AnyNet" as the product to trace.
   b. Select the "MPTN_Trace" component.
   c. Select one of the following MPTN trace options:

Internal Calls: Traces AnyNet internal function calls

Transport Provider Calls: Captures data entering and leaving the local node and socket interface calls

d. Click on “OK” to return to the “Networking Services Trace Facility” window.

7. Click on the “Start” or “Resume” push button to begin the collection of trace data.

8. Click on the “Suspend...” push button to stop tracing.

Note: If you are tracing **both** SNA over TCP/IP and APPC Networking Services, you must specify trace options for one product then return to the “Networking Services Trace Facility” window before specifying trace options for the other product.

To trace **both** SNA over TCP/IP and APPC Networking Services:

1. Double-click on the “Trace” icon in the “APPC Networking Services” program group. The “Networking Services Trace Facility” window is displayed.

2. Click on the “Initialize” push button. The “Trace Initialization” window is displayed.

3. Click on “OK” to initialize the tracing function.

4. In the “Networking Services Trace Facility” window, click on the “Configure” push button. The “Trace Configuration” window is displayed.

5. In the “Trace Configuration” window:
   a. Select a product to trace.
   b. Specify your trace options.
   c. Click on “OK” to return to the “Networking Services Trace Facility” window.

6. In the “Networking Services Trace Facility” window, click on the “Configure” push button again. The “Trace Configuration” window is displayed.

7. In the “Trace Configuration” window:
   a. Select another product to trace.
   b. Specify your trace options.
   c. Click on “OK” to return to the “Networking Services Trace Facility” window.

8. Click on the “Start” or “Resume” push button to begin the collection of trace data.

9. Click on the “Suspend...” push button to stop tracing.

To view the trace output, follow the steps in “Formatting and Viewing Trace Output” on page 67.

Formatting and Viewing Trace Output

To view trace output, you must format the trace data:

1. If tracing has not stopped, click on the "Suspend..." push button in the "Networking Services Trace Facility" window.
2. Click on the "Format..." push button. The path and file name for the formatted trace output displays.
3. Click on "OK" to begin formatting the trace data.

Reading Trace Data

The formatted trace data is in the FMTNS.LOG file. This file contains three different components that provide information about AnyNet.

A description of each trace component follows:

Component	Description
MPTNASNA	Intercepts SNA flows and issues transport-layer protocol boundary (TLPB) verbs

Trace Call Type	Description
FUNC CALL	Indicates a function was entered
FUNC RET	Indicates an exit from a function

For an example, see "MPTNASNA Trace" on page 100.

Component	Description
MPTNCMM	Traces general MPTN functions

Trace Call Type	Description
FUNC CALL	Indicates a function was entered
FUNC RET	Indicates an exit from a function
ERROR	Error condition detected

For an example, see "MPTNCMM Trace" on page 101.

Component	Description
MPTNBTCP	Traces TCP-specific MPTN functions. This is the only component that makes socket calls.

Trace Call Type	Description
<==SEND=====	Data sent over the TCP connection
===RECV====>	Data received on the TCP connection
<==SEND=DG==	Data sent over the datagram connection
===RECV=DG=>	Data received on the datagram connection
FUNC CALL	Indicates a function was entered
FUNC RET	Indicates an exit from a function
ERROR	Error condition detected
PROV CALL	Indicates a call to the socket interface
PROV RET	Indicates the return from a call to the socket interface

For an example, see "MPTNBTCP Trace" on page 101.

Troubleshooting

Session Initiation Trace Examples

See Appendix D, “Trace Examples for APPC Networking Services” on page 99 for examples of a successful and unsuccessful session initiation.

Abinfo Mapping Utility

The abinfo utility can be used if you have selected APPC Networking Services as your SNA transport user. Using abinfo, you can display the associations between the transport user (SNA) and the transport provider (TCP/IP).

For each session, abinfo helps you determine the local and remote IP addresses and LU names (mapping SNA to TCP/IP). The following information is displayed for each partner in all currently active sessions:

An LU name and an IP address (and its associated port number)

The procedure correlator ID (PCID), which is a 16-character hexadecimal number that identifies an LU-LU session

To use the abinfo utility:

1. Create an “Abinfo Mapping Utility” icon:
   a. In the Windows Program Manager window, click on the “AnyNet” icon to open the “AnyNet” program group.
   b. Click on the “File” option.
   c. Select “New...,” then select the “Program Item” option.
   d. Click on “OK.”
   e. In the “Description” field, enter ABINFO Mapping Utility.
   f. In the “Command Line” field, enter *c:\anynet*\BIN\ABINFO.EXE, where *c:\anynet* is the directory in which you installed SNA over TCP/IP.
   g. In the “Working Directory” field, enter *c:\anynet*\BIN.
   h. Click on “OK.”
2. Double-click on the “ABINFO Mapping Utility” icon to display the abinfo menu.
3. Click on the “Command” option.
4. Select “Run” on the “Command” pulldown menu:

   The “Session Information” window is displayed and contains detailed information on the first active session. The total number of active sessions appears in the lower right hand corner of the window.
5. If there is more than one active session, use the “Next” and “Previous” push buttons to view session information on other active sessions.

Troubleshooting Aids for Personal Communications

This section includes the following information:

"Personal Communications Message and Trace Log"
"Personal Communications Trace Utility" on page 70

Personal Communications Message and Trace Log

If tracing has been turned on, messages and trace data are logged in log file PCSMON.TRC. See Figure 18 for an example. To turn tracing on, see the steps in "Starting the Trace Utility" on page 70.

```
1 :34: 2.653 PIU size: 2 48
1 :34: 2.655 INB:    [LH:] l=-7968 LSid=e e
                          [TH:] OIS 1<-1 EFI ODAI SNF=
                          [RH:] REQ SC FI OIC DR1I
                         [RU:]  BIND
                                ..'..J...77.............
                                USIBMNR.WINSNA5....SNASVCMG.....{.....USIBMNR.WINSNA5.....
                                 1 bbdb  ff8          12   eeccddd4ecdedcf3   edceecdc
   c   1 eeccddd4ecdedcf 1
                                 37  131f77f162       43  f4292459b69525151 92251253479
314 3 1 1 44292459b6952515a3 4 3
                                {.....USIBMNR.BADNAME-....{.....USIBMNR.WINSNA5
                                c      eeccddd4cccdcdc61   c     eeccddd4ecdedcf
                                 1    f4292459b2145145 84 3 1   f4292459b6952515
1 :34: 5.83  MPTNBTCP ANY  15W: LU name BADNAME.USIBMNR.SNA.IBM.COM not found in
 domain name server.
1 :34: 5.83  Data:
                                ........


1 :34: 5.831 OUTB: [LH:] l=  LSid=
                          [TH:] OIS 1->1 EFI ODAI SNF=   1
                          [RH:] REQ SC FI OIC DR1I
                         [RU:]  UNBIND
                                .....

                                f
1 :34: 5.842 INB:    [LH:] l=  LSid=
                          [TH:] OIS 1<-1 EFI ODAI SNF=
                          [RH:] +RSP SC FI OIC DR1I
                         [RU:]  UNBIND
```

Figure 18. Example of a Personal Communications Message and Trace Log

Notes:

1. Messages in log file PCSMON.TRC with a prefix of "ANY" are issued during the initialization and operation of SNA over TCP/IP. For a description of these messages, see "Message and Error Code Descriptions" on page 73.
2. If message ERROR 100 is recorded repeatedly in the log file, this indicates that there is a problem with the AnyNet message file. SNA over TCP/IP must be reinstalled to correct the problem. For additional information, see the description of message ANY0999E in "Message and Error Code Descriptions" on page 73. This message is issued during the initialization of SNA over TCP/IP.

Troubleshooting

Personal Communications Trace Utility

To use the Personal Communications trace utility, follow the procedures outlined in the following sections:

"Starting the Trace Utility" explains how to start the PCSMON trace utility.

"Reading Trace Data" on page 71 explains how to format raw trace data and create the trace and message log file (PCSMON.TRC).

"Stopping the Trace Utility" on page 71 explains how to stop tracing. Note that messages are not logged when the tracing function is turned off. For more information, see "Personal Communications Message and Trace Log" on page 69.

Starting the Trace Utility

To trace data for AnyNet, you can use the PCSMON utility provided by Personal Communications. For details on using PCSMON, see the Personal Communications online help.

To start the trace utility:

1. Add the following lines to the workstation profile configuration file (*.WS):

   ```
   [AnyNet]
   DebugMask=1111
   ```

2. Double-click on the "Administration Tools" icon in the "Personal Communications" program group.

3. Click on the "Trace Set-up Utility (PCSTRSET)" option.

4. Set the "Communication-Line" option to "On," and click on "OK" to save the setting.

5. Click on the "Trace-Capture Program (PCSMON)" option. The "Trace Start" window is displayed.

6. Enter anytrace in the "Trace Name" field.

7. Select one or both of the following options:

Option	Description
File	Record trace data in a file
Watch	Display trace options

   If displayed trace records are truncated, save the trace to a file and view.

To stop the display of trace records:

a. Click on the “Trace” option on the menu bar.
b. Click on “Stop watch....”

8 Click on “OK.”

Reading Trace Data

The formatted trace data is in the PCSMON.TRC file. This file contains two different components that provide information about AnyNet.

A description of each trace component follows:

Component	Description
MPTNCMM	Traces general MPTN functions

Trace Call Type	Description
FUNC CALL	Indicates a function was entered
FUNC RET	Indicates an exit from a function
ERROR	Error condition detected

MPTNBTCP — Traces TCP-specific MPTN functions. This is the only component that makes socket calls.

Trace Call Type	Description
<==SEND=====	Data sent over the TCP connection
===RECV====>	Data received on the TCP connection
<==SEND=DG==	Data sent over the datagram connection
===RECV=DG=>	Data received on the datagram connection
FUNC CALL	Indicates a function was entered
FUNC RET	Indicates an exit from a function
ERROR	Error condition detected
PROV CALL	Indicates a call to the socket interface
PROV RET	Indicates the return from a call to the socket interface

Stopping the Trace Utility

To stop the Personal Communications trace utility:

Note: Messages are not logged when the tracing function is turned off. For more information, see “Personal Communications Message and Trace Log” on page 69.

1 Double-click on the “Administration Tools” icon in the “Personal Communications” program group.

2 Click on the “Trace Set-up Utility (PCSTRSET)” option.

3 Set the “Communication-Line” option to “Off,” and click on “OK” to save the setting.

Session Initiation Trace Example

See Appendix E, “Trace Example for Personal Communications” on page 117 for an example of a successful session initiation.

Using IBM Support Services

For product warranty and IBM service information, see the license information provided with this product. In addition to the service information described in the license information, you can query the following locations:

Internet: http://www.ibm.com/Products/support.html

- Search tips and techniques database
- Download fixes, device drivers

Telephone: 1-800-237-5511 (U.S. and Puerto Rico only)

- Top Frequently Asked Questions (FAQs)
- Faxback System

Contract Support

If you are a licensed customer with a support contract, call one of the following support numbers:

From U.S. or Puerto Rico 1-800-237-5511
From Canada 1-800-465-2222

Please have the following information available:

Registration number from your proof of license card
Complete product name and version number
Hardware and software used
Description of the problem and circumstances under which it occurred
Description of any troubleshooting efforts (if any)
Exact wording of any messages received

If you are a licensed customer without a support contract, you can call one of the following phone numbers to purchase a service contract.

For single-license or multiple-license customers, call 1-800-237-5511 (8:00 am to 8:00 pm eastern time, Monday through Friday)

For corporate customers in the U.S. and Puerto Rico, call 1-800-799-7765 (from Canada, call 1-800-465-9600) 8:00 am to 5:00 pm eastern time, Monday through Friday

Informal Technical Assistance

Technical information and informal assistance can be obtained from the following forums:

The AnyNet customer forum of the OS/2 Bulletin Board System through IBM Information Network and the TalkLink facility

The MPTN forum of the CompuServe Bulletin Board Service

Appendix A. AnyNet Messages and Error Codes

This appendix includes the following message descriptions:

"Message and Error Code Descriptions"
"Installation and Configuration Message Descriptions" on page 86

The last character of the message identifier indicates the type of message. See Table 8 for an explanation of message types.

Table 8. Message Types

Last Character	Message Type	Description
I	Information	The message is for your information.
W	Warning	The message informs you of a situation that may or may not be an error. Actions may be recommended or required.
E	Error	The message informs you of an error, and you should take action to correct the problem.

If there are major and minor error codes associated with one of these messages, these codes are recorded in the message log.

The major code is the first code that is displayed; it contains the number of the message that is associated with the error.

The minor code indicates the specific cause of the error.

Figure 19 shows an example of a major and minor error code combination for message ANY0005E.

```
  Major      Minor
  Code       Code
-------- ---------
    5-  3   4
```

Figure 19. Example of Error Codes for Message ANY0005E

Message and Error Code Descriptions

This section includes descriptions of AnyNet messages and error codes that are issued during the initialization and operation of SNA over TCP/IP.

If a message is logged, the SNA transport user you select determines how logging is handled. For additional information, see:

"Networking Services Message Log" on page 64
"Personal Communications Message and Trace Log" on page 69

© Copyright IBM Corp. 1996

ANY0001E ANY0004E

ANY0001E AnyNet storage unavailable - critical error.

Error code: 00000001-00010001

Indicates: AnyNet function is not available because SNA over TCP/IP storage is exhausted. Communication is no longer possible using SNA-over-TCP/IP connections.

User response: Reduce the number of sockets in use by closing SNA sessions or other socket applications. To reactivate SNA over TCP/IP, stop the SNA transport user and then restart your system using normal procedures.

The SNA transport user is either APPC Networking Services or Personal Communications. For information on restarting, see “Starting SNA over TCP/IP” on page 49.

If this does not correct your problem, you might have to stop Windows and then restart using normal procedures.

ANY0002E AnyNet internal error - critical.

Error code: 00000002-00010002
Error code: 00000002-000C0001

Indicates: A critical SNA over TCP/IP error occurred in the AnyNet manager component. Communication is no longer possible using SNA over TCP/IP connections.

User response: To reactivate SNA over TCP/IP, stop the SNA transport user and then restart your system using normal procedures.

The SNA transport user is either APPC Networking Services or Personal Communications. For information on restarting, see “Starting SNA over TCP/IP” on page 49.

If this does not correct your problem, you may have to stop Windows and then restart using normal procedures.

If problems persist, save your message log. Note the timestamp on this message and investigate messages with timestamps close to this one. For additional assistance, contact IBM for service. See “Using IBM Support Services” on page 72 for information.

ANY0003E AnyNet internal error - noncritical.

Error code: 00000003-00010003
Error code: 00000003-000C0002

Indicates: An unexpected internal condition was detected that is considered a noncritical error. An SNA over TCP/IP connection may have failed or been closed.

User response: If a connection failed, attempt to restart it.

ANY0004E AnyNet configuration or definition error.

Error code: 00000004-00010004

Indicates: An AnyNet configuration or definition error was logged because the configuration file ANYNET.CFN could not be opened. It may not be possible to establish communications using AnyNet connections.

User response: Make sure the file exists in the MISC subdirectory of the directory in which AnyNet was installed.

Error code: 00000004-00010005

Indicates: An AnyNet configuration or definition error was logged because there is an error in the configuration file ANYNET.CFN. It may not be possible to establish communications using AnyNet connections.

User response: Make sure that the file exists in the MISC subdirectory of the directory in which AnyNet was installed. If it has been corrupted, the file must be restored from backup, or SNA over TCP/IP must be reinstalled.

Error code: 00000004-00010008

Indicates: An AnyNet configuration or definition error was logged because the SNA transport user trace facilities could not be loaded.

User response: Make sure that the SNA transport user is active and that it is possible to activate trace facilities.

Error code: 00000004-00010009

Indicates: An AnyNet configuration or definition error was logged because SNA over TCP/IP could not load the AnyNet transport provider, TCPPMM.DLL.

User response: Ensure that the ANYNET.CFN file has not been corrupted. If it has been corrupted, the file must be restored from backup, or SNA over TCP/IP must be reinstalled.

Error code: 00000004-0001000A

Indicates: An AnyNet configuration or definition error was logged because SNA over TCP/IP could not load the SNA transport provider.

User response: Ensure that the SNA transport provider is installed correctly and is active.

Error code: 00000004-000C0003

Indicates: An AnyNet configuration or definition error was logged because an error was detected in an SNA over TCP/IP environment setting.

User response: Check the message log for the name of the environment variable. Ensure that the environment setting has the correct value and that the value is within the valid range.

ANY0005E AnyNet transport provider connection setup failed. Partner transport provider address=*address***. Partner transport user address=***address***.**

Error code: 00000005-00030001

Indicates: The SNA over TCP/IP connection setup attempt failed because no route was available to the partner.

User response: Determine whether a route is available from this node to the partner TCP/IP address. Using the TCP/IP ping command with the appropriate partner address may provide additional information.

Error code: 00000005-00030002

Indicates: The SNA over TCP/IP connection setup attempt failed. Possible reasons are:

The partner node may be down.
TCP/IP may not be running in the partner node.
AnyNet may not be active in the partner node.
The partner may be overloaded by connection requests.

User response: Retry the connection request. If repeated connection attempts fail, verify that the partner node has SNA over TCP/IP and TCP/IP active. See “Is

ANY0005E

TCP/IP Operational on the Remote System?" on page 57 for information on testing the partner node.

Error code: 00000005-00030003

Indicates: The SNA over TCP/IP connection setup attempt failed because local resources, such as available sockets, have been temporarily exhausted.

User response: If necessary, reduce the number of active connections and retry.

Error code: 00000005-00030004

Indicates: The SNA over TCP/IP connection setup attempt failed because there was a failure in the TCP/IP connection setup process.

User response: Attempt to restart the connection.

Error code: 00000005-00030005

Indicates: The SNA over TCP/IP connection setup attempt failed because there was a failure in the TCP/IP connection setup process with the partner node.

User response: Attempt to restart the connection when the partner node is reachable.

Error code: 00000005-00030006

Indicates: The SNA over TCP/IP connection setup attempt failed because there was a temporary shortage of TCP/IP buffers.

User response: This error can occur if too many applications are processing large data transfers at the same time. You might have to close one or more of the applications.

Error code: 00000005-00030007

Indicates: The SNA over TCP/IP connection setup attempt failed because the transport provider connection limit was exceeded.

User response: Close some sessions. This results in the corresponding transport provider connections being closed.

Error code: 00000005-00030008

Indicates: The SNA over TCP/IP connection setup attempt failed. Possible reasons are:

The partner node may be down.
The partner may be overloaded by connection requests.
The network may be overloaded.
The value of the connection wait time limit environment variable may be set too low.

User response: Retry the connection request. If repeated connection attempts fail, verify that the partner node is active and that connectivity is possible. See "Is TCP/IP Operational on the Remote System?" on page 57 for suggested actions.

If you cannot find problems with the partner node, you may need to increase the value of the connection wait time limit environment variable. For additional information, see "Connection Wait Time Limit" on page 37.

Error code: 00000005-00FFFFFF

Indicates: The SNA over TCP/IP connection setup attempt failed because of a previously logged error.

User response: Note the timestamp on this message and investigate messages with timestamps close to this one. For additional assistance, contact IBM for service. See "Using IBM Support Services" on page 72 for information.

ANY0006E AnyNet transport provider connection failed. Partner transport provider address=*address*. Partner transport user address=*address*.

Error code: 00000006-00050001
Indicates: An established SNA over TCP/IP connection failed because the partner TCP/IP closed the connection.

User response: Attempt to restart the connection.

Error code: 00000006-00050002
Indicates: An established SNA over TCP/IP connection failed because the partner node is not available.

User response: Attempt to restart the connection after approximately 5 minutes. If the connection fails to restart, check the current status of the remote system.

Error code: 00000006-00050003
Indicates: An established AnyNet connection failed because there was an MPTN protocol error.

User response: Check whether the partner node is equipped with compatible AnyNet capabilities and apply any corrective service package that might be applicable. If the problem persists, save your message log and contact IBM for service. See "Using IBM Support Services" on page 72 for information.

Error code: 00000006-00050004
Indicates: An established SNA over TCP/IP connection failed because there was a temporary shortage of TCP/IP buffers.

User response: This error can occur if too many applications are processing large data transfers at the same time. You might have to close one or more of the applications.

Error code: 00000006-00050005
Indicates: An established SNA over TCP/IP connection failed because there was a temporary shortage of AnyNet buffers.

User response: This error can occur if too many applications are processing large data transfers at the same time. You might have to close one or more of the applications.

Error code: 00000006-00FFFFFF
Indicates: An established SNA over TCP/IP connection failed because of a previously logged error.

User response: Note the timestamp on this message and investigate messages with timestamps close to this one. For additional assistance, contact IBM for service. See "Using IBM Support Services" on page 72 for information.

ANY0007E AnyNet connection setup failed. Partner transport provider address=*address*. Partner transport user address=*address*.

Error code: 00000007-00040001
Indicates: An SNA over TCP/IP connection setup attempt failed because the partner address (LU name) was not reachable in the destination node.

User response: Determine whether the partner node contains the destination address that was used in the connection request. For example, ensure that the destination address exists in the SNA local LU definitions of the partner node. If the partner node contains the address, then check the domain name server to ensure the address exists there.

ANY0008E ANY0009E

Error code: 00000007-00040002

Indicates: An SNA over TCP/IP connection setup attempt failed because although the partner application address was found in the destination node, the partner application was not ready to accept the connection request.

User response: Ensure that the partner application is activated. The problem may be a temporary shortage of resources in the partner node. Retry the connection request. If the problem persists it may be necessary to free resources in the partner node, for example, by deactivating other connections.

Error code: 00000007-00040003

Indicates: An SNA over TCP/IP connection setup attempt failed because the partner rejected the connection request.

User response: Examine the SNA sense codes for further information about why the partner rejected the connection.

Error code: 00000007-00040004

Indicates: SNA-over-TCP/IP connection setup attempt failed because the request was rejected by AnyNet in the partner node.

User response: Determine whether the partner node is equipped with compatible AnyNet capabilities. Apply any corrective service package that might be applicable. If the nodes are compatible, save the message log and contact IBM for service. See "Using IBM Support Services" on page 72 for information.

Error code: 00000007-00040005

Indicates: The SNA over TCP/IP connection setup attempt failed because the AnyNet connection limit of 20 connections has been reached.

User response: Close other sessions and retry the connection.

Error code: 00000007-00FFFFFF

Indicates: An SNA over TCP/IP connection setup attempt failed because of a previously logged error.

User response: Note the timestamp on this message and investigate messages with timestamps close to this one. For additional assistance, contact IBM for service. See "Using IBM Support Services" on page 72 for more information.

ANY0008E Incorrect MPTN command.

Error code: 00000008-00060001
Error code: 00000008-00060002
Error code: 00000008-00060003

Indicates: There was a protocol violation.

User response: Determine whether the partner node is equipped with compatible AnyNet capabilities. Apply any corrective service package that might be applicable. If the nodes are compatible, save the message log and contact IBM for service. See "Using IBM Support Services" on page 72 for information.

ANY0009E Incorrect MPTN format.

Error code: 00000009-00070001
Error code: 00000009-00070002
Error code: 00000009-00070003
Error code: 00000009-00070004
Error code: 00000009-00070005
Error code: 00000009-00070006

Error code: 00000009-00070007
Error code: 00000009-00070008
Error code: 00000009-00070009
Error code: 00000009-0007000A
Error code: 00000009-0007000B
Error code: 00000009-0007000C
Error code: 00000009-0007000D
Error code: 00000009-0007000E
Error code: 00000009-0007000F
Error code: 00000009-00070010
Error code: 00000009-000D0001
Error code: 00000009-000E0001
Error code: 00000009-000F0001
Error code: 00000009-000F0002
Error code: 00000009-000F0003
Error code: 00000009-000F0004
Error code: 00000009-000F0005
Error code: 00000009-000F0006
Error code: 00000009-000F0007
Error code: 00000009-000F0008
Error code: 00000009-000F0009
Error code: 00000009-000F000A

Indicates: There was a protocol error.

User response: Determine whether the partner node is equipped with compatible AnyNet capabilities. Apply any corrective service package that might be applicable. If the nodes are compatible, save the message log and contact IBM for service. See "Using IBM Support Services" on page 72 for information.

ANY0010E AnyNet load error.

Error code: 0000000A-00010006

Indicates: SNA over TCP/IP failed to initialize because no transport provider was loaded.

User response: Save your message log. Note the timestamp on this message and investigate messages with timestamps close to this one. Ensure that TCP/IP is installed and has been started. If so, reinstall SNA over TCP/IP to ensure that the AnyNet DLL and configuration files are not corrupted.

Error code: 0000000A-00010007

Indicates: An installation error occurred because SNA over TCP/IP could not load SNA.

User response: Ensure that the ANYNET.CFN file exists and has not been corrupted. If it has been corrupted, the file must be restored from backup or SNA over TCP/IP reinstalled. Ensure that SNA is being loaded and initialized.

ANY0011E Transport protocol unavailable.

Error code: 0000000B-00080001

Indicates:

TCP/IP is not installed or is not started.

TCP/IP is installed, but the installed version is not compatible with SNA over TCP/IP.

User response: Ensure that TCP/IP has been installed and is started, and that a compatible version is installed. For additional information, see “Software Requirements” on page 9 and “TCP/IP Support Services” on page 11.

Error code: 0000000B-00080002

Indicates: The local host name is not defined, or cannot be found in the HOSTS file or domain name server database.

User response: Ensure that the local host name is defined and has been added to either the local HOSTS file or domain name server database. For more information, see “Defining Domain Names and IP Addresses” on page 14.

ANY0012E Resource limit exceeded.

Error code: 0000000C-00090001

Indicates: The local resource limit has been exceeded because no more sockets are available to create TCP connections or send datagrams.

User response: Reduce the number of sockets in use by closing SNA sessions or other socket applications.

If message ANY0002E was logged after this message, stop SNA over TCP/IP and restart.

Otherwise, wait a few minutes and retry the connection.

If problems persist, you may have retried the connection before TCP/IP freed the sockets.

Refer to your TCP/IP documentation:

Ensure that the value of the FILES parameter in your CONFIG.SYS file is sufficient. This message indicates that the value might be too low.

Check to see if your TCP/IP product allows you to increase the number of available sockets or decrease the time you must wait before reusing sockets.

Error code: 0000000C-00090002

Indicates: The local resource limit has been exceeded because there is not enough memory to create a new process.

User response: Too many applications might already be running. Remove some of the applications.

Error code: 0000000C-0001000B

Indicates: The local resource limit has been exceeded because SNA over TCP/IP internal storage for out-of-band (OOB) datagrams (expedited data) has been exhausted.

User response: Too many connections that require expedited data are active. If possible, reduce the number of active SNA over TCP/IP connections.

ANY0013E Function not supported.

Error code: 0000000D-000A0001

Indicates: An internal function was requested that is not supported.

User response: Save your message log and contact IBM for service. See “Using IBM Support Services” on page 72 for information.

ANY0014E Domain name server not available.

Error code: 0000000E-000B0001

Indicates: The TCP/IP domain name server could not be located.

User response: Ensure that a domain name server is accessible within the IP network.

If you are using IBM TCP/IP, one problem could be incorrect entries in the TCP/IP RESOLV file. This file is located in the ETC subdirectory. The default is \TCPDOS\ETC. If you are using a non-IBM TCP/IP product, see your product documentation for troubleshooting information.

Error code: 0000000E-000B0002

Indicates: An unrecoverable error was detected while trying to access the domain name server.

User response: Notify your network administrator.

ANY0015W LU name *luname* not found in domain name server.

Indicates: The specified LU name could not be found in the TCP/IP domain name server.

User response: Ensure that the mapping from the destination LU name to the IP address where the LU currently resides is entered in the database of a domain name server accessible from the node where this error was logged.

ANY0017E TCP/UDP port number reserved by MPTN already in use.

Error code: 00000011-00020001

Indicates: This message indicates that the AnyNet well-known port (397) is in use. Possible causes are:

SNA over TCP/IP was stopped and restarted before port number 397 could be freed by TCP/IP.

Another socket application program is using port number 397.

This port number is reserved for AnyNet and should not be used by any other application programs. They cannot coexist with AnyNet in the same IP network.

User response:

If SNA over TCP/IP was stopped and restarted, stop SNA over TCP/IP again and wait a few minutes before restarting.

If other applications are using port number 397, remove them. Then, stop and restart SNA over TCP/IP.

ANY0018W A timeout has occurred.

Error code: 00000012-0001000C

Indicates: Too much time elapsed without a response to an AnyNet connection request.

User response: Check the network for any problems. Ensure that the intended partner is reachable. If necessary, increase the value for the connection retry duration environment setting.

ANY0019I An AnyNet module (*module_name*) failed to load properly. AnyNet is not active.

Indicates: AnyNet module *module_name* is missing or corrupted, or the system does not have enough memory to load the module.

User response: If files are missing or corrupted, reinstall SNA over TCP/IP.

If there is a memory problem, you might have to close one or more applications and try again.

ANY0020E Unknown connection alias or connection identifier.

Error code: 00000014-00110001

Indicates: An out-of-band (OOB) datagram (expedited data) contained a connection alias or a connection identifier that does not correspond to an existing connection. The connection was probably removed by the receiving side, because it assumed that the partner was not active.

User response: This condition may occur in the system, and no user response is necessary.

ANY0021E AnyNet initialization failed.

Error code: 00000015-0001000E

Indicates: SNA over TCP/IP failed to initialize because no transport provider is available.

User response: See previous messages in the message log for more information about transport provider problems.

Error code: 00000015-00010010

Indicates: SNA over TCP/IP failed to initialize due to insufficient storage.

User response: Too many applications might already be running. Remove some of the applications.

ANY0024E Communication with a partner node has been lost. Partner transport provider address=*address*.

Error code: 00000018-00050006

Indicates: The remote node inactivity polling has detected a loss of communication with the specified partner.

User response: Check the network for any problems. Ensure that the specified partner can be reached and is active. If necessary increase the value for the remote node inactivity poll interval or the unacknowledged datagram retry interval.

ANY0351E There should be exactly one [TRACE] entry.

Indicates: There is more than one [TRACE] entry in the AnyNet configuration file (ANYNET.CFN).

This message is issued during initialization and is not logged.

User Response: This error can occur if an SNA transport user has not been selected or if a system error occurred when a SNA transport user was selected. Use the AnyNet configuration tool to select the SNA transport user again. For additional information, see “Selecting the SNA Transport User” on page 35.

If you continue to have problems, reinstall SNA over TCP/IP. For additional assistance, contact IBM for service. See “Using IBM Support Services” on page 72 for more information.

ANY0352E Error *number* **while loading module** *module_name***.**

Indicates: AnyNet detected an error while loading the trace and logging module *module_name*. Error *number* is an internal error code.

This message is issued during initialization and is not logged.

User Response: Ensure that there is not a module of the same name earlier in the path. The most common cause of this error is that module *module_name* is corrupted or has been deleted.

To correct this problem, reinstall SNA over TCP/IP to obtain the correct configuration file. For additional assistance, contact IBM for service. See “Using IBM Support Services” on page 72 for more information. Error *number* is used by IBM service.

ANY0353E Error *number* **while getting address for procedure** *procedure_name***.**

Indicates: AnyNet detected an error while obtaining the address for procedure *procedure_name*. Error *number* is an internal error code.

This message is issued during initialization and is not logged.

User Response: To correct this problem, reinstall SNA over TCP/IP to obtain the correct configuration file. For additional assistance, contact IBM for service. See “Using IBM Support Services” on page 72 for more information. Error *number* is used by IBM service.

ANY0354E Error in opening configuration file ANYNET.CFN.

Indicates: AnyNet detected an error when opening the AnyNet configuration file (ANYNET.CFN).

This message is issued during initialization and is not logged.

User Response: Make sure that the ANYNET.CFN file exists in the directory that is specified on the ANYNETPATH environment variable.

This error can occur if an SNA transport user has not been selected or if a system error occurred when a SNA transport user was selected. Use the AnyNet configuration tool to select the SNA transport user again. For additional information, see “Selecting the SNA Transport User” on page 35.

If you continue to have problems, reinstall SNA over TCP/IP. For additional assistance, contact IBM for service. See “Using IBM Support Services” on page 72 for more information.

ANY0355E Configuration file syntax error in line number *number***.**

Indicates: The AnyNet configuration file (ANYNET.CFN) contains a syntax error. Messages may follow to provide more information about this error.

This message is issued during initialization and is not logged.

User Response: This error can occur if an SNA transport user has not been selected or if a system error occurred when a SNA transport user was selected. Use the AnyNet configuration tool to select the SNA transport user again. For additional information, see “Selecting the SNA Transport User” on page 35.

If you continue to have problems, reinstall SNA over TCP/IP. For additional assistance, contact IBM for service. See “Using IBM Support Services” on page 72 for more information.

ANY0356E Line with error = *line*

Indicates: The AnyNet configuration file (ANYNET.CFN) contains a syntax error in the displayed line.

This message is issued during initialization and is not logged.

User Response: See message ANY0355E.

ANY0357E Error in string “*string***.”**

Indicates: The AnyNet configuration file (ANYNET.CFN) contains a syntax error in the displayed string.

This message is issued during initialization and is not logged.

User Response: See message ANY0355E.

ANY0358E Line is longer than *number* **bytes.**

Indicates: A line in the AnyNet configuration file (ANYNET.CFN) is too long. *number* is the maximum number of bytes allowed in a line.

This message is issued during initialization and is not logged.

User Response: See message ANY0355E.

ANY0359E Error in reading line.

Indicates: The AnyNet configuration file (ANYNET.CFN) contains a syntax error.

This message is issued during initialization and is not logged.

User Response: See message ANY0355E.

ANY0360E Unrecognized keyword in configuration file.

Indicates: The AnyNet configuration file (ANYNET.CFN) contains an unrecognized keyword.

This message is issued during initialization and is not logged.

User Response: See message ANY0355E.

ANY0361E More parameters expected for this or previous entry.

Indicates: The AnyNet configuration file (ANYNET.CFN) contains a syntax error.

This message is issued during initialization and is not logged.

User Response: See message ANY0355E.

ANY0362E More parameters expected for this entry.

Indicates: The AnyNet configuration file (ANYNET.CFN) contains a syntax error.

This message is issued during initialization and is not logged.

User Response: See message ANY0355E.

ANY0363E Too many parameters for this entry.

Indicates: The AnyNet configuration file (ANYNET.CFN) contains a syntax error.

This message is issued during initialization and is not logged.

User Response: See message ANY0355E.

ANY0364E Fully qualified name of ANYNET.CFN cannot be longer than *number*.

Indicates: The path name that AnyNet uses to determine the location of the AnyNet configuration file (ANYNET.CFN) is too long. *number* is the maximum number of bytes allowed in the path name. This directory is set during installation and saved in the ANYNETPATH environment variable.

This message is issued during initialization and is not logged.

User Response: Reinstall SNA over TCP/IP in a directory closer to the root directory, or use shorter directory names for the directories in the path to the AnyNet directory.

ANY0365E ANYNETPATH environment variable not defined properly.

Indicates: The path name that AnyNet uses to determine the location of the AnyNet configuration file (ANYNET.CFN) is not defined correctly.

This message is issued during initialization and is not logged.

User Response: Correct the ANYNETPATH environment variable by using **one** of the following methods:

Ensure that the following line is in the AUTOEXEC.BAT file:

SET ANYNETPATH=*c:\anynet*

c:\anynet is the directory where SNA over TCP/IP is installed. The default is C:\ANYNET.

Reinstall SNA over TCP/IP.

ANY0366E ANYNETPATH environment variable not defined.

Indicates: The pathname that AnyNet uses to determine the location of the AnyNet configuration file (ANYNET.CFN) is not defined.

This message is issued during initialization and is not logged.

User Response: Define the ANYNETPATH environment variable by using **one** of the following methods:

Add the following line to the AUTOEXEC.BAT file:

SET ANYNETPATH=*c:\anynet*

c:\anynet is the directory where SNA over TCP/IP is installed. The default is C:\ANYNET.

Reinstall SNA over TCP/IP.

ANY0367E Out of memory.

Indicates: This message is issued during initialization and indicates that the workstation has run out of useable memory.

User Response: Reduce the number of active applications, or erase unnecessary files on the disk. If this does not correct the problem, try rebooting your system.

ANY0999E Message file not found.

Indicates: The AnyNet message or trace file is missing or corrupted.

This message is issued during initialization and is not logged. Initialization continues.

User response: Reinstall SNA over TCP/IP. If you do not reinstall SNA over TCP/IP, message and trace logging will be affected:

If AnyNet message file ANY.MR0 is missing or corrupted, the following message is recorded in the message log every time AnyNet attempts to log a message:

ERROR 1 : 2nd level message retrieval failure: Check message files.

If AnyNet trace file ANYTRC.MR0 is missing or corrupted, ERROR 100 is recorded repeatedly in the trace log file.

Installation and Configuration Message Descriptions

If a problem occurs during installation or configuration, messages are displayed in a window describing the problem. These messages are not logged by APPC Networking Services for Windows.

ANY0450W AnyNet Configuration Tool is already active.

Indicates: The AnyNet Configuration Tool is already running on your system.

User response: No action is required unless you want to close the first instance of the tool and restart it.

ANY0451E Unrecoverable system error occurred. See messages in logfile *logfile* for details.

Indicates: An unrecoverable system error was detected while running the installation program, AnyNet Configuration Tool, or the CID Setup Tool.

logfile is *x*:\ANYCFGW.LOG, where *x* is the system file drive.

User response: Use the messages logged in *logfile* to assist you in correcting the problem. If you need additional assistance, contact IBM for service. See “Using IBM Support Services” on page 72 for information.

ANY0452E Version *number* of IBM APPC Networking Services for Windows is required. Version *number* is installed.

Indicates: The required version of APPC Networking Services for Windows is not installed on your system.

User response: Install the required version number.

You can continue the installation or configuration of SNA over TCP/IP. When the installation or configuration is completed:

1. Install APPC Networking Services.
2. Open the AnyNet configuration tool.
3. Select APPC Networking Services as your SNA transport user.
4. Press the "OK" push button to save.

ANY0453W Version *number* **of IBM TCP/IP for DOS is required. Version** *number* **is installed.**

Indicates: The required version of TCP/IP for DOS is not installed on your system.

User response: Install the required version number.

You can continue the installation of SNA over TCP/IP. When the installation is completed, install the correct version of TCP/IP for DOS before you use AnyNet.

ANY0454E Version *number* **of Microsoft Windows is required. Version** *number* **is installed.**

Indicates: The required version of Microsoft Windows is not installed on your system.

User response: Install the required version number.

ANY0459E IBM APPC Networking Services for Windows could not be found on this workstation.

Indicates: AnyNet cannot find APPC Networking Services for Windows on your system. Possible causes of this error are:

Networking Services is not installed.

Networking Services is installed, but the VERSION file has been corrupted or deleted. AnyNet checks this file to confirm that Networking Services is installed.

User response:

Install Networking Services if it is not installed.

You can continue the installation or configuration of SNA over TCP/IP. When the installation or configuration is completed:

1. Install APPC Networking Services.
2. Open the AnyNet configuration tool.
3. Select APPC Networking Services as your SNA transport user.
4. Press the "OK" push button to save.

If Networking Services is installed ensure that the VERSION file exists and includes the version number.

ANY0460W IBM TCP/IP for DOS could not be found on this workstation. A TCP/IP product is required.

Indicates: TCP/IP for DOS cannot be found on your system.

Note: If you are using a non-IBM TCP/IP product, you can ignore this message.

See “Software Requirements” on page 9 for a list of non-IBM TCP/IP products that can be used with APPC over TCP/IP for Windows.

Possible causes of this error are:

TCP/IP for DOS is not installed.

TCP/IP for DOS is installed, but an incorrect directory was specified.

TCP/IP for DOS is installed, but the ETC environment variable has been deleted from the AUTOEXEC.BAT file.

TCP/IP for DOS is installed, but the TCPDOS.INI file has been corrupted or deleted. AnyNet checks this file to confirm that IBM TCP/IP is installed.

User response:

Install TCP/IP for DOS if it is not installed.

You can continue the installation of SNA over TCP/IP. When the installation is completed, install TCP/IP for DOS before you use AnyNet.

If TCP/IP for DOS is installed, ensure that the ETC environment variable is in the AUTOEXEC.BAT file and is pointing to the ETC subdirectory of TCP/IP for DOS.

ANY0461W The program and group *program_name, group_name* **could not be added to the Windows Program Manager. See messages in logfile** *logfile* **for details.**

Indicates: The installation program attempted to add *program_name* to *group_name* under the Windows Program Manager.

User response: Use the messages logged in *logfile* for a description of how to manually add the program and group.

ANY0462W Language DLL *dll_name* **could not be loaded. English will be used.**

Indicates: There was an error loading the language dynamic link library (DLL) *dll_name*.

User response: Ensure that *dll_name* exists on your system and run the tool again. If problems persist, contact IBM for service. See “Using IBM Support Services” on page 72 for information.

ANY0465E Program cannot continue due to insufficient memory.

Indicates: This message indicates that you are out of memory or that a program error occurred.

User response: Increase memory. If problems persist, contact IBM for service. See “Using IBM Support Services” on page 72 for information.

ANY0467W Library Reader or BookManager READ could not be found on this workstation.

Indicates: This message is issued when you install the online documentation, and AnyNet cannot find the Library Reader or BookManager READ on your system.

Possible causes are:

The Library Reader or BookManager READ is not installed.

The directory specified for the Library Reader or BookManager READ during installation is not correct.

User response: Do one of the following:

Install the Library Reader.

Specify the correct path and file name for the Library Reader or BookManager READ:

1. Click mouse button 2 on the “AnyNet Books” icon.
2. Click mouse button 1 on “Open.”
3. Click on “Settings” and select the “File” option.
4. Specify the correct path and file name in the space provided.
   - For the Library Reader, the file name is READIBM.PIF.
   - For BookManager READ, the file name is BOOKMGR.PIF.
5. Close the window to save.

ANY0478E IBM Personal Communications for Windows could not be found on this workstation.

Indicates: AnyNet cannot find Personal Communications on your system. Possible causes of this error are:

Personal Communications is not installed.

Personal Communications is installed, but the PCSMAINT.TXT file has been corrupted or deleted. AnyNet checks this file to confirm that Personal Communications is installed.

User response:

Install Personal Communications if it is not installed.

You can continue the installation or configuration of SNA over TCP/IP. When the installation or configuration is completed:

1. Install Personal Communications.
2. Open the AnyNet configuration tool.
3. Select Personal Communications as your SNA transport user.
4. Press the “OK” push button to save.

If Personal Communications is installed:

- Ensure that you specified the correct directory.
- Ensure that the PCSMAINT.TXT file exists in the Personal Communications directory and contains a VERSION section with one of the following definitions:
  E3270VER=*version_number*
  E5250VER=*version_number*

ANY0479E ANY0480E

ANY0479E Version *number* **of IBM Personal Communications for Windows is required. Version** *number* **is installed.**

Indicates: The required version of Personal Communications is not installed on your system.

User response: Install the required version number.

You can continue the installation or configuration of SNA over TCP/IP. When the installation or configuration is completed:

1. Install Personal Communications.
2. Open the AnyNet configuration tool.
3. Select Personal Communications as your SNA transport user.
4. Press the "OK" push button to save.

ANY0480E The specified value for response file keyword *keyword_value* **is not valid.**

Indicates: The response file used by the AnyNet configuration, installation, and distribution (CID) setup tool contains an error. This error occurs for the following reasons:

The response file was edited manually and contains a keyword value that is not valid.

The response file variable SNA_TRANSPORT_USER was set to PCS (Personal Communications) or NSW (APPC Networking Services), but the required version of the selected SNA transport user is not installed on your workstation.

User response:

If the response file contains a keyword value that is not valid, use the AnyNet CID setup tool to recreate the response file. For more information, see "Using the CID Setup Tool to Automate Installation" on page 24.

If the required version of the selected SNA transport user is not installed, install either APPC Networking Services or Personal Communications on your workstation before installing AnyNet. For information on product versions and releases, see "Software Requirements" on page 9.

Appendix B. Automatic Installation Keywords, Parameters, and Return Codes

This appendix includes the following sections:

"Response File Keywords"

"Parameters for the AnyNet INSTALLW.EXE and ANYNVDMD.EXE Programs" on page 93

"Return Codes from the AnyNet ANYNVDMD.EXE Program" on page 94

Response File Keywords

This section describes keywords that are generated by the AnyNet configuration, install, and distribution (CID) setup tool for an automatic installation of the SNA-over-TCP/IP product, online and PostScript documentation, and IBM Library Reader. To specify keywords, use the following syntax:

keyword=user-specified value

Attention

It is recommended that you use the CID setup tool to generate response file keywords. See "Using the CID Setup Tool to Automate Installation" on page 24 for more information on this tool.

Install, Update, Restore, and Delete Keywords

The following keywords are used for an automatic install, update, restore, or delete of the SNA-over-TCP/IP product, online and PostScript documentation, and IBM Library Reader:

CFGUPDATE=AUTO | MANUAL

This keyword is required for Install and Update.

If you specify **AUTO**, the AUTOEXEC.BAT file is automatically changed during installation, if necessary.

If you specify **MANUAL**, the file AUTOEXEC.ADD is created on the workstation and contains the updates you need to make to AUTOEXEC.BAT.

Note: If you specify **MANUAL** and reboot the workstation after an automatic install, the product will not be enabled until you change the AUTOEXEC.ADD file and reboot the workstation.

© Copyright IBM Corp. 1996

Response File Keywords

COMP=SNA over TCP/IP for Windows Product
| **SNA over TCP/IP for Windows Online Documentation**
| **SNA over TCP/IP for Windows Printable Documentation**
| **Library Reader for Windows**
This keyword is required for an Install, Update, Restore, or Delete and indicates the specified components.

DELETEBACKUP=YES | NO
This keyword is required for Delete.

YES indicates that only the backup copy of the product that was saved during an update should be deleted.

NO indicates that the entire product should be deleted from the workstation.

FILE=*path*
This keyword is required for Install.

path is the directory in which the SNA over TCP/IP product, documentation, and Library Reader are to be installed.

IGNOREPREREQERRORS=YES | NO
This keyword is required for Install.

NO is the default, and indicates that the installation should terminate if the required version of IBM TCP/IP for DOS is not installed on the workstation.

YES indicates that the product should be installed even if the required version of IBM TCP/IP for DOS is not installed.

Note: If you are using a non-IBM TCP/IP product, this keyword should be set to **YES**. See "Software Requirements" on page 9 for a list of TCP/IP products that can be used with SNA over TCP/IP.

OVERWRITE=YES | NO
This keyword is required for Install and Update.

YES indicates that files with the same names as product files are overwritten by the installation.

NO indicates that existing files on the workstation are not overwritten. If **NO** is selected, the product may not be installed properly.

SAVEBACKUP=YES | NO
This keyword is required for Update.

YES indicates that a backup copy of the product will be saved on the workstation before the update is done.

NO indicates that no backup copy will be saved.

TCPDIR=*path*
This keyword is required for Install if you are using IBM TCP/IP for DOS.

Note: If you are using a non-IBM TCP/IP product, the IGNOREPREREQERRORS

keyword should be set to **YES**. See "Software Requirements" on page 9 for a list of TCP/IP products that can be used with SNA over TCP/IP.

path is the directory in which IBM TCP/IP for DOS is installed.

Configuration Keywords

The following keywords are used for an automatic installation of the SNA-over-TCP/IP product and specify configuration settings:

Keyword:	Configuration Setting:
SNASUFFIX	SNA Domain Name Suffix
CONN_RETRY_SECS	Connection Retry Duration
CONNWAIT_SECS	Connection Wait Time Limit
INACTIVITY_TIMER_SECS	Remote Node Inactivity Poll Interval
SNA_TRANSPORT_USER:	SNA Transport User PCS Personal Communications NSW APPC Networking Services NCF Not configured
UNACKED_DG_RETRY_SECS	Unacknowledged Datagram Retry Interval
UNSENT_DG_RETRY_SECS	Unsent Datagram Retry Interval

For additional information on the SNA transport user, see "Selecting the SNA Transport User" on page 35. See "Configuring SNA-over-TCP/IP Environment Settings" on page 36 for a description of all other configuration settings.

Parameters for the AnyNet INSTALLW.EXE and ANYNVDMD.EXE Programs

The AnyNet INSTALLW.EXE program is executed during automatic installation.

If you are using NVDM/2, the ANYNVDMD.EXE program is executed. The parameters for this program are specified in the change file profile. See "Creating a Change File Profile for NetView Distribution Manager/2" on page 26 for information about change files.

The following parameters are valid for the INSTALLW.EXE and ANYNVDMD.EXE programs:

/A:*action*
/L1:*installation error log file*
/L2:*installation history log file*
/L3:*configuration error log file*
/R:*response file*
/S:*source directory*
/X

Parameter descriptions follow:

Parameters and Return Codes

/**A:***action* Specifies the action that is being performed.

Valid values for this parameter are:

I Install an AnyNet product for the first time
U Update a previously installed AnyNet product
R Restore a previous version of an AnyNet product
D Delete an installed AnyNet product

Note: This parameter is required.

/**L1:***installation error log file*
Specifies the fully qualified path for the installation error log file.

Note: This parameter is required for the ANYNVDMD.EXE program only.

/**L2:***installation history log file*
Specifies the fully qualified path for the installation history log file.

/**L3:***configuration error log file*
Specifies the fully qualified path for the configuration error log file.

Note: This parameter is required for the ANYNVDMD.EXE program only.

/**R:***response file*
Specifies the fully qualified path to the response file.

Note: This parameter is required.

/**S:***source directory*
Specifies the directory that contains the source files to be installed. This should be the directory on your code server that contains the AnyNet product images.

Note: This parameter is required.

/**X:** Specifies that the action is unattended.

Note: This parameter is required.

Return Codes from the AnyNet ANYNVDMD.EXE Program

The Anynet ANYNVDMD.EXE program issues the following 2-byte hexadecimal return codes. When using NetView DM, these codes are returned to the NetView DM message log.

00 00 Action was successful.

00 04 Action was successful. Warning messages were logged.

12 00 Log file specified on the \L1 or \L3 parameter is not valid.

16 00 Possible causes of this error are:

\L1 or \L3 parameter was not specified.

The INSTALLW.EXE program was not found in the same directory as the ANYNVDMD.EXE program.

There was an error starting the installation program, and messages were logged.

16 04 Possible causes of this error are:

Unexpected condition detected. Messages were logged.
Installation was stopped. Messages were logged.
Action was unsuccessful. Messages were logged.

FE 00 Action was successful. Rebooting the workstation is required.

FE 04 Action was successful. Rebooting the workstation is required. Warning messages were logged.

Appendix C. Syntax Conventions

This appendix describes the syntax conventions that are used in this book.

To help you locate and identify information easily, this book uses the following visual cues and standard text formats:

Font Type	Used for
bold	Return codes Socket call names
CAPS	File names Default file paths
ITALICS	User-supplied file paths
italics	User-supplied variables other than file paths Message variables Titles of books
monospace	Required commands or file path specifications Examples
“quotes”	Icon names Window names or options Menu titles or options

© Copyright IBM Corp. 1996

Appendix D. Trace Examples for APPC Networking Services

This appendix contains the following trace examples:

"MPTNASNA Trace" on page 100
"MPTNCMM Trace" on page 101
"MPTNBTCP Trace" on page 101
"Successful Session Initiation" on page 104
"Unsuccessful Session Initiation" on page 112

© Copyright IBM Corp. 1996

MPTNASNA Example

Component Trace Examples

This section includes examples of MPTNASNA, MPTNCMM, and MPTNBTCP component traces.

MPTNASNA Trace

Figure 20 shows an example of an MPTNASNA trace. Notes following the example provide additional information.

```
                                    (1)
[2]  2/1  11:24:25.69 src=ANYNET, len=172, rec_id= 4 1h, thr_id=9886:1
   (2)                (3)
FUNC CALL          MPTNASNA
              (4)
SEND: BIND FROM A_SNA,  1 1 3
            (5)                                        (6)                    (7)
   2F   1 3     6B8                        </.....k..          >    <.......,..          >

[9]  2/1  11:24:25.8  src=ANYNET, len=94, rec_id= 4 1h, thr_id=9886:1
FUNC CALL          MPTNASNA
                                (8)
M_CONNECT_DC,  1 1 3 348F B

[14]  2/1  11:24:25.96 src=ANYNET, len=94, rec_id= 4 1h, thr_id=9886:1
  (9)
FUNC RET           MPTNASNA
                      (1 )
M_CONNECT_DC, RC_PENDING,  1 1 3 348F B
```

Figure 20. MPTNASNA Component Trace Example for APPC Networking Services

Note	Description
(1)	Networking Services Trace Facility trace log header.
(2)	FUNC CALL indicates that a function was just entered.
(3)	MPTNASNA is the AnyNet component name creating the trace entry.
(4)	Send is the function name. An SNA BIND request was received FOR LFSID 010103.
(5)	This is a hexadecimal dump of the SNA transmission header (2F000103 0000) and request/response header (6B80 00).
(6)	The hexadecimal dump translated to ASCII.
(7)	The hexadecimal dump translated to EBCDIC.
(8)	The control block associated with the FUNC CALL m_connect_dc is at address 348F0B00 (in segment:offset order) and the LFSID IS 010103.
(9)	FUNC RET indicates exit from a function.
(10)	The m_connect_dc FUNC CALL returned a return code of pending for the control block at 348F0B00 and LFSID OF 010103.

MPTNBTCP Example

MPTNCMM Trace

Figure 21 shows an example of an MPTNCMM trace. Notes following the example provide additional information.

```
                                    (1)
[1 ]  2/1  11:24:25.86 src=ANYNET, len=11 8, rec_id= 4 1h, thr_id=9886:1
   (2)                (3)
FUNC CALL          MPTNCMM
                 (4)
CMMDREQ(M_CONNECT) 348F B
                        (5)                                  (6)                          (7)
          B 11 E4 E2C9C2D4 D5D94BD5    <..............K.>    <.......USIBMNR.N>
  D9C9F4F  F1F F     1  88             <................>    <RI4 1  ...h.....>
        6B8   31   13 7B  B D1B3 1    <....k..1........>    <...........J..>
   85858   1 6 2                423    <...............#>    <.ee............>
     DE4 E2C9C2D4 D5D94BD3 E4D7F1F4    <..........K.....>    <...USIBMNR.LUP14>
  2F   9 2 E2D5C1E2 E5C3D4C7  9 3 1DC    </...............>    <....SNASVCMG....>
  1BF 69B4 7FA8 E 4 E4E2C9C2 D4D5D94B    <..i............K>    <. .."y..USIBMNR.>
  D3E4D7F1 F4 A13   DF2E9F26 1   41      <...........&..A.>    <LUP14...........>
   1 E4E2 C9C2D4D5 D94BD5D9 C9F4F F1    <.........K......>    <..USIBMNR.NRI4 1>
  F F 6 16 8FDC1BF  69B47FA8  DE4E2C9    <.. .....i.......>    <  -.... .."y.USI>
  C2D4D5D9 4BD3E4D7 F1F4 4   473913      <....K.......G9..>    <BMNR.LUP14......>
  534E4138 46444331 42463 36 39423437    <SNA8FDC1BF 69B47>    <.+..............>
 464138                                  <FA8..                >    <.....                >

[13]  2/1  11:24:25.96 src=ANYNET, len=94, rec_id= 4 1h, thr_id=9886:1
   (8)
FUNC RET           MPTNCMM
                 (9)
CMMDREQ(M_CONNECT) 348F B   pending
```

Figure 21. MPTNCMM Component Trace Example for APPC Networking Services

Note	Description
(1)	Networking Services Trace Facility trace log header.
(2)	FUNC CALL indicates that a function was just entered.
(3)	MPTNCMM is the AnyNet component name creating the trace entry.
(4)	CMMDREQ is the function name. An M_CONNECT downcall was made. The control block is at address 348F0B00 (in segment:offset order).
(5)	This is a hexadecimal dump of the M_CONNECT downcall.
(6)	The hexadecimal dump translated to ASCII.
(7)	The hexadecimal dump translated to EBCDIC.
(8)	FUNC RET indicates exit from a function.
(9)	The CMMDREQ FUNC RET returns a status of pending for the control block at 348F0B00.

MPTNBTCP Trace

Figure 22 on page 102 shows an example of an MPTNBTCP trace. Notes following the example provide additional information.

MPTNBTCP Example

```
                                                  (1)
[15]  2/1  11:24:26. 7 src=ANYNET, len=94, rec_id= 4 1h, thr_id=1 118:1
    (2)                  (3)
PROV CALL          MPTNBTCP
         (4)
sock_gethostbyname
                        (5)                                  (6)                    (7)
   4E524934 3 313 3  2E555349 424D4E52    <NRI4 1  .USIBMNR>    <+............(+.>
   2E534E41 2E49424D 2E434F4D                <.SNA.IBM.COM       >    <..+....(...(       >

[16]  2/1  11:24:26.23 src=ANYNET, len=94, rec_id= 4 1h, thr_id=1 118:1
    (8)
PROV RET           MPTNBTCP
          (9)
 =sock_gethostbyname
   4E524934 3 313 3  2E555349 424D4E52    <NRI4 1  .USIBMNR>    <+............(+.>
   2E534E41 2E49424D 2E434F4D                <.SNA.IBM.COM       >    <..+....(...(       >

[2 ]  2/1  11:24:26.77 src=ANYNET, len=94, rec_id= 4 1h, thr_id=2 78:1
    (1 )
FUNC CALL          MPTNBTCP
           (11)
ConnectFSM(362F A18)

[32]  2/1  11:24:27.2  src=ANYNET, len=172, rec_id= 4 2h, thr_id=2 78:1
      (12)
<==SEND=====   MPTNBTCP
       (13)              (14)          (15)                (16)  (17)   (18)   (19)
9.67.98.162         1 24    9.67.98.198            397      7 CMDLPREF 4
        (2 )
        EA                                          <....                    >   <....                    >

[33]  2/1  11:24:27.31 src=ANYNET, len=1264, rec_id= 4 2h, thr_id=2 78:1
<==SEND=====   MPTNBTCP
                                                                                  (21)
9.67.98.162         1 24    9.67.98.198            397      7 MPTNCONN 23
                       (22)
   8  A  E6 2A B 111 E4E2C9C2 D4D5D94B    <.... ..........K>    <...W....USIBMNR.>
   D5D9C9F4 F F1F F   1 B 1 E E4E2C9C2    <................>    <NRI4 1  ....USIB>
   D4D5D94B D3E4D7F1 F4 1 712 5B 2        <...K........ffl...>    <MNR.LUP14...$...>
     F      1 5       6       1 52       <......... .... .>    <.........-......>
    5 8   6  1 1 A 2   8C     6B8   31    <............k..1>    <..............,...>
     13 7B  B D1B3 1   85858   1 6 2      <................>    <.....J...ee.....>
                423      DE4 E2C9C2D4    <.......#........>    <...........USIBM>
   D5D94BD3 E4D7F1F4 2F   9 2 E2D5C1E2    <..K...../.......>    <NR.LUP14....SNAS>
   E5C3D4C7  9 3 1DC 1BF 69B4 7FA8 E 4    <..........i.....>    <VCMG..... .."y..>
   E4E2C9C2 D4D5D94B D3E4D7F1 F4 A13      <.......K........>    <USIBMNR.LUP14...>
   DF2E9F26 1   41     1 E4E2 C9C2D4D5    <...&..A.........>    <...........USIBMN>
   D94BD5D9 C9F4F F1 F F 6 16 8FDC1BF     <.K........ .....>    <R.NRI4 1  -.... >
   69B47FA8  DE4E2C9 C2D4D5D9 4BD3E4D7    <i...........K...>    <.."y.USIBMNR.LUP>
   F1F418     5 119  A   A    1 32 1     <............. .>    <14.............>
  831A       584                                  <......                  >   <c....d                  >

[23]  2/1  11:24:26.83 src=ANYNET, len=94, rec_id= 4 1h, thr_id=2 78:1
  (23)
FUNC RET           MPTNBTCP
        (24)
GO=ConnectFSM
```

Figure 22. MPTNBTCP Component Trace Example for APPC Networking Services

MPTNBTCP Example

Note	Description
(1)	Networking Services Trace Facility trace log header.
(2)	PROV CALL indicates this is a call to the sockets interface.
(3)	MPTNBTCP is the AnyNet component name creating the trace entry.
(4)	The PROV CALL name is sock_gethostbyname. MPTNBTCP is requesting TCP/IP make the LUNAME-to-IP address mapping using the domain name server or the HOST file.
(5)	This is a hexadecimal dump of the domain name.
(6)	The hexadecimal dump translated to ASCII.
(7)	The hexadecimal dump translated to EBCDIC.
(8)	PROV RET indicates the return from a socket call.
(9)	The sock_gethostbyname socket call returns a 0 which indicates success. The domain name was found in either the DNS or HOSTS file.
(10)	FUNC CALL indicates that a function was just entered.
(11)	ConnectFSM is the name of the function. The control block is at address 362F0A18 (in segment:offset order).
(12)	Indicates data being sent on the TCP connection.
(13)	The source IP address for the connection.
(14)	The source port number for the connection.
(15)	The destination IP address for the connection.
(16)	The destination port number for the connection.
(17)	The socket number for the connection.
(18)	The data being traced is an MPTN command prefix.
(19)	Length of the data being sent.
(20)	A hexadecimal dump of the four bytes of data being sent.
(21)	The data being traced is an MPTN connect request.
(22)	A hexadecimal dump of the MPTN connect.
(23)	FUNC RET indicates exit from a function.
(24)	The function ConnectFSM is returning a successful state.

Successful Initiation Example

Session Initiation Trace Examples

This section includes an example of a successful and unsuccessful session initiation.

Successful Session Initiation

The following example shows an AnyNet trace generated during a successful outbound session initiation. An explanation of the trace output, by line number, follows the example.

```
[1]  2/1  11:24:25.65 src=ANYNET, len=94, rec_id= 4 1h, thr_id=9886:1
FUNC CALL         MPTNASNA
GETBTUS LOCAL 82 1 REMOTE 82 1

[2]  2/1  11:24:25.69 src=ANYNET, len=172, rec_id= 4 1h, thr_id=9886:1
FUNC CALL         MPTNASNA
SEND: BIND FROM A_SNA,  1 1 3
   2F   1 3     6B8                          </.....k..            >    <......,..            >

[3]  2/1  11:24:25.69 src=ANYNET, len=94, rec_id= 4 1h, thr_id=9886:1
FUNC CALL         MPTNASNA
M_CREATE_DC,  1 1 3


[4]  2/1  11:24:25.69 src=ANYNET, len=64 , rec_id= 4 1h, thr_id=9886:1
FUNC CALL         MPTNCMM
CMMDREQ(M_CREATE)
           B         2AA61 1715 1      <...........a....>    <............./....>
    1   5 1     6        5 1     2      <...... ....... .>    <......-.........>
    5 1      5 1                      <................>    <................>
    1   1                      1      <................>    <................>
                                   <................>    <................>
                                   <................>    <................>
                                                  <....                  >   <....                  >


[5]  2/1  11:24:25.75 src=ANYNET, len=94, rec_id= 4 1h, thr_id=9886:1
FUNC RET          MPTNCMM
CMMDREQ(M_CREATE) 348F B


[6]  2/1  11:24:25.8  src=ANYNET, len=94, rec_id= 4 1h, thr_id=9886:1
FUNC CALL         MPTNASNA
M_BIND_DC,  1 1 3 348F B

[7]  2/1  11:24:25.8  src=ANYNET, len=25 , rec_id= 4 1h, thr_id=9886:1
FUNC CALL         MPTNCMM
CMMDREQ(M_BIND) 348F B
           B 1 DE4 E2C9C2D4 D5D94BD3    <..............K.>    <.......USIBMNR.L>
   E4D7F1F4                                <...........          >    <UP14.......          >
```

Successful Initiation Example

```
[8]  2/1  11:24:25.8  src=ANYNET, len=94, rec_id= 4 1h, thr_id=9886:1
FUNC RET          MPTNCMM
CMMDREQ(M_BIND) 348F B

[9]  2/1  11:24:25.8  src=ANYNET, len=94, rec_id= 4 1h, thr_id=9886:1
FUNC CALL         MPTNASNA
M_CONNECT_DC,  1 1 3 348F B


[1 ]  2/1  11:24:25.86 src=ANYNET, len=11 8, rec_id= 4 1h, thr_id=9886:1
FUNC CALL         MPTNCMM
CMMDREQ(M_CONNECT) 348F B
          B 11 E4 E2C9C2D4 D5D94BD5    <.............K.>    <.......USIBMNR.N>
   D9C9F4F  F1F F     1  88             <.............>    <RI4 1  ...h.....>
         6B8   31   13 7B  B D1B3 1    <....k..1.......>    <....,.....J..>
    85858   1 6 2              423    <.............#>    <.ee.............>
      DE4 E2C9C2D4 D5D94BD3 E4D7F1F4    <.........K.....>    <...USIBMNR.LUP14>
   2F   9 2 E2D5C1E2 E5C3D4C7  9 3 1DC    </.............>    <....SNASVCMG....>
   1BF 69B4 7FA8 E 4 E4E2C9C2 D4D5D94B    <..i...........K>    <. .."y..USIBMNR.>
   D3E4D7F1 F4 A13   DF2E9F26 1   41      <..........&..A.>    <LUP14..........>
    1 E4E2 C9C2D4D5 D94BD5D9 C9F4F F1    <.........K......>    <..USIBMNR.NRI4 1>
   F F 6 16 8FDC1BF  69B47FA8  DE4E2C9    <.. .....i.......>    <  -.... .."y.USI>
   C2D4D5D9 4BD3E4D7 F1F4 4   473913      <....K.......G9..>    <BMNR.LUP14......>
   534E4138 46444331 42463 36 39423437    <SNA8FDC1BF 69B47>    <.+.............>
  464138                                 <FA8..                >   <.....                >


[11]  2/1  11:24:25.91 src=ANYNET, len=94, rec_id= 4 1h, thr_id=9886:1
FUNC CALL         MPTNCMM
CMMDCALL(P_LOCATE) 348F B

[12]  2/1  11:24:25.96 src=ANYNET, len=94, rec_id= 4 1h, thr_id=9886:1
FUNC RET          MPTNCMM
CMMDCALL(P_LOCATE) 348F B


[13]  2/1  11:24:25.96 src=ANYNET, len=94, rec_id= 4 1h, thr_id=9886:1
FUNC RET          MPTNCMM
CMMDREQ(M_CONNECT) 348F B   pending

[14]  2/1  11:24:25.96 src=ANYNET, len=94, rec_id= 4 1h, thr_id=9886:1
FUNC RET          MPTNASNA
M_CONNECT_DC, RC_PENDING,  1 1 3 348F B


[15]  2/1  11:24:26. 7 src=ANYNET, len=94, rec_id= 4 1h, thr_id=1 118:1
PROV CALL         MPTNBTCP
sock_gethostbyname
   4E524934 3 313 3  2E555349 424D4E52    <NRI4 1  .USIBMNR>    <+...........(+.>
   2E534E41 2E49424D 2E434F4D                 <.SNA.IBM.COM       >    <..+....(...(       >


[16]  2/1  11:24:26.23 src=ANYNET, len=94, rec_id= 4 1h, thr_id=1 118:1
PROV RET          MPTNBTCP
 =sock_gethostbyname
   4E524934 3 313 3  2E555349 424D4E52    <NRI4 1  .USIBMNR>    <+...........(+.>
   2E534E41 2E49424D 2E434F4D                 <.SNA.IBM.COM       >    <..+....(...(       >

[17]  2/1  11:24:26.23 src=ANYNET, len=94, rec_id= 4 1h, thr_id=1 118:1
FUNC CALL         MPTNCMM
CMMUREQ(P_LOCATE_UC) 348F B
```

Successful Initiation Example

```
[18]  2/1  11:24:26.5  src=ANYNET, len=94, rec_id= 4 1h, thr_id=1 118:1
FUNC CALL          MPTNCMM
CMMDCALL(P_CONNECT) 348F B

[19]  2/1  11:24:26.56 src=ANYNET, len=94, rec_id= 4 1h, thr_id=1 118:1
FUNC RET           MPTNCMM
CMMDCALL(P_CONNECT) 348F B    returned pending


[2 ]  2/1  11:24:26.77 src=ANYNET, len=94, rec_id= 4 1h, thr_id=2 78:1
FUNC CALL          MPTNBTCP
ConnectFSM(362F A18)

[21]  2/1  11:24:26.77 src=ANYNET, len=94, rec_id= 4 1h, thr_id=2 78:1
PROV CALL          MPTNBTCP
socket


[22]  2/1  11:24:26.83 src=ANYNET, len=94, rec_id= 4 1h, thr_id=2 78:1
PROV RET           MPTNBTCP
socket(7)

[23]  2/1  11:24:26.83 src=ANYNET, len=94, rec_id= 4 1h, thr_id=2 78:1
FUNC RET           MPTNBTCP
GO=ConnectFSM

[24]  2/1  11:24:26.83 src=ANYNET, len=94, rec_id= 4 1h, thr_id=2 78:1
FUNC CALL          MPTNBTCP
ConnectFSM1(362F A18)


[25]  2/1  11:24:26.88 src=ANYNET, len=94, rec_id= 4 1h, thr_id=2 78:1
PROV CALL          MPTNBTCP
sock_connect(7, 9.67.98.198)

[26]  2/1  11:24:26.88 src=ANYNET, len=94, rec_id= 4 1h, thr_id=2 78:1
PROV RET           MPTNBTCP
-1=sock_connect(7, 9.67.98.198)

[27]  2/1  11:24:26.93 src=ANYNET, len=94, rec_id= 4 1h, thr_id=2 78:1
FUNC RET           MPTNBTCP
YIELD=ConnectFSM1


[28]  2/1  11:24:27.1  src=ANYNET, len=94, rec_id= 4 1h, thr_id=2 78:1
PROV RET           MPTNBTCP
sock_select( ,1)

[29]  2/1  11:24:27.1  src=ANYNET, len=94, rec_id= 4 1h, thr_id=2 78:1
FUNC CALL          MPTNBTCP
ConnectFSM2(362F A18)
```

Successful Initiation Example

```
[3 ]  2/1  11:24:27.15 src=ANYNET, len=94, rec_id= 4 1h, thr_id=2 78:1
PROV CALL          MPTNBTCP
sock_getsockname(7)

[31]  2/1  11:24:27.15 src=ANYNET, len=94, rec_id= 4 1h, thr_id=2 78:1
PROV RET           MPTNBTCP
sock_getsockname(7)

[32]  2/1  11:24:27.2  src=ANYNET, len=172, rec_id= 4 2h, thr_id=2 78:1
<==SEND=====   MPTNBTCP
9.67.98.162         1 24    9.67.98.198            397      7 CMDLPREF 4
       EA                                          <....                    >   <....                    >

[33]  2/1  11:24:27.31 src=ANYNET, len=1264, rec_id= 4 2h, thr_id=2 78:1
<==SEND=====   MPTNBTCP
9.67.98.162         1 24    9.67.98.198            397      7 MPTNCONN 23
   8  A  E6 2A B 111 E4E2C9C2 D4D5D94B    <.... ..........K>    <...W....USIBMNR.>
   D5D9C9F4 F F1F F   1 B 1 E E4E2C9C2    <................>    <NRI4 1  ....USIB>
   D4D5D94B D3E4D7F1 F4 1 712 5B 2        <...K........ffl...>    <MNR.LUP14...$...>
     F     1 5      6      1 52      <......... .... .>    <.........-......>
    5 8   6  1 1 A 2   8C     6B8   31    <............k..1>    <.............,...>
     13 7B  B D1B3 1   85858   1 6 2      <................>    <.....J...ee.....>
              423      DE4 E2C9C2D4    <.......#........>    <...........USIBM>
   D5D94BD3 E4D7F1F4 2F   9 2 E2D5C1E2    <..K...../.......>    <NR.LUP14....SNAS>
   E5C3D4C7  9 3 1DC 1BF 69B4 7FA8 E 4    <..........i.....>    <VCMG..... .."y..>
   E4E2C9C2 D4D5D94B D3E4D7F1 F4 A13      <.......K........>    <USIBMNR.LUP14...>
   DF2E9F26 1   41     1 E4E2 C9C2D4D5    <...&..A.........>    <..........USIBMN>
   D94BD5D9 C9F4F F1 F F 6 16 8FDC1BF     <.K........ .....>    <R.NRI4 1  -.... >
   69B47FA8  DE4E2C9 C2D4D5D9 4BD3E4D7    <i...........K...>    <.."y.USIBMNR.LUP>
   F1F418      5 119  A   A    1 32 1     <............. .>    <14............>
  831A       584                                  <......                  >   <c....d                  >

[34]  2/1  11:24:27.31 src=ANYNET, len=94, rec_id= 4 1h, thr_id=2 78:1
FUNC RET           MPTNBTCP
GO=ConnectFSM2

[35]  2/1  11:24:27.31 src=ANYNET, len=94, rec_id= 4 1h, thr_id=2 78:1
FUNC CALL          MPTNBTCP
PutData(362F A18)

[36]  2/1  11:24:27.37 src=ANYNET, len=94, rec_id= 4 1h, thr_id=2 78:1
PROV CALL          MPTNBTCP
sock_send(7, 4)

[37]  2/1  11:24:27.37 src=ANYNET, len=94, rec_id= 4 1h, thr_id=2 78:1
PROV RET           MPTNBTCP
4=sock_send(7)

[38]  2/1  11:24:27.37 src=ANYNET, len=94, rec_id= 4 1h, thr_id=2 78:1
FUNC RET           MPTNBTCP
GO=PutData

[39]  2/1  11:24:27.37 src=ANYNET, len=94, rec_id= 4 1h, thr_id=2 78:1
FUNC CALL          MPTNBTCP
PutData1(362F A18)
```

Successful Initiation Example

```
[4 ]  2/1  11:24:27.37 src=ANYNET, len=94, rec_id= 4 1h, thr_id=2 78:1
FUNC RET          MPTNBTCP
GO=PutData1

[41]  2/1  11:24:27.37 src=ANYNET, len=94, rec_id= 4 1h, thr_id=2 78:1
FUNC CALL         MPTNBTCP
PutData(362F A18)

[42]  2/1  11:24:27.42 src=ANYNET, len=94, rec_id= 4 1h, thr_id=2 78:1
PROV CALL         MPTNBTCP
sock_send(7, 23 )

[43]  2/1  11:24:27.42 src=ANYNET, len=94, rec_id= 4 1h, thr_id=2 78:1
PROV RET          MPTNBTCP
23 =sock_send(7)

[44]  2/1  11:24:27.42 src=ANYNET, len=94, rec_id= 4 1h, thr_id=2 78:1
FUNC RET          MPTNBTCP
GO=PutData

[45]  2/1  11:24:27.42 src=ANYNET, len=94, rec_id= 4 1h, thr_id=2 78:1
FUNC CALL         MPTNBTCP
ConnectFSM3(362F A18)

[46]  2/1  11:24:27.47 src=ANYNET, len=94, rec_id= 4 1h, thr_id=2 78:1
FUNC RET          MPTNBTCP
YIELD=ConnectFSM3

[47]  2/1  11:24:27.96 src=ANYNET, len=94, rec_id= 4 1h, thr_id=2 78:1
PROV RET          MPTNBTCP
sock_select(1, )

[48]  2/1  11:24:27.96 src=ANYNET, len=94, rec_id= 4 1h, thr_id=2 78:1
FUNC CALL         MPTNBTCP
GetData(362F A18)

[49]  2/1  11:24:27.96 src=ANYNET, len=94, rec_id= 4 1h, thr_id=2 78:1
PROV CALL         MPTNBTCP
sock_recv(7)

[5 ]  2/1  11:24:27.96 src=ANYNET, len=94, rec_id= 4 1h, thr_id=2 78:1
PROV RET          MPTNBTCP
4=sock_recv(7)

[51]  2/1  11:24:27.96 src=ANYNET, len=94, rec_id= 4 1h, thr_id=2 78:1
FUNC RET          MPTNBTCP
GO=GetData

[52]  2/1  11:24:28. 1 src=ANYNET, len=94, rec_id= 4 1h, thr_id=2 78:1
FUNC CALL         MPTNBTCP
GetData1(362F A18)
```

Successful Initiation Example

```
[53]  2/1  11:24:28. 1 src=ANYNET, len=172, rec_id= 4 2h, thr_id=2 78:1
===RECV====>   MPTNBTCP
9.67.98.162         1 24    9.67.98.198            397      7 CMDLPREF 4
       C9                                          <....                    >   <...I                    >

[54]  2/1  11:24:28. 1 src=ANYNET, len=94, rec_id= 4 1h, thr_id=2 78:1
FUNC RET           MPTNBTCP
GO=GetData1


[55]  2/1  11:24:28. 1 src=ANYNET, len=94, rec_id= 4 1h, thr_id=2 78:1
FUNC CALL          MPTNBTCP
GetData(362F A18)

[56]  2/1  11:24:28. 1 src=ANYNET, len=94, rec_id= 4 1h, thr_id=2 78:1
PROV CALL          MPTNBTCP
sock_recv(7)


[57]  2/1  11:24:28. 1 src=ANYNET, len=94, rec_id= 4 1h, thr_id=2 78:1
PROV RET           MPTNBTCP
197=sock_recv(7)

[58]  2/1  11:24:28. 7 src=ANYNET, len=94, rec_id= 4 1h, thr_id=2 78:1
FUNC RET           MPTNBTCP
GO=GetData

[59]  2/1  11:24:28. 7 src=ANYNET, len=94, rec_id= 4 1h, thr_id=2 78:1
FUNC CALL          MPTNBTCP
ConnectFSM4(362F A18)


[6 ]  2/1  11:24:28.18 src=ANYNET, len=11 8, rec_id= 4 2h, thr_id=2 78:1
===RECV====>   MPTNBTCP
9.67.98.162         1 24    9.67.98.198            397      7 POSMCRSP 197
   8 8A  C5 2A B 111 E4E2C9C2 D4D5D94B    <.... ..........K>    <...E....USIBMNR.>
   D5D9C9F4 F F1F F   1 B 1 E E4E2C9C2    <................>    <NRI4 1  ....USIB>
   D4D5D94B D3E4D7F1 F4 1 712 5B 2        <...K........ffl...>    <MNR.LUP14...$...>
     F      1 5       6        762       <......... ...v .>    <.........-......>
    5 8   6  1 1 A 2   6B     EB8   31    <.........k.....1>    <..........,......>
     13 7B  B 5 B3 1   85858   1 6 2      <.....P..........>    <.....&...ee.....>
              1423       2B    9 2E2    <.......#...+....>    <................S>
   D5C1E2E5 C3D4C7 2  3 211 5 E4E2C9C2    <................>    <NASVCMG.....USIB>
   D4D5D94B D5D9C9F4 F F1F F   A13  2     <...K........... >    <MNR.NRI4 1  ....>
   1D27 867 49A395     6 168F DC1BF 69    <.'.gI.... .....i>    <.....tn..-.... .>
   B47FA8 D E4E2C9C2 D4D5D94B D3E4D7F1    <...........K....>    <."y.USIBMNR.LUP1>
   F418      5 119 A    A   1  32 1 83    <............ ..>    <4..............c>
  1A     5  84                                      <.....                    >   <....d                    >


[61]  2/1  11:24:28.28 src=ANYNET, len=94, rec_id= 4 1h, thr_id=2 78:1
FUNC CALL          MPTNCMM
CMMUREQ(P_CONNECT_UC) 348F B   OK

[62]  2/1  11:24:28.34 src=ANYNET, len=94, rec_id= 4 1h, thr_id=2 78:1
FUNC CALL          MPTNCMM
CMMUCALL(M_CONNECT_UC) 348F B   connection setup succeeded
```

Successful Initiation Example

```
[63]  2/1  11:24:28.34 src=ANYNET, len=172, rec_id= 4 1h, thr_id=2 78:1
FUNC CALL         MPTNASNA
M_CONNECT_UC,  1 1 3 348F B
   2F   3 1     EB8                          </.......            >    <.........            >

[64]  2/1  11:24:28.39 src=ANYNET, len=94, rec_id= 4 1h, thr_id=2 78:1
FUNC RET          MPTNASNA
M_CONNECT_UC, RETURN FROM A_SNA OK,  1 1 3 348F B

[65]  2/1  11:24:28.39 src=ANYNET, len=94, rec_id= 4 1h, thr_id=2 78:1
FUNC RET          MPTNCMM
CMMUCALL(M_CONNECT_UC) 348F B

[66]  2/1  11:24:28.5  src=ANYNET, len=94, rec_id= 4 1h, thr_id=2 78:1
FUNC RET          MPTNCMM
CMMUREQ(P_CONNECT_UC) 348F B   MPTN connection established
```

Trace Line Number	Explanation
[1]	Networking Services calls the transport user (MPTNASNA) to obtain the default maximum BIU (Basic Information Unit) size.
[2]	MPTNASNA receives a SNA BIND request from Networking Services to initiate a session using LFSID 010103.
[3-4]	The transport user calls MPTNCMM to create an MPTN control block for the session with LFSID 010103.
[5]	MPTNCMM creates an MPTN control block and returns to MPTNASNA. The pointer to the control block is 348F0B00.
[6-7]	The transport user requests to BIND the local LU name (USIBMNR.LUP14) to the MPTN control block at 348F0B00.
[8]	MPTNCMM successfully BINDs the name and returns.
[9-10]	The transport user requests a session be set up to the partner LU (USIBMNR.NRI40100) by issuing an M_CONNECT downcall.
[11-12]	MPTNCMM makes a P_LOCATE downcall, requesting MPTNBTCP obtain the LUNAME-to-IP address mapping.
[13-14]	MPTNCMM returns pending for the M_CONNECT downcall to MPTNASNA, informing it that the connection setup is in progress.
[15-16]	MPTNBTCP makes a gethostbyname socket call for domain name (NRI40100.USIBMNR.SNA.IBM.COM). The socket call successfully returns the partner node's IP address.
[17]	MPTNBTCP makes a P_LOCATE upcall to pass the results of the gethostbyname call to MPTNCMM.
[18-19]	MPTNCMM makes a P_CONNECT downcall to set up the connection. MPTNBTCP returns pending, indicating that the connection setup is in progress.
[20]	MPTNBTCP begins the connection setup process.
[21-22]	MPTNBTCP makes a socket call and receives socket selector 7.
[23-24]	MPTNBTCP continues the connection setup process.
[25]	MPTNBTCP makes a socket call to connect to IP address 9.67.98.198 on socket 7.

[26] The socket connect call returns a -1. This indicates either the connect is pending completion or that an error occurred. In this case, the connect call is pending completion. If an error had occurred, subsequent trace entries would indicate a connection setup failure and a message would be logged in the message log.

[27-31] MPTNBTCP continues the connection setup process.

[32] MPTNBTCP sends the MPTN command prefix on the TCP connection.

[33] MPTNBTCP sends the MPTN command prefix on the TCP connection.

[34-46] MPTNBTCP continues the connection setup process.

[47-52] MPTNBTCP prepares to receive 4 bytes of data on socket 7.

[53] MPTNBTCP receives an MPTN command prefix. The extra data shows the 4-bytes length field of the MPTN command.

[54-58] MPTNBTCP prepares to receive 197 bytes of data on socket 7.

[59-60] MPTNBTCP receives a positive MPTN connect response from the TCP connection. The extra data shows the MPTN connect response beginning at the command type field.

[61] MPTNBTCP makes a P_CONNECT upcall to MPTNCMM with a status of OK indicating the connection setup succeeded.

[62] MPTNCMM makes an M_CONNECT upcall to MPTNASNA indicating the connection setup succeeded.

[63-65] MPTNASNA forwards the SNA Bind response to Networking Services and returns to MPTNCMM.

[66] MPTNCMM then returns to MPTNBTCP and indicates that the MPTN connection is established.

Unsuccessful Initiation Example

Unsuccessful Session Initiation

The following example shows an AnyNet trace generated during an unsuccessful session initiation. An explanation of the trace output, by line number, follows the example.

```
[1]  2/11 13:3 : 9.45 src=ANYNET, len=94, rec_id= 4 1h, thr_id=9886:1
FUNC CALL         MPTNASNA
GetBTUs local 82 1 remote 82 1

[2]  2/11 13:3 : 9.45 src=ANYNET, len=172, rec_id= 4 1h, thr_id=9886:1
FUNC CALL         MPTNASNA
SEND: BIND FROM A_SNA,  1 1 3
   2F   1 3     6B8                          </.....k..            >    <.......,..            >

[3]  2/11 13:3 : 9.45 src=ANYNET, len=94, rec_id= 4 1h, thr_id=9886:1
FUNC CALL         MPTNASNA
M_CREATE_DC,  1 1 3


[4]  2/11 13:3 : 9.45 src=ANYNET, len=64 , rec_id= 4 1h, thr_id=9886:1
FUNC CALL         MPTNCMM
CMMDREQ(M_CREATE)
          B         2AA61 1715 1      <...........a....>    <............./....>
    1   5 1     6        5 1     2      <...... ....... .>    <......-.........>
    5 1      5 1                      <................>    <................>
    1   1                      1      <................>    <................>
                                    <................>    <................>
                                    <................>    <................>
                                                  <....                    >   <....                    >

[5]  2/11 13:3 : 9.45 src=ANYNET, len=94, rec_id= 4 1h, thr_id=9886:1
FUNC RET          MPTNCMM
CMMDREQ(M_CREATE) 348F B


[6]  2/11 13:3 : 9.45 src=ANYNET, len=94, rec_id= 4 1h, thr_id=9886:1
FUNC CALL         MPTNASNA
M_BIND_DC,  1 1 3 348F B


[7]  2/11 13:3 : 9.45 src=ANYNET, len=25 , rec_id= 4 1h, thr_id=9886:1
FUNC CALL         MPTNCMM
CMMDREQ(M_BIND) 348F B
          B 1 DE4 E2C9C2D4 D5D94BD3    <..............K.>    <.......USIBMNR.L>
   E4D7F1F4                                <...........          >    <UP14.......          >


[8]  2/11 13:3 : 9.45 src=ANYNET, len=94, rec_id= 4 1h, thr_id=9886:1
FUNC RET          MPTNCMM
CMMDREQ(M_BIND) 348F B

[9]  2/11 13:3 : 9.45 src=ANYNET, len=94, rec_id= 4 1h, thr_id=9886:1
FUNC CALL         MPTNASNA
M_CONNECT_DC,  1 1 3 348F B
```

Unsuccessful Initiation Example

```
[1 ]  2/11 13:3 : 9.5  src=ANYNET, len=1 3 , rec_id= 4 1h, thr_id=9886:1
FUNC CALL          MPTNCMM
CMMDREQ(M_CONNECT) 348F B
           B 1 BE4 E2C9C2D4 D5D94BE7    <.............K.>    <.......USIBMNR.X>
   E7E7   1   83                 6B    <.............k>    <XX...c.........,>
   8   31   13 7B B  D1B3 1   85858  1    <..1.............>    <........J...ee..>
    6 2               423      DE4E2    <..........#.....>    <..............US>
   C9C2D4D5 D94BD3E4 D7F1F42F    9 2E2    <.....K...../....>    <IBMNR.LUP14....S>
   D5C1E2E5 C3D4C7 9  3 1DC1B F 8B2381    <..............#.>    <NASVCMG..... ..a>
   A8 E 4E4 E2C9C2D4 D5D94BD3 E4D7F1F4    <..........K.....>    <y..USIBMNR.LUP14>
    A13  DF 2E9F261    41      BE4E2C9    <......&..A......>    <.............USI>
   C2D4D5D9 4BE7E7E7 6 168FDC 1BF 8B23    <....K... ......#>    <BMNR.XXX-.... ..>
   81A8 DE4 E2C9C2D4 D5D94BD3 E4D7F1F4    <..........K.....>    <ay.USIBMNR.LUP14>
    4  4739 13  534E 41384644 43314246    <..G9..SNA8FDC1BF>    <.......+........>
   3 384232 33383141 38                              < 8B2381A8..          >    <...........          >


[11]  2/11 13:3 : 9.5  src=ANYNET, len=94, rec_id= 4 1h, thr_id=9886:1
FUNC CALL          MPTNCMM
CMMDCALL(P_LOCATE) 348F B


[12]  2/11 13:3 : 9.5  src=ANYNET, len=94, rec_id= 4 1h, thr_id=9886:1
FUNC RET           MPTNCMM
CMMDCALL(P_LOCATE) 348F B


[13]  2/11 13:3 : 9.5  src=ANYNET, len=94, rec_id= 4 1h, thr_id=9886:1
FUNC RET           MPTNCMM
CMMDREQ(M_CONNECT) 348F B   pending


[14]  2/11 13:3 : 9.5  src=ANYNET, len=94, rec_id= 4 1h, thr_id=9886:1
FUNC RET           MPTNASNA
M_CONNECT_DC, RC_PENDING,  1 1 3 348F B


[15]  2/11 13:3 : 9.55 src=ANYNET, len=25 , rec_id= 4 1h, thr_id=1 118:1
PROV CALL          MPTNBTCP
sock_gethostbyname
    5858582E 55534942 4D4E522E 534E412E    <XXX.USIBMNR.SNA.>    <........(+...+..>
   49424D2E  434F4D                                  <IBM.COM                 >   <..(...(                 >
```

Unsuccessful Initiation Example

```
[16]  2/11 13:3 : 9.72 src=ANYNET, len=94, rec_id= 4 1h, thr_id=1 118:1
PROV RET          MPTNBTCP
-1=sock_gethostbyname
   5858582E 55534942 4D4E522E 534E412E    <XXX.USIBMNR.SNA.>    <........(+...+..>
  49424D2E  434F4D                          <IBM.COM              >   <..(...(              >

[17]  2/11 13:3 : 9.88 src=ANYNET, len=94, rec_id= 4 1h, thr_id=1 118:1
FUNC CALL         MPTNCMM
CMMUREQ(P_LOCATE_UC) 348F B

[18]  2/11 13:3 : 9.88 src=ANYNET, len=94, rec_id= 4 1h, thr_id=1 118:1
FUNC CALL         MPTNCMM
CMMUCALL(M_CONNECT_UC) 348F B   connection setup failed

[19]  2/11 13:3 : 9.88 src=ANYNET, len=172, rec_id= 4 1h, thr_id=1 118:1
FUNC CALL         MPTNASNA
SENDUNBIND REQUEST TO A_SNA,  1 1 3 348F B
   2F   3 1    16B8                          </.....k..            >    <.......,..            >


[2 ]  2/11 13:3 : 9.88 src=ANYNET, len=94, rec_id= 4 1h, thr_id=1 118:1
FUNC CALL         MPTNASNA
M_CONNECT_UC, ERROR, CLOSING CMM CB,  1 1 3 348F B

[21]  2/11 13:3 : 9.88 src=ANYNET, len=25 , rec_id= 4 1h, thr_id=1 118:1
FUNC CALL         MPTNCMM
CMMDREQ(M_CLOSE) 348F B
                                  <................>    <................>
   5                                          <......                >   <......                >

[22]  2/11 13:3 : 9.88 src=ANYNET, len=94, rec_id= 4 1h, thr_id=1 118:1
FUNC RET          MPTNCMM
CMMDREQ(M_CLOSE) 348F B

[23]  2/11 13:3 : 9.88 src=ANYNET, len=94, rec_id= 4 1h, thr_id=1 118:1
FUNC RET          MPTNCMM
CMMUCALL(M_CONNECT_UC) 348F B


[24]  2/11 13:3 : 9.94 src=ANYNET, len=172, rec_id= 4 1h, thr_id=9886:1
FUNC CALL         MPTNASNA
SEND: UNBIND_RESP FROM A_SNA,  1 1 3
   2F   1 3    1EB8                          </........            >    <.........            >
```

Trace Line Number	Explanation
[1]	Networking Services calls the transport user (MPTNASNA) to obtain the default maximum BIU (Basic Information Unit) size.
[2]	MPTNASNA receives a SNA BIND request from Networking Services to initiate a session using LFSID 010103.
[3-4]	The transport user calls MPTNCMM to create an MPTN control block for the session with LFSID 010103.
[5]	MPTNCMM creates an MPTN control block and returns to MPTNASNA. The pointer to the control block is 348F0B00.
[6-7]	The transport user requests to BIND the local LU name (USIBMNR.LUP14) to the MPTN control block at 348F0B00.

[8]	MPTNCMM successfully BINDs the name and returns.
[9-10]	The transport user requests a session be set up to the partner LU (USIBMNR.XXX) by issuing an M_CONNECT downcall.
[11-12]	MPTNCMM makes a P_LOCATE downcall, requesting MPTNBTCP obtain the LUNAME-to-IP address mapping.
[13-14]	MPTNCMM returns pending for the M_CONNECT downcall to MPTNASNA, informing it that the connection setup is in progress.
[15]	MPTNBTCP makes a gethostbyname socket call for domain name (XXX.USIBMNR.SNA.IBM.COM).
[16]	The gethostbyname socket call was not successful, indicating the domain name was not found in the DNS or HOSTS file.
[17]	MPTNBTCP makes a P_LOCATE upcall to pass the results of the gethostbyname call to MPTNCMM.
[18]	MPTNCMM makes an M_CONNECT upcall to MPTNASNA to report the connection setup failure.
[19]	MPTNASNA sends an SNA UNBIND request to Networking Services to indicate the unsuccessful session initiation for LFSID 010103. The SNA transmission header and request/response header is shown in the extra data.
[20-21]	MPTNASNA requests MPTNCMM to close the control block at address 348F0B00.
[22-23]	MPTNCMM returns.
[24]	MPTNASNA receives an SNA UNBIND response from Networking Services for LFSID 010103.

Unsuccessful Initiation Example

Appendix E. Trace Example for Personal Communications

The following example shows an AnyNet trace generated during a successful outbound session initiation.

Note: An actual trace contains trace calls that relate to acquiring and freeing memory ("MemGet" and "MemFree"). To reduce the size of this trace example, those lines have been removed.

```
//------------------------------
// 1 -11-95  4:1 :   USA CONV
//------------------------------
1 :14:3 .663 MPTNCMM FUNC CALL ANY  54I: TCPPMM successfully loaded
1 :14:3 .665 MPTNCMM FUNC CALL ANY  54I: PCSMPTN successfully loaded
1 :14:3 .669 MPTNCMM FUNC CALL ANY  88I: CMMDCALL(P_INIT)

1 :14:3 .678 MPTNBTCP FUNC CALL PMMGlobalInit
1 :14:3 .678 MPTNBTCP FUNC CALL InitGlobals
1 :14:3 .678 MPTNBTCP FUNC CALL InitializeRetry
1 :14:3 .678 MPTNBTCP FUNC CALL InitializePartnerNodeHTable
1 :14:3 .679 MPTNBTCP FUNC CALL InitializeCbList
1 :14:3 .679 MPTNBTCP FUNC CALL InitializeCbArray
1 :14:3 .679 MPTNBTCP FUNC CALL InitializeReadableWritable
1 :14:3 .679 MPTNBTCP FUNC CALL ResetStubQueue
1 :14:3 .679 MPTNBTCP FUNC CALL InitializeClosedQueue

1 :14:3 .69  MPTNBTCP FUNC CALL MoatMain
1 :14:3 .69  MPTNBTCP FUNC CALL MoatInit
1 :14:3 .797 MPTNBTCP FUNC CALL IsTCPInstalled
1 :14:3 .797 MPTNBTCP PROV CALL       sock_sock_init
1 :14:3 .918 MPTNBTCP PROV RET        sock_sock_init
1 :14:3 .918 MPTNBTCP PROV CALL       socket
1 :14:3 .919 MPTNBTCP PROV RET        socket(66)
1 :14:3 .919 MPTNBTCP FUNC CALL sock_getlocalhostaddrs
1 :14:3 .919 MPTNBTCP PROV CALL       sock_gethostbyname
1 :14:3 .919 Data:
                                ..>.>/..
                                76676673
                                79E3E161
1 :14:3 .985 MPTNBTCP PROV RET        =sock_gethostbyname
1 :14:3 .985 Data:
                                ..>.>/..
                                76676673
                                79E3E161

1 :14:3 .985 MPTNBTCP FUNC RET  =sock_getlocalhostaddrs
1 :14:3 .985 MPTNBTCP PROV CALL       sock_close(66)
1 :14:3 .986 MPTNBTCP PROV RET        sock_close(66)
1 :14:3 .986 MPTNBTCP FUNC RET 38FF 1 4=IsTCPInstalled
1 :14:3 .986 MPTNBTCP FUNC CALL ReadEnvVars
1 :14:3 .986 MPTNBTCP FUNC CALL DatagramInit
1 :14:3 .987 MPTNBTCP FUNC CALL InitPmmCb(39 F  12)
1 :14:3 .987 MPTNBTCP FUNC CALL InitLocalData(39 F  12)
1 :14:3 .987 MPTNBTCP FUNC CALL InitLocalData(39 F  5 )
1 :14:3 .987 MPTNBTCP PROV CALL       socket
1 :14:3 .988 MPTNBTCP PROV RET        socket(66)
1 :14:3 .988 MPTNBTCP PROV CALL       sock_set_snd_buf_size(66)
1 :14:3 .988 MPTNBTCP PROV RET        sock_set_snd_buf_size(66)
```

© Copyright IBM Corp. 1996

Trace Example

```
1 :14:3 .988 MPTNBTCP PROV CALL       sock_set_rcv_buf_size(66)
1 :14:3 .988 MPTNBTCP PROV RET        sock_set_rcv_buf_size(66)
1 :14:3 .988 MPTNBTCP PROV CALL       sock_set_no_blocking(66)
1 :14:3 .988 MPTNBTCP PROV RET        sock_set_no_blocking(66)
1 :14:3 .989 MPTNBTCP PROV CALL       sock_bind(66)
1 :14:3 .934 MPTNBTCP PROV RET        sock_bind(66)
1 :14:3 .934 MPTNBTCP PROV CALL       CheckReadable(39 F  12,66)
1 :14:3 .934 MPTNBTCP PROV RET        CheckReadable(39 F  12,66)
1 :14:3 .934 MPTNBTCP FUNC CALL ListenInit
1 :14:3 .935 MPTNBTCP FUNC CALL InitPmmCb(39 F 294)
1 :14:3 .935 MPTNBTCP FUNC CALL InitLocalData(39 F 294)
1 :14:3 .935 MPTNBTCP FUNC CALL InitLocalData(39 F 2D2)
1 :14:3 .935 MPTNBTCP PROV CALL       socket
1 :14:3 .936 MPTNBTCP PROV RET        socket(67)
1 :14:3 .936 MPTNBTCP PROV CALL       sock_bind(67)
1 :14:3 .936 MPTNBTCP PROV RET        sock_bind(67)
1 :14:3 .936 MPTNBTCP PROV CALL       sock_listen(67)

1 :14:3 .937 MPTNBTCP PROV RET        sock_listen(67)
1 :14:3 .937 MPTNBTCP PROV CALL       sock_set_no_blocking(67)
1 :14:3 .937 MPTNBTCP PROV RET        sock_set_no_blocking(67)
1 :14:3 .937 MPTNBTCP PROV CALL       CheckReadable(39 F 294,67)
1 :14:3 .938 MPTNBTCP PROV RET        CheckReadable(39 F 294,67)
1 :14:3 .938 MPTNBTCP FUNC CALL KeepaliveInit
1 :14:3 .938 MPTNBTCP FUNC CALL InitLocateThreads
1 :14:3 .944 MPTNBTCP FUNC CALL LocateProc
1 :14:3 .944 MPTNBTCP PROV CALL       sock_sock_init
1 :14:3 .949 MPTNBTCP PROV RET        sock_sock_init
1 :14:3 .949 MPTNCMM FUNC RET ANY  89I: CMMDCALL(P_INIT) returned pending
1 :14:3 .95  MPTNCMM FUNC RET ANY 147I: CMMDREQ(CMM_INIT)
1 :14:3 .95  MPTNCMM FUNC CALL ANY  7 I: CMMMGMT(DEFINE_ATTRIBUTES)
1 :14:3 .951 MPTNCMM FUNC RET ANY  7 I: CMMMGMT(DEFINE_ATTRIBUTES)

1 :14:3 .951 MPTNCMM FUNC CALL ANY 148I: CMMDREQ(M_ENABLED)
1 :14:3 .951 Data:
                                ....@..'..................
                                   734 2
                                   CC277
1 :14:3 .951 MPTNCMM FUNC RET ANY 148I: CMMDREQ(M_ENABLED)
1 :14:3 .952 MPTNBTCP FUNC CALL GetBspecInfo
1 :14:3 .952 MPTNCMM FUNC CALL ANY  99I: CMMUREQ(P_INIT_UC)
1 :14:3 .953 MPTNCMM FUNC RET ANY 1 1I: CMMUREQ(P_INIT_UC) received from TCPPMM
1 :14:3 .953 MPTNCMM FUNC CALL ANY 16 I: CMMUCALL(M_ENABLED_UC)
1 :14:3 .953 M_ENABLED received
1 :14:3 .953 MPTNCMM FUNC RET ANY 16 I: CMMUCALL(M_ENABLED_UC)
1 :14:3 .953 MPTNBTCP FUNC CALL GetNextLocateRequest
1 :14:3 .954 MPTNBTCP FUNC RET  =GetNextLocateRequest

1 :14:3 .99  M_CREATE    issued
1 :14:3 .99  MPTNCMM FUNC CALL ANY 129I: CMMDREQ(M_CREATE)
1 :14:3 .991 Data:
                                ...........'......"..-..................................
                                ..................................
                                        34 2    F7  6      2

                                   B     C2771 1 FF     51    51  51         1 1
     1      1
                                .......................


1 :14:3 .991 MPTNCMM FUNC CALL ANY 176I: CreateMptnCb: MPTN_CB 38FF 2A8 created
1 :14:3 .992 MPTNCMM FUNC RET ANY 131I: CMMDREQ(M_CREATE) 38FF 2A8
1 :14:3 .992 M_BIND       issued
```

Trace Example

```
1 :14:3 .992 MPTNCMM FUNC CALL ANY 132I: CMMDREQ(M_BIND) 38FF 2A8
1 :14:3 .992 Data:
                                    ....@.............
                                       7  131      3
                                       C142  1 1 1 3
1 :14:3 .992 MPTNCMM FUNC CALL ANY 214I: BindName
1 :14:3 .993 MPTNCMM FUNC CALL ANY 214I: ProcessRegister
1 :14:3 .993 MPTNCMM FUNC RET ANY 132I: CMMDREQ(M_BIND) 38FF 2A8
1 :14:3 .993 M_LISTEN    issued
1 :14:3 .993 MPTNCMM FUNC CALL ANY 139I: CMMDREQ(M_LISTEN) 38FF 2A8
1 :14:3 .993 Data:
                                    ........

                                       1
1 :14:3 .994 MPTNCMM FUNC RET ANY 139I: CMMDREQ(M_LISTEN) 38FF 2A8

1 :14:3 .994 M_CREATE    issued
1 :14:3 .994 MPTNCMM FUNC CALL ANY 129I: CMMDREQ(M_CREATE)
1 :14:3 .994 Data:
                                    ...........'.....".-.....................................
                                    ..............................
                                           34 2    F7  6       2

                                       B     C277  1 FF      51    51  51   8      1 1
    1     1 1
                                    ......................


1 :14:3 .994 MPTNCMM FUNC CALL ANY 176I: CreateMptnCb: MPTN_CB 38FF 4A6 created
1 :14:3 .995 MPTNCMM FUNC RET ANY 131I: CMMDREQ(M_CREATE) 38FF 4A6
1 :14:3 .995 M_BIND       issued

1 :14:3 .995 MPTNCMM FUNC CALL ANY 132I: CMMDREQ(M_BIND) 38FF 4A6
1 :14:3 .995 Data:
                                    ....@.............
                                       7  131      3
                                       C142  1    1 3
1 :14:3 .995 MPTNCMM FUNC CALL ANY 214I: BindName:
1 :14:3 .995 MPTNCMM FUNC RET ANY 132I: CMMDREQ(M_BIND) 38FF 4A6
1 :14:3 .995 M_RCVDG_I issued
1 :14:3 .996 MPTNCMM FUNC CALL ANY 219I: CMMDREQ(M_RCV_DG_INIT) 38FF 4A6
1 :14:3 .996 Data:
                                    ..........


1 :14:3 .996 MPTNCMM FUNC RET ANY 219I: CMMDREQ(M_RCV_DG_INIT) 38FF 4A6
1 :14:3 .996 M_SEND_DG issued
1 :14:3 .996 MPTNCMM FUNC CALL ANY 22 I: CMMDREQ(M_SEND_DG) 38FF 4A6
1 :14:3 .996 Data:
                                    ..............USIBMNR.SNAIPGW.............7.......
                                                EECCDDD4EDCCDCE           2 F2
                                           2 B1F4292459B2519776 1 4       8376

1 :14:3 .996 MPTNCMM FUNC CALL ANY 214I: CreateDgStruct
1 :14:3 .997 MPTNCMM FUNC CALL ANY 214I: CheckABCache
1 :14:3 .997 MPTNCMM FUNC CALL ANY 214I: SelectBSpec
1 :14:3 .998 MPTNCMM FUNC CALL ANY  96I: CMMDCALL(P_LOCATE) 38FF 4A6
1 :14:3 .998 MPTNBTCP FUNC CALL TCPMREQ: P_LOCATE
1 :14:3 .998 MPTNBTCP FUNC CALL InitiateLocate
1 :14:3 .998 MPTNCMM FUNC RET ANY  96I: CMMDCALL(P_LOCATE) 38FF 4A6
1 :14:3 .999 MPTNCMM FUNC RET ANY 22 I: CMMDREQ(M_SEND_DG) 38FF 4A6 27
1 :14:31.135 MPTNCMM FUNC CALL ANY 214I: WorkerThread
1 :14:31.135 MPTNCMM FUNC CALL ANY 214I: ProcessReportInfo
```

Trace Example

```
1 :14:31.135 MPTNCMM FUNC RET ANY 214I: WorkerThread
1 :14:31.651 MPTNBTCP FUNC CALL GetNextLocateRequest
1 :14:31.651 MPTNBTCP FUNC RET 38FF 6EA=GetNextLocateRequest
1 :14:31.651 MPTNBTCP FUNC CALL SnaNodeAddrToStr
1 :14:31.652 MPTNBTCP FUNC RET  =SnaNodeAddrToStr
1 :14:31.652 MPTNBTCP PROV CALL       sock_gethostbyname
1 :14:31.652 Data:
                                .+..&.......(+...+....(..](
                                5444545255444454254424442444
                                3E19 77E5392DE2E3E1E92DE3FD
1 :14:31.732 MPTNBTCP PROV RET        =sock_gethostbyname
1 :14:31.732 Data:
                                .+..&.......(+...+....(..](
                                5444545255444454254424442444
                                3E19 77E5392DE2E3E1E92DE3FD

1 :14:31.732 MPTNCMM FUNC CALL ANY 113I: CMMUREQ(P_LOCATE_UC) 38FF 4A6
1 :14:31.733 MPTNCMM FUNC CALL ANY 214I: FSMUpcall
1 :14:31.733 MPTNCMM FUNC CALL ANY 214I: CacheName
1 :14:31.733 MPTNCMM FUNC CALL ANY  65I: CacheName: name cache initialized
1 :14:31.734 MPTNCMM FUNC CALL ANY 214I: LRUAdjust
1 :14:31.734 MPTNCMM FUNC CALL ANY 214I: SelectBADDR
1 :14:31.734 MPTNCMM FUNC CALL ANY 214I: LRUAdjust
1 :14:31.734 MPTNCMM FUNC CALL ANY 214I: LRUAdjustA
1 :14:31.735 MPTNCMM FUNC CALL ANY 228I: CMMDCALL(P_SEND_DG) 38FF 4A6
1 :14:31.736 MPTNBTCP FUNC CALL TCPMREQ: P_SEND_DG

1 :14:31.737 MPTNBTCP FUNC CALL AddToStubQueue
1 :14:31.737 MPTNBTCP FUNC RET  =AddToStubQueue
1 :14:31.737 MPTNCMM FUNC CALL ANY 214I: FreeDgStruct
1 :14:31.738 MPTNCMM FUNC RET ANY 228I: CMMDCALL(P_SEND_DG) 38FF 4A6 2
1 :14:31.738 MPTNBTCP FUNC CALL GetNextLocateRequest
1 :14:31.738 MPTNBTCP FUNC RET  =GetNextLocateRequest
1 :14:31.877 MPTNBTCP FUNC CALL ProcessMessage
1 :14:31.877 MPTNBTCP FUNC CALL SendUserDg
1 :14:31.878 MPTNBTCP FUNC CALL AddrToSockaddr
1 :14:31.878 MPTNBTCP PROV CALL       sock_inet_ntoa(998 43 9)
1 :14:31.878 MPTNBTCP PROV RET        sock_inet_ntoa(9.67.128.153)
1 :14:31.878 MPTNBTCP PROV CALL       sock_send_dg(66, 9.67.128.153, 61)
1 :14:31.878 MPTNBTCP <==SEND=DG==                                  9.67.128.153          397
66 MPTNDGRM 61

1 :14:31.878 Data:
                                .....USIBMNR.SNAIPGW.@.....'.....'.....  ..
                                  38  32  1EECCDDD4EDCCDCE 7  131   19
  4
                                   DE  5AB1 4292459B25197761C152  1178C    58 61172 8
11  4

1 :14:31.884 MPTNBTCP PROV RET        61 = sock_send_dg(66, 9.67.128.153)
1 :14:31.885 MPTNBTCP FUNC CALL FreeAddrFlds
1 :14:31.885 MPTNBTCP FUNC CALL FreeAddrFlds
1 :14:31.886 Unknown-    received
1 :14:32. 15 MPTNBTCP PROV RET        sock_select(1, )
1 :14:32. 15 MPTNBTCP FUNC CALL RecvDgrm
1 :14:32. 15 MPTNBTCP PROV CALL       sock_recv_dg
1 :14:32. 15 MPTNBTCP PROV RET        53=sock_recv_dg (66)
1 :14:32. 15 MPTNBTCP PROV CALL       sock_inet_ntoa(998 43 9)
1 :14:32. 16 MPTNBTCP PROV RET        sock_inet_ntoa(9.67.128.153)
1 :14:32. 16 MPTNBTCP ===RECV=DG=>                                  9.67.128.153          111
66 MPTNDGRM 53
```

Trace Example

```
1 :14:32. 16 Data:
                                        .........@.........USIBMNR.SNAIPGW.'¬cX.%.....
                                          38  227  131    1EECCDDD4EDCCDCE  58E36
                                          5E  DAC152  11B1 4292459B251977617F37DC 58 6112  4
1 :14:32. 16 MPTNCMM FUNC CALL ANY 116I: CMMUREQ(P_PARSE_DG_UC)  A
1 :14:32. 18 MPTNCMM FUNC RET ANY 116I: CMMUREQ(P_PARSE_DG_UC) 38FF 4A6
1 :14:32. 18 MPTNBTCP FUNC CALL ProcessUserDg
1 :14:32. 18 Unknown-    received
1 :14:32. 18 MPTNBTCP FUNC CALL SockaddrToAddr
1 :14:32. 19 MPTNCMM FUNC CALL ANY 216I: CMMUREQ(P_RCV_DG_UC) 38FF 4A6

1 :14:32. 19 MPTNCMM FUNC CALL ANY 221I: CMMUCALL(M_RCV_DG_UC) 38FF 4A6
1 :14:32. 19 M_RCV_DG    received
1 :14:32. 19 MPTNCMM FUNC RET ANY 221I: CMMUCALL(M_RCV_DG_UC) 38FF 4A6
1 :14:32. 2  MPTNCMM FUNC RET ANY 216I: CMMUREQ(P_RCV_DG_UC) 38FF 4A6
1 :14:32. 21 MPTNBTCP FUNC CALL FreeAddrFlds
1 :14:32. 21 MPTNBTCP FUNC CALL FreeAddrFlds
1 :14:32. 21 MPTNBTCP PROV CALL       sock_recv_dg
1 :14:32. 22 MPTNBTCP PROV RET        -1=sock_recv_dg (66, 1  35)
1 :14:32. 22 MPTNBTCP FUNC RET YIELD=RecvDgrm
1 :14:32.149 MPTNBTCP PROV RET        sock_select(1, )
1 :14:32.149 MPTNBTCP FUNC CALL ListenFSM(39 F 294)
1 :14:32.149 MPTNBTCP PROV CALL       sock_accept(67)
1 :14:32.149 MPTNBTCP PROV RET        sock_accept(68)
1 :14:32.15  MPTNBTCP FUNC CALL InitPmmCb(39 F 918)
1 :14:32.15  MPTNBTCP FUNC CALL InitLocalData(39 F 918)
1 :14:32.15  MPTNBTCP FUNC CALL InitLocalData(39 F 956)
1 :14:32.15  MPTNBTCP FUNC CALL AddToCbList(39 F 918)
1 :14:32.15  MPTNBTCP FUNC CALL AddToCbArray(39 F 918)
1 :14:32.15  MPTNBTCP PROV CALL       sock_getsockname(68)

1 :14:32.151 MPTNBTCP PROV RET        sock_getsockname(68)
1 :14:32.151 MPTNBTCP PROV CALL       sock_inet_ntoa(AB8 43 9)
1 :14:32.151 MPTNBTCP PROV RET        sock_inet_ntoa(9.67.128.171)
1 :14:32.151 MPTNBTCP PROV CALL       sock_getpeername(68)
1 :14:32.151 MPTNBTCP PROV RET        sock_getpeername(68)
1 :14:32.151 MPTNBTCP PROV CALL       sock_inet_ntoa(998 43 9)
1 :14:32.151 MPTNBTCP PROV RET        sock_inet_ntoa(9.67.128.153)
1 :14:32.151 MPTNBTCP PROV CALL       sock_set_no_blocking(68)
1 :14:32.151 MPTNBTCP PROV RET        sock_set_no_blocking(68)
1 :14:32.152 MPTNBTCP PROV CALL       CheckReadable(39 F 918,68)
1 :14:32.152 MPTNBTCP PROV RET        CheckReadable(39 F 918,68)
1 :14:32.152 MPTNBTCP FUNC RET GO=ListenFSM
1 :14:32.152 MPTNBTCP FUNC CALL ListenFSM(39 F 294)
1 :14:32.152 MPTNBTCP PROV CALL       sock_accept(67)
1 :14:32.152 MPTNBTCP PROV RET        sock_accept(-1)
1 :14:32.152 MPTNBTCP FUNC RET YIELD=ListenFSM
1 :14:32.267 MPTNBTCP PROV RET        sock_select(1, )
1 :14:32.268 MPTNBTCP FUNC CALL GetData(39 F 918)
1 :14:32.268 MPTNBTCP PROV CALL       sock_recv(68)
1 :14:32.268 MPTNBTCP PROV RET        4=sock_recv(68)
1 :14:32.268 MPTNBTCP FUNC RET GO=GetData
1 :14:32.268 MPTNBTCP FUNC CALL GetData1(39 F 918)
1 :14:32.268 MPTNBTCP ===RECV====> 9.67.128.171                9.67.128.153          271
68 CMDLPREF 4

1 :14:32.268 Data:
                                      ....
                                        7
                                        1
1 :14:32.269 MPTNBTCP FUNC RET GO=GetData
1 :14:32.269 MPTNBTCP FUNC CALL GetData(39 F 918)
1 :14:32.269 MPTNBTCP PROV CALL       sock_recv(68)
1 :14:32.269 MPTNBTCP PROV RET        1 9=sock_recv(68)
1 :14:32.269 MPTNBTCP FUNC RET GO=GetData
1 :14:32.269 MPTNBTCP FUNC CALL ListenFSM1(39 F 918)
1 :14:32.27  MPTNBTCP ===RECV====> 9.67.128.171                9.67.128.153          271
68 MPTNCONN 1 9
```

Trace Example

```
1 :14:32.27  Data:
                                ..._.@..........USIBMNR.SNAIPGW.'|cX._....  ............
                                ..............,..................
                                8  627  131    1EECCDDD4EDCCDCE  B8E36    4    7   72
      1    68 1        1    1
                                 A DAC152  11B1 4292459B251977617D37DD F   1   6   6  5
8 611A2 4    B  1214     8  519A
                                .......c....d.....  ...
                                    2181   8A    4
                                 A 13  3A  54F2 98  51

1 :14:32.27  MPTNBTCP PROV CALL        StopReadable(39 F 918,68)
1 :14:32.27  MPTNBTCP PROV RET         StopReadable(39 F 918,68)
1 :14:32.27  MPTNBTCP FUNC CALL SockaddrToAddr
1 :14:32.27  MPTNBTCP FUNC CALL SockaddrToAddr
1 :14:32.271 MPTNCMM FUNC CALL ANY 1 8I: CMMUREQ(P_ACCEPT_NN_UC)  A
1 :14:32.274 MPTNCMM FUNC CALL ANY 176I: CreateMptnCb: MPTN_CB 38FF 996 created

1 :14:32.276 MPTNCMM FUNC CALL ANY 214I: FSMUpcall
1 :14:32.277 MPTNCMM FUNC CALL ANY 154I: CMMUCALL(M_ACCEPT_UC) 38FF 996
1 :14:32.277 M_ACCEPT    received
1 :14:32.277 Payload received
1 :14:32.277 Data:
                                .  ...
                                 4
                                8  51

1 :14:32.277 OUTB: [LH:] l=-7968 LSid=e e
                        [TH:] OIS  ->  EFI SNF=
                        [RH:] REQ SC FI OIC DR1I
                       [RU:]  ACTPU
                                ........

                                214

1 :14:32.277 MPTNCMM FUNC RET ANY 154I: CMMUCALL(M_ACCEPT_UC) 38FF 996
1 :14:32.277 MPTNCMM FUNC CALL ANY 214I: FSMUpcall
1 :14:32.278 MPTNCMM FUNC RET ANY 1 8I: CMMUREQ(P_ACCEPT_NN_UC)
1 :14:32.278 Data:
                                ...._...='......'....'......'...%'.......o.............
                                ........................
                                   6   7 F3   7 F3 6 F3   7 F3 6 F3 1  39 F3

                                   D   E7F821 27F8 67F821 87F8 C7F8289F969F8


1 :14:32.278 MPTNBTCP FUNC RET YIELD=ListenFSM1(
1 :14:32.279 INB:    [LH:] l=-7968 LSid=e e
                        [TH:] OIS  <-  EFI SNF=
                        [RH:] +RSP SC FI OIC DR1I
                       [RU:]  ACTPU
                                .              ..'.......
                                144444444
                                2         71
1 :14:32.279 M_ACCEPT    issued
1 :14:32.279 MPTNCMM FUNC CALL ANY 233I: CMMDREQ(M_ACCEPT) 38FF 996

1 :14:32.28  Data:
                                ........................              ..'.......F......  ..................
                                     1   1      E8 1144444444         C F2   4  7
7
                                     B   B      B  12         71      63F1  1   6
6   2
1 :14:32.28  MPTNCMM FUNC CALL ANY 234I: CMMDCALL(P_ACCEPT_NN) 38FF 996
```

Trace Example

```
1 :14:32.282 MPTNCMM FUNC CALL ANY 214I: 38FF 996
1 :14:32.282 Data:
                                .....'......cd.................2............'.........
                                .f.......o............  .........
                                        2188        7   7   F F3   7 F3 9 F3
 8 F3    9 F3        4  7
                                     7  13  34        8   8   2AF8B1F4BF8 E7F86   6
 6BF8    69F81 1     1   6
                                .......'...o.........................................
                                ..........................
                                    34 2  9 F3

                                    C277  69F8


1 :14:32.282 MPTNBTCP FUNC CALL TCPMREQ: P_ACCEPT_NN
1 :14:32.283 MPTNBTCP FUNC CALL CopyPCreateFlds(39 F 918)
1 :14:32.283 MPTNBTCP FUNC CALL ProcessCompHdrs
1 :14:32.283 MPTNBTCP FUNC RET 1=ProcessCompHdrs
1 :14:32.283 MPTNBTCP FUNC CALL AddToStubQueue
1 :14:32.283 MPTNBTCP FUNC RET  =AddToStubQueue
1 :14:32.283 MPTNCMM FUNC RET ANY 234I: CMMDCALL(P_ACCEPT_NN) 38FF 996
1 :14:32.283 MPTNCMM FUNC CALL ANY 214I: CleanupMptnCb
1 :14:32.284 MPTNCMM FUNC CALL ANY 214I: FreeConnInfoBufs
1 :14:32.285 MPTNCMM FUNC CALL ANY 214I: FreeM_INFO

1 :14:32.285 MPTNCMM FUNC RET ANY 233I: CMMDREQ(M_ACCEPT) 38FF 996
1 :14:32.459 MPTNBTCP FUNC CALL ProcessMessage
1 :14:32.459 MPTNBTCP FUNC CALL AcceptFSM(39 F 918)
1 :14:32.46  MPTNBTCP <==SEND===== 9.67.128.171                9.67.128.153          271
 68 CMDLPREF 4
1 :14:32.46  Data:
                                ...@
                                  7
                                  C
1 :14:32.46  MPTNBTCP <==SEND===== 9.67.128.171                9.67.128.153          271
 68 POSMCRSP 12
1 :14:32.46  Data:
                                ....@..........USIBMNR.SNAIPGW.|cX._....  ............
                                ..................              .....
                                88 727  131    1EECCDDD4EDCCDCE  B8E36    4    7   72
      1    E8 1144444444
                                 A 8AC152  11B1 4292459B251977617D37DD F   1   6   6  5
8 611A2 F    B  12         71
                                ..................c....d.........
                                   1    1      2181   8A
                                   8  519A A 13  3A  54F1 98  51
```

Trace Example

```
1 :14:32.46  MPTNBTCP PROV CALL       Writable(39 F 918,68)
1 :14:32.46  MPTNBTCP PROV RET        Writable(39 F 918,68)
1 :14:32.46  MPTNBTCP FUNC RET GO=AcceptFSM
1 :14:32.46  MPTNBTCP FUNC CALL PutData(39 F 918)
1 :14:32.46  MPTNBTCP PROV CALL       sock_send(68, 4)
1 :14:32.461 MPTNBTCP PROV RET        4=sock_send(68)
1 :14:32.462 MPTNBTCP FUNC RET GO=PutData
1 :14:32.462 MPTNBTCP FUNC CALL PutData1(39 F 918)
1 :14:32.462 MPTNBTCP FUNC RET GO=PutData1
1 :14:32.462 MPTNBTCP FUNC CALL PutData(39 F 918)
1 :14:32.462 MPTNBTCP PROV CALL       sock_send(68, 12 )
1 :14:32.462 MPTNBTCP PROV RET        -1=sock_send(68, 1  35)
1 :14:32.462 MPTNBTCP FUNC RET YIELD=PutData
1 :14:32.463 MPTNBTCP FUNC CALL PutData(39 F 918)
1 :14:32.463 MPTNBTCP PROV CALL       sock_send(68, 12 )
1 :14:32.463 MPTNBTCP PROV RET        -1=sock_send(68, 1  35)
1 :14:32.463 MPTNBTCP FUNC RET YIELD=PutData
1 :14:32.97  MPTNBTCP PROV RET        sock_select( ,1)
1 :14:32.97  MPTNBTCP FUNC CALL PutData(39 F 918)
1 :14:32.97  MPTNBTCP PROV CALL       sock_send(68, 12 )
1 :14:32.972 MPTNBTCP PROV RET        12 =sock_send(68)
1 :14:32.972 MPTNBTCP FUNC RET GO=PutData
1 :14:32.972 MPTNBTCP FUNC CALL AcceptFSM1(39 F 918)
1 :14:32.972 MPTNBTCP PROV CALL       CheckReadable(39 F 918,68)
1 :14:32.972 MPTNBTCP PROV RET        CheckReadable(39 F 918,68)
1 :14:32.972 MPTNBTCP PROV CALL       sock_keep_alive(68)
1 :14:32.973 MPTNBTCP PROV RET        sock_keep_alive(68)
1 :14:32.973 MPTNBTCP FUNC CALL InitiateKeepalive(39 F 918)
1 :14:32.973 MPTNBTCP FUNC CALL FindPartnerNode(39 F 918)
1 :14:32.973 MPTNBTCP FUNC RET NULL=FindPartnerNode
1 :14:32.973 MPTNBTCP FUNC CALL CreatePartnerNode(39 F 918)
1 :14:32.973 MPTNBTCP FUNC RET 38FF 976=CreatePartnerNode

1 :14:32.974 MPTNBTCP PROV CALL       socket
1 :14:32.974 MPTNBTCP PROV RET        socket(7 )
1 :14:32.974 MPTNBTCP PROV CALL       sock_set_no_blocking(7 )
1 :14:32.974 MPTNBTCP PROV RET        sock_set_no_blocking(7 )
1 :14:32.974 MPTNBTCP PROV CALL       sock_bind(7 )
1 :14:32.975 MPTNBTCP PROV RET        sock_bind(7 )
1 :14:32.975 MPTNBTCP PROV CALL       sock_inet_ntoa(998 43 9)
1 :14:32.975 MPTNBTCP PROV RET        sock_inet_ntoa(9.67.128.153)
1 :14:32.976 MPTNBTCP FUNC CALL InsertPartnerNode
1 :14:32.976 MPTNBTCP FUNC RET GO=InitiateKeepalive
1 :14:32.976 MPTNBTCP PROV CALL       StopWritable(39 F 918,68)
1 :14:32.976 MPTNBTCP PROV RET        StopWritable(39 F 918,68)
1 :14:32.976 MPTNBTCP FUNC RET YIELD=AcceptFSM1
1 :14:33.535 MPTNBTCP PROV RET        sock_select(1, )
1 :14:33.536 MPTNBTCP FUNC CALL ListenFSM(39 F 294)
1 :14:33.536 MPTNBTCP PROV CALL       sock_accept(67)
1 :14:33.536 MPTNBTCP PROV RET        sock_accept(69)
1 :14:33.537 MPTNBTCP FUNC CALL InitPmmCb(39 F B9A)
1 :14:33.537 MPTNBTCP FUNC CALL InitLocalData(39 F B9A)
1 :14:33.537 MPTNBTCP FUNC CALL InitLocalData(39 F BD8)
1 :14:33.537 MPTNBTCP FUNC CALL AddToCbList(39 F B9A)
1 :14:33.537 MPTNBTCP FUNC CALL AddToCbArray(39 F B9A)
1 :14:33.537 MPTNBTCP PROV CALL       sock_getsockname(69)
1 :14:33.537 MPTNBTCP PROV RET        sock_getsockname(69)
1 :14:33.537 MPTNBTCP PROV CALL       sock_inet_ntoa(AB8 43 9)
```

Trace Example

```
1 :14:33.538 MPTNBTCP PROV RET         sock_inet_ntoa(9.67.128.171)
1 :14:33.538 MPTNBTCP PROV CALL        sock_getpeername(69)
1 :14:33.538 MPTNBTCP PROV RET         sock_getpeername(69)
1 :14:33.538 MPTNBTCP PROV CALL        sock_inet_ntoa(998 43 9)
1 :14:33.538 MPTNBTCP PROV RET         sock_inet_ntoa(9.67.128.153)
1 :14:33.538 MPTNBTCP PROV CALL        sock_set_no_blocking(69)
1 :14:33.538 MPTNBTCP PROV RET         sock_set_no_blocking(69)
1 :14:33.538 MPTNBTCP PROV CALL        CheckReadable(39 F B9A,69)
1 :14:33.538 MPTNBTCP PROV RET         CheckReadable(39 F B9A,69)
1 :14:33.539 MPTNBTCP FUNC RET GO=ListenFSM
⋮
1 :14:33.676 MPTNBTCP FUNC CALL GetData(39 F B9A)
1 :14:33.676 MPTNBTCP PROV CALL        sock_recv(69)
1 :14:33.676 MPTNBTCP PROV RET         4=sock_recv(69)
1 :14:33.676 MPTNBTCP FUNC RET GO=GetData
1 :14:33.676 MPTNBTCP FUNC CALL GetData1(39 F B9A)
1 :14:33.677 MPTNBTCP ===RECV====> 9.67.128.171                9.67.128.153           2711
69 CMDLPREF 4

1 :14:33.677 Data:
                                    ...,
                                     6
                                     B
1 :14:33.677 MPTNBTCP FUNC RET GO=GetData1
1 :14:33.677 MPTNBTCP FUNC CALL GetData(39 F B9A)
1 :14:33.677 MPTNBTCP PROV CALL        sock_recv(69)
1 :14:33.677 MPTNBTCP PROV RET         1 3=sock_recv(69)
1 :14:33.678 MPTNBTCP FUNC RET GO=GetData
1 :14:33.678 MPTNBTCP FUNC CALL ListenFSM1(39 F B9A)
1 :14:33.678 MPTNBTCP ===RECV====> 9.67.128.171                9.67.128.153           2711
69 MPTNCONN 1 3

1 :14:33.678 Data:
                                    .....@.....USIBMNR.SNAIPGW.'TgX.>....  .....
                                    ..................................
                                    8  627  131    1EECCDDD4EDCCDCE  E8E36    4    7   72
        68    1    1     2
                                     A 7AC152  11B1 4292459B251977617377DE F   1   6   6  5
8 611A2 E2   B  D218  519A A 13
                                    .c....d.....  ...
                                    181   8A    4
                                     3A  54F2 98  51

1 :14:33.678 MPTNBTCP PROV CALL        StopReadable(39 F B9A,69)
1 :14:33.678 MPTNBTCP PROV RET         StopReadable(39 F B9A,69)
1 :14:33.678 MPTNBTCP FUNC CALL SockaddrToAddr
1 :14:33.679 MPTNBTCP FUNC CALL SockaddrToAddr
1 :14:33.679 MPTNCMM FUNC CALL ANY 1 8I: CMMUREQ(P_ACCEPT_NN_UC)  A

1 :14:33.627 MPTNCMM FUNC CALL ANY 176I: CreateMptnCb: MPTN_CB 38FF A38 created
1 :14:33.63  MPTNCMM FUNC CALL ANY 214I: FSMUpcall
1 :14:33.63  MPTNCMM FUNC CALL ANY 154I: CMMUCALL(M_ACCEPT_UC) 38FF A38
1 :14:33.63  M_ACCEPT    received
1 :14:33.63  Payload received

1 :14:33.63  Data:
                                    .  ...
                                     4
                                    8  51
1 :14:33.63  OUTB: [LH:] l=-7968 LSid=e e
                          [TH:] OIS  ->2 EFI SNF=
                          [RH:] REQ SC FI OIC DR1I
                         [RU:]  ACTLU
                                    ..

                                    21
1 :14:33.63  MPTNCMM FUNC RET ANY 154I: CMMUCALL(M_ACCEPT_UC) 38FF A38
1 :14:33.63  MPTNCMM FUNC CALL ANY 214I: FSMUpcall
1 :14:33.631 MPTNCMM FUNC RET ANY 1 8I: CMMUREQ(P_ACCEPT_NN_UC)
```

Trace Example

```
1 :14:33.631 Data:
                                  .....2.....%.....{.....
                                  .....
                                    6   F F3   6 F3 7 F3   C F3 E F3 9  33 F3

                                    7   2AF821 C7F8 87F821  6F8 CAF82ABF98AF8

1 :14:33.631 MPTNBTCP FUNC RET YIELD=ListenFSM1
⋮
1 :14:33.669 INB:    [LH:] l=-7968 LSid=e e
                       [TH:] OIS  <-2 EFI SNF=
                       [RH:] +RSP SC FI OIC DR1I
                      [RU:]  ACTLU
                                  ...e.....
                                    8
                                  11 5   c63 1
1 :14:33.669 M_ACCEPT    issued=
1 :14:33.67  MPTNCMM FUNC CALL ANY 233I: CMMDREQ(M_ACCEPT) 38FF A38

1 :14:33.67  Data:
                                  .....e.....p....  .....
                                      1   1       E8     8          8792   4  7   7

                                      7   7    2  B  D11 5   C63 1   E47D  1   6   6
2
1 :14:33.725 MPTNCMM FUNC CALL ANY 234I: CMMDCALL(P_ACCEPT_NN) 38FF A38
1 :14:33.728 MPTNCMM FUNC CALL ANY 214I: 38FF A38

1 :14:33.728 Data:
                                  .....cd.....=.....
                                  .....  .....
                                           2188         7   7   7 F3   B1F3 1 F3
                                   51F3    3 F3         4  7
                                      7  13  34         4   4   E7F8B1FA5F8 2BF86   6
61F8    8AF81 1     1   6
                                  .....
                                  .....
                                    34 2  3 F3

                                    C277  8AF8


1 :14:33.728 MPTNBTCP FUNC CALL TCPMREQ: P_ACCEPT_NN
1 :14:33.728 MPTNBTCP FUNC CALL CopyPCreateFlds(39 F B9A)
1 :14:33.728 MPTNBTCP FUNC CALL ProcessCompHdrs
1 :14:33.728 MPTNBTCP FUNC RET 1=ProcessCompHdrs
1 :14:33.728 MPTNBTCP FUNC CALL AddToStubQueue
1 :14:33.728 MPTNBTCP FUNC RET  =AddToStubQueue
1 :14:33.728 MPTNCMM FUNC RET ANY 234I: CMMDCALL(P_ACCEPT_NN) 38FF A38
1 :14:33.729 MPTNCMM FUNC CALL ANY 214I: CleanupMptnCb
1 :14:33.729 MPTNCMM FUNC CALL ANY 214I: FreeConnInfoBufs

1 :14:33.73  MPTNCMM FUNC CALL ANY 214I: FreeM_INFO
1 :14:33.731 MPTNCMM FUNC RET ANY 233I: CMMDREQ(M_ACCEPT) 38FF A38
1 :14:33.78  MPTNBTCP PROV RET          sock_select(1, )
⋮
1 :14:33.787 MPTNBTCP FUNC CALL ProcessMessage
1 :14:33.787 MPTNBTCP FUNC CALL AcceptFSM(39 F B9A)
1 :14:33.787 MPTNBTCP <==SEND===== 9.67.128.171                  9.67.128.153           2711
69 CMDLPREF 4

1 :14:33.787 Data:
                                  ....
                                    7
                                    8
1 :14:33.787 MPTNBTCP <==SEND===== 9.67.128.171                  9.67.128.153           2711
69 POSMCRSP 116            {
```

```
1 :14:33.787 Data:
                                ....@.........USIBMNR.SNAIPGW.TgX.>....  .............
                                .....................e...........
                                88 727  131    1EECCDDD4EDCCDCE  E8E36    4    7   72
      1    E8     8
                                 A 4AC152  11B1 4292459B251977617377DE F   1   6   6  5
8 611A2 B 2  B  D11 5   C63 1
                                ..............c....d.........
                                1    1      2181   8A
                                8  519A A 13  3A  54F1 98  51

1 :14:33.788 MPTNBTCP PROV CALL        Writable(39 F B9A,69)
1 :14:33.788 MPTNBTCP PROV RET         Writable(39 F B9A,69)
1 :14:33.788 MPTNBTCP FUNC RET GO=AcceptFSM
1 :14:33.788 MPTNBTCP FUNC CALL PutData(39 F B9A)
1 :14:33.788 MPTNBTCP PROV CALL        sock_send(69, 4)
1 :14:33.789 MPTNBTCP PROV RET         4=sock_send(69)
1 :14:33.789 MPTNBTCP FUNC RET GO=PutData
1 :14:33.789 MPTNBTCP FUNC CALL PutData1(39 F B9A)
1 :14:33.789 MPTNBTCP FUNC RET GO=PutData1

1 :14:33.789 MPTNBTCP FUNC CALL PutData(39 F B9A)
1 :14:33.79  MPTNBTCP PROV CALL        sock_send(69, 116)
1 :14:33.79  MPTNBTCP PROV RET         -1=sock_send(69, 1  35)
1 :14:33.79  MPTNBTCP FUNC RET YIELD=PutData
⋮
1 :14:33.736 MPTNBTCP FUNC CALL PutData(39 F B9A)
1 :14:33.737 MPTNBTCP PROV CALL        sock_send(69, 116)

1 :14:33.737 MPTNBTCP PROV RET         -1=sock_send(69, 1  35)
1 :14:33.737 MPTNBTCP FUNC RET YIELD=PutData
⋮
1 :14:34. 6  MPTNBTCP FUNC CALL PutData(39 F B9A)
1 :14:34. 6  MPTNBTCP PROV CALL        sock_send(69, 116)
1 :14:34. 61 MPTNBTCP PROV RET         116=sock_send(69)
1 :14:34. 61 MPTNBTCP FUNC RET GO=PutData
1 :14:34. 62 MPTNBTCP FUNC CALL AcceptFSM1(39 F B9A)
1 :14:34. 62 MPTNBTCP PROV RET         CheckReadable(39 F B9A,69)

1 :14:34. 62 MPTNBTCP PROV CALL        sock_keep_alive(69)
1 :14:34. 62 MPTNBTCP PROV RET         sock_keep_alive(69)
1 :14:34. 62 MPTNBTCP FUNC CALL InitiateKeepalive(39 F B9A)
1 :14:34. 63 MPTNBTCP FUNC CALL FindPartnerNode(39 F B9A)
1 :14:34. 63 MPTNBTCP FUNC RET 38FF 976=FindPartnerNode
1 :14:34. 63 MPTNBTCP FUNC RET GO=InitiateKeepalive

1 :14:34. 63 MPTNBTCP PROV CALL        StopWritable(39 F B9A,69)
1 :14:34. 63 MPTNBTCP PROV RET         StopWritable(39 F B9A,69)
1 :14:34. 63 MPTNBTCP FUNC RET YIELD=AcceptFSM1
⋮
1 :14:34.323 MPTNBTCP FUNC CALL RecvFSM(39 F B9A)
1 :14:34.324 MPTNBTCP FUNC RET GO=RecvFSM
1 :14:34.324 MPTNBTCP FUNC CALL GetData(39 F B9A)
1 :14:34.324 MPTNBTCP PROV CALL        sock_recv(69)
1 :14:34.324 MPTNBTCP PROV RET         5=sock_recv(69)
1 :14:34.324 MPTNBTCP FUNC RET GO=GetData
1 :14:34.324 MPTNBTCP FUNC CALL RecvFSM1(39 F B9A)
1 :14:34.324 MPTNBTCP ===RECV====> 9.67.128.171                9.67.128.153           2711
 69 DATALPHD 5
```

Trace Example

```
1 :14:34.324 Data:
                                    .....
                                      4
                                      4
1 :14:34.325 MPTNBTCP FUNC CALL GetBuffer(39 F B9A)
1 :14:34.325 M_GETBUF    received
1 :14:34.325 MPTNBTCP FUNC RET GO=GetBuffer
1 :14:34.325 MPTNBTCP FUNC RET GO=RecvFSM1
1 :14:34.325 MPTNBTCP FUNC CALL GetData(39 F B9A)
1 :14:34.325 MPTNBTCP PROV CALL        sock_recv(69)
1 :14:34.325 MPTNBTCP PROV RET         63=sock_recv(69)
1 :14:34.325 MPTNBTCP FUNC RET GO=GetData
1 :14:34.326 MPTNBTCP FUNC CALL RecvFSM2(39 F B9A)
1 :14:34.326 MPTNBTCP ===RECV====> 9.67.128.171                9.67.128.153          2711
 69 DATA         63

1 :14:34.326 Data:
                                    ......USSMSG1 : WINLU2      ENTER LOGON APPLID() DATA() LO
GMODE().
                                      8 1EEEDECFF74ECDDEF444CDECD4DDCDD4CDDDCC454CCEC454DD
CDDCC451
                                    33  54224271 A 695342   55359 36765 177394DD 4131DD 36
74645DD5

1 :14:34.326 MPTNCMM FUNC CALL ANY 11 I: CMMUREQ(P_RECEIVE_UC) 38FF A38
1 :14:34.326 MPTNCMM FUNC CALL ANY 157I: CMMUCALL(M_RECEIVE_UC) 38FF A38
1 :14:34.326 M_RECEIVE receivedN
1 :14:34.326 OUTB: [LH:] l=-7968 LSid=e e
                           [TH:] OIS  ->2 SNF=   3
                           [RH:] REQ FMD OIC DR1I
                           [RU:] .USSMSG1 : WINLU2      ENTER LOGON APPLID() DATA
().
                                 1eeedecff74ecddef444cdecd4ddcdd4cdddcc454ccec454ddcddcc
451
                                 54224271 a 695342   55359 36765 177394dd 4131dd 3674645
dd5
1 :14:34.327 MPTNCMM FUNC RET ANY 157I: CMMUCALL(M_RECEIVE_UC) 38FF A38
1 :14:34.327 MPTNCMM FUNC RET ANY 11 I: CMMUREQ(P_RECEIVE_UC) 38FF A38
1 :14:34.327 MPTNBTCP FUNC RET GO=RecvFSM2
1 :14:34.327 MPTNBTCP FUNC CALL RecvFSM(39 F B9A)
1 :14:34.327 MPTNBTCP FUNC RET GO=RecvFSM
1 :14:34.327 MPTNBTCP FUNC CALL GetData(39 F B9A)
1 :14:34.327 MPTNBTCP PROV CALL        sock_recv(69)
1 :14:34.328 MPTNBTCP PROV RET         -1=sock_recv(69, 1  35)
1 :14:34.328 MPTNBTCP FUNC RET YIELD=GetData
1 :14:34.332 INB:    [LH:] l=-7968 LSid=e e
                           [TH:] OIS  <-2 SNF=   3
                           [RH:] +RSP FMD OIC DR1I

1 :14:34.332 M_SEND        issued
1 :14:34.333 MPTNCMM FUNC CALL ANY 143I: CMMDREQ(M_SEND) 38FF A38
1 :14:34.333 Data:
                                    ......................
                                           D F2
                                      5       C3F1    1
1 :14:34.333 MPTNCMM FUNC CALL ANY  94I: CMMDCALL(P_SEND) 38FF A38
1 :14:34.333 MPTNBTCP FUNC CALL TCPMREQ: P_SEND
1 :14:34.333 MPTNBTCP FUNC CALL AddToStubQueue
1 :14:34.333 MPTNBTCP FUNC RET  =AddToStubQueue
1 :14:34.334 MPTNCMM FUNC RET ANY  94I: CMMDCALL(P_SEND) 38FF A38
1 :14:34.334 MPTNCMM FUNC RET ANY 143I: CMMDREQ(M_SEND) 38FF A38
```

Trace Example

```
1 :14:34.482 MPTNBTCP FUNC CALL ProcessMessage
1 :14:34.483 MPTNBTCP PROV CALL       Writable(39 F B9A,69)
1 :14:34.483 MPTNBTCP PROV RET        Writable(39 F B9A,69)
1 :14:34.483 MPTNBTCP FUNC CALL SendFSM(39 F B9A)
1 :14:34.483 MPTNBTCP <==SEND===== 9.67.128.171                9.67.128.153          2711
69 DATALPHD 5
1 :14:34.483 Data:
                                    .....

                                     A
1 :14:34.484 MPTNBTCP <==SEND===== 9.67.128.171                9.67.128.153          2711
69 DATA        5

1 :14:34.484 Data:
                                   ..c..
                                    88
                                   33
1 :14:34.484 MPTNBTCP FUNC RET GO=SendFSM
1 :14:34.484 MPTNBTCP FUNC CALL PutData(39 F B9A)
1 :14:34.484 MPTNBTCP PROV CALL       sock_send(69, 5)
1 :14:34.485 MPTNBTCP PROV RET        5=sock_send(69)
1 :14:34.485 MPTNBTCP FUNC RET GO=PutData
1 :14:34.485 MPTNBTCP FUNC CALL PutData1(39 F B9A)
1 :14:34.485 MPTNBTCP FUNC RET GO=PutData1
1 :14:34.486 MPTNBTCP FUNC CALL PutData(39 F B9A)
1 :14:34.486 MPTNBTCP PROV CALL       sock_send(69, 5)
1 :14:34.486 MPTNBTCP PROV RET        -1=sock_send(69, 1  35)

1 :14:34.486 MPTNBTCP FUNC RET YIELD=PutData
1 :14:34.59  MPTNBTCP PROV RET        sock_select( ,1)
1 :14:34.59  MPTNBTCP FUNC CALL PutData(39 F B9A)
1 :14:34.59  MPTNBTCP PROV CALL       sock_send(69, 5)
1 :14:34.591 MPTNBTCP PROV RET        5=sock_send(69)
1 :14:34.592 MPTNBTCP FUNC RET GO=PutData
1 :14:34.592 MPTNBTCP FUNC CALL SendFSM1(39 F B9A)
1 :14:34.592 M_FREEBUF received;
1 :14:34.592 MPTNBTCP PROV CALL       StopWritable(39 F B9A,69)
1 :14:34.592 MPTNBTCP PROV RET        StopWritable(39 F B9A,69)
1 :14:34.592 MPTNBTCP FUNC RET YIELD=SendFSM
```

Trace Example

Glossary

This glossary includes terms and definitions from:

The *American National Standard Dictionary for Information Systems*, ANSI X3.172-1990, copyright 1990 by the American National Standards Institute (ANSI). Copies may be purchased from the American National Standards Institute, 11 West 42nd Street, New York, New York 10036. Definitions are identified by the symbol (A) after the definition.

The ANSI/EIA Standard—440-A, *Fiber Optic Terminology*. Copies may be purchased from the Electronic Industries Association, 2001 Pennsylvania Avenue, N.W., Washington, DC 20006. Definitions are identified by the symbol (E) after the definition.

The *Information Technology Vocabulary*, developed by Subcommittee 1, Joint Technical Committee 1, of the International Organization for Standardization and the International Electrotechnical Commission (ISO/IEC JTC1/SC1). Definitions of published parts of this vocabulary are identified by the symbol (I) after the definition; definitions taken from draft international standards, committee drafts, and working papers being developed by ISO/IEC JTC1/SC1 are identified by the symbol (T) after the definition, indicating that final agreement has not yet been reached among the participating National Bodies of SC1.

The *IBM Dictionary of Computing*, New York: McGraw-Hill, 1994.

Internet Request for Comments: 1208, *Glossary of Networking Terms*.

Internet Request for Comments: 1392, *Internet Users' Glossary*.

The *Object-Oriented Interface Design: IBM Common User Access Guidelines*, Carmel, Indiana: Que, 1992.

The following cross-references are used in this glossary:

Contrast with: This refers to a term that has an opposed or substantively different meaning.

Synonym for: This indicates that the term has the same meaning as a preferred term, which is defined in its proper place in the glossary.

Synonymous with: This is a backward reference from a defined term to all other terms that have the same meaning.

See: This refers the reader to multiple-word terms that have the same last word.

See also: This refers the reader to terms that have a related, but not synonymous, meaning.

Deprecated term for: This indicates that the term should not be used. It refers to a preferred term, which is defined in its proper place in the glossary.

A

ACDI. Asynchronous Communications Device Interface.

adapter. A part that electrically or physically connects a device to a computer or to another device.

address class. In Internet communications, the categorization by the part of an IP address that distinguishes the network address from the host address. Class A addresses allocate 7 bits to the network ID and 24 bits to the host ID. Class B addresses allocate 14 bits to the network ID and 16 bits to the host ID. Class C addresses allocate 21 bits to the network ID and 8 bits to the host ID. Class D addresses contain 1110 in the first 4 bits and identify the address as a multicast. The remaining 28 bits in the class D address specify a particular multicast group.

address-mapper function. In MPTN architecture, a component that maps nonnative transport-user addresses to a form used in the native transport network.

address mask. For internet subnetworking, a 32-bit mask used to identify the subnetwork address bits in the host portion of an IP address. Synonymous with *subnet mask* and *subnetwork mask*.

address resolution. A method for mapping network-layer addresses to media-specific addresses.

address space. In MPTN architecture, the set of all legal transport addresses that may be formed according to the rules of a given address type. These rules include the maximum number of characters that can be in the address and the permissible characters. Each transport protocol has its own set of rules. Since addresses in one protocol may also be legitimate in another protocol, MTPN qualifies all transport addresses with an address type.

© Copyright IBM Corp. 1996

address space identifier (ASID). A unique, system-assigned identifier for an address space.

address type. In MPTN architecture, an identifier in an MPTN header that indicates the transport protocol category (for example, OSI or TCP/IP) and hence the specific syntax and structure of the accompanying address. A given transport-user address plus its address type form an MPTN-qualified address.

Administrative Domain. A collection of hosts and routers, and the interconnecting networks, managed by a single administrative authority.

Advanced Peer-to-Peer Networking (APPN). An extension to SNA featuring (a) greater distributed network control that avoids critical hierarchical dependencies, thereby isolating the effects of single points of failure; (b) dynamic exchange of network topology information to foster ease of connection, reconfiguration, and adaptive route selection; (c) dynamic definition of network resources; and (d) automated resource registration and directory lookup. APPN extends the LU 6.2 peer orientation for end-user services to network control and supports multiple LU types, including LU 2, LU 3, and LU 6.2.

Advanced Peer-to-Peer Networking (APPN) end node. A node that provides a broad range of end-user services and supports sessions between its local control point (CP) and the CP in an adjacent network node. It uses these sessions to dynamically register its resources with the adjacent CP (its network node server), to send and receive directory search requests, and to obtain management services. An APPN end node can also attach to a subarea network as a peripheral node or to other end nodes.

Advanced Peer-to-Peer Networking (APPN) network. A collection of interconnected network nodes and their client end nodes.

Advanced Peer-to-Peer Networking (APPN) network node. A node that offers a broad range of end-user services and that can provide the following:

- Distributed directory services, including registration of its domain resources to a central directory server
- Topology database exchanges with other APPN network nodes, enabling network nodes throughout the network to select optimal routes for LU-LU sessions based on requested classes of service
- Session services for its local LUs and client end nodes
- Intermediate routing services within an APPN network

Advanced Peer-to-Peer Networking (APPN) node. An APPN network node or an APPN end node.

advanced program-to-program communication (APPC). (1) The general facility characterizing the LU 6.2 architecture and its various implementations in products. (2) Sometimes used to refer to the LU 6.2 architecture and its product implementations as a whole, or to an LU 6.2 product feature in particular, such as an APPC application programming interface.

AIX. Advanced Interactive Executive.

AIX operating system. IBM's implementation of the UNIX operating system. The RISC System/6000 system, among others, runs the AIX operating system.

alert. A message sent to a management services focal point in a network to identify a problem or an impending problem.

AnyNet product family. A group of IBM products that implement the multiprotocol transport networking (MPTN) architecture, thus enabling application programs to communicate independently of the underlying network transport protocol.

APAR. Authorized program analysis report.

API. Application programming interface.

APPC. Advanced program-to-program communication.

application program interface. See *application programming interface (API)*.

application programming interface (API). (1) The set of programming language constructs or statements that can be coded in an application program to obtain the specific functions and services provided by an underlying operating system or service program. (2) In VTAM, the language structure used in control blocks so that application programs can reference them and be identified to VTAM.

APPN. Advanced Peer-to-Peer Networking.

ARPANET. A network established by the United States Department of Defense Advanced Research Projects Agency (now the Defense Advanced Research Projects Agency).

ASID. Address space identifier.

Asynchronous Communications Device Interface (ACDI). An application programming interface or service that is used by application programs. Application programs use the ACDI service to communicate with programs on other systems that use asynchronous communications.

authorized program analysis report (APAR). A report of a problem caused by a suspected defect in a current unaltered release of a program.

AUTOEXEC.BAT file. In the DOS operating system, a batch file that resides in the root directory of the boot drive. AUTOEXEC.BAT contains commands that DOS executes every time the PC is booted.

B

below-specific. In MPTN architecture, pertaining to a specific transport provider.

below-specific protocol boundary (BSPB). In MPTN architecture, the interface between the common MPTN manager (CMM) and the protocol-specific MPTN manager (PMM).

Berkeley Software Distribution (BSD). Pertaining to any of the series of UNIX specifications or implementations distributed by the University of California at Berkeley. The mnemonic “BSD” is usually followed by a number to specify the particular version of UNIX that was distributed (for example, BSD 4.3). Many vendors use BSD specifications as standards for their UNIX products.

BGP. Border Gateway Protocol.

boot. To prepare a computer system for operation by loading an operating system.

Border Gateway Protocol (BGP). An Internet Protocol (IP) routing protocol used between domains and autonomous systems. Contrast with *Exterior Gateway Protocol (EGP)*.

bridge. (1) A functional unit that interconnects two local area networks that use the same logical link control protocol but may use different medium access control protocols. (T) (2) A functional unit that interconnects multiple LANs (locally or remotely) that use the same logical link control protocol but that can use different medium access control protocols. A bridge forwards a frame to another bridge based on the medium access control (MAC) address. (3) In the connection of local loops, channels, or rings, the equipment and techniques used to match circuits and to facilitate accurate data transmission. (4) Contrast with *gateway* and *router*.

broadcast. (1) Transmission of the same data to all destinations. (T) (2) Simultaneous transmission of data to more than one destination. (3) Contrast with *multicast*.

BSD. Berkeley Software Distribution.

BSPB. Below-specific protocol boundary.

button. See *mouse button*, *push button*, *radio button*, and *spin button*.

C

CICS. Customer Information Control System.

CID. Configuration, installation, and distribution. See *CID methodology*.

CID methodology. An IBM-specified way to install and configure products on, or remove products from, remote workstations and hosts. Response files and redirected installation and configuration may be used by a CID-enabled product to eliminate or reduce user interaction with the CID-enabled product.

class A network. In Internet communications, a network in which the high-order (most significant) bit of the IP address is set to 0 and the host ID occupies the three low-order octets.

class B network. In Internet communications, a network in which the two high-order (most significant and next-to-most significant) bits of the IP address are set to 1 and 0, respectively, and the host ID occupies the two low-order octets.

class C network. In Internet communications, a network in which the two high-order (most significant and next-to-most significant) bits of the IP address are both set to 1 and the next high-order bit is set to 0. The host ID occupies the low-order octet.

client. (1) A functional unit that receives shared services from a server. (T) (2) A user. (3) In an AIX distributed file system environment, a system that is dependent on a server to provide it with programs or access to programs. (4) Synonymous with *requester*.

client/server. In communications, the model of interaction in distributed data processing in which a program at one site sends a request to a program at another site and awaits a response. The requesting program is called a client; the answering program is called a server.

CMIP. Common Management Information Protocol.

CMISE. Common Management Information Service Element.

CMM. Common MPTN manager.

Common Management Information Protocol (CMIP). The OSI standard protocol defined in ISO/IEC 9596-1 for the interaction between managers and agents that use the Common Management Information Service Element (CMISE).

Common Management Information Service Element (CMISE). The particular application service element defined in ISO/IEC 9595.

common MPTN manager (CMM). The component of the MPTN architecture that provides services independent of any transport protocol. Examples include registering transport users with the MPTN address-mapper component, selecting a transport provider, and establishing MPTN connections.

Common Programming Interface for Communications (CPI-C). An evolving application programming interface (API), embracing functions to meet the growing demands from different application environments and to achieve openness as an industry standard for communications programming. CPI-C provides access to interprogram services such as (a) sending and receiving data, (b) synchronizing processing between programs, and (c) notifying a partner of errors in the communication.

common transport semantics (CTS). The layer of the Networking Blueprint above the transport layer that makes the services of transport providers available to the transport user. See also *transport-layer protocol boundary (TLPB)*.

Communications Manager/2. An IBM licensed program that supports the development and use of OS/2 applications involving two or more connected systems or workstations. Communications Manager/2 provides multiple concurrent connectivities using different protocols for IBM 3270 and 5250 emulation sessions, printer sessions, and file transfers. It supports a range of application programming interfaces (APIs), which may be called concurrently and are designed for a variety of applications. Communications Manager/2 includes the necessary interfaces for network management.

Communications Manager/2 Kernel. In Communications Manager/2, the program primarily responsible for starting and stopping communication services and for tracking the various resources and services used by different emulators and user transaction programs (TPs). It also tracks the status of user sessions, and it allocates and initializes the system shared segment.

compensation. In MPTN architecture, the function of making up for differences in functions requested by the transport user and those provided by the transport provider.

composite network. In MPTN architecture, a single-protocol transport network made up of multiple individual networks running the same transport protocol, each with its own unique net ID.

CONFIG.SYS file. In the OS/2 operating system, a file used by the base operating system that describes the devices, system parameters, and resource options of a workstation. See also *configuration file*.

configuration file. A file that specifies the characteristics of a system device or network.

connection. (1) In data communication, an association established between functional units for conveying information. (I) (A) (2) In SNA, the network path that links together two logical units (LUs) in different nodes to enable them to establish communications. (3) In TCP/IP, the path between two protocol applications that provides reliable data stream delivery service. In the Internet, a connection extends from a TCP application on one system to a TCP application on another system.

connection-oriented service. A service that establishes a logical connection between two partners for the duration that they want to communicate. Data transfer takes place in a reliable, sequenced manner. Contrast with *connectionless service*.

connectionless service. A network service that treats each packet or datagram as a separate entity that contains the source address and destination address and for which no acknowledgment is returned to the originating source. Connectionless services are on a best-effort basis and do not guarantee reliable or

in-sequence delivery. Contrast with *connection-oriented service*.

conversation. A logical connection between two transaction programs using an LU 6.2 session. Conversations are delimited by brackets to gain exclusive use of a session.

CPI-C. Common Programming Interface for Communications.

CTS. Common transport semantics.

Customer Information Control System (CICS). An IBM licensed program that enables transactions entered at remote terminals to be processed concurrently by user-written application programs. It includes facilities for building, using, and maintaining databases.

D

data link control (DLC). A set of rules used by nodes on a data link (such as an SDLC link or a token ring) to accomplish an orderly exchange of information.

datagram. (1) In packet switching, a self-contained packet, independent of other packets, that carries information sufficient for routing from the originating data terminal equipment (DTE) to the destination DTE without relying on earlier exchanges between the DTEs and the network. (I) (2) In TCP/IP, the basic unit of information passed across the Internet environment. A datagram contains a source and destination address along with the data. An Internet Protocol (IP) datagram consists of an IP header followed by the transport layer data. (3) See also *packet* and *segment*.

DCE. Distributed Computing Environment.

dependent LU. See *SSCP-dependent LU*.

dependent LU requester (DLUR). An APPN end node or an APPN network node that owns dependent LUs, but requests that a dependent LU server provide the SSCP services for those dependent LUs.

dependent LU server (DLUS). An APPN network node that provides SSCP services for a dependent LU in its own or another APPN network. Contrast with *dependent LU requester*.

destination address. A code that identifies the location to which information is to be sent.

Distributed Computing Environment (DCE). The Open Software Foundation (OSF) specification (or a product derived from this specification) that assists in networking. DCE provides such functions as authentication, directory service (DS), and remote procedure call (RPC).

DLC. Data link control.

DLL. Dynamic link library.

DLUR. Dependent LU requester.

DLUS. Dependent LU server.

DNS. Domain Name System.

domain. (1) That part of a computer network in which the data processing resources are under common control. (T) (2) In SNA, see *end node domain*, *network node domain*, and *system services control point (SSCP) domain*. (3) See *Administrative Domain* and *domain name*.

domain name. In the Internet suite of protocols, a name of a host system. A domain name consists of a sequence of subnames separated by a delimiter character. For example, if the fully qualified domain name (FQDN) of a host system is ralvm7.vnet.ibm.com, each of the following is a domain name:

ralvm7.vnet.ibm.com
vnet.ibm.com
ibm.com

domain name server. In the Internet suite of protocols, a server program that supplies name-to-address translation by mapping domain names to IP addresses. Synonymous with *name server*.

Domain Name System (DNS). In the Internet suite of protocols, the distributed database system used to map domain names to IP addresses.

DOS. Disk Operating System. See *IBM Disk Operating System*.

dotted decimal notation. The syntactical representation for a 32-bit integer that consists of four 8-bit numbers written in base 10 with periods (dots) separating them. It is used to represent IP addresses.

download. (1) To transfer programs or data from a computer to a connected device, typically a personal computer. (T) (2) To transfer data from a computer to a

connected device, such as a workstation or microcomputer. Contrast with *upload*.

duplex. Pertaining to communication in which data can be sent and received at the same time. Synonymous with *full-duplex*. Contrast with *half-duplex*.

dynamic link library (DLL). A file containing executable code and data bound to a program at load time or run time, rather than during linking. The code and data in a dynamic link library can be shared by several applications simultaneously.

E

EGP. Exterior Gateway Protocol.

EN. End node.

end node (EN). (1) See *Advanced Peer-to-Peer Networking (APPN) end node* and *low-entry networking (LEN) end node*. (2) In communications, a node that is frequently attached to a single data link and cannot perform intermediate routing functions.

end node domain. An end node control point, its attached links, and its local LUs.

environment variable. Any of a number of variables that describe the way an operating system is going to run and the devices it is going to recognize.

Exterior Gateway Protocol (EGP). In the Internet suite of protocols, a protocol, used between domains and autonomous systems, that enables network reachability information to be advertised and exchanged. IP network addresses in one autonomous system are advertised to another autonomous system by means of EGP-participating routers. Contrast with *Border Gateway Protocol (BGP)*.

F

FDX. Full-duplex.

FFST. First Failure Support Technology.

File Transfer Protocol (FTP). In the Internet suite of protocols, an application layer protocol that uses TCP and Telnet services to transfer bulk-data files between machines or hosts.

file transfer, access, and management (FTAM). An application service element that enables user application processes to manage and access a file system, which may be distributed. (T)

first failure data capture. A problem diagnosis aid that identifies certain errors, gathers information about these errors, and presents this information for problem resolution.

First Failure Support Technology (FFST). Technology and software that provide first failure data capture in a complex environment. FFST is available with various operating systems, such as MVS, VM, and OS/2.

FQDN. Fully qualified domain name.

FTAM. File transfer, access, and management.

FTP. File Transfer Protocol.

full-duplex (FDX). Synonym for *duplex*.

fully qualified domain name (FQDN). In the Internet suite of protocols, the name of a host system that includes all of the subnames of the domain name. An example of a fully qualified domain name is ralvm7.vnet.ibm.com. See also *host name*.

fully qualified name. (1) In SNA, synonym for *network-qualified name*. (2) In the Internet suite of protocols, see *fully qualified domain name (FQDN)*.

G

gateway. (1) A functional unit that interconnects two computer networks with different network architectures. A gateway connects networks or systems of different architectures. A bridge interconnects networks or systems with the same or similar architectures. (T) (2) The combination of machines and programs that provide address translation, name translation, and system services control point (SSCP) rerouting between independent SNA networks to allow those networks to communicate. A gateway consists of one gateway NCP and at least one gateway VTAM. (3) In the IBM Token-Ring Network, a device and its associated software that connect a local area network to another local area network or a host that uses different logical link protocols. (4) In the AIX operating system, an entity that operates above the link layer and translates, when required, the interface and protocol used by one network into those used by another distinct network. (5) In TCP/IP, synonym for *router*.

group address. A single transport address identifying a collection of users. The collection of users is formed so that they can all receive common multicast datagrams.

H

half-duplex (HD, HDX). In data communication, pertaining to transmission in only one direction at a time. Contrast with *duplex*.

half-duplex operation. A mode of operation of a data link in which data can be transmitted in both directions, one way at a time. (T)

half-duplex transmission. Data transmission in either direction, one direction at a time. (I) (A)

HD. Half-duplex.

HDX. Half-duplex.

header file. Synonym for *include file*.

heterogeneous network. In MPTN architecture, a single logical network formed by MPTN gateways joining individual networks that support different transport protocols.

hexadecimal. (1) Pertaining to a selection, choice, or condition that has 16 possible different values or states. (I) (2) Pertaining to a fixed-radix numeration system, with radix of 16. (I) (3) Pertaining to a system of numbers to the base 16; hexadecimal digits range from 0 through 9 and A through F, where A represents 10 and F represents 15.

host. In the Internet suite of protocols, an end system. The end system can be any workstation; it does not have to be a mainframe.

host address. See *IP address*.

host ID. In the Internet suite of protocols, that part of the IP address that defines the host system on the network. The length of the host ID depends on the type of network or network class (A, B, or C).

host name. In the Internet suite of protocols, the name given to a machine. Sometimes, “host name” is used to mean *fully qualified domain name (FQDN)*; other times, it is used to mean the most specific subname of a fully qualified domain name. For example, if ralvm7.vnet.ibm.com is the fully qualified domain name, either of the following may be considered the host name:

ralvm7.vnet.ibm.com
ralvm7

I

IAB. Internet Architecture Board.

IBM Disk Operating System (DOS). A disk operating system based on MS-DOS that operates with all IBM personal computers.

IBM Operating System/2 (OS/2). Pertaining to the IBM licensed program that can be used as the operating system for personal computers. The OS/2 licensed program can perform multiple tasks at the same time.

ICMP. Internet Control Message Protocol.

IETF. Internet Engineering Task Force.

include file. A text file that contains declarations used by a group of functions, programs, or users. Synonymous with *header file*.

internet. A collection of networks interconnected by a set of routers that allow them to function as a single, large network. See also *Internet*.

Internet. The internet administered by the Internet Architecture Board (IAB), consisting of large national backbone networks and many regional and campus networks all over the world. The Internet uses the Internet suite of protocols.

Internet address. See *IP address*.

Internet Architecture Board (IAB). The technical body that oversees the development of the Internet suite of protocols that are known as TCP/IP.

Internet Control Message Protocol (ICMP). The protocol used to handle errors and control messages in the Internet Protocol (IP) layer. Reports of problems and incorrect datagram destinations are returned to the original datagram source. ICMP is part of the Internet Protocol.

Internet drafts. Proposals, techniques, and mechanisms that document the Internet Engineering Task Force (IETF) work in progress and that define protocols and their characteristics in an internet. After the drafts are approved, they become Request for Comments (RFC) standards.

Internet Engineering Task Force (IETF). The task force of the Internet Architecture Board (IAB) that is responsible for solving the short-term engineering needs of the Internet.

internet layer. In Internet communications, the Internet Protocol (IP) layer that provides transparency over the physical network (transmission media) and over the internet topology. The internet layer is equivalent to the network layer in Open Systems Interconnection (OSI) architecture.

Internet Protocol (IP). A connectionless protocol that routes data through a network or interconnected networks. IP acts as an intermediary between the higher protocol layers and the physical network. However, this protocol does not provide error recovery and flow control and does not guarantee the reliability of the physical network.

Internet router. A device that enables an internet host to act as a gateway for routing data between separate networks that use a specific adapter.

Internet suite of protocols. A set of protocols developed for use on the Internet and published as Requests for Comments (RFCs).

Internetwork Packet Exchange (IPX). The network protocol used to connect Novell's servers, or any workstation or router that implements IPX, with other workstations. Although similar to the Internet Protocol (IP), IPX uses different packet formats and terminology.

IP. Internet Protocol.

IP address. The 32-bit address defined by the Internet Protocol, standard 5, Request for Comment (RFC) 791. It is usually represented in dotted decimal notation.

IP datagram. In the Internet suite of protocols, the fundamental unit of information transmitted through an internet. It contains source and destination addresses, user data, and control information such as the length of the datagram, the header checksum, and flags indicating whether the datagram can be or has been fragmented.

IPX. Internetwork Packet Exchange.

K

keepalive message. In Internet communications, a message sent among the Border Gateway Protocol (BGP) neighbors to ensure the vitality of the session.

kernel. (1) The part of an operating system that performs basic functions such as allocating hardware resources. (2) In Communications Manager/2, see also *Communications Manager/2 Kernel*.

L

LAN. Local area network.

LEN. Low-entry networking.

local area network (LAN). (1) A computer network located on a user's premises within a limited geographical area. Communication within a local area network is not subject to external regulations; however, communication across the LAN boundary may be subject to some form of regulation. (T) (2) A network in which a set of devices are connected to one another for communication and that can be connected to a larger network.

local host. In TCP/IP, the host on the network at which a particular operator is working.

logical unit (LU). A type of network accessible unit that enables end users to gain access to network resources and communicate with each other.

loopback interface. (1) An interface that bypasses unnecessary communications functions when the information is addressed to an entity within the same system. (2) See also *loopback test*.

loopback test. A test in which signals from a tester are looped at a modem or other network element back to the tester for measurements that determine or verify the quality of the communications path.

low-entry networking (LEN). A capability of nodes to attach directly to one another using basic peer-to-peer protocols to support multiple and parallel sessions between logical units.

low-entry networking (LEN) end node. A LEN node receiving network services from an adjacent APPN network node.

low-entry networking (LEN) node. A node that provides a range of end-user services, attaches directly to other nodes using peer protocols, and derives network services implicitly from an adjacent APPN network node, that is, without the direct use of CP-CP sessions.

LU. Logical unit.

LU type. The classification of an LU in terms of the specific subset of SNA protocols and options it supports for a given session, namely:

The mandatory and optional values allowed in the session activation request

The usage of data stream controls, function management headers (FMHs), request unit parameters, and sense data values

Presentation services protocols such as those associated with FMH usage

LU types 0, 1, 2, 3, 4, 6.1, 6.2, and 7 are defined.

LU 6.2. A type of logical unit that supports general communication between programs in a distributed processing environment. LU 6.2 is characterized by (a) a peer relationship between session partners, (b) efficient utilization of a session for multiple transactions, (c) comprehensive end-to-end error processing, and (d) a generic application programming interface (API) consisting of structured verbs that are mapped into a product implementation.

M

matching. In MPTN architecture, pertaining to the relationship between peer transport users or peer transport providers that use the same user protocols or the same transport protocols.

mode name. The name used by the initiator of a session to designate the characteristics desired for the session, such as traffic pacing values, message-length limits, sync point and cryptography options, and the class of service within the transport network.

mouse button. A mechanism on a mouse pointing device used to select objects or choices, initiate actions, or directly manipulate objects; a user presses a mouse button to interact with a computer system. The button makes a “clicking” sound when pressed and released.

MPTN. Multiprotocol transport networking.

MPTN access node. A node that has MPTN components installed, allowing transport users to use nonnative transport providers.

MPTN connection. An end-to-end connection through the MPTN network that may traverse multiple networks running different protocols. If the network consists of multiple MPTN segments, the MPTN connection is formed by having MPTN transport gateways concatenate the MPTN segments into one logical connection.

MPTN datagram. A datagram that carries an MPTN header as part of the data.

MPTN network. A network consisting of a mixture of native nodes, MPTN access nodes, MPTN address-mapper nodes, and MPTN transport-gateway nodes. The resulting network has the appearance to the user of one logical network. An MPTN network that consists of just a single transport network does not contain an MPTN transport gateway.

MPTN-qualified transport address. A transport address that is qualified by its corresponding address type. The address conforms to the syntax and meaning of the specified address type. An example of an MPTN-qualified transport address is the pair (type=SNA, transport address=network-qualified LU name).

MPTN segment. A connection across a single-protocol transport network between an MPTN node (either an MPTN access node or gateway node) and another node that may or may not be an MPTN node.

MPTN transport gateway. An MPTN component that concatenates two or more single-protocol networks to form an integrated heterogeneous network.

multicast. (1) Transmission of the same data to a selected group of destinations. (T) (2) A special form of broadcast in which copies of a packet are delivered to only a subset of all possible destinations.

multiprotocol node. An implementation of MPTN architecture that supports more than one transport protocol.

multiprotocol transport network. A collection of single-protocol transport networks, each using a different transport protocol.

multiprotocol transport networking (MPTN). A networking architecture that allows application programs using common upper-layer protocols and expecting the same transport services to communicate over transport

networks that may use protocols different from the transport network the programs were designed to use. For example, socket application programs that were originally designed to communicate over a TCP/IP transport network can, using MPTN support, communicate over an SNA transport network.

N

name resolution. In Internet communications, the process of mapping a machine name to the corresponding Internet Protocol (IP) address. See also *Domain Name System (DNS)*.

name server. In the Internet suite of protocols, synonym for *domain name server*.

native. In MPTN architecture, pertaining to the relationship between a transport user and a transport provider that are both based on the same transport protocol.

native MPTN segment. An MPTN segment that spans only a single transport protocol type and does not use any MPTN flows or headers. A native MPTN segment may form part of an MPTN connection.

native network. In MPTN architecture, with respect to a particular transport user, a transport network that provides the address type and transport characteristics assumed in the design of the transport user. No MPTN address mapping or compensation protocols are used for data transfer.

native node. In MPTN architecture, a node with no MPTN capability.

native routing gateway. In MPTN architecture, with respect to a given transport network, a protocol-specific gateway that implements the same routing protocol as that network.

native transport address. In MPTN architecture, a transport-user address having the address type that corresponds to the type employed by the transport network underlying the transport user, for example, an SNA name that is being registered within an SNA network.

net ID. See *network identifier*.

NetBEUI. NetBIOS Extended User Interface.

NetBIOS. Network Basic Input/Output System. A standard interface to networks, IBM personal computers (PCs), and compatible PCs, that is used on LANs to provide message, print-server, and file-server functions. Application programs that use NetBIOS do not need to handle the details of LAN data link control (DLC) protocols.

NetBIOS Extended User Interface (NetBEUI). The application programming interface (API) to the NetBIOS transport protocol.

NETID. See *network identifier*.

NetView program. An IBM licensed program used to monitor and manage a network and to diagnose network problems.

network identifier. (1) In TCP/IP, that part of the IP address that defines a network. The length of the network ID depends on the type of network class (A, B, or C). (2) A 1- to 8-byte customer-selected name or an 8-byte IBM-registered name that uniquely identifies a specific subnetwork. (3) In MPTN architecture, the address qualifier of a transport provider address that identifies a group of nodes according to the network in which they reside.

network node domain. An APPN network-node control point, its attached links, the network resources for which it answers directory search requests (namely, its local LUs and adjacent LEN end nodes), the adjacent APPN end nodes with which it exchanges directory search requests and replies, and other resources (such as a local storage device) associated with its own node or an adjacent end node for which it provides management services.

network-qualified name. In SNA, a name that uniquely identifies a specific resource (such as an LU or a CP) within a specific network. It consists of a network identifier and a resource name, each of which is a 1- to 8-byte symbol string. Synonymous with *fully qualified name*.

Networking Blueprint. An open, highly modular framework for networking support using industry-wide standards. The Networking Blueprint (a) incorporates multiple protocols and multiple vendor components; (b) enables comprehensive systems management and application choices independent of the network; and (c) facilitates the support of new technologies.

nonmatching. In MPTN architecture, pertaining to the relationship between peer transport users or peer transport providers that use different protocols.

nonnative. In MPTN architecture, pertaining to the relationship between a transport user and a transport provider that are based on different transport protocols.

nonnative MPTN segment. A segment of an MPTN connection in which an MPTN protocol was used to establish the segment.

nonnative network. In MPTN architecture, with respect to a particular transport user, a transport network whose addressing structure and transport service are different from that assumed in the design of that transport user.

nonnative transport address. In MPTN architecture, a transport-user address having an address type different from the transport-network address type, for example, an OSI address for the target of a connection request carried on an SNA transport network.

O

Open Software Foundation (OSF). A nonprofit research and development organization whose goals are (a) to develop specifications and software for use in an open software environment and (b) to make the specifications and software available to information technology vendors under fair and equitable licensing terms. For example, OSF developed the Distributed Computing Environment (DCE).

OS/2 operating system. IBM Operating System/2.

OSF. Open Software Foundation.

P

packet. In data communication, a sequence of binary digits, including data and control signals, that is transmitted and switched as a composite whole. The data, control signals, and, possibly, error control information are arranged in a specific format. (I)

packet internet groper (PING). (1) In Internet communications, a program used in TCP/IP networks to test the ability to reach destinations by sending the destinations an Internet Control Message Protocol (ICMP) echo request and waiting for a reply. (2) In communications, a test of reachability.

PING. Packet internet groper.

ping command. The command that sends an Internet Control Message Protocol (ICMP) echo-request packet to a gateway, router, or host with the expectation of receiving a reply.

PMM. Protocol-specific MPTN manager.

port. (1) An access point for data entry or exit. (2) A connector on a device to which cables for other devices such as display stations and printers are attached. (3) The representation of a physical connection to the link hardware. A port is sometimes referred to as an adapter; however, there can be more than one port on an adapter. There may be one or more ports controlled by a single DLC process. (4) In the Internet suite of protocols, a 16-bit number used to communicate between TCP or the User Datagram Protocol (UDP) and a higher-level protocol or application. Some protocols, such as File Transfer Protocol (FTP) and Simple Mail Transfer Protocol (SMTP), use the same well-known port number in all TCP/IP implementations. (5) An abstraction used by transport protocols to distinguish among multiple destinations within a host machine. (6) Synonymous with *socket*.

port number. In Internet communications, the identification of an application entity to the transport service.

program. (1) A sequence of instructions suitable for processing by a computer. Processing may include the use of an assembler, a compiler, an interpreter, or a translator to prepare the program for execution, as well as to execute it. (I) (2) In programming languages, a logical assembly of one or more interrelated modules. (I) (3) To design, write, and test computer programs. (I) (A)

protocol boundary. The signals and rules governing interactions between two components within a node.

protocol-specific MPTN manager (PMM). A component of the MPTN architecture that performs management, routing, and binding functions that are performed differently for the different transport providers.

push button. A button, labeled with text, graphics, or both, that represents an action that will be initiated when a user selects it.

R

radio button. A circle with text beside it. Radio buttons are combined to show a user a fixed set of choices from which the user can select one. The circle becomes partially filled when a choice is selected.

record data format. A format that maintains record boundaries for the data being transmitted.

redirect. To define or use a logical device name as a reference to another device or file that may be local or remote.

Remote Execution Protocol (REXEC). A protocol that allows the execution of a command or program on any host in the network. The local host receives the results of the command execution.

remote procedure call (RPC). A facility that a client uses to request the execution of a procedure call from a server. This facility includes a library of procedures and an external data representation.

Request for Comments (RFC). In Internet communications, the document series that describes a part of the Internet suite of protocols and related experiments. All Internet standards are documented as RFCs.

requester. Synonym for *client*.

resolver. In TCP/IP, a program or subroutine that obtains information from a domain name server or a local table for use by an application program.

response file. A file that contains a set of predefined answers to questions asked by a program and that is used in place of user dialog. See also *CID methodology*.

REXEC. Remote Execution Protocol.

RFC. Request for Comments.

rlogin (remote login). A service, offered by Berkeley UNIX-based systems, that allows authorized users of one machine to connect to other UNIX systems across an internet and interact as if their terminals were connected directly. The rlogin software passes information about the user's environment (for example, terminal type) to the remote machine.

route. (1) An ordered sequence of nodes and transmission groups (TGs) that represent a path from an origin node to a destination node traversed by the traffic exchanged between them. (2) The path that network traffic uses to get from source to destination.

route daemon. A program that runs under 4BSD UNIX to propagate route information among machines on a local area network. Also referred to as *routed* (pronounced “route-d”).

route metrics. In the Internet suite of protocols, the method used by the Routing Information Protocol (RIP) to determine the relative desirability of internet routes. The unit of measure is the number of hops between the origin and destination. In practice, the metric for a route is often assigned based on factors such as link transmission speed and line cost, not on the actual number of hops.

routed. Pronounced “route-d.” See *route daemon*.

router. (1) A computer that determines the path of network traffic flow. The path selection is made from several paths based on information obtained from specific protocols, algorithms that attempt to identify the shortest or best path, and other criteria such as metrics or protocol-specific destination addresses. (2) An attaching device that connects two LAN segments, which use similar or different architectures, at the reference model network layer. (3) In OSI terminology, a function that determines a path by which an entity can be reached. (4) In TCP/IP, synonymous with *gateway*. (5) Contrast with *bridge*.

routing. (1) The process of determining the path to be used for transmission of a message over a network. (T) (2) The assignment of the path by which a message is to reach its destination. (3) In SNA, the forwarding of a message unit along a particular path through a network, as determined by parameters carried in the message unit, such as the destination network address in a transmission header.

routing domain. In Internet communications, a group of intermediate systems that use a routing protocol so that the representation of the overall network is the same within each intermediate system. Routing domains are connected to each other by exterior links.

routing table. A collection of routes used to direct datagram forwarding or to establish a connection. The information is passed among routers to identify network topology and destination feasibility.

RPC. Remote procedure call.

S

segment. In Internet communications, the unit of transfer between TCP functions in different machines. Each segment contains control and data fields; the current byte-stream position and actual data bytes are identified along with a checksum to validate received data.

segmentation. A process by which path control (PC) divides basic information units (BIUs) into smaller units, called BIU segments, to accommodate smaller buffer sizes in adjacent nodes. Both segmentation and segment assembly are optional PC features. The support for either or both is indicated in the BIND request and response.

serial line Internet Protocol (SLIP). A protocol used to run Internet Protocol (IP) over serial lines, such as telephone circuits or RS-232 cables, interconnecting two systems.

server. (1) A functional unit that provides shared services to workstations over a network; for example, a file server, a print server, a mail server. (T) (2) In a network, a data station that provides facilities to other stations; for example, a file server, a print server, a mail server. (A) (3) In the AIX operating system, an application program that usually runs in the background and is controlled by the system program controller. (4) In the Enhanced X-Windows Toolkit, a program that provides the basic windowing mechanism. It handles interprocess communication (IPC) connections from clients, demultiplexes graphics requests onto screens, and multiplexes input back to clients.

Server-Requester Programming Interface (SRPI). An IBM application programming interface (API) used by requester and server programs to communicate with the personal computer or host routers.

service mode. In MPTN architecture, the designation by a transport user of the characteristics that must be maintained for a given connection or datagram transmission. Each networking protocol has its own way of requesting these characteristics, which must be mapped to the MPTN service mode.

service point (SP). An entry point that supports applications that provide network management for resources not under the direct control of itself as an entry point. Each resource is either under the direct control of another entry point or not under the direct control of any entry point. A service point accessing these resources is not required to use SNA sessions (unlike a focal point). A service point is needed when entry point support is not yet available for some network management function.

Simple Network Management Protocol (SNMP). In the Internet suite of protocols, a network management protocol that is used to monitor routers and attached networks. SNMP is an application layer protocol. Information on devices managed is defined and stored in the application's Management Information Base (MIB).

single-protocol transport network (SPTN). In MPTN architecture, a collection of physically connected nodes that implement a single common transport protocol. A single-protocol transport network may span multiple net IDs. Contrast with *multiprotocol transport network*.

SLIP. Serial line Internet Protocol.

SNA. Systems Network Architecture.

SNA management services (SNA/MS). The services provided to assist in management of SNA networks.

SNA network. The part of a user-application network that conforms to the formats and protocols of Systems Network Architecture. It enables reliable transfer of data among end users and provides protocols for controlling the resources of various network configurations. The SNA network consists of network accessible units (NAUs), boundary function, gateway function, and intermediate session routing function components; and the transport network.

SNA network interconnection (SNI). The connection, by gateways, of two or more independent SNA networks to allow communication between logical units in those networks. The individual SNA networks retain their independence.

SNA/MS. SNA management services.

SNI. SNA network interconnection.

SNMP. Simple Network Management Protocol.

socket. An endpoint for communication between processes or application programs.

socket address. The address of an application program that uses the socket interface on the network. In Internet format, it consists of the IP address of the socket's host and the port number of the socket. The application program is usually not aware of the structure of the address.

socket interface. A Berkeley Software Distribution (BSD) application program interface that allows users to easily write their own communication application programs.

Software Installer for OS/2. An OS/2-based tool that is used to install workstation functions such as the NetView Graphic Monitor Facility.

SP. Service point.

spin button. A component used to display, in sequence, a ring of related but mutually exclusive choices. A user can accept the value displayed in the entry field or can type a valid choice into the entry field.

SPTN. Single-protocol transport network.

SRPI. Server-Requester Programming Interface.

SSCP. System services control point.

SSCP-dependent LU. An LU that requires assistance from a system services control point (SSCP) in order to initiate an LU-LU session. It requires an SSCP-LU session.

stream data format. Data that has no record boundaries. The data is simply a stream of bits.

subnet. In TCP/IP, a part of a network that is identified by a portion of the IP address.

subnet address. In Internet communications, an extension to the basic IP addressing scheme where a portion of the host address is interpreted as the local network address.

subnet mask. Synonym for *address mask*.

subnetwork. (1) Any group of nodes that have a set of common characteristics, such as the same network ID. (2) In the AIX operating system, one of a group of multiple logical network divisions of another network, such as can be created by the Transmission Control Protocol/Internet Protocol (TCP/IP) interface program. (3) Synonymous with *subnet*.

subnetwork mask. Synonym for *address mask*.

system services control point (SSCP). A component within a subarea network for managing the configuration, coordinating network operator and problem determination requests, and providing directory services and other session services for end users of the network. Multiple SSCPs, cooperating as peers with one another, can divide the network into domains of control, with each SSCP having a hierarchical control relationship to the physical units and logical units within its own domain.

system services control point (SSCP) domain. The system services control point, the physical units (PUs), the logical units (LUs), the links, the link stations, and all the resources that the SSCP has the ability to control by means of activation and deactivation requests.

Systems Network Architecture (SNA). The description of the logical structure, formats, protocols, and operational sequences for transmitting information units through, and controlling the configuration and operation of, networks. The layered structure of SNA allows the ultimate origins and destinations of information, that is, the end users, to be independent of and unaffected by the specific SNA network services and facilities used for information exchange.

T

TCP. Transmission Control Protocol.

TCP/IP. Transmission Control Protocol/Internet Protocol.

Telnet. In the Internet suite of protocols, a protocol that provides remote terminal connection service. It allows users of one host to log on to a remote host and interact as directly attached terminal users of that host.

TLPB. Transport-layer protocol boundary.

translation table. A table used to replace one or more characters with alternative characters; for example, to translate characters representing a virtual address to those representing a real address, characters representing an event to those representing a procedure call, characters of a national character set to those of another national language, or characters representing a relocated address to those representing an absolute address.

Transmission Control Protocol (TCP). A communications protocol used in the Internet and in any network that follows the U.S. Department of Defense standards for internetwork protocol. TCP provides a reliable host-to-host protocol between hosts in packet-switched communications networks and in interconnected systems of such networks. It uses the Internet Protocol (IP) as the underlying protocol.

Transmission Control Protocol/Internet Protocol (TCP/IP). A set of communications protocols that support peer-to-peer connectivity functions for both local and wide area networks.

transport characteristics. In MPTN architecture, the set of transport services that a transport user expects, for example, whether data will be sent using connections or datagrams, and formatted as streams or records.

transport-layer protocol boundary (TLPB). In MPTN architecture, the protocol boundary that provides access in a protocol-independent fashion to multiple transport protocols.

transport network. An implementation of transport networking. Examples are parts of SNA, TCP/IP, OSI, IPX, NetBIOS, DECnet, and Appletalk.

transport networking. In MPTN architecture, the communications services provided at the transport layer and below.

transport protocol. A specification of the rules governing the exchange of information between components of a transport network.

transport provider. In MPTN architecture, a component that provides the transport functions associated with a particular transport protocol stack.

transport-provider address. In MPTN architecture, a transport address used to identify a transport provider.

transport service access point (TSAP). In MPTN architecture, the address that each transport-layer user uses to access the services of the transport layer.

transport user. In MPTN architecture, an application program or application support element that uses transport services to convey data through a network. A program that directly requests transport services.

transport-user address. In MPTN architecture, a transport address used to identify a transport user.

TSAP. Transport service access point.

U

UDP. User Datagram Protocol.

unicast. Transmission of data to a single destination. Contrast with *broadcast* and *multicast*.

UNIX operating system. An operating system developed by Bell Laboratories that features multiprogramming in a multiuser environment. The UNIX operating system was originally developed for use on minicomputers but has been adapted for mainframes and microcomputers. The AIX operating system is IBM's implementation of the UNIX operating system.

upload. (1) To transfer programs or data from a connected device, typically a personal computer, to a computer with greater resources. (T) (2) To transfer data from a device, such as a workstation or a microcomputer, to a computer. Contrast with *download*.

User Datagram Protocol (UDP). In the Internet suite of protocols, a protocol that provides unreliable, connectionless datagram service. It enables an application program on one machine or process to send a datagram to an application program on another machine or process. UDP uses the Internet Protocol (IP) to deliver datagrams.

V

Virtual Telecommunications Access Method (VTAM). An IBM licensed program that controls communication and the flow of data in an SNA network. It provides single-domain, multiple-domain, and interconnected network capability.

VTAM. Virtual Telecommunications Access Method.

VTAM application program. A program that has opened an access method control block (ACB) to identify itself to VTAM and that can therefore issue VTAM macroinstructions.

W

well-known port. In Internet communications, one of a set of preassigned protocol port numbers that address specific functions used by transport level protocols (for example, TCP and UDP).

window. An area with visible boundaries that presents

a view of an object or with which a user conducts a dialog with a computer system.

X

X Window System. A network-transparent windowing system developed by the Massachusetts Institute of Technology (MIT). It is the basis for the Enhanced X-Windows Toolkit.

X.400. The message-handling services standard of CCITT and ISO.

Bibliography

AnyNet Product Family Publications

AIX SNA Server/6000 V2R1.1 AnyNet Feature: Guide to Sockets over SNA, SC31-8065

AIX SNA Server/6000 V2R1.1 AnyNet Feature: Guide to APPC over TCP/IP, SV40-0212

AnyNet/2: NetBEUI over SNA Version 1.0 Administrator's Guide, GV40-0402

AnyNet/2: NetBEUI over SNA Version 1.0 User's Guide, GV40-0377

IBM AnyNet/2 Version 2.0.2: Guide to SNA over TCP/IP, GV40-0375

IBM AnyNet: Guide to SNA over TCP/IP Version 1.0 for Windows, GV40-0113

IBM AnyNet/2 Version 2.0.2: Guide to Sockets over SNA, GV40-0376

IBM AnyNet/2: Guide to Sockets over SNA Gateway Version 1.1.6, GV40-0374

IBM AnyNet: Guide to APPC over TCP/IP Version 1.0 for Windows, SV40-0215

IBM AnyNet: Guide to IPX over SNA Gateway Version 1.0 for OS/2, GV40-0405

IBM AnyNet: Guide to SNA over TCP/IP Gateway Version 1.0 for OS/2, GV40-0216

IBM AnyNet/2 Version 2.0.2: Guide to Sockets over NetBIOS, SV40-0111

VTAM AnyNet Feature for V4R3 Publications

VTAM AnyNet Feature: Guide to SNA over TCP/IP, SC31-6560

VTAM AnyNet Feature: Guide to Sockets over SNA, SC31-6559

VTAM AnyNet Feature for V4R2: Guide to Sockets over SNA Gateway for OS/2, SC31-6528

Softcopy Information

IBM Networking Systems Softcopy Collection Kit CD-ROM, SK2T-6012

APPC Publications

API and Technical Reference, SC41-3531

Communications Server Publications

Up and Running!, GC31-8189

Network Administration and Subsystem Management Guide, SC31-8181

A CM/2 APPC/APPN Tutorial, GG24-2537

Guide to SNA over TCP/IP, GC31-8192

Guide to SNA over TCP/IP Gateway, GC31-8193

Guide to Sockets over SNA, GC31-8190

Guide to Sockets over SNA Gateway, GC31-8191

CPI-C Publications

Common Programming Interface Communications: CPI-C 2.0 Reference, SC26-4399

Common Programming Interface Communications: CPI-C 2.0 Specification, SC31-6180

MPTN Publications

Networking Blueprint Executive Overview, GC31-7057

Multiprotocol Transport Networking: Technical Overview, GC31-7073

Multiprotocol Transport Networking: Formats, GC31-7074

Multiprotocol Transport Networking: Pocket Guide, GC31-7072

The Networking Blueprint Pocket Guide, SX33-6090

© Copyright IBM Corp. 1996

Bibliography

NetView Publications

IBM NetView for OS/2 User's Guide, Version 2, SC31-8099-01

IBM NetView General Information, Version 2 Release 4, GC31-7098

IBM NetView: Administration Reference, Version 2 Release 4, SC31-7080

IBM NetView at a Glance, Version 2 Release 3, GC31-7016

IBM NetView: Administration Reference, Version 2 Release 3, SC31-6128

IBM NetView for AIX Concepts: A General Information Manual, GC31-6234

IBM NetView for AIX: User's Guide for Beginners, SC31-6232

IBM NetView for AIX: Administrator's Guide, SC31-7192

IBM NetView for AIX: Administrator's Reference, SC31-8104

IBM NetView for AIX and the Host Connection, SC31-6235

IBM NetView Distribution Manager/2: Change Distribution Manager User's Guide, SH19-5048

IBM NetView Distribution Manager/2: Concepts and Overview, GH19-4009

IBM NetView Distribution Manager/2: Installation and Customization, SH19-6915

IBM NetView Distribution Manager/2: Getting Started, SH19-4067

IBM NetView Distribution Manager/2: Messages and Error Recovery Guide, SH19-6924

Networking Services Publications

IBM APPC Networking Services for Windows: Getting Started, SC31-8124

IBM APPC Networking Services for Windows: Administrator's Guide, SC31-8125

IBM APPC Networking Services for Windows: Application Programmer's Reference, SC31-8126

IBM APPC Networking Services for Windows: Configuration Parameters Reference for Administrators and Application Programmers, SC31-8138

Personal Communications Publications

IBM Personal Communications: Up and Running for Windows, S20H-1770

IBM Personal Communications: Reference for Windows, S20H-1773

IBM Personal Communications: Programmer's Guide for Full-Function DOS and Windows Modes, SC18-2286

TCP/IP for DOS Publications

IBM TCP/IP Version 2.1.1 for DOS: Command Reference, SX75-0083

IBM TCP/IP Version 2.1.1 for DOS: Installation and Administration, SC31-7047

IBM TCP/IP Version 2.1.1 for DOS: Programmer's Reference, SC31-7046

IBM TCP/IP Version 2.1.1 for DOS: User's Guide, SC31-7045

Requests for Comments (RFCs)

This section describes how to obtain electronic and printed copies of Internet RFCs.

Obtaining Electronic Copies through FTP:

If you have FTP running on a workstation that is connected to the Internet, you may retrieve RFCs from the Network Information Center by using FTP to connect to ds.internic.net.

1. Use FTP to connect to host nic.ddn.mil.
2. Issue the command **user anonymous** to identify yourself to the host.
3. When prompted for a password, type **guest**.
4. Type the command **cd rfc** to change to the RFC directory.

5. Type the command **get rfcnnn.txt**, where *nnn* is the number of the RFC you are requesting.

Obtaining Electronic Copies through Electronic Mail: SERVICE@NIC.DDN.MIL is an automated service provided by the Network Information Center. It allows access to RFCs (and other documents) through ordinary electronic mail. This is especially useful for users who do not have access to the Network Information Center through a direct Internet link. Follow this procedure to obtain an RFC through electronic mail:

1. Send a mail message to SERVICE@NIC.DDN.MIL.
2. In the **Subject** field, type **RFC.nnn.**, where *nnn* is the RFC number. To obtain a list of all of the RFCs available, substitute the word **index** for *nnn*.

Large files are broken into smaller separate messages. The information you request is sent back to you as soon as possible.

Obtaining Printed Copies: Printed copies of RFCs are available for a fee from:

SRI International, Room EJ291
333 Ravenswood Avenue
Menlo Park, CA 94025
(415) 859-3695
(415) 859-6387
FAX (415) 859-6028

Bibliography

Index

Numerics

3270 session configuration 16
5250 session configuration 16

A

abinfo mapping utility 68
address mapping
  abinfo mapping utility 68
  example, how a destination LU gets an IP address 13
  mapping methods 14
  overview 13
ANY0999E message 86
AnyNet CID setup tool 24
AnyNet configuration tool 35
AnyNet messages and error codes 73—90
AnyNet software
  and MPTN architecture xi
  APPC over TCP/IP xi
  benefits of xi
  IPX over SNA Gateway xi
  NetBEUI over SNA xi
  overview xi
  SNA over TCP/IP xii, 1
  SNA over TCP/IP Gateway xii
  Sockets over IPX xii
  Sockets over NetBIOS xii
  Sockets over SNA xii
  Sockets over SNA Gateway xii
AnyNet well-known port (397) 8, 62
ANYNETPATH environment variable 22, 29
ANYNVDMD.EXE automatic install program 93
APARs and ServicePaks
  automating installation with a response file 25
  installing on a workstation or code server 30
APING program, verifying connectivity 63
APIs supported 10
APPC and CPI-C application support 1, 10
APPC Networking Services for Windows
  applications supported 1
  configuration examples 1, 2, 39
  DLCTYPE setting as MPTN 59
  initializing LU to start SNA over TCP/IP 49
  message log 64
APPC Networking Services for Windows *(continued)*
  publications 148
  selecting as the SNA transport user 35
  starting SNA over TCP/IP 49
  stopping SNA over TCP/IP 51
  system file changes during installation 29
  trace examples 99—115
  trace utility, starting 65
  troubleshooting 59
  verifying status 59
application program interfaces (APIs) supported 10
applications supported 1, 10
AUTOEXEC.ADD file 22
AUTOEXEC.BAT file
  ANYNETPATH environment variable 22, 29
  changes during installation 29
  option to update during installation 22
  SNA-over-TCP/IP environment variables 36
  TCP/IP host name definition 14
  TCP/IP HOSTS file definition 14
AUTOINST.EXE, automatic install program 26
automatic installation
  AUTOINST.EXE program 26
  creating a change file profile 26
  creating a response file 25
  keywords, parameters, and return codes 91
  overview of procedure 24
  steps 24
  using NetView Distribution Manager/2 (NVDM/2) 26

C

change file profile, creating for automatic installation 26
CID install 24
CID setup tool, using for automatic installation 24
Client Access/400 for Windows API 10
code server
  setting up 23
  using to install SNA over TCP/IP 23
configuration examples
  connecting a Windows workstation running APPC Networking Services with other workstations 2
  connecting a Windows workstation running Personal Communications to a subarea network 4, 42
  connecting a Windows workstation running Personal Communications to an APPN network 5

© Copyright IBM Corp. 1996

Index

configuration examples *(continued)*
  connecting a Windows workstation running Personal Communications to an AS/400 Host 6, 44
  connecting two Windows workstations running APPC Networking Services 1, 39
configuration tool, AnyNet 35
configuration, install, and distribution (CID) setup tool 24
configuring SNA over TCP/IP for Windows
  address mapping methods 14
  environment settings 36
  selecting a SNA transport user 35
  using the AnyNet configuration tool 35
CONN_RETRY_SECS environment variable 37
connection retry duration, environment setting 37
connection wait time limit, environment setting 37
CONNWAIT_SECS environment variable 37
CPI-C and APPC application support 1, 10

D

data link control type (DLCTYPE) 30
defining LU name mappings 14
deleting a prior version of APPC over TCP/IP for Windows 20
DLCTYPE 30
documentation for SNA over TCP/IP
  disk space requirement 9
  installing 21
  online, how to view xiii
  ordering xiii
  PostScript, how to print xiii
  printing xiii
domain name format 13, 36
domain name server 14

E

EHNAPPCNonblocking variable, NSD.INI file 30
environment settings
  ANYNETPATH 22
  connection retry duration 37
  connection wait time limit 37
  remote node inactivity poll interval 37
  SNA domain name suffix 15, 36
  unacknowledged datagram retry interval 38
  unsent datagram retry interval 38
ERROR 100 message 64, 65, 69, 86
examples
  checking TCP/IP on a local system 55
examples *(continued)*
  checking TCP/IP on a remote system 57
  configuration with a Windows workstation and other workstations on different platforms 2
  configuration with a Windows workstation running Personal Communications 4, 5, 6, 42, 44
  configuration with two Windows workstations 1, 39
  mapping of a destination LU name to an IP address 13
  trace examples
    APPC Networking Services 99—115
    Personal Communications 117—129
  tracing AnyNet data
    using APPC Networking Services 65
    using Personal Communications 70
  verifying that AnyNet is initialized correctly 62
  verifying that APPC Networking Services is active 59
  verifying that Personal Communications is active 61
  Windows node structure
    with APPC Networking Services 7
    with Personal Communications 7

F

FMTNSD.LOG trace output file, APPC Networking Services 67

G

getting help
  IBM support services 72
  messages and error code descriptions 73—90
  online help during SNA-over-TCP/IP installation 21
  online help for the AnyNet configuration tool 35
  other publications 147

H

hardcopy documentation xiii
hardware requirements 9
help
  IBM support services 72
  messages and error code descriptions 73—90
  online help during SNA-over-TCP/IP installation 21
  online help for the AnyNet configuration tool 35
  other publications 147
host command 57
HOSTNAME environment variable 14

HOSTS file 14

I

IBM APPC Networking Services for Windows
  applications supported 1
  configuration examples 1, 2, 39
  DLCTYPE setting as MPTN 59
  initializing LU to start SNA over TCP/IP 49
  message log 64
  publications 148
  selecting as the SNA transport user 35
  starting SNA over TCP/IP 49
  stopping SNA over TCP/IP 51
  system file changes during installation 29
  trace examples 99—115
  trace utility, starting 65
  troubleshooting 59
  verifying status 59
IBM Personal Communications for Windows
  accessing AnyNet configuration options 16
  configuration examples 4, 5, 42, 44
  description of emulator package support 1
  message and trace log 69
  planning considerations
    CP name, IDBLK, IDNUM, and SNASUFFIX 15
    creating a workstation profile 16
    system location and PC location names 16
  publications 148
  selecting as the SNA transport user 35
  starting SNA over TCP/IP 50
  stopping SNA over TCP/IP 51
  system file change during installation 29
  trace example 117—129
  trace utility, starting 70
  troubleshooting 61
  verifying status 61
  workstation profile, creating 16
IBM support services 72
INACTIVITY_TIMER_SECS environment variable 37
initializing the APPC Networking Services LU 49
installing APARs and ServicePaks 25, 30
installing SNA over TCP/IP for Windows
  automating installation 24
  deleting a prior version of APPC over TCP/IP for Windows 20
  files created and changed 27
  hardware requirements 9
  installing an APAR or ServicePak 25, 30
  installing on a workstation from a CD 21
installing SNA over TCP/IP for Windows *(continued)*
  installing on a workstation from a code server 21
  installing on a workstation from diskettes 21
  prerequisite information 19
  program group and programs created 29
  setting up a code server 23
  software requirements 9
  verifying initialization 62
INSTALLW.EXE automatic install program 93

L

licensing agreement ix
LU name mappings, defining 14

M

managing a network, tools 51
mapping LU names to IP addresses
  abinfo mapping utility 68
  example, how a destination LU gets an IP address 13
  mapping methods 14
  overview 13
message 5239, APPC Networking Services initialization error 50
message and trace log, Personal Communications 69
message ANY0999E 86
message ERROR 100 64, 65, 69, 86
message log, APPC Networking Services 64
mixed protocol networking xi
MPLD variable, NSD.INI file 30
MPTN architecture xi
MSGLOG.NSD, message log for APPC Networking Services 64
multiprotocol transport networking (MPTN) architecture xi

N

netstat command 56, 62
NetView Distribution Manager/2 26
NetView, using for network management 51
network interface protocols 11
network management tools 51
Networking Services for Windows
  applications supported 1
  configuration examples 1, 2, 39
  DLCTYPE setting as MPTN 59
  initializing LU to start SNA over TCP/IP 49

Index

Networking Services for Windows *(continued)*
  message log 64
  publications 148
  selecting as the SNA transport user 35
  starting SNA over TCP/IP 49
  stopping SNA over TCP/IP 51
  system file changes during installation 29
  trace examples 99—115
  trace utility, starting 65
  troubleshooting 59
  verifying status 59
NSD.INI file 30
NSTRACE trace utility, APPC Networking Services
  FMTNS.LOG output file 67
  formatting a trace 67
  NSTRACE.CFG file 30
  starting a trace 65
NSTRACE.CFG file 30
NVDM/2 26

O

OIA status messages and indicators, Personal Communications 61
online help
  AnyNet configuration tool 35
  SNA-over-TCP/IP installation 21

P

PCID, procedure correlator ID 68
PCSMON trace utility, Personal Communications
  PCSMON.TRC output file 69
  reading trace data 71
  starting a trace 70
  stopping the trace utilty 71
Personal Communications for Windows
  accessing AnyNet configuration options 16
  configuration examples 4, 5, 42, 44
  description of emulator package support 1
  message and trace log 69
  planning considerations
    CP name, IDBLK, IDNUM, and SNASUFFIX 15
    creating a workstation profile 16
    system location and PC location names 16
  publications 148
  selecting as the SNA transport user 35
  starting SNA over TCP/IP 50
  stopping SNA over TCP/IP 51
  system file change during installation 29
Personal Communications for Windows *(continued)*
  trace example 117—129
  trace utility, starting 70
  troubleshooting 61
  verifying status 61
  workstation profile, creating 16
ping command 56, 58
planning considerations
  defining host names, domain names, and IP addresses 14
  for Personal Communications 15
  hardware requirements 9
  mapping LU names to IP addresses 13
  software requirements 9
PostScript file
  installing 21
  printing xiii
printing SNA-over-TCP/IP documentation
  formatted PostScript file xiii
  softcopy xiii
problem determination
  APPC Networking Services troubleshooting aids
    abinfo mapping utility 68
    message log 64
    trace log 67
    trace utility 65
  checking product status
    APPC Networking Services 59
    Personal Communications 61
    SNA over TCP/IP 62
    TCP/IP 55
  IBM support services 72
  introduction to troubleshooting problems 53
  messages and error codes 73—90
  Personal Communications troubleshooting aids
    message and trace log 69
    trace utility 70
  step-by-step procedure 54
  summary, troubleshooting steps 54
procedure correlator ID (PCID) 68

R

README file for SNA over TCP/IP 20, 29
reconfiguring SNA over TCP/IP for Windows 35
remote installation with the CID setup tool 24
remote node inactivity poll interval, environment setting 37
removing the AnyNet product from a workstation
  APPC over TCP/IP for Windows, before installing SNA over TCP/IP 20

removing the AnyNet product from a workstation *(continued)*
SNA over TCP/IP for Windows 31
requirements
address mapping 13
hardware 9
selecting a SNA transport user 35
software 9
RESOLV file 14
response file for automatic installation
creating 25
keyword descriptions 91

S

ServicePak, installing 25, 30
ServicePaks and APARs
automating installation with a response file 25
installing on a workstation or code server 30
settings for environment variables
ANYNETPATH 22
connection retry duration 37
connection wait time limit 37
remote node inactivity poll interval 37
SNA domain name suffix 15, 36
unacknowledged datagram retry interval 38
unsent datagram retry interval 38
simple network management protocol (SNMP) 51
SNA Domain Name Suffix environment setting 15, 36
SNA over TCP/IP for Windows
applications supported 1, 7
configuring 36
documentation xiii, 9
environment variables 36
getting started 19
IBM support services 72
installing from a CD 21
installing from a code server 21
installing from diskettes 21
installing on a code server 23
installing on a workstation 21
introducing 1
managing 51
messages and codes 73—90
network management 51
removing from a code server 32
removing from a workstation 31
starting 49
stopping 51
troubleshooting problems 53
SNA over TCP/IP for Windows *(continued)*
updating on a workstation or code server 30
SNA transport users
APPC Networking Services for Windows 35
Personal Communications for Windows 35
SNASUFFIX environment variable 15, 36
SNMP, simple network management protocol 51
software requirements 9
starting SNA over TCP/IP for Windows
using APPC Networking Services 49
using Personal Communications 50
stopping SNA over TCP/IP for Windows
using APPC Networking Services 51
using Personal Communications 51
support services, IBM 72
syntax conventions used 97
SYSTEM.INI file 29

T

TCP/IP
configuring 38
HOSTS file 14
products and versions compatible with SNA over TCP/IP 9
RESOLV file 14
support services 11
TCPDOS.INI file 14
troubleshooting 55
tools for managing networks 51
trace and message log, Personal Communications 69
tracing AnyNet data
using APPC Networking Services
examples, component traces 100—104
examples, session initiation traces 104—115
formatting and viewing trace output 67
overview 65
reading trace data 67
starting the trace utility 65
using Personal Communications
example, session initiation trace 117—129
overview 70
reading trace data 71
starting the trace utility 70
stopping the trace utility 71
transport users, SNA
APPC Networking Services for Windows 35
Personal Communications for Windows 35
troubleshooting
APPC Networking Services troubleshooting aids
abinfo mapping utility 68

troubleshooting *(continued)*
APPC Networking Services troubleshooting aids *(continued)*
message log 64
trace log 67
trace utility 65
checking product status
APPC Networking Services 59
Personal Communications 61
SNA over TCP/IP 62
TCP/IP 55
IBM support services 72
introduction to troubleshooting problems 53
messages and error codes 73—90
Personal Communications troubleshooting aids
message and trace log 69
trace utility 70
step-by-step procedure 54
summary, troubleshooting steps 54

U

UNACKED_DG_RETRY_SECS environment variable 38
unacknowledged datagram retry interval, environment setting 38
unsent datagram retry interval, environment setting 38
UNSENT_DG_RETRY_SECS environment variable 38
updating SNA over TCP/IP for Windows on a workstation or code server 30

V

verifying connectivity with the APING program 63
verifying correct LU-name-to-IP-address mapping 57
verifying that APPC Networking Services is active 60
verifying that Personal Communications is active 61

W

well-known port (397), AnyNet 8, 62
Windows node, structure 7
Windows socket API 11
WINNSD.EXE program, APPC Networking Services 60
workstation profile, Personal Communications 16

Communicating Your Comments to IBM

IBM AnyNet
Guide to SNA over TCP/IP for Windows
Version 1.0

Publication No. GV40-0113-00

If you especially like or dislike anything about this book, please use one of the methods listed below to send your comments to IBM. Whichever method you choose, make sure you send your name, address, and telephone number if you would like a reply.

Feel free to comment on specific errors or omissions, accuracy, organization, subject matter, or completeness of this book. However, the comments you send should pertain to only the information in this manual and the way in which the information is presented. To request additional publications, or to ask questions or make comments about the functions of IBM products or systems, you should talk to your IBM representative or to your IBM authorized remarketer.

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you.

If you are mailing a readers' comment form (RCF) from a country other than the United States, you can give the RCF to the local IBM branch office or IBM representative for postage-paid mailing.

If you prefer to send comments by mail, use the RCF at the back of this book.

If you prefer to send comments by FAX, use this number:

United States and Canada: **1-800-227-5088**

If you prefer to send comments electronically, use this network ID:

- IBM Mail Exchange: **USIB2HPD at IBMMAIL**
- IBMLink: **CIBMORCF at RALVM13**
- Internet: **USIB2HPD@VNET.IBM.COM**

Make sure to include the following in your note:

Title and publication number of this book
Page number or topic to which your comment applies.

Help us help you!

IBM AnyNet
Guide to SNA over TCP/IP for Windows
Version 1.0

Publication No. GV40-0113-00

We hope you find this publication useful, readable and technically accurate, but only you can tell us! Your comments and suggestions will help us improve our technical publications. Please take a few minutes to let us know what you think by completing this form.

Overall, how satisfied are you with the information in this book?	Satisfied	Dissatisfied

How satisfied are you that the information in this book is:	Satisfied	Dissatisfied
Accurate		
Complete		
Easy to find		
Easy to understand		
Well organized		
Applicable to your task		

Specific Comments or Problems:

Please tell us how we can improve this book:

Thank you for your response. When you send information to IBM, you grant IBM the right to use or distribute the information without incurring any obligation to you. You of course retain the right to use the information in any way you choose.

Name

Address

Company or Organization

Phone No.

Help us help you!
GV40-0113-00

IBM®

Cut or Fold
Along Line

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

Information Development
Department CGMD
International Business Machines Corporation
PO BOX 12195
RESEARCH TRIANGLE PARK NC 27709-9990

Fold and Tape **Please do not staple** Fold and Tape

GV40-0113-00

Cut or Fold
Along Line

IBM®

Printed in the United States of America
on recycled paper containing 10%
recovered post-consumer fiber.

GV4 - 113-