

z/OS Communications Server



AnyNet SNA over TCP/IP

Version 1 Release 2

z/OS Communications Server



AnyNet SNA over TCP/IP

Version 1 Release 2

Note:

Before using this information and the product it supports, be sure to read the general information under "Appendix D. Notices" on page 93.

First Edition (October 2001)

This edition applies to Version 1 Release 2 of z/OS (5694-A01) and to all subsequent releases and modifications until otherwise indicated in new editions.

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

Address comments to:

IBM Corporation
Software Reengineering
Department G71A/ Bldg 503
Research Triangle Park, North Carolina 27709-9990
U.S.A.

If you prefer to send comments electronically, use one of the following methods:

Fax (USA and Canada):

1-800-227-5088

Internet e-mail:

usib2hpd@vnet.ibm.com

World Wide Web:

<http://www.ibm.com/servers/eserver/zseries/zos>

IBMLink:

CIBMORCF at RALVM17

IBM Mail Exchange:

tkinlaw@us.ibm.com

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

© Copyright International Business Machines Corporation 1995, 2001. All rights reserved.

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Figures	vii
Tables	ix
About This Book	xi
Artwork Used in This Book	xi
Where to Find More Information	xii
Where to Find Related Information on the Internet	xiii
Licensed Documents	xiii
LookAt, an Online Message Help Facility	xiv
How to Contact IBM® Service	xv
z/OS Communications Server Information	xv
Summary of Changes	xxiii
Chapter 1. Introducing SNA over TCP/IP	1
How SNA over TCP/IP Works	1
Software Requirements	2
TCP/IP Support Services	2
Application Program Interfaces Supported by SNA over TCP/IP	2
Attachments Supported by SNA over TCP/IP	3
Example Network Scenarios	3
Communicating from VTAM to VTAM over TCP/IP	3
Communicating from VTAM to OS/2 over TCP/IP	3
Communicating from VTAM to OS/2 through an Intermediate VTAM Node	4
Communicating from SNA Node to SNA Node through an Intermediate IP Network	5
Managing SNA over TCP/IP	6
Managing the TCP/IP Network with NetView/6000	7
Managing SNA over TCP/IP in a TCP/IP Network that Connects to an SNA Network	7
Chapter 2. Defining the IP Network to VTAM	9
TCP/IP Major Node	9
Coding Notes, References, and Terminology	10
Full Syntax	10
Operand Descriptions	15
Defining Independent and Dependent Logical Units for SNA over TCP/IP	22
Defining Independent Logical Units for SNA over TCP/IP	22
Defining Dependent Logical Units for SNA over TCP/IP	23
Sample Configurations and Required Definitions	24
VTAM to VTAM over a Single IP Network	24
VTAM to Communications Server/2 over a Single IP Network	25
VTAM as an MPTN Transport Gateway	26
VTAMs as Chained MPTN Transport Gateways	28
Dependent Logical Units Initiating Sessions	29
Considerations for Using SNA over TCP/IP with LUs in Different SNA Networks	30
Mapping LU Names to IP Addresses	30
Mapping LU Names to IP Addresses in the DNS	31
Mapping LU Names to IP Addresses in the HOSTS File	31
Configuration Requirements for OS/2-to-MVS Communications	31
Configuration on MVS	32
Configuration on OS/2	33

Optimizing SNA over TCP/IP Performance	33
Recommendations for Specifying VTAM and TCP/IP Tuning Parameters	33
Recommendations for Specifying TCP/IP Major Node Timers	33
Recommendations for Specifying TCB Values	34
Chapter 3. Initializing and Operating AnyNet SNA over TCP/IP	35
Initializing SNA over TCP/IP	35
Monitoring SNA over TCP/IP	35
DISPLAY ID	35
DISPLAY SESSIONS	36
DISPLAY STATS	36
Messages That Can Contain IP Transport-Related Text	37
Chapter 4. Diagnosing AnyNet SNA over TCP/IP	39
I/O Tracing for SNA over TCP/IP	39
VTAM Internal Tracing for SNA over TCP/IP	39
TCP/IP Module Name in VIT Records	40
Sense Codes That Contain IP-Related Information	40
Service Information	42
Appendix A. Syntax Conventions	43
How to Read a Syntax Diagram	43
Symbols and Punctuation	43
Parameters	43
Syntax Examples	44
Appendix B. Flows for SNA over TCP/IP	47
Abbreviations Used in The Flow Diagrams	47
Connection Setup: No Transport Gateway, New Session Required	48
Connection Setup: No Transport Gateway, Existing Session Available	49
Connection Setup: From Native SNA Network, Existing Session Available	50
Connection Setup: No Transport Gateway, Trying Native First, SNA Fails, IP Succeeds	51
Connection Setup: No Transport Gateway, Trying Nonnative First, IP Fails (Name Not in DNS), SNA Succeeds	53
Connection Setup: No Transport Gateway, Trying Nonnative First, DNS Not Found	55
Connection Setup: No Transport Gateway, Trying Nonnative First, No IP Connectivity	57
Connection Setup: No Transport Gateway, BIND Rejected	59
Connection Setup: From Native SNA Subarea Network, New Session Required	61
Connection Setup: From IP Network to native SNA APPN Network, New Session Required	63
Connection Setup: From IP Network, Existing Session Available	64
Send Data: Normal LU 6.2 Buffering	65
Send Data: Sending Data with Flushing	66
Send Data: Segmented Data from Native SNA	66
Send Data: Expedited Data, Datagram Copy Arrives First	67
Send Data: Expedited Data through a Transport Gateway	69
Send Data: Expedited Data, Connection Copy Acknowledgement Arrives Before Datagram Copy Sent	70
Connection Takedown: Deallocate, No Transport Gateway	71
Connection Takedown: UNBIND, No Transport Gateway, AnyNet Environments	72
Connection Takedown: UNBIND, No Transport Gateway, AnyNet/2 Environments	73
Connection Takedown: UNBIND through a Transport Gateway	74
Connection Takedown: Outage in the SNA Network	75

Connection Takedown: Deallocate through a Transport Gateway	76
Connection Takedown: Outage in an IP Network	77
DLUR-Initiated CPSVRMGR Pipe Activation	78
DLUS-Initiated CPSVRMGR Pipe Activation	81
SLU-Initiated LU-LU Session to PLU LUp	84
SLU-Initiated LU-LU Session Deactivation	86
DLUR-Initiated CPSVRMGR Pipe Deactivation	88
Appendix C. Information Apars	91
IP Information Apars	91
SNA Information Apars	92
Appendix D. Notices	93
Programming Interfaces	95
Trademarks	96
Index	99

Figures

1. Conventions Used in Network Illustrations	xii
2. Structure of an AnyNet SNA over TCP/IP Node	2
3. VTAM Application-to-VTAM Application over an IP Network	3
4. VTAM Application-to-OS/2 Application over an IP Network	4
5. VTAM Application-to-OS/2 Application Using an Intermediate VTAM Node	5
6. SNA Application-to-SNA Application through an Intermediate IP Network	6
7. VTAM Sessions over an IP Network	25
8. Sessions between VTAM and Communications Server/2 Workstations over an IP Network	26
9. Sessions Using VTAM as an MPTN Transport Gateway	27
10. Sessions Using VTAMs as MPTN Transport Gateways	28
11. Workstation to VTAM Dependent LU Communication	29
12. Dependent LU Communication through an Intermediate VTAM	30
13. Configuring SNA over TCP/IP to Support OS/2-to-MVS Communications	32
14. Connection Setup: No Transport Gateway, New Session Required	48
15. Connection Setup: No Transport Gateway, Existing Session Available	49
16. Connection Setup: From Native SNA Network, Existing Session Available	50
17. Connection Setup: No Transport Gateway, Trying Native First, SNA Fails, IP Succeeds	51
18. Connection Setup: No Transport Gateway, Trying Nonnative First, IP Fails (Name Not in DNS), SNA Succeeds	53
19. Connection Setup: No Transport Gateway, Trying Nonnative First, DNS Not Found	55
20. Connection Setup: No Transport Gateway, Trying Nonnative First, No IP Connectivity	57
21. Connection Setup: No Transport Gateway, BIND Rejected	59
22. Connection Setup: From Native SNA Subarea Network, New Session Required.	61
23. Connection Setup: From IP Network to native SNA APPN Network, New Session Required	63
24. Connection Setup: From IP Network, Existing Session Available	64
25. Send Data: Normal LU 6.2 Buffering	65
26. Send Data: Sending Data with Flushing	66
27. Send Data: Segmented Data from Native SNA	66
28. Send Data: Expedited Data, Datagram Copy Arrives First	67
29. Send Data: Expedited Data through a Transport Gateway	69
30. Send Data: Expedited Data, Connection Copy Acknowledgement Arrives Before Datagram Copy Sent	70
31. Connection Takedown: Deallocate, No Transport Gateway	71
32. Connection Takedown: UNBIND, No Transport Gateway, AnyNet Environments	72
33. Connection Takedown: UNBIND, No Transport Gateway, AnyNet/2 Environments	73
34. Connection Takedown: UNBIND through a Transport Gateway	74
35. Connection Takedown: Outage in the SNA Network	75
36. Connection Takedown: Deallocate through a Transport Gateway	76
37. Connection Takedown: Outage in an IP Network	77
38. DLUR-Initiated CPSVRMGR Pipe Activation	78
39. DLUS-Initiated CPSVRMGR Pipe Activation	81
40. SLU-Initiated LU-LU Session to PLU LUp	84
41. SLU-Initiated LU-LU Session Deactivation	86
42. DLUR-Initiated CPSVRMGR Pipe Deactivation	88

Tables

1. TCP/IP Major Node	9
2. IP Information Apars	91
3. SNA Information Apars.	92

About This Book

This publication describes the primary tasks involved with installing, configuring, using, and diagnosing AnyNet SNA over TCP/IP.

Artwork Used in This Book

Figure 1 shows the conventions used in this book to illustrate the parts of a network.

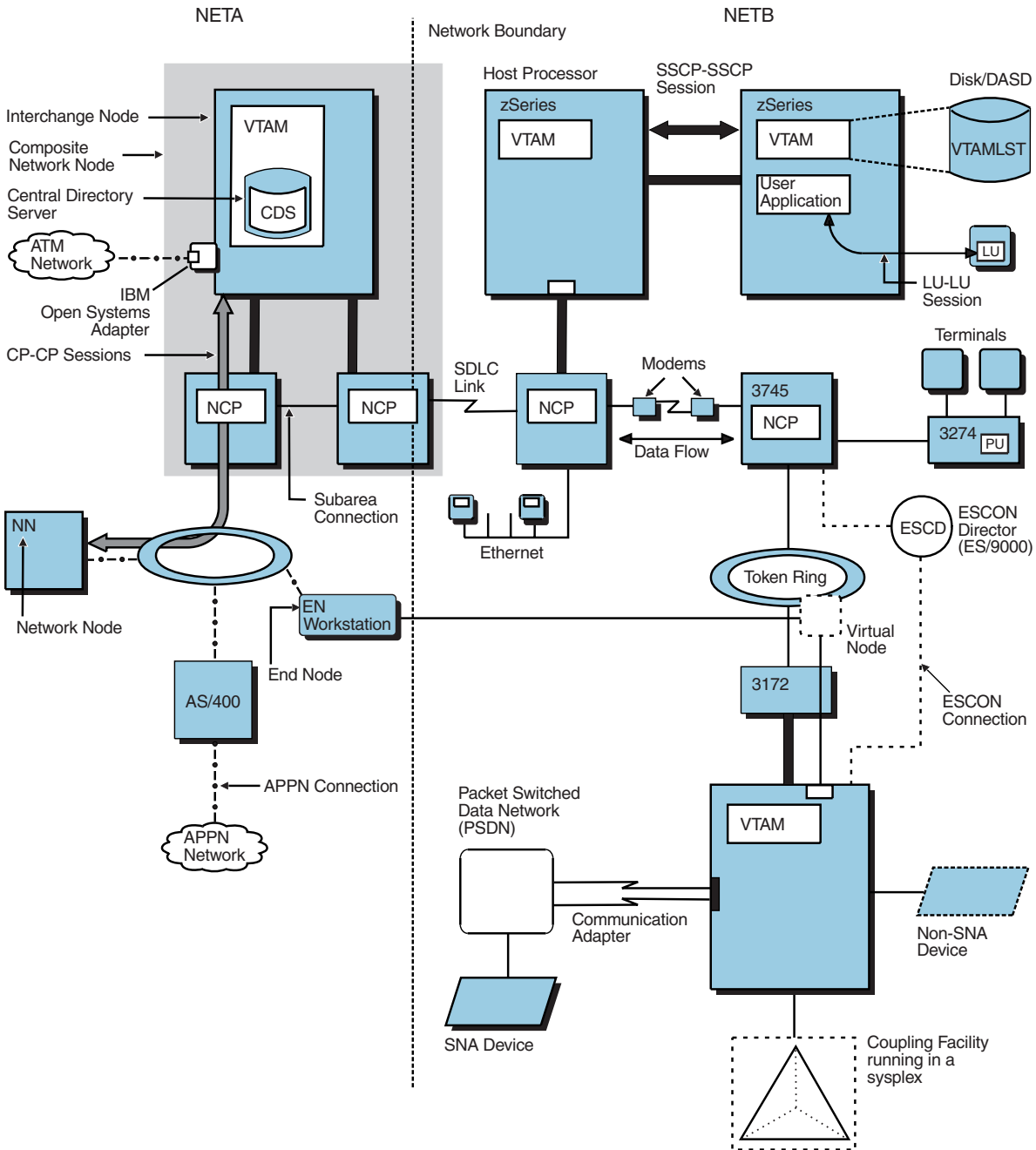


Figure 1. Conventions Used in Network Illustrations

Where to Find More Information

This section contains:

- Pointers to information available on the Internet
- Information about licensed documentation
- Information about LookAt, the online message tool

- A set of tables that describes the books in the z/OS Communications Server (z/OS CS) library, along with related publications

Where to Find Related Information on the Internet

Home Page	Web address
z/OS	http://www.ibm.com/servers/eserver/zseries/zos/
z/OS Internet Library	http://www.ibm.com/servers/eserver/zseries/zos/bkserv/
IBM Communications Server product	http://www.software.ibm.com/network/commserver/
IBM Communications Server support	http://www.software.ibm.com/network/commserver/support/
IBM Systems Center publications	http://www.redbooks.ibm.com/
IBM Systems Center flashes	http://www-1.ibm.com/support/techdocs/atmastr.nsf
VTAM and TCP/IP	http://www.software.ibm.com/network/commserver/about/csos390.html
IBM	http://www.ibm.com
RFC	http://www.ietf.org/rfc.html

Information about Web addresses can also be found in informational APAR II11334.

DNS Web Sites

For information about DNS, see the following Web sites:

USENET news groups:

comp.protocols.dns.bind

For BIND mailing lists, see:

- <http://www.isc.org/ml-archives/>
 - BIND Users
 - Subscribe by sending mail to bind-users-request@isc.org
 - Submit questions or answers to this forum by sending mail to bind-users@isc.org
 - BIND 9 Users (Note: This list may not be maintained indefinitely.)
 - Subscribe by sending mail to bind9-users-request@isc.org
 - Submit questions or answers to this forum by sending mail to bind9-users@isc.org

For definitions of the terms and abbreviations used in this book, you can view or download the latest *IBM Glossary of Computing Terms* at the following Web address:

<http://www.ibm.com/ibm/terminology>

Note: Any pointers in this publication to Web sites are provided for convenience only and do not in any manner serve as an endorsement of these Web sites.

Licensed Documents

z/OS Communications Server licensed documentation in PDF format is available on the Internet at the IBM Resource Link Web site at <http://www.ibm.com/servers/resourcelink>. Licensed books are available only to customers with a z/OS Communications Server license. Access to these books requires an IBM Resource Link Web user ID and password, and a key code. With

your z/OS Communications Server order, you received a memo that includes this key code. To obtain your IBM Resource Link Web user ID and password, log on to <http://www.ibm.com/servers/resourcelink>. To register for access to the z/OS licensed books perform the following steps:

1. Log on to Resource Link using your Resource Link user ID and password.
2. Click on **User Profiles** located on the left-hand navigation bar.
3. Click on **Access Profile**.
4. Click on **Request Access to Licensed books**.
5. Supply your key code where requested and click on the **Submit** button.

If you supplied the correct key code, you will receive confirmation that your request is being processed. After your request is processed, you will receive an e-mail confirmation.

You cannot access the z/OS licensed books unless you have registered for access to them and received an e-mail confirmation informing you that your request has been processed. To access the licensed books:

1. Log on to Resource Link using your Resource Link user ID and password.
2. Click on **Library**.
3. Click on **zSeries**.
4. Click on **Software**.
5. Click on **z/OS Communications Server**.
6. Access the licensed book by selecting the appropriate element.

LookAt, an Online Message Help Facility

LookAt is an online facility that allows you to look up explanations for z/OS CS messages and system abends.

Using LookAt to find information is faster than a conventional search because LookAt goes directly to the explanation.

LookAt can be accessed from the Internet or from a TSO command line.

To use LookAt as a TSO command, LookAt must be installed on your host system. You can obtain the LookAt code for TSO from the LookAt Web site by clicking on **News and Help** or from the z/OS V1R2 Collection, SK3T-4269.

To find a message explanation from a TSO command line, simply enter **lookat+message ID**, as in the following example:

```
lookat ezz8477i
```

This results in direct access to the message explanation for message EZZ8477I.

You can use LookAt on the Internet at the following Web site:
www.ibm.com/servers/eserver/zseries/zos/bkserv/lookat/lookat.html

To find a message explanation from the LookAt Web site, simply enter the message ID. You can select the release, if applicable.

How to Contact IBM® Service

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

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

z/OS Communications Server Information

This section contains descriptions of the books in the z/OS Communications Server library.

z/OS Communications Server publications are available:

- Online at the z/OS Internet Library web page at <http://www.ibm.com/servers/eserver/zseries/zos/>
- In hardcopy and softcopy
- In softcopy only

Softcopy Information

Softcopy publications are available in the following collections:

Titles	Order Number	Description
<i>z/OS V1R2 Collection</i>	SK3T-4269	This is the CD collection shipped with the z/OS product. It includes the libraries for z/OS V1R2, in both BookManager and PDF formats.
<i>z/OS Software Products Collection</i>	SK3T-4270	This CD includes, in both BookManager and PDF formats, the libraries of z/OS software products that run on z/OS but are not elements and features, as well as the <i>Getting Started with Parallel Sysplex</i> bookshelf.
<i>z/OS V1R2 and Software Products DVD Collection</i>	SK3T-4271	This collection includes the libraries of z/OS (the element and feature libraries) and the libraries for z/OS software products in both BookManager and PDF format. This collection combines SK3T-4269 and SK3T-4270.
<i>z/OS Licensed Product Library</i>	SK3T-4307	This CD includes the licensed books in both BookManager and PDF format.
<i>System Center Publication IBM S/390 Redbooks Collection</i>	SK2T-2177	This collection contains over 300 ITSO redbooks that apply to the S/390 platform and to host networking arranged into subject bookshelves.

z/OS Communications Server Library

The following abbreviations follow each order number in the tables below.

HC/SC — Both hardcopy and softcopy are available.

SC — Only softcopy is available. These books are available on the CD Rom accompanying z/OS (SK3T-4269 or SK3T-4307). Unlicensed books can be viewed at the z/OS Internet library site.

Updates to books are available on RETAIN and in the document called *OS/390 DOC APARs and ++HOLD DOC data* which can be found at

http://www.s390.ibm.com/os390/bkserv/new_tech_info.html. See “Appendix C. Information Apars” on page 91 for a list of the books and the informational apars (info apars) associated with them.

Planning and Migration:

Title	Number	Format	Description
<i>z/OS Communications Server: SNA Migration</i>	GC31-8774	HC/SC	This book is intended to help you plan for SNA, whether you are migrating from a previous version or installing SNA for the first time. This book also identifies the optional and required modifications needed to enable you to use the enhanced functions provided with SNA.
<i>z/OS Communications Server: IP Migration</i>	GC31-8773	HC/SC	This book is intended to help you plan for TCP/IP Services, whether you are migrating from a previous version or installing IP for the first time. This book also identifies the optional and required modifications needed to enable you to use the enhanced functions provided with TCP/IP Services.

Resource Definition, Configuration, and Tuning:

Title	Number	Format	Description
<i>z/OS Communications Server: IP Configuration Guide</i>	SC31-8775	HC/SC	This book describes the major concepts involved in understanding and configuring an IP network. Familiarity with the z/OS operating system, IP protocols, z/OS UNIX System Services, and IBM Time Sharing Option (TSO) is recommended. Use this book in conjunction with the <i>z/OS Communications Server: IP Configuration Reference</i> .
<i>z/OS Communications Server: IP Configuration Reference</i>	SC31-8776	HC/SC	This book presents information for people who want to administer and maintain IP. Use this book in conjunction with the <i>z/OS Communications Server: IP Configuration Guide</i> . The information in this book includes: <ul style="list-style-type: none"> • TCP/IP configuration data sets • Configuration statements • Translation tables • SMF records • Protocol number and port assignments
<i>z/OS Communications Server: SNA Network Implementation Guide</i>	SC31-8777	HC/SC	This book presents the major concepts involved in implementing an SNA network. Use this book in conjunction with the <i>z/OS Communications Server: SNA Resource Definition Reference</i> .
<i>z/OS Communications Server: SNA Resource Definition Reference</i>	SC31-8778	HC/SC	This book describes each SNA definition statement, start option, and macroinstruction for user tables. It also describes NCP definition statements that affect SNA. Use this book in conjunction with the <i>z/OS Communications Server: SNA Network Implementation Guide</i> .
<i>z/OS Communications Server: SNA Resource Definition Samples</i>	SC31-8836	SC	This book contains sample definitions to help you implement SNA functions in your networks, and includes sample major node definitions.
<i>z/OS Communications Server: AnyNet SNA over TCP/IP</i>	SC31-8832	SC	This guide provides information to help you install, configure, use, and diagnose SNA over TCP/IP.

Title	Number	Format	Description
<i>z/OS Communications Server AnyNet Sockets over SNA</i>	SC31-8831	SC	This guide provides information to help you install, configure, use, and diagnose sockets over SNA. It also provides information to help you prepare application programs to use sockets over SNA.

Operation:

Title	Number	Format	Description
<i>z/OS Communications Server: IP User's Guide and Commands</i>	SC31-8780	HC/SC	This book describes how to use TCP/IP applications. It contains requests that allow a user to: log on to a remote host using Telnet, transfer data sets using FTP, send and receive electronic mail, print on remote printers, and authenticate network users.
<i>z/OS Communications Server: IP System Administrator's Commands</i>	SC31-8781	HC/SC	This book describes the functions and commands helpful in configuring or monitoring your system. It contains system administrator's commands, such as NETSTAT, PING, TRACERTE and their UNIX counterparts. It also includes TSO and MVS commands commonly used during the IP configuration process.
<i>z/OS Communications Server: SNA Operation</i>	SC31-8779	HC/SC	This book serves as a reference for programmers and operators requiring detailed information about specific operator commands.
<i>z/OS Communications Server: Operations Quick Reference</i>	SX75-0124	HC/SC	This book contains essential information about SNA and IP commands.

Customization:

Title	Number	Format	Description
<i>z/OS Communications Server: SNA Customization</i>	LY43-0092	SC	This book enables you to customize SNA, and includes the following: <ul style="list-style-type: none"> • Communication network management (CNM) routing table • Logon-interpret routine requirements • Logon manager installation-wide exit routine for the CLU search exit • TSO/SNA installation-wide exit routines • SNA installation-wide exit routines
<i>z/OS Communications Server: IP Network Print Facility</i>	SC31-8833	SC	This book is for system programmers and network administrators who need to prepare their network to route SNA, JES2, or JES3 printer output to remote printers using TCP/IP Services.

Writing Application Programs:

Title	Number	Format	Description
<i>z/OS Communications Server: IP Application Programming Interface Guide</i>	SC31-8788	SC	This book describes the syntax and semantics of program source code necessary to write your own application programming interface (API) into TCP/IP. You can use this interface as the communication base for writing your own client or server application. You can also use this book to adapt your existing applications to communicate with each other using sockets over TCP/IP.
<i>z/OS Communications Server: IP CICS Sockets Guide</i>	SC31-8807	SC	This book is for people who want to set up, write application programs for, and diagnose problems with the socket interface for CICS using z/OS TCP/IP.
<i>z/OS Communications Server: IP IMS Sockets Guide</i>	SC31-8830	SC	This book is for programmers who want application programs that use the IMS TCP/IP application development services provided by IBM's TCP/IP Services.
<i>z/OS Communications Server: IP Programmer's Reference</i>	SC31-8787	SC	This book describes the syntax and semantics of a set of high-level application functions that you can use to program your own applications in a TCP/IP environment. These functions provide support for application facilities, such as user authentication, distributed databases, distributed processing, network management, and device sharing. Familiarity with the z/OS operating system, TCP/IP protocols, and IBM Time Sharing Option (TSO) is recommended.
<i>z/OS Communications Server: SNA Programming</i>	SC31-8829	SC	This book describes how to use SNA macroinstructions to send data to and receive data from (1) a terminal in either the same or a different domain, or (2) another application program in either the same or a different domain.
<i>z/OS Communications Server: SNA Programmers LU 6.2 Guide</i>	SC31-8811	SC	This book describes how to use the SNA LU 6.2 application programming interface for host application programs. This book applies to programs that use only LU 6.2 sessions or that use LU 6.2 sessions along with other session types. (Only LU 6.2 sessions are covered in this book.)
<i>z/OS Communications Server: SNA Programmers LU 6.2 Reference</i>	SC31-8810	SC	This book provides reference material for the SNA LU 6.2 programming interface for host application programs.
<i>z/OS Communications Server: CSM Guide</i>	SC31-8808	SC	This book describes how applications use the communications storage manager.
<i>z/OS Communications Server: CMIP Services and Topology Agent Guide</i>	SC31-8828	SC	This book describes the Common Management Information Protocol (CMIP) programming interface for application programmers to use in coding CMIP application programs. The book provides guide and reference information about CMIP services and the SNA topology agent.

Diagnosis:

Title	Number	Format	Description
<i>z/OS Communications Server: IP Diagnosis</i>	GC31-8782	HC/SC	This book explains how to diagnose TCP/IP problems and how to determine whether a specific problem is in the TCP/IP product code. It explains how to gather information for and describe problems to the IBM Software Support Center.
<i>z/OS Communications Server: SNA Diagnosis V1 Techniques and Procedures</i> and <i>z/OS Communications Server: SNA Diagnosis V2 FFST Dumps and the VIT</i>	LY43-0088 LY43-0089	HC/SC	These books help you identify an SNA problem, classify it, and collect information about it before you call the IBM Support Center. The information collected includes traces, dumps, and other problem documentation.
<i>z/OS Communications Server: SNA Data Areas Volume 1</i> and <i>z/OS Communications Server: SNA Data Areas Volume 2</i>	LY43-0090 LY43-0091	SC	These books describe SNA data areas and can be used to read an SNA dump. They are intended for IBM programming service representatives and customer personnel who are diagnosing problems with SNA.

Messages and Codes:

Title	Number	Format	Description
<i>z/OS Communications Server: SNA Messages</i>	SC31-8790	HC/SC	This book describes the ELM, IKT, IST, ISU, IUT, IVT, and USS messages. Other information in this book includes: <ul style="list-style-type: none"> • Command and RU types in SNA messages • Node and ID types in SNA messages • Supplemental message-related information
<i>z/OS Communications Server: IP Messages Volume 1 (EZA)</i>	SC31-8783	HC/SC	This volume contains TCP/IP messages beginning with EZA.
<i>z/OS Communications Server: IP Messages Volume 2 (EZB)</i>	SC31-8784	HC/SC	This volume contains TCP/IP messages beginning with EZB.
<i>z/OS Communications Server: IP Messages Volume 3 (EZY)</i>	SC31-8785	HC/SC	This volume contains TCP/IP messages beginning with EZY.
<i>z/OS Communications Server: IP Messages Volume 4 (EZZ-SNM)</i>	SC31-8786	HC/SC	This volume contains TCP/IP messages beginning with EZZ and SNM.
<i>z/OS Communications Server: IP and SNA Codes</i>	SC31-8791	HC/SC	This book describes codes and other information that appear in z/OS Communications Server messages.

APPC Application Suite:

Title	Number	Format	Description
<i>z/OS Communications Server: APPC Application Suite User's Guide</i>	GC31-8809	SC	This book documents the end-user interface (concepts, commands, and messages) for the AFTP, ANAME, and APING facilities of the APPC application suite. Although its primary audience is the end user, administrators and application programmers may also find it useful.

Title	Number	Format	Description
<i>z/OS Communications Server APPC Application Suite Administration</i>	SC31-8835	SC	This book contains the information that administrators need to configure the APPC application suite and to manage the APING, ANAME, AFTP, and A3270 servers.
<i>z/OS Communications Server: APPC Application Suite Programming</i>	SC31-8834	SC	This book provides the information application programmers need to add the functions of the AFTP and ANAME APIs to their application programs.

Redbooks

The following Redbooks may help you as you implement z/OS Communications Server.

Title	Number
<i>TCP/IP Tutorial and Technical Overview</i>	GG24-3376
<i>SNA and TCP/IP Integration</i>	SG24-5291
<i>IBM Communication Server for OS/390 V2R10 TCP/IP Implementation Guide: Volume 1: Configuration and Routing</i>	SG24-5227
<i>IBM Communication Server for OS/390 V2R10 TCP/IP Implementation Guide: Volume 2: UNIX Applications</i>	SG24-5228
<i>IBM Communication Server for OS/390 V2R10 TCP/IP Implementation Guide: Volume 3: MVS Applications</i>	SG24-5229
<i>OS/390 SecureWay Communication Server V2R8 TCP/IP Guide to Enhancements</i>	SG24-5631
<i>TCP/IP in a Sysplex</i>	SG24-5235
<i>Managing OS/390 TCP/IP with SNMP</i>	SG24-5866
<i>Security in OS/390-based TCP/IP Networks</i>	SG24-5383
<i>IP Network Design Guide</i>	SG24-2580

Related Information

For information about z/OS products, refer to *z/OS Information Roadmap* (SA22-7500). The Roadmap describes what level of documents are supplied with each release of z/OS Communications Server, as well as describing each z/OS publication.

The table below lists books that may be helpful to readers.

Title	Number
<i>z/OS SecureWay Security Server Firewall Technologies</i>	SC24-5922
<i>S/390: OSA-Express Customer's Guide and Reference</i>	SA22-7403
<i>z/OS MVS Diagnosis: Procedures</i>	GA22-7587
<i>z/OS MVS Diagnosis: Reference</i>	GA22-7588
<i>z/OS MVS Diagnosis: Tools and Service Aids</i>	GA22-7589

Determining If a Publication Is Current

As needed, IBM updates its publications with new and changed information. For a given publication, updates to the hardcopy and associated BookManager softcopy are usually available at the same time. Sometimes, however, the updates to hardcopy and softcopy are available at different times. Here is how to determine if you are looking at the most current copy of a publication:

1. At the end of a publication's order number there is a dash followed by two digits, often referred to as the dash level. A publication with a higher dash level is more current than one with a lower dash level. For example, in the publication order number GC28-1747-07, the dash level 07 means that the publication is more current than previous levels, such as 05 or 04.
2. If a hardcopy publication and a softcopy publication have the same dash level, it is possible that the softcopy publication is more current than the hardcopy publication. Check the dates shown in the Summary of Changes. The softcopy publication might have a more recently dated Summary of Changes than the hardcopy publication.
3. To compare softcopy publications, you can check the last two characters of the publication's filename (also called the book name). The higher the number, the more recent the publication. Also, next to the publication titles in the CD-ROM booklet and the readme files, there is an asterisk (*) that indicates whether a publication is new or changed.

Summary of Changes

| **Summary of Changes**
| **for SC31-8832-00**
| **z/OS Version 1 Release 2**

| This book contains information previously presented in *OS/390 V2R5 eNetwork*
| *Communications Server: AnyNet SNA over TCP/IP*, SC31-8578.

| **New Information**

- | • A new operand, TCPIPJOB, has been added for the TCP/IP major node. See
| “TCPIPJOB” on page 21.
- | • A new software requirement was added. See “Software Requirements” on
| page 2.
- | • A new item has been added to the introductory section of Chapter 2 regarding
| the LE/370 run-time library and LU name to IP address translation. See
| “Chapter 2. Defining the IP Network to VTAM” on page 9.

| **Changed Information**

- | • The limit on the number of sessions an MVS subtask can handle has been
| removed.
- | • The recommendations for specifying TCB values have been simplified. See
| “Recommendations for Specifying TCB Values” on page 34.

| This book contains terminology, maintenance, and editorial changes. Technical
| changes or additions to the text and illustrations are indicated by a vertical line to
| the left of the change.

Chapter 1. Introducing SNA over TCP/IP

Using SNA over TCP/IP, an application program designed to use SNA protocols can now use the TCP/IP transport to communicate with another SNA application program.

This chapter describes:

- How SNA over TCP/IP works
- System software required to use SNA over TCP/IP
- Hardware requirements
- Support services provided by z/OS Communications Server and used by SNA over TCP/IP
- Application program interfaces (APIs) supported by SNA over TCP/IP
- Attachments supported by SNA over TCP/IP
- How applications in various network configurations can use SNA over TCP/IP to communicate over an IP network

The term “TCP/IP” is used in this book to refer to the transport layers (TCP or UDP architectural layers) of the TCP/IP suite of communication protocols.

How SNA over TCP/IP Works

SNA over TCP/IP enables applications using SNA protocols to send and receive information over an IP network using the MPTN architecture. Application programs, such as Customer Information Control System (CICS[®]) and Information Management System (IMS), can access an IP network without modification.

SNA over TCP/IP uses protocols that bypass the lower transport layers of the SNA architecture (see Figure 2 on page 2). Instead of encapsulating an entire SNA path information unit (PIU) within a TCP transmission frame, SNA over TCP/IP builds a basic information unit (BIU) that TCP can accept from SNA over TCP/IP when sending and receiving data.

The process of building the unique transmission frame is transparent to an SNA application program. Using a supported API, an application program passes data to the presentation services of VTAM (z/OS) or Communications Manager/2 (CM/2). The data, in turn, is passed through the SNA architectural layers and presented to SNA over TCP/IP instead of the usual SNA path control. When VTAM or CM/2 initiates a session for an application program, SNA over TCP/IP translates the SNA routing information (fully qualified network name) into IP routing information (IP address) and uses the IP address to create a TCP connection to the appropriate system.

The configuration information you define enables SNA over TCP/IP to determine:

- Whether to route the data using SNA transport or to route the data using TCP/IP transport
- The IP address associated with the fully qualified LU name

A TCP or UDP port designated for the exclusive use of SNA over TCP/IP enables TCP/IP to determine whether to route the transmission frame it receives to a TCP/IP application or to SNA over TCP/IP. For more information on the MPTN architecture, refer to the MPTN publications *Multiprotocol Transport Networking: Formats* and *Multiprotocol Transport Networking: Technical Overview*.

Figure 2 illustrates the structure of an AnyNet SNA over TCP/IP node.

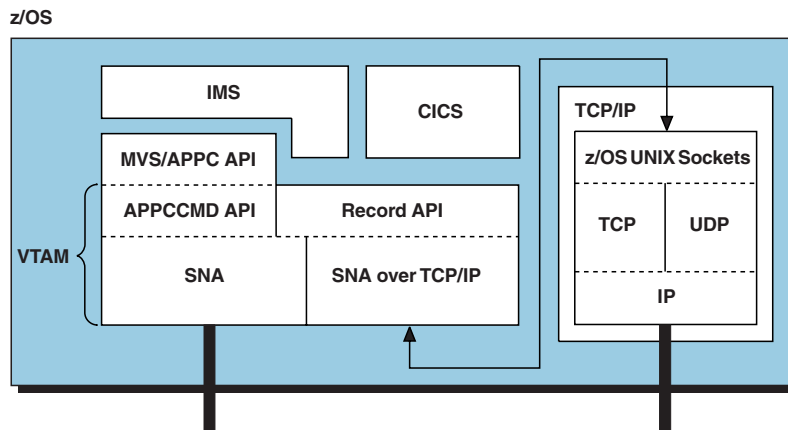


Figure 2. Structure of an AnyNet SNA over TCP/IP Node

Software Requirements

SNA partners and AnyNet partners not configured with SNA over TCP/IP that communicate through intermediate nodes that have SNA over TCP/IP must have a currently supported release of VTAM or a level of SNA software capable of communicating with the currently supported release of VTAM. See Figure 6 on page 6 for an example of a network scenario using SNA-over-TCP/IP intermediate nodes to connect nodes that do not have SNA over TCP/IP.

You will need to add SCEERUN to the VTAM startup procedure to use ANYNET SNA over TCP/IP.

AnyNet uses UNIX System Services and the VTAM started task must be registered as a UNIX System Services user using UID security.

TCP/IP Support Services

SNA over TCP/IP uses the following support services provided by z/OS Communications Server:

- z/OS UNIX socket interface

Refer to *z/OS UNIX System Services Programming: Assembler Callable Services Reference*, *z/OS Communications Server: IP Migration*, and *z/OS Communications Server: IP User's Guide and Commands* for information on z/OS UNIX.

- Socket API (for C language programs)
- TCP/IP protocol support
- Domain name server (DNS) resolver function or HOSTS file.

Application Program Interfaces Supported by SNA over TCP/IP

SNA over TCP/IP supports application programs and subsystems that use the following application program interfaces (APIs):

- VTAM APPCCMD
- VTAM record application program interface (RAPI)

- MVS/APPC including CPI-C (for example, IMS uses MVS/APPC and CPI-C). These APIs use LU 6.2 for networking support.

Attachments Supported by SNA over TCP/IP

Because AnyNet SNA over TCP/IP uses z/OS Communications Server, all attachments supported by this level of IBM TCP/IP are supported by SNA over TCP/IP. Refer to *z/OS Communications Server: IP Configuration Reference* to determine which attachments are supported.

Example Network Scenarios

This section illustrates some SNA and TCP/IP network configurations that would enable communication among SNA programs using IP transport. For information on the VTAM definition requirements for specific VTAM-to-TCP/IP configurations, see “Sample Configurations and Required Definitions” on page 24.

You can communicate with dependent logical units in any of the following host-to-workstation configurations provided that the host is defined as a dependent LU server, and the workstation is defined as a dependent LU requestor. Refer to “Defining Independent and Dependent Logical Units for SNA over TCP/IP” on page 22 for more information.

Communicating from VTAM to VTAM over TCP/IP

Figure 3 illustrates a simple IP network where VTAM Application A communicates with VTAM Application B over an IP network. VTAM and TCP/IP for z/OS must be installed on Node A and Node B. All LU protocols are supported in this network configuration if all sessions are initiated by the primary LU.

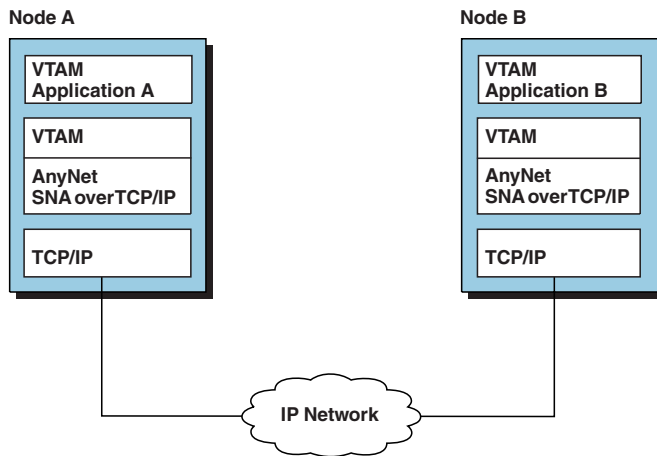


Figure 3. VTAM Application-to-VTAM Application over an IP Network

Communicating from VTAM to OS/2 over TCP/IP

Figure 4 on page 4 illustrates a simple IP network where VTAM Application A communicates with OS/2 SNA Application 1. VTAM and TCP/IP for MVS must be installed on Node A. Communications Server/2 must be installed on Node B.

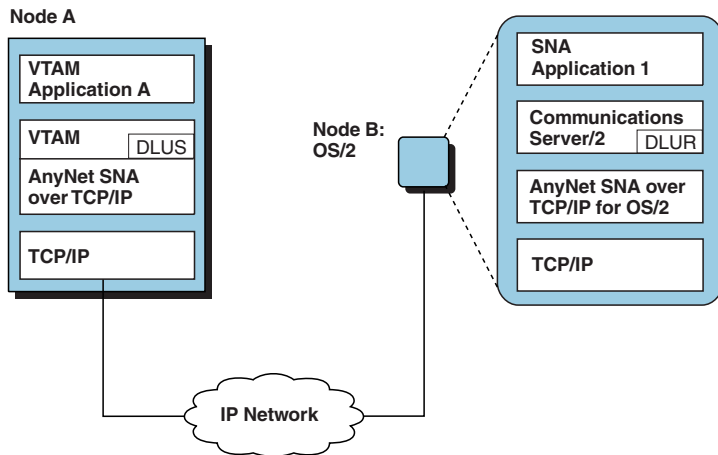


Figure 4. VTAM Application-to-OS/2 Application over an IP Network

Communicating from VTAM to OS/2 through an Intermediate VTAM Node

Figure 5 on page 5 illustrates how VTAM Application A, using SNA transport, can communicate with Dependent LU 1. Node A *does not* require TCP/IP Services.

VTAM and TCP/IP Services must be installed on Node C. Since SNA Application 1 is a non-LU 6.2 application program, Node C must be defined as a dependent LU server and Node B must be a dependent LU requestor. Communication Server/2 must be installed on Node B.

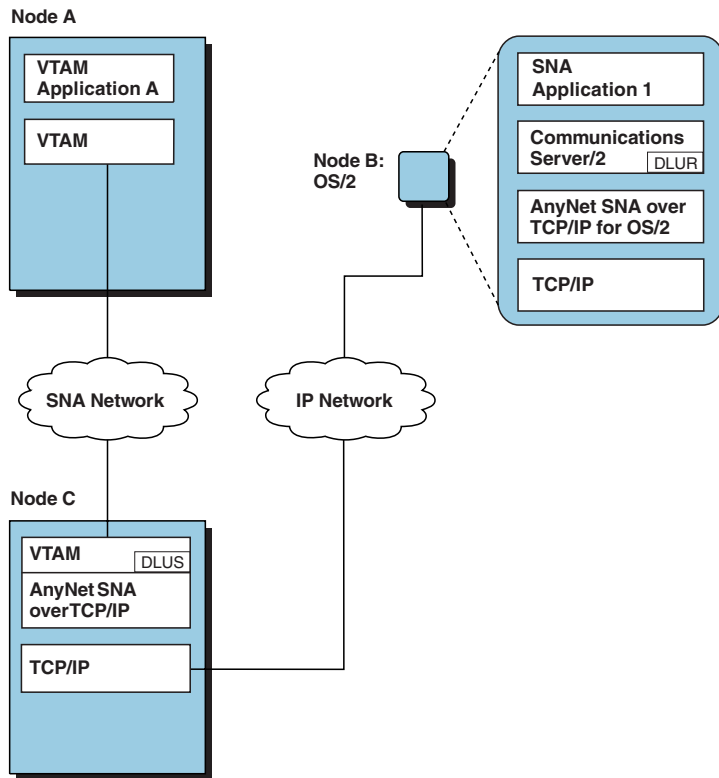


Figure 5. VTAM Application-to-OS/2 Application Using an Intermediate VTAM Node

Communicating from SNA Node to SNA Node through an Intermediate IP Network

Figure 6 on page 6 illustrates how SNA Application 1, using SNA transport, can communicate with SNA Application 2 through intermediate nodes C and D. Nodes A and B can be configured with any operating system that supports SNA, for example, VM/ESA[®] or OS/400[®]. VTAM and TCP/IP Services must be installed on Node C and Node D.

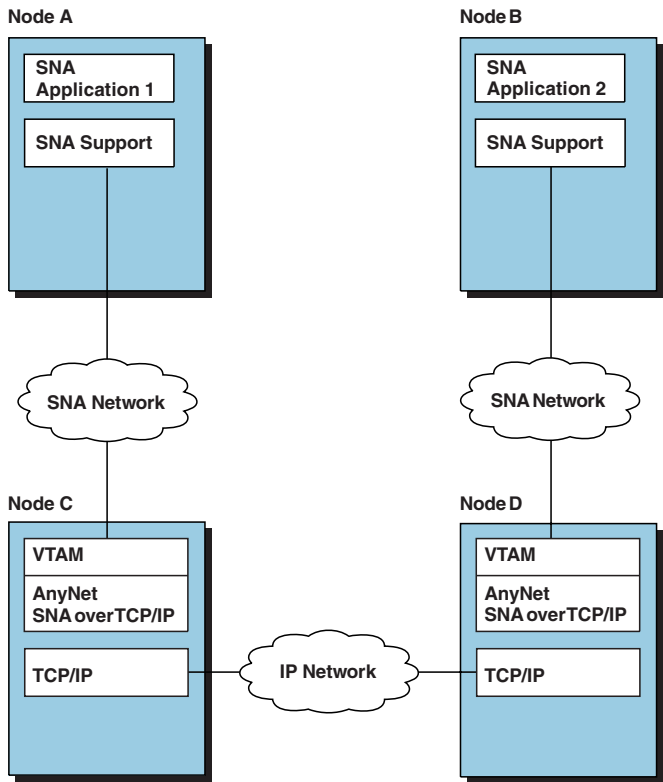


Figure 6. SNA Application-to-SNA Application through an Intermediate IP Network

Managing SNA over TCP/IP

This section describes the tools you can use to manage network communication for MVS hosts.

The SNA-over-TCP/IP support allows IP devices to be addressed using LU names. In the domain name server (DNS) database or in the local HOSTS file, a fully qualified LU name is added to a user-specified IP subdomain name to form the full IP domain name. The IP name is mapped to an IP address used to route network data. Because the IP domain name contains an LU name, TCP/IP commands can now work with LU names defined in the DNS file to test network connections.

See the *z/OS Communications Server: IP User's Guide and Commands* for information on TCP/IP commands.

You can define a host as a central site for receiving SNA over TCP/IP alerts on problems in your network. You can define a workstation that performs a key support role in your network, for example, a network server or the workstation of someone responsible for monitoring network status, and define it as a critical workstation. If a workstation is defined as critical, then it sends both conditional and unconditional alerts.

Because SNA over TCP/IP operates as a TCP/IP application that uses TCP/IP transport, you can use NetView® Version 2 Release 1 (or later), NetView/6000 Version 1 (or later), and the NetView family of products to manage SNA-over-TCP/IP network activity. The underlying TCP/IP network is managed by NetView/6000, which is a service point to NetView.

Managing the TCP/IP Network with NetView/6000

NetView/6000 provides the following TCP/IP management:

- Generic support for Management Information Base 2 (MIB2) devices
- Support for fault and performance applications
- Support for Trouble Ticket/6000* applications
- Support for TCP/IP routers and SNMP devices.

Managing SNA over TCP/IP in a TCP/IP Network that Connects to an SNA Network

In an interconnected networking environment where SNA and IP networks use SNA over TCP/IP on z/OS, NetView and NetView/6000 work together to do the following:

- Manage SNA using NetView
- Manage TCP/IP using NetView/6000
- Filter SNMP traps
- Convert SNMP traps to NetView alerts
- Customize alerts at NetView/6000
- Use RUN commands from NetView to NetView/6000 for any SNMP and TCP/IP commands
- Access SNMP commands through an z/OS TCP/IP server when IBM TCP/IP Services is installed
- Access z/OS TCP/IP commands through TSO.

Chapter 2. Defining the IP Network to VTAM

This chapter explains what you have to do to configure AnyNet SNA over TCP/IP. The following are the two major configuration tasks:

- Define the following to the VTAM program:

TCP/IP major node

Use the TCP/IP major node to define the IP network to VTAM. See “TCP/IP Major Node” for details.

CDRSC

Use the CDRSC definition statement to define remote independent logical unit (LU) names. See “Defining Independent Logical Units for SNA over TCP/IP” on page 22 for details.

Switched major node

Use the switched major node to define the dependent LU names. See “Defining Dependent Logical Units for SNA over TCP/IP” on page 23 for details.

- For each LU that uses the IP network, an IP address is used to establish a TCP connection for LU-LU sessions involving that LU. Each IP address can have many associated LUs; each LU can have many associated IP addresses. See “Mapping LU Names to IP Addresses” on page 30 to determine how one or more IP addresses are specified for an LU.

AnyNet uses the LE/370 environment to perform the LU name to IP address translation. The SCEERUN (LE/370 run time library) must be added to the LINKLST(LNKLST xx) or be loaded during VTAM startup using a STEPLIB DD card.

TCP/IP Major Node

The TCP/IP major node defines the TCP/IP network to VTAM. You can activate multiple TCP/IP major nodes; however only one PU and one LINE within those TCP/IP major nodes can be active at one time. The number of TCP/IP major nodes you need to define depends on the configuration of your network.

Table 1. TCP/IP Major Node

Name	Definition Statement	Operands	Page	Req	Default
name	VBUILD	TYPE=TCP	22	R	
		CONTIMER	15		30
		DGTIMER	15		30
		DNSUFFIX	16		sna.ibm.com
		EXTIMER	17		3
		IATIMER	18		120
		PORT	20		397
		TCB	21		10
name	GROUP	ISTATUS	18		ACTIVE
		SPAN	22		
name	LINE	ISTATUS	18		ACTIVE
		SPAN	22		
name	PU	ISTATUS	18		ACTIVE
		NETID	20		
		SPAN	22		

Coding Notes, References, and Terminology

You must define a TCP/IP major node:

- As a single, nonswitched major node
- With no switched LINES
- With no LUs

To define a TCP/IP major node to VTAM, code at least one of each of the following definition statements:

- VBUILD
- GROUP
- LINE
- PU

Notes:

1. You can define more than one GROUP, LINE, and PU in a single TCP/IP major node; however, only one LINE and one PU can be active at one time.
2. More than one TCP/IP major node can be active; however, only one LINE and one PU among all the TCP/IP major nodes can be active at one time.

See “Sample Configurations and Required Definitions” on page 24 for sample network configurations and corresponding definition requirements.

VBUILD

The VBUILD definition statement is required to define a TCP/IP major node.

GROUP

The GROUP definition statement defines the group for the line in the TCP/IP major node.

LINE

A LINE definition statement defines the line for the TCP/IP major node. You can define more than one line; however, only one line can be active at one time.

PU

The PU definition statement defines the physical unit for a TCP/IP major node line. You can define more than one physical unit; however, only one PU can be active at one time. You must define the TCP/IP PU as type 2.1; however, it cannot be an APPN[®] PU.

Full Syntax

The full syntax for this major node follows. The syntax for each operand is repeated in “Operand Descriptions” on page 15. Refer to “Appendix A. Syntax Conventions” on page 43 for a description of the syntax conventions used in this book.

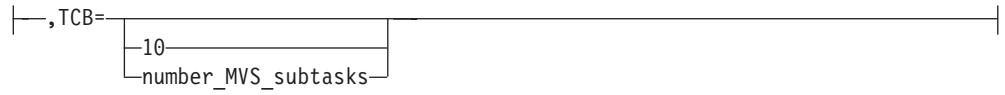
VBUILD

```
▶▶—name—VBUILD—TYPE=TCP—| CONTIMER | DGTIMER | DNSUFFIX | EXTIMER |  
▶| IATIMER | PORT | TCB |
```

TCB



TCB



PORT



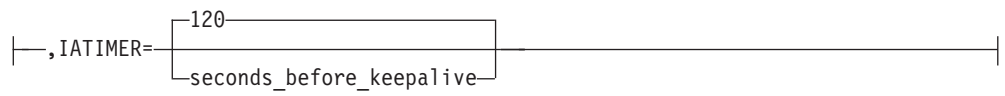
PORT



IATIMER



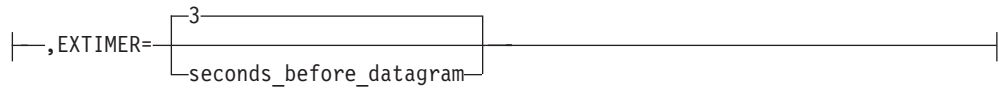
IATIMER



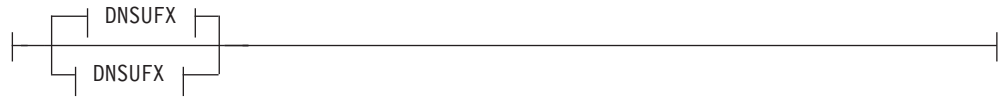
EXTIMER



EXTIMER



DNSUFIX



DNSUFIX



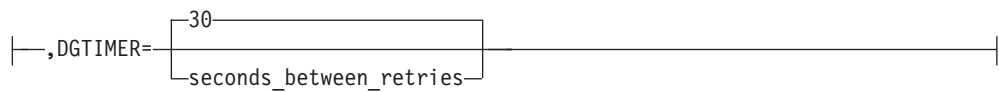
DNSUFIX



DGTIMER



DGTIMER



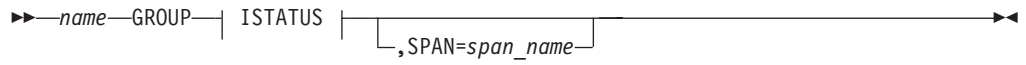
CONTIMER



CONTIMER



GROUP



ISTATUS



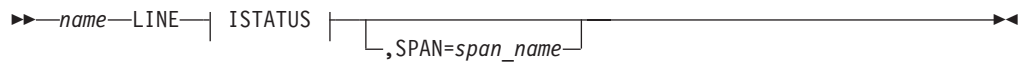
ISTATUS



ISTATUS



LINE



ISTATUS



ISTATUS



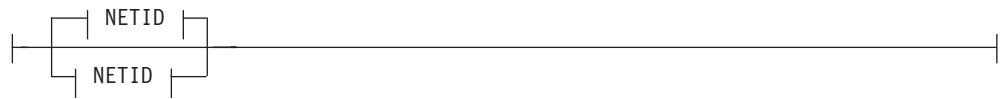
ISTATUS



PU



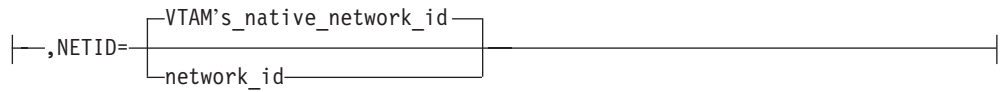
NETID



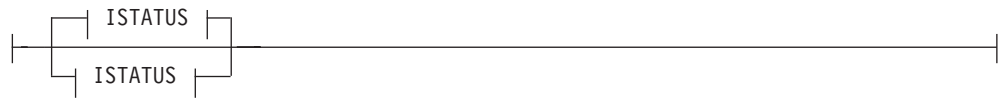
NETID



NETID



ISTATUS



ISTATUS



ISTATUS



Operand Descriptions

The following sections describe the operands you can code for this major node.

CONTIMER



CONTIMER



CONTIMER

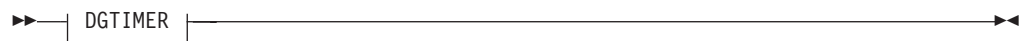


statements:
range:

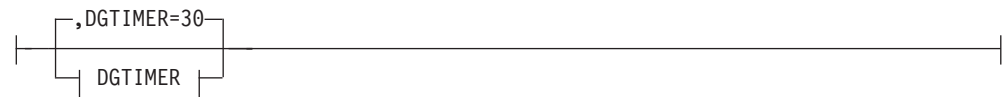
VBUILD
1-65535

Specifies the length of time, in seconds, that VTAM should wait for the MPTN connection to be established after the TCP connection has been activated. If the MPTN connection is not established within the time specified by CONTIMER, the session initiation fails. If you do not specify a value for CONTIMER, a default of 30 seconds is assumed.

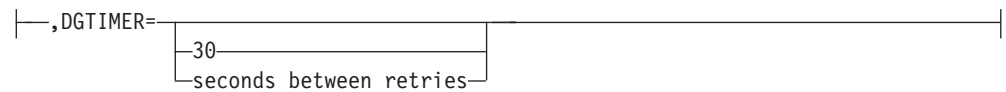
DGTIMER



DGTIMER



DGTIMER



statements:

VBUILD

range:

1-65535

Specifies the length of time, in seconds, between retries of sending a datagram for SNA expedited data, SNA session termination requests, or an MPTN KEEPALIVE datagram to a remote system using SNA over TCP/IP. The datagrams are continually sent at regular intervals dictated by the DGTIMER setting until a response is received from the remote system or until five datagrams have been sent without any response.

Retries are exceeded when five datagrams have been sent without any response; the action taken depends on the type of datagram that was sent:

- For SNA expedited data, the session that the data is being sent on is deactivated.
- For SNA session termination requests, the TCP connection that was being used by the SNA session is closed.
- For MPTN KEEPALIVE datagrams, all sessions using the IP address pair that the MPTN KEEPALIVE datagram represents are deactivated.

If you do not specify a value for DGTIMER, a default of 30 seconds is assumed.

DNSUFFIX



DNSUFFIX



DNSUFFIX



DNSUFFIX



statements:

VBUILD

Specifies the domain name suffix to be used when VTAM creates an IP domain name from an SNA LU name and SNA network ID. The DNSUFFIX operand ensures that any SNA LU name and SNA network ID specified in the form *luname.netid* is distinct from any existing IP domain name.

Because the generated domain name must be a valid IP domain name, the following conditions apply:

- The domain name suffix must be less than or equal to 237 characters in length. The maximum allowable length for a TCP/IP fully qualified name is 255 characters. Subtract 8 characters each for the LU name and network ID and 2 characters for each period (.) delimiter, and the difference is the 237-character limitation.
- Each label (the characters between the periods) in the 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

The domain name created by VTAM is in the form *luname.netid.cccc.cccc.cccc* where:

luname

is the SNA LU name

netid

is the SNA network ID

cccc.cccc.cccc

is the domain name suffix specified on the DNSUFFIX operand.

For example, assume that you specify DNSUFFIX=SNA.IBM.COM for an LU named LU1 that is in the SNA network named NETA. VTAM searches for the domain name LU1.NETA.SNA.IBM.COM in the DNS or in the HOSTS file. See “Mapping LU Names to IP Addresses in the DNS” on page 31 for more information on how an LU name is mapped to an IP address.

Note: TCP/IP domain names are formatted in increasing scope (for example, LU1.NETA.SNA.IBM.COM); SNA network-qualified LU names are formatted in decreasing scope (for example, NETA.LU1).

The default value for this operand is SNA.IBM.COM.

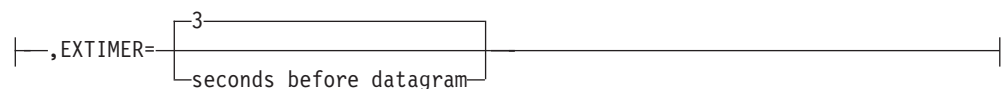
EXTIMER



EXTIMER



EXTIMER



statements:
range:

VBUILD
1-65535

Specifies the length of time, in seconds, between sending SNA expedited data over the TCP connection associated with an SNA session and sending the SNA expedited data using a datagram. When a response is received from the remote system either as a datagram or on the TCP connection, the process of sending the datagram is complete and no additional retries are performed. If you do not specify a value for EXTIMER, a default of 3 seconds is assumed.

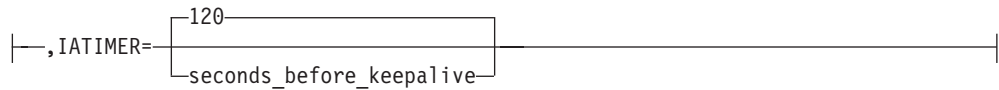
IATIMER



IATIMER



IATIMER



statements:
range:

VBUILD
1-65535

Specifies the length of time, in seconds, that VTAM allows two IP addresses to remain inactive before sending an MPTN KEEPALIVE datagram to test the connectivity between the two IP addresses.

Note: *Inactive* means that no data has been received from the remote IP address.

If you do not specify a value for IATIMER, a default of 120 seconds (2 minutes) is assumed.

ISTATUS



ISTATUS



ISTATUS



ISTATUS



statements:

GROUP, LINE, PU

Determines whether this minor node is to be activated after the first activation of the TCP/IP major node.

If you do not code the ISTATUS operand, or if you code ISTATUS=ACTIVE, the minor node becomes active when its major node becomes active. If you code ISTATUS=INACTIVE, the minor node remains inactive until the operator activates it. The operator can override this value by specifying the SCOPE operand on the VARY ACT command when the major node is activated. Refer to *z/OS Communications Server: SNA Operation* for information on using the VARY ACT command.

name



statements:

VBUILD

On VBUILD: Assigns a symbolic name to the VBUILD definition statement. The *name* is required.

On GROUP: Provides the minor node name of the line group represented by this definition statement. The *name* is required.

On LINE: Provides the minor node name of the link represented by this definition statement. The *name* is required.

On PU: Provides the minor node name of the physical unit represented by this definition statement. This name is used to represent an adjacent link station (ALS) for an LU to which a session is to be established using TCP over an IP network. This ALS name is used by the following:

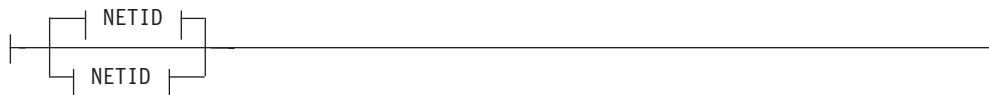
- ALSLIST operand on the CDRSC definition statement
- ALS selection function of the session management exit routine.

The *name* is required.

NETID



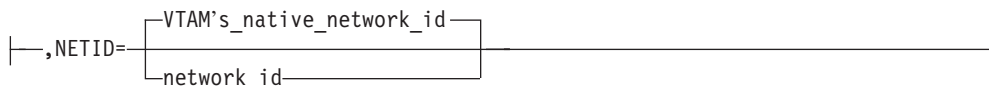
NETID



NETID



NETID



statements:

PU

Specifies a 1–8 character network ID. NETID is the network ID for all LUs that access VTAM through this PU which acts as the adjacent link station. If the NETID operand was not specified on the PU definition statement, VTAM's native network ID is used for the LU.

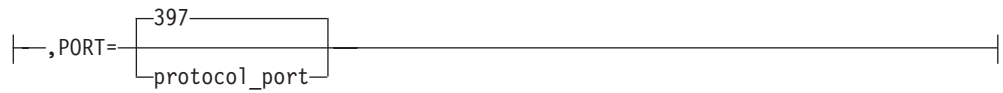
PORT



PORT



PORT



statements: *VBUILD*
 range: *1-65535*

Specifies the TCP and UDP protocol port that VTAM uses to support SNA sessions over an IP network. All nodes that establish SNA sessions over an IP network must use the same PORT number if those nodes are to communicate with each other. Specify an integer from 1 to 65535. The default value is 397. This value is a well-known port registered with the Internet Activities Board (IAB). It is recommended that you use the default value under normal operating conditions. If you specify a value other than the default, this value cannot be in use by any other TCP/IP application program in your network. Typically in TCP/IP environments values ranging from 5000 to 32000 are available for socket applications.

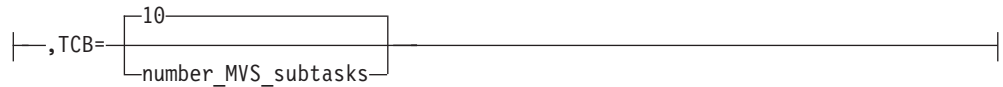
TCB



TCB



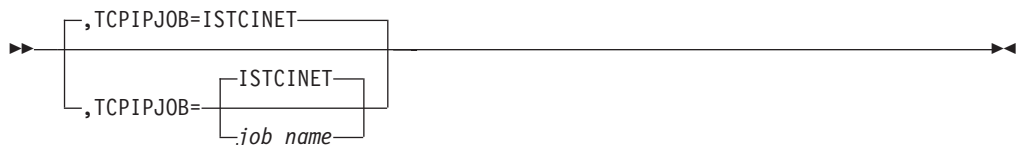
TCB



statements: *VBUILD*
 range: *1-99*

Specifies the number of MVS subtasks that can be used by VTAM to access TCP/IP. The value for TCB can range from 1 to 99. If you do not specify a value for TCB, a default of 10 is assumed.

TCPIPJOB



statements: VBUILD

Specifies the 1–8 character TCP/IP job name used to start the TCP/IP address space or the value ISTCINET for common inet implementation. When using SNA over TCP/IP, VTAM acts as a TCP/IP application.

If specified, the `job_name` variable must be the same name entered on the TCP/IPJOBNAME statement in the TCP/IP data file. If a job name is not specified, the default name used is ISTCINET. This is not a job name, but causes VTAM to connect to all available TCP/IP jobs on the host. The ISTCINET is derived from the common inet implementation of multiple TCP/IP stacks running concurrently.

TYPE=TCP

▶▶—TYPE=TCP—▶▶

statements: VBUILD

Indicates a TCP/IP major node. This operand is required.

SPAN

▶▶—`[,SPAN=span_name]`—▶▶

statements: GROUP, LINE, PU

Defines a span of control for VTAM minor node resources. Code this operand if you are using the NetView program.

The NetView program checks the SPAN value, but VTAM ignores it, even if you code it incorrectly.

Defining Independent and Dependent Logical Units for SNA over TCP/IP

SNA over TCP/IP supports communication for both independent logical units and dependent logical units.

The following section outlines definition guidelines that are specific for resources in an SNA over TCP/IP environment. For general information about VTAM resource definition, refer to *z/OS Communications Server: SNA Resource Definition Reference* and *z/OS Communications Server: SNA Network Implementation Guide*.

Defining Independent Logical Units for SNA over TCP/IP

Use the following guidelines when defining independent logical units in an IP network to VTAM:

- If a remote independent logical unit in an IP network initiates a session with VTAM, VTAM can define the LU dynamically provided that DYNLU=YES is coded as a VTAM start option.

- A remote independent logical unit in an IP network is defined to VTAM as a CDRSC.
- If VTAM initiates an LU-LU session to an LU over an IP network, you have to either:
 - Dynamically define the CDRSC by using the ALS selection function of the session management exit routine. (Refer to *z/OS Communications Server: SNA Customization* for more information.)
 - Predefine the CDRSC to VTAM using a CDRSC definition statement. (Refer to *z/OS Communications Server: SNA Resource Definition Reference* for more information.)

The following rules apply to the CDRSC definition for independent LUs:

- The name label on the CDRSC definition statement must be the remote independent logical unit name.
- The PU defined within the TCP/IP major node is the adjacent link station (ALS) for each session to the remote independent logical unit. Use the following to specify the PU:
 - ALSLIST operand on the CDRSC definition statement.
 - ALS selection function of the session management exit routine. For more information on the ALS selection function, see *z/OS Communications Server: SNA Customization*.

Defining Dependent Logical Units for SNA over TCP/IP

The dependent LU server (DLUS) function and the dependent LU requester (DLUR) function enable dependent LU communication across networks. SNA over TCP/IP uses the DLUS-DLUR functions to provide support for dependent LU communication across IP networks. Dependent logical units can communicate over the TCP/IP network provided that the following is true:

- The host is defined as a dependent LU server
- The workstation is defined as a dependent LU requester.

Refer to the *z/OS Communications Server: SNA Resource Definition Reference* and the *z/OS Communications Server: SNA Network Implementation Guide* for information about VTAM's DLUS function and the workstation product documentation for the DLUR support.

Consider the following when planning the use of the DLUS-DLUR support:

- VTAM is started as either of the following:
 - An interchange node (by using the NODETYPE and HOSTSA start options)
 - A network node (by using the NODETYPE start option)
- The DLUS must be adjacent to the TCP/IP network
- The LU-LU session must flow through the DLUS node.

Use the following guidelines when defining dependent LUs in an IP network to VTAM:

- The DLUR in an IP network is defined to VTAM as a CDRSC.
- The dependent LUs are defined within a switched major node.
- For outbound connections to the workstation, the DLURNAME operand on the PATH definition statement specifies the name of the CDRSC that represents the DLUR.

The following rules apply to the CDRSC definition for the DLUR.

- The name label on the CDRSC definition statement must be the CP name of the DLUR.
- The PU defined within the TCP/IP major node is the adjacent link station (ALS) for each session established between the DLUR and its supported dependent LUs. Use the following to specify the PU activated within the TCP/IP major node:
 - ALSLIST operand on the CDRSC definition statement.
 - ALS selection function of the session management exit routine. For more information on the ALS selection function, see *z/OS Communications Server: SNA Customization*.

Sample Configurations and Required Definitions

This section describes several configurations that enable communications between LUs using SNA over TCP/IP. Read this section to:

- Understand how SNA over TCP/IP is integrated into an SNA-based network
- Determine the VTAM system definition requirements for your network's configuration.

SNA over TCP/IP networks can be simple or complex. An example of a simple configuration is one in which the communicating LUs reside within VTAM nodes that are directly attached to the IP network. An example of a complex configuration is where communicating LUs use both SNA and IP transports requiring one or more MPTN transport gateways. A node that is attached to both an IP network and an SNA network and enables LUs attached to one network to communicate with LUs in the other network is an example of an MPTN transport gateway.

The configurations in this section assume the following characteristics:

- A DNS is attached to the network and accessible from all VTAM nodes. Different DNS configurations are possible, including placing the DNS on one of the IP hosts.
- One or more IP addresses must be associated with each node and defined in a DNS.
- One or more IP addresses are defined for each MPTN transport gateway. This IP address applies to all LUs within the SNA network that communicate with LUs over the IP network.
- In the DNS, you must define the domain name associated with any destination LU.
- Unless otherwise stated, communication flows between independent logical units.

See "Mapping LU Names to IP Addresses" on page 30 for more information on what you must define in a domain name server.

VTAM to VTAM over a Single IP Network

In Figure 7 on page 25, two VTAM hosts are connected only by an IP network. The LU application on each VTAM node communicates using an LU-LU session. An SNA network ID is still used and resolved as a qualifier of the SNA LU names, even though no SNA network is used. The IP address of each LU is the IP address of the node containing the LU. Because the session is between VTAM hosts, any LU protocol can be used. Sessions between LUs must be initiated by the primary LU.

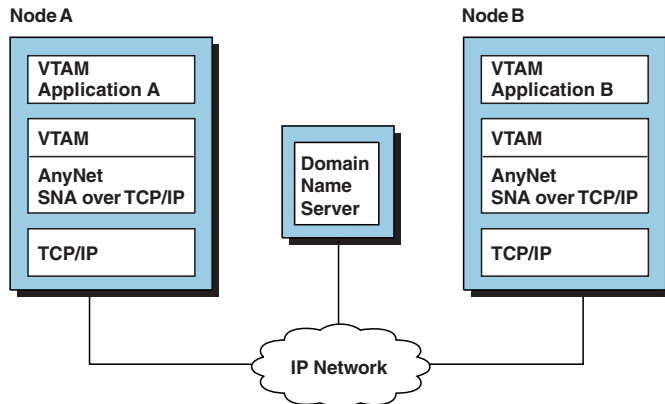


Figure 7. VTAM Sessions over an IP Network

The following definitions and guidelines apply to the configuration in Figure 7:

- Each VTAM must have a TCP/IP major node defined.
- For each LU accessed over the IP network, the VTAM that initiates the LU-LU session must either:
 - Contain a predefined CDRSC definition statement for the destination logical unit. You can use one of the following methods to define the adjacent link station:
 - Code the ALSLIST operand on the CDRSC definition statement
 - Use the ALS selection function of the session management exit routine.
 - Use the ALS selection function of the session management exit routine to dynamically define the CDRSC for the destination LU.
- The domain name associated with any destination LU must be defined in the DNS along with one or more corresponding IP addresses.
- The originating LU can be predefined by defining a CDRSC for the originating LU, or dynamically defined by coding DYNLU=YES on the VTAM start option list.

VTAM to Communications Server/2 over a Single IP Network

In Figure 8 on page 26, the VTAM node is connected only by an IP network to several IBM PCs running Communications Server/2. SNA application programs reside on the VTAM node and on each workstation. In this example, a domain name server (DNS) is accessible from the VTAM node and from each Communications Server/2 node. The IP address of the LU is the IP address of the node containing the LU.

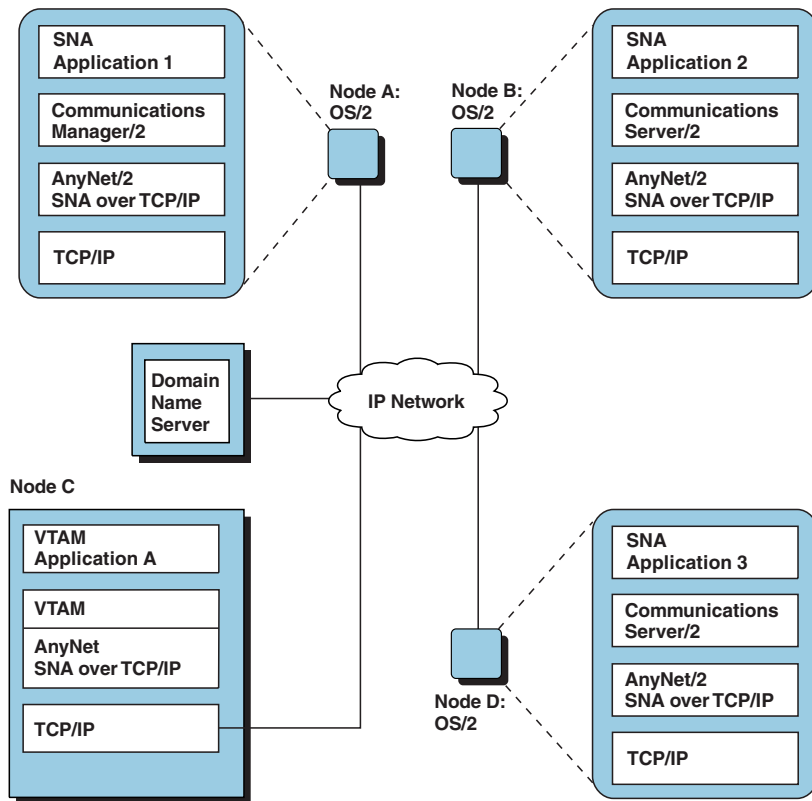


Figure 8. Sessions between VTAM and Communications Server/2 Workstations over an IP Network

The following definitions and guidelines apply to the configuration in Figure 8:

- VTAM must have a TCP/IP major node defined.
- For each LU accessed over the IP network, the VTAM that initiates the LU-LU session must either:
 - Contain a predefined CDRSC definition statement for the destination logical unit in the PC. You can use one of the following methods to define the adjacent link station:
 - Code the ALSLIST operand on the CDRSC definition statement
 - Use the ALS selection function of the session management exit routine.
 - Use the ALS selection function of the session management exit routine to dynamically define the CDRSC for the destination LU in the PC.
- The domain name associated with any destination LU must be defined in the DNS along with one or more corresponding IP addresses.
- The originating LU can be predefined by defining a CDRSC for the originating LU or dynamically defined by coding DYNLU=YES on the VTAM start option list.

VTAM as an MPTN Transport Gateway

In Figure 9 on page 27, VTAM2, configured with the AnyNet SNA over TCP/IP code, operates as an MPTN transport gateway between an SNA network and an IP network. The IP network contains a VTAM node and a Communications Server/2 node. The SNA network nodes support LUs that contact the MPTN transport gateway to setup sessions with LUs in the IP network.

All existing SNA network components, for example, other SSCPs, type 4 PUs, and type 2.1 nodes, can exist in an SNA-to-IP environment.

If two or more MPTN transport gateways connect an IP network and an SNA network, these transport gateways have no direct awareness of each other. From the SNA network, the transport gateway used is chosen using existing SNA session routing. From the IP network, the transport gateway is chosen from the list of IP addresses returned from the DNS.

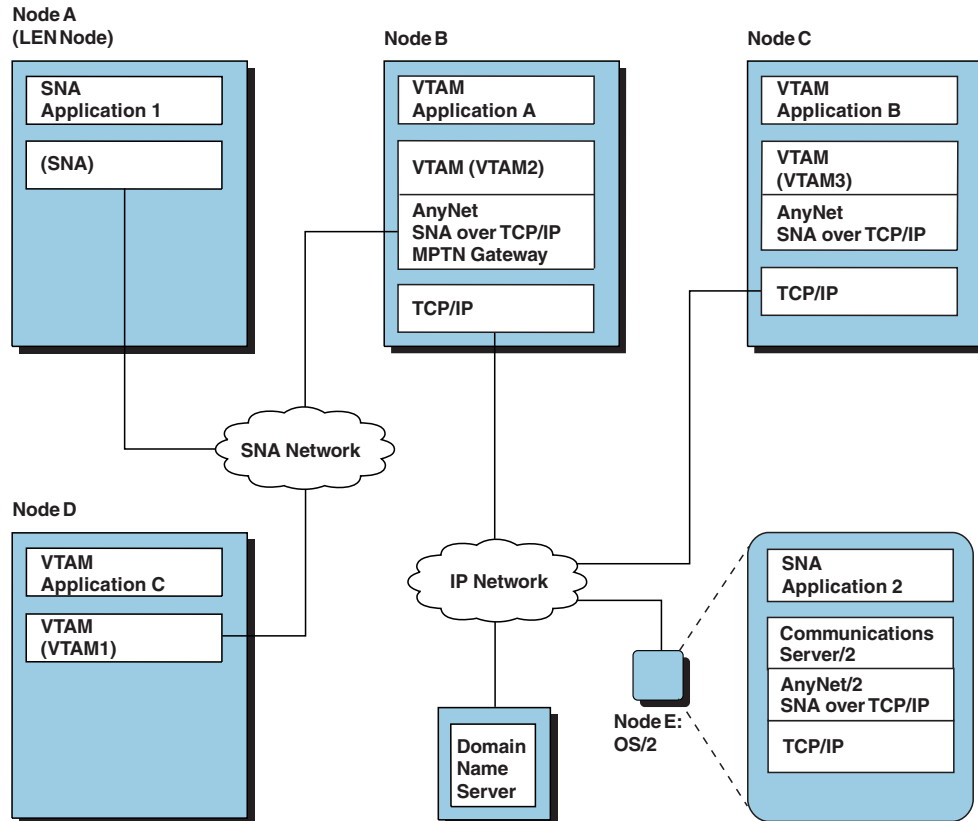


Figure 9. Sessions Using VTAM as an MPTN Transport Gateway

The following definitions and guidelines apply to the configuration in Figure 9:

- The MPTN transport gateway must have a TCP/IP major node defined.
- For each LU accessed over the IP network, the MPTN transport gateway that initiates the LU-LU session must either:
 - Contain a predefined CDRSC definition statement for the destination logical unit. You can use one of the following methods to define the adjacent link station:
 - Code the ALSLIST operand on the CDRSC definition statement
 - Use the ALS selection function of the session management exit routine.
 - Use the ALS selection function of the session management exit routine to dynamically define the CDRSC for the destination LU.
- The originating LU can be predefined by defining a CDRSC for the originating LU or dynamically defined by coding DYNLU=YES on the VTAM start option list.
- VTAM3, the IP-attached VTAM that is not an MPTN transport gateway, has the same definition requirements as the VTAM node in “VTAM to Communications Server/2 over a Single IP Network” on page 25.

VTAMs as Chained MPTN Transport Gateways

In Figure 10, two MPTN transport gateways, VTAM1 and VTAM3, attach separate SNA networks to one IP network. These nodes support LUs that contact the MPTN transport gateway to set up sessions with LUs in the IP network. The IP network contains a VTAM node and a Communications Server/2 node. Additional VTAM nodes and Communications Server/2 nodes can also exist on the IP network.

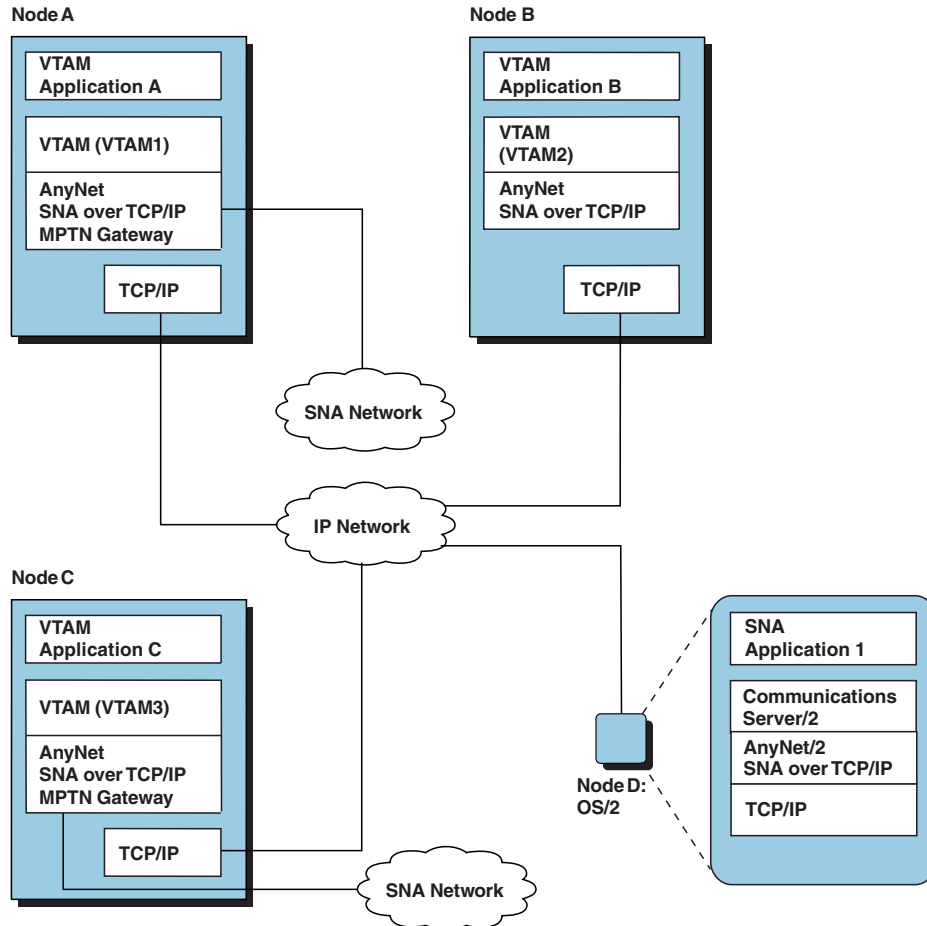


Figure 10. Sessions Using VTAMs as MPTN Transport Gateways

The following definitions and guidelines apply to the configuration in Figure 10:

- Each MPTN transport gateway must have a TCP/IP major node defined.
- For each LU accessed over the IP network, the MPTN transport gateway that initiates the LU-LU session must either:
 - Contain a predefined CDRSC definition statement for the destination logical unit. You can use one of the following methods to define the adjacent link station:
 - Code the ALSLIST operand on the CDRSC definition statement
 - Use the ALS selection function of the session management exit routine.
 - Use the ALS selection function of the session management exit routine to dynamically define the CDRSC for the destination LU.
- The domain name associated with any destination LU must be defined in the DNS or a HOSTS file along with one or more corresponding IP addresses.

- The originating LU can be predefined by defining a CDRSC for the originating LU or dynamically defined by coding DYNLU=YES on the VTAM start option list.
- Existing LUs in the SNA networks can operate as LUs that access the IP network.
- VTAM2, the IP-attached VTAM that is not an MPTN transport gateway, has the same definition requirements as the VTAM node in “VTAM to Communications Server/2 over a Single IP Network” on page 25.

Dependent Logical Units Initiating Sessions

In any of the previous configurations, you can use dependent logical units providing the host serves as the dependent LU server and the dependent LU requester function is enabled at the workstation.

Figure 11 illustrates how DEPLU1, a dependent logical unit, can initiate a session with SNA Application 2 over an IP network. This configuration requires that node A is configured as a DLU requester and Node B is configured as a DLU server.

DEPLU1 initiates a session with the help of VTAM, which sets up the session between DEPLU1 and SNA Application 2. When VTAM helps DEPLU1 set up the session, it acts as DEPLU1’s DLUS.

The following definitions and guidelines apply to the configuration in Figure 11.

- Node B is defined as a dependent LU server.
- Node A is configured as a DLUR.

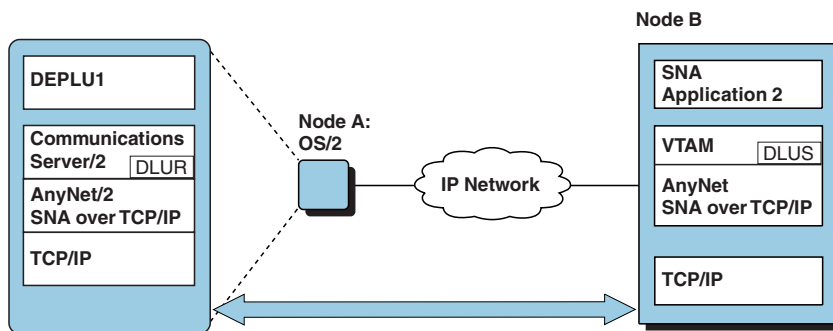


Figure 11. Workstation to VTAM Dependent LU Communication

Figure 12 on page 30 illustrates how DEPLU1, a dependent logical unit, can communicate with SNA Application 2 over an IP network. In this example, Node B is the DLUS.

Note: Be aware of the following restriction: the communication flows between SNA Application 2 and DEPLU1 are always routed through the DLUS, Node B.

For more information about VTAM’s dependent LU server support, refer to the *z/OS Communications Server: SNA Network Implementation Guide*.

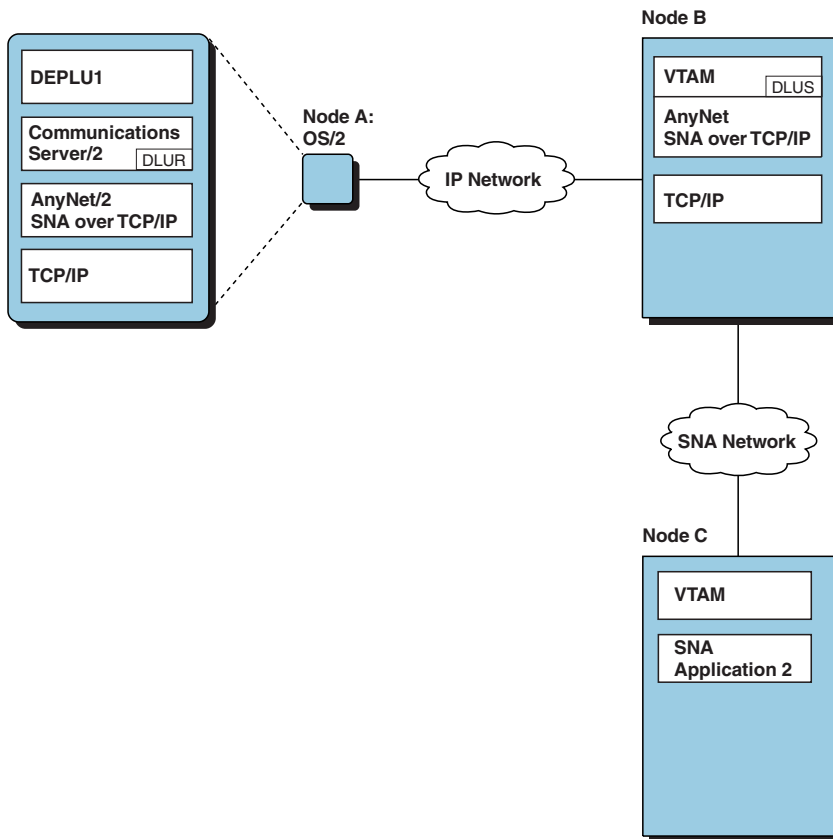


Figure 12. Dependent LU Communication through an Intermediate VTAM

Considerations for Using SNA over TCP/IP with LUs in Different SNA Networks

SNA over TCP/IP provides cross-network support to allow logical units (LUs) with different SNA network IDs (NETIDs) to communicate over the TCP/IP network. To enable LUs to do this, you must code YES on the VTAM XNETALS start option. See the XNETALS start option in the *z/OS Communications Server: SNA Resource Definition Reference* for more information.

Mapping LU Names to IP Addresses

This section explains the relationship between LU names and IP addresses in an SNA over TCP/IP environment. For SNA application programs to communicate across an IP network, LU names are reformatted into IP domain names that, in turn, are mapped to IP addresses. To understand how LU names are reformatted into IP domain names, read the explanation of the DNSUFFIX operand in “DNSUFFIX” on page 16. You should also be familiar with the following:

- Defining an LU to VTAM (refer to the *z/OS Communications Server: SNA Resource Definition Reference*)
- Setting up and using TCP/IP domain name systems (refer to the *z/OS Communications Server: IP Configuration Reference*)
- Defining host names in the HOSTS.LOCAL file of TCP/IP (refer to the *z/OS Communications Server: SNA Resource Definition Reference*)

You can use the HOSTS.LOCAL file, the DNS, or both HOSTS.LOCAL and the DNS to define domain names. These domain names must be consistent with the domain names built by SNA over TCP/IP support using the DNSUFFIX operand. Using the gethostbyname call of the socket API of TCP/IP, SNA over TCP/IP accesses the DNS to obtain the IP address associated with a particular SNA LU. The DNS can be located anywhere in the IP network so long as the DNS resolver can locate the node on which the desired IP address resides. If the DNS is not accessible, the HOSTS.LOCAL file is searched for the address.

Mapping LU Names to IP Addresses in the DNS

Each destination LU name is mapped to a corresponding internet address in the DNS. Consequently, each LU name must have a corresponding resource record (RR) defined in a DNS resource set.

You do not need to define the domain name associated with the LU in any particular DNS. However, you must define the LU name in a DNS that can be reached from the node originating the LU-LU session over the IP network.

Attention: If you try to initiate a session with an LU that is not defined in a DNS, that session fails with sense code 80040000.

If you are using z/OS TCP/IP Services domain name server, refer to the *z/OS Communications Server: IP User's Guide and Commands* or the *z/OS Communications Server: IP Configuration Reference* for information on defining resource records. For information on how to obtain RFCs, check out the InterNIC home page at this URL:

<http://ds.internic.net>

Mapping LU Names to IP Addresses in the HOSTS File

Each destination LU name is mapped to a corresponding internet address in the HOSTS file. If you are using a HOSTS file and a DNS to map LU names to IP addresses, you have the additional overhead of maintaining the same names in two files.

For more information on the format of the HOSTS file, see RFC 952.

Configuration Requirements for OS/2-to-MVS Communications

This section describes some of the steps that you may need to take to configure your network communications. This example illustrates how to define SNA over TCP/IP to enable the nodes in Figure 13 on page 32 to communicate.

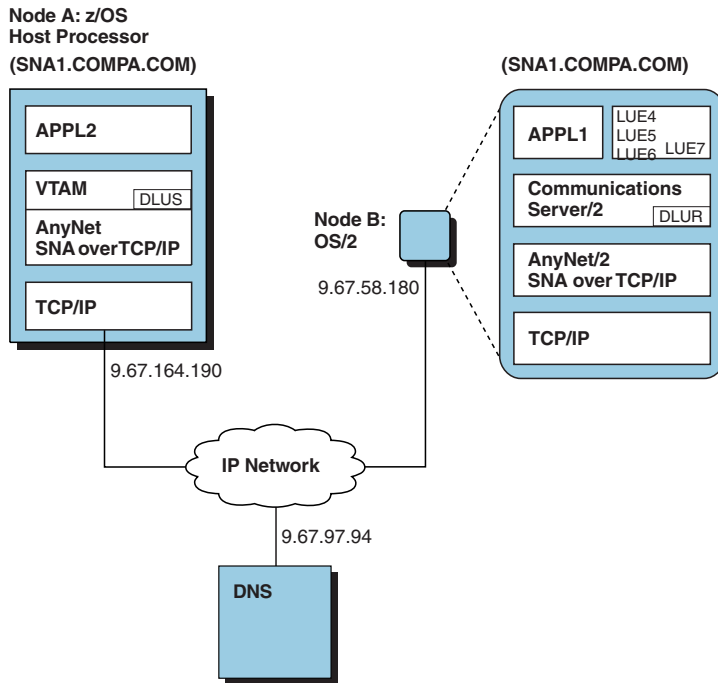


Figure 13. Configuring SNA over TCP/IP to Support OS/2-to-MVS Communications

Configuration on MVS

To configure SNA over TCP/IP on MVS, do the following:

1. Define the IP network to VTAM by coding the TCP/IP major node. The following example shows how to code the TCP major node:

```
TCP1      VBUILD TYPE=TCP
          DNSUFFIX=(SNA.COMPA.COM)
TCPGR     GROUP
TCPLI     LINE
TCPPU     PU      NETID=NETA
```

2. Create a CDRSC.

If the OS/2 session partner initiates the session, you can specify that VTAM dynamically create a CDRSC major node for that LU. If VTAM initiates the session, you can either use the ALS selection function of VTAM's session management exit routine to dynamically define a CDRSC for the LU, or you can code a CDRSC major node. The following example shows how to code the CDRSC major node.

```
CDRSC1   VBUILD TYPE=CDRSC
CDRSCG   GROUP
APPL1    CDRSC  ALSLIST=TCPPU
DLUR     CDRSC  ALSLIST=TCPPU
```

3. Create a switched major node for all dependent LUs.

The following example shows how to define the switched major node.

```
SWDLR   VBUILD TYPE=SWNET,MAXDLUR=10
LOGPU    PU      ADDR=01,IDBLK=05D,IDNUM=00001
PATH1    PATH    DLURNAME=DLUR,DLCADDR=(1,C,INTPU),
              DLCADDR=(2,X,05D00001)
```

LUE4	LU	LOCADDR=4
LUE5	LU	LOCADDR=5
LUE6	LU	LOCADDR=6
LUE7	LU	LOCADDR=7

For more information on VTAM coding definitions, refer to the *z/OS Communications Server: SNA Resource Definition Reference*.

Configuration on OS/2

For the steps used to configure the OS/2 workstation, refer to *IBM Communications Server for OS/2 Warp: Guide to AnyNet Sockets over SNA*.

Optimizing SNA over TCP/IP Performance

The following sections provide guidelines and recommendations for optimizing SNA over TCP/IP performance.

Recommendations for Specifying VTAM and TCP/IP Tuning Parameters

Performance of SNA over TCP/IP is sensitive to tuning parameters that are specified by VTAM and by TCP/IP. Additionally, SNA over TCP/IP performance is affected by how the VTAM and TCP/IP tuning parameters affect each other.

For sessions that flow across a TCP/IP network, you should modify the maximum RU size that is used for the SNA over TCP/IP sessions. Maximum RU size is specified on the RUSIZES parameter on the MODEENT macroinstruction. Refer to the *z/OS Communications Server: SNA Resource Definition Reference* for more information.

When specifying maximum RU size, consider the values that have been chosen for TCP/IP window size and maximum transmission unit (MTU). SNA over TCP/IP performance could be enhanced by ensuring the maximum RU plus 10 bytes of header information equals the MTU size. It is also recommended that the TCP/IP window size be a multiple of the MTU size. For more information about the TCP/IP window size and the maximum transmission unit (MTU), refer to *z/OS Communications Server: IP Configuration Reference*.

In some environments, such as high-speed transfer of bulk data, increasing the maximum RU size can lower CPU usage and increase system throughput.

Recommendations for Specifying TCP/IP Major Node Timers

The TCP/IP major node provides four timers that can be used to tune or optimize SNA over TCP/IP:

- CONTIMER
- DGTIMER
- EXTIMER
- IATIMER

Specifying CONTIMER

When specifying CONTIMER, ensure that the value is longer than the expected round-trip response time between the host and the remote partner. Allow additional time to permit temporary congestion in the IP network to subside.

Specifying DGTIMER

If you code DGTIMER, code a value that is longer than the expected round-trip response time between this system and the remote system. Allow additional time between retries of sending datagrams to permit temporary congestion in the IP network to subside.

Specifying EXTIMER

The value for EXTIMER should be slightly larger than the round-trip response time between this system and the remote systems to which it communicates. You can specify a larger value to reduce the number of datagrams sent with the same information (SNA expedited data) that flows over the TCP connection; however, a larger value also increases the response time for SNA expedited data when the TCP connection is congested.

Specifying IATIMER

When determining the value of IATIMER, consider the balance between CPU usage and storage. If IATIMER is set too low, VTAM spends CPU cycles by routinely sending MPTN KEEPALIVE datagrams to test IP address connectivity for idle, but active, connections. If IATIMER is set too high, resources and storage could remain dedicated to IP addresses that are inactive.

Recommendations for Specifying TCB Values

| If the TCB operand is less than the number of processors in your node, you can
| optimize throughput by coding the value of the TCB equal to the number of
| processors.

Chapter 3. Initializing and Operating AnyNet SNA over TCP/IP

This chapter describes the steps and commands used to start and monitor SNA over TCP/IP.

Before you activate and use SNA over TCP/IP, you must:

- Install, configure, and initialize TCP/IP over the z/OS product.
- Install VTAM using the procedures described in the *z/OS Program Directory*.
- Define the IP network to VTAM using the procedures and guidelines described in “Chapter 2. Defining the IP Network to VTAM” on page 9.

Initializing SNA over TCP/IP

To begin using SNA over TCP/IP, activate the network by doing the following:

1. Start the following:
 - VTAM
 - TCP/IP
2. Activate the TCP/IP major node
3. After TCP/IP is started, activate the PU of the TCP/IP major node.
4. Activate any CDRSCs or switched major nodes you defined for SNA over TCP/IP.
5. Establish sessions.

If all the software components start up without failures, begin normal operation procedures.

Monitoring SNA over TCP/IP

Use the DISPLAY ID, DISPLAY SESSIONS, and DISPLAY STATS operator commands to obtain information related to SNA over TCP/IP.

DISPLAY ID

Use the DISPLAY ID command to determine:

- Status of a TCP/IP major node
- Settings of VBUILD parameters specified for a TCP/IP major node.

The DISPLAY ID command returns the information as part of the IST075I message group. Refer to *z/OS Communications Server: SNA Messages* for more information about VTAM messages.

►►—DISPLAY NET, ID=*name*—◄◄

id=*name*

Specify the symbolic *name* of the TCP/IP major node to display settings for associated VBUILD parameters.

The following is an example of the messages that are returned when a TCP/IP major node is activated and then displayed. In this example, the major node is named TCPMN1 and the associated PU is named TCPPU1.

```

V NET,ACT,ID=TCPMN1,ALL
IST097I VARY ACCEPTED
IST093I TCPMN1 ACTIVE
IST093I TCPPU1 ACTIVE
D NET,ID=TCPMN1,SCOPE=ALL
IST097I DISPLAY ACCEPTED
IST075I NAME = TCPMN1, TYPE = TCP/IP MAJOR NODE
IST486I STATUS= ACTIV, DESIRED STATE= ACTIV
IST1342I DNSUFFIX = SNA.IBM.COM
IST1400I DGTIMER = 30 EXTIMER = 3
IST1406I CONTIMER = 30 IATIMER = 120
IST1692I TCB = 10 TCP PORT = 397
IST654I I/O TRACE = OFF, BUFFER TRACE = OFF
IST170I LINES:
IST232I TCPLINE1, ACTIV
IST314I END

```

For more information about this command, see *z/OS Communications Server: SNA Operation*.

DISPLAY SESSIONS

For SNA sessions over an IP network, the DISPLAY SESSIONS command enables you to determine the IP address and port number corresponding to the session with that session ID. DISPLAY SESSIONS returns the new message IST1165I as part of the IST879I message group.

►►—DISPLAY NET,SESSIONS,SID=*session_id*—◄◄

Sid=*sessionid*

Returns the remote and local network types and IP connection addresses for the session partners.

The following is sample output for the DISPLAY SESSIONS command.

```

D NET,SESSIONS,SID=F6ABEEC34A665F8D
IST097I DISPLAY ACCEPTED
IST350I DISPLAY TYPE = SESSIONS
IST879I PLU/OLU REAL = NETA.APPC2A02 ALIAS = ***NA***
IST879I SLU/DLU REAL = NETA.TCPLU3 ALIAS = ***NA***
IST880I SETUP STATUS = ACTIV
IST875I ALSNAME TOWARDS SLU = TCPPU1
IST933I LOGMODE=BATCH , COS=*BLANK*
IST1165I LOCAL TCP/IP ADDRESS = 9.67.58.164..1025
IST1165I REMOTE TCP/IP ADDRESS = 9.67.58.180..397
IST314I END

```

For more information about this command, see *z/OS Communications Server: SNA Operation*. See *z/OS Communications Server: SNA Messages* for a description of the text in message IST1165I.

DISPLAY STATS

The DISPLAY STATS command enables you to determine:

- Whether AnyNet SNA over TCP/IP is installed on this host
- Number of TCP/IP major nodes defined to this VTAM
- Maximum TCB value for any of the TCP/IP major nodes activated by this VTAM
- Number of LU-LU sessions that have been established across the IP network

The following is sample output for the DISPLAY STATS command.

```
IST097I DISPLAY ACCEPTED
IST350I DISPLAY TYPE = STATS,TYPE=VTAM
IST1345I ID      VALUE      DESCRIPTION
IST1227I 130     YES = ANYNET/MVS SNA OVER TCP/IP INSTALLED
IST1227I 127     1 = TCP/IP MAJOR NODES
IST1227I 128     1 = MAXIMUM TCB VALUE FOR TCP/IP MAJOR NODES
IST1227I 129     5 = TCP/IP LU-LU SESSIONS
IST314I END
```

Attention: Depending on the size and configuration of your network, issuing this command from the console or from the ISTSTATS program operator may affect system performance. Issuing the DISPLAY STATS command from the console is not recommended.

For more information about this command, refer to *z/OS Communications Server: SNA Operation*.

Messages That Can Contain IP Transport-Related Text

The variable text for some existing IST messages now includes TCP/IP as follows:

```
IST075I
    command FOR ID = nodename FAILED - INSUFFICIENT STORAGE
IST879I
    {PLU{lutype}|SLU{lutype}}: REAL = realname ALIAS = aliasname
IST1165I
    REMOTE|LOCAL network-type ADDRESS = connection_address
IST1342I
    DNSUFFIX = dnsuffix
IST1343I
    dnsuffix_continuation
IST1387I
    TCP PU puname IS UNABLE TO ACCEPT CONNECTION REQUESTS
IST1388I
    SOCKET callname CALL FAILED, TCP ERROR NUMBER = error
IST1389I
    NO TCB IS AVAILABLE FOR SOCKET
IST1390I
    NO SOCKET DESCRIPTOR IS AVAILABLE
IST1400I
    DGTIMER = dgtimer EXTIMER = extimer
IST1406I
    CONTIMER = contimer IATIMER = iatimer
IST1692I
    TCB = taskno TCP PORT = portno
```

See *z/OS Communications Server: SNA Messages* for a description of all messages related to SNA over TCP/IP.

Chapter 4. Diagnosing AnyNet SNA over TCP/IP

This chapter describes the traces you can use to help diagnose problems that could arise when you install, configure, or use z/OS SNA over TCP/IP.

I/O Tracing for SNA over TCP/IP

For SNA over TCP/IP, the I/O trace providing tracing of all MPTN formats that pass through the TCP major node. To activate the trace for this interface, specify TYPE=IO as follows on the VTAM MODIFY TRACE command:

```
►►—MODIFY procname,TRACE,TYPE=IO,ID=node_name—◄◄
```

TYPE=IO

starts a trace of the MPTN formats associated with the node identified by the ID operand. For the format of the MPTN trace header, see *z/OS Communications Server: SNA Diagnosis V1 Techniques and Procedures*.

ID=*name*

specifies the name of the resource for which tracing is to be done.

For **TYPE=IO** or **TYPE=BUF**, either of the following names can be specified:

- The name of a TCP/IP major node (TYPE=IO,SCOPE=ALL only)
- The name of an active line attached to a TCP/IP major node
- The name of an active PU defined within a TCP/IP major node

For more information about this command, see *z/OS Communications Server: SNA Operation*.

You can also activate MPTN tracing when you start VTAM. Specify the TRACE and TYPE options on the VTAM START command. For more information on specifying VTAM START options, see the *z/OS Communications Server: SNA Resource Definition Reference*.

VTAM Internal Tracing for SNA over TCP/IP

The VIT option, TCP, traces socket API calls and completions and z/OS UNIX calls and completions. The following VIT entries can be generated for SNA over TCP/IP when you specify the TCP option:

- IPI
- IPC
- IPE
- MPT/MPT2
- NMPT/NMPT2
- SOCA
- SOCD
- SOCF
- SOCR

The following VIT entries can be generated for SNA over TCP/IP when you specify the PSS option:

- ATSK
- BTSK

- DTSK
- ETSK
- VPST
- VRSM
- VWAI.

The following VIT entries can be generated for SNA over TCP/IP when you specify the NRM option:

- BSPA
- BSPD
- BSPF.

To activate the VIT with the TCP option, specify TYPE=VTAM,OPTION=TCP as follows on the VTAM MODIFY TRACE command:

```
▶▶—MODIFY procname,TRACE,TYPE=VTAM,OPTION=TCP—▶▶
```

TYPE=VTAM

Activates VTAM internal tracing.

OPTION=TCP

Causes tracing of the activity associated with SNA over TCP/IP.

For more information about this command, see *z/OS Communications Server: SNA Operation*.

You can also activate VIT tracing for the TCP option when you start VTAM. Specify TRACE, TYPE=VTAM and OPTION=TCP on VTAM START command. For more information on specifying VTAM START options, see the *z/OS Communications Server: SNA Resource Definition Reference* .

For the formats of all VIT entries related to SNA over TCP/IP, see *z/OS Communications Server: SNA Diagnosis V1 Techniques and Procedures*.

TCP/IP Module Name in VIT Records

Some VTAM internal trace records include the associated module names in EBCDIC, without the IST prefix and the sixth letter. For TCP/IP interface modules, the first two letters are "IP". For more information on identifying VTAM components in trace output, see *z/OS Communications Server: SNA Diagnosis V1 Techniques and Procedures*.

Sense Codes That Contain IP-Related Information

08010000

Resource not available. The LU, PU, link station, or link specified in an RU is not available.

For SNA over TCP/IP, this sense code can be set when the PU of the TCP/IP major node is activated or when a session is activated.

- When it is set during PU activation, it implies one of the following:
 - TCP/IP is not active
 - TCP/IP has rejected VTAM's initial steps to start communication to TCP/IP.

If TCP/IP is active, this could be a TCP/IP or a VTAM error. It is recommended that you verify that the TCP option of the VTAM Internal Trace is set and dump for problem determination. If you cannot determine the cause of the problem from the output provided, take the following actions:

- If you have access to IBMLink^{*}, search for known problems in this area. If no applicable matches are found, report the problem to IBM by using the Electronic Technical Report (ETR) option on IBMLink.
- If you do not have access to IBMLink, report the problem to the IBM software support center.
- When it is set during session activation, it implies that the connection timer (CONTIMER on the TCP major node) has expired. This could mean that an intermediate node in the TCP connection route has failed or that there is an error at the endpoint of the TCP connection. Use the TCP/IP ping command to verify that you have connectivity to the destination node.

081C0000

Request not executable: The requested function cannot be executed, because of a permanent error condition in the receiver.

For SNA over TCP/IP, this sense code usually occurs when TCP/IP fails a VTAM socket API request.

This could mean that an intermediate node in the TCP connection route has failed or that there is an error at the endpoint of the TCP connection. Use the TCP/IP ping command to verify that you have connectivity to the destination node.

This sense code could also indicate a VTAM or a TCP/IP error. If the ping command completes successfully, it is recommended that you verify that the TCP option of the VTAM Internal Trace is set and dump for problem determination. If you cannot determine the cause of the problem from the output provided, take the following actions:

- If you have access to IBMLink, search for known problems in this area. If no applicable matches are found, report the problem to IBM by using the Electronic Technical Report (ETR) option on IBMLink.
- If you do not have access to IBMLink, report the problem to the IBM software support center.

08640002

Excessive Elapsed Time- No Retry: Excessive time has elapsed while waiting for a required action or event.

For SNA over TCP/IP, this sense code is set when the inactivity timer (IATIMER on the TCP major node) has expired, and all attempts to contact the remote IP address have failed. See "DGTIMER" on page 15 and "IATIMER" on page 18 for more information. When attempts to contact the address fail, all the SNA sessions using that specific IP address pair are deactivated with this sense code.

This sense code usually means that an intermediate node in the TCP connection route has failed or that there is an error at the endpoint of the TCP connection. Use the TCP/IP ping command to test connectivity to the destination node.

80040000

Unrecognized destination: A node in the path has no routing information for the destination specified by the SLU name in the BIND request.

For SNA over TCP/IP, this sense code is set during session activation when an attempt to find the LU name in the TCP/IP domain name server fails.

The failure could result from one of the following:

- The domain name server is not accessible.
- The LU name has not been properly added to the domain name server.

Use the TCP/IP ping command to verify connectivity to the domain name server address. If you do have connectivity to the domain name server, ping using the fully-qualified network name of the destination LU (LU1.NETA.SNA.IBM.COM) to verify that the address that the domain name server returns is the correct IP address.

Refer to *z/OS Communications Server: IP and SNA Codes* for information about other VTAM sense codes.

Service Information

If you cannot determine the cause of the problem from the output provided, take the following actions:

- If you have access to IBMLink, search for known problems in this area. If no applicable matches are found, report the problem to IBM by using the Electronic Technical Report (ETR) option on IBMLink.
- If you do not have access to IBMLink, report the problem to the IBM Support Center.

Appendix A. 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

monospace

Commands, command options, examples, messages, text that you type or enter

CAPS File names and file paths

italics User-supplied variables, message variables, titles of books

bold Socket call names, return codes

“quotes”

Titles in a menu

How to Read a Syntax Diagram

The syntax diagram shows you how to specify a command so that the operating system can correctly interpret what you type. Read the syntax diagram from left to right and from top to bottom, following the horizontal line (the main path).

Symbols and Punctuation

The following symbols are used in syntax diagrams:

- ▶▶ Marks the beginning of the command syntax.
- ▶ Indicates that the command syntax is continued.
- | Marks the beginning and end of a fragment or part of the command syntax.
- ◀◀ Marks the end of the command syntax.

You must include all punctuation such as colons, semicolons, commas, quotation marks, and minus signs that are shown in the syntax diagram.

Parameters

The following types of parameters are used in syntax diagrams.

Required

Required parameters are displayed on the main path.

Optional

Optional parameters are displayed below the main path.

Default

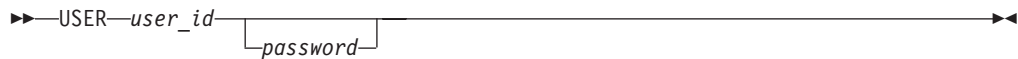
Default parameters are displayed above the main path.

Parameters are classified as keywords or variables. Keywords are displayed in uppercase letters and can be entered in uppercase or lowercase. For example, a command name is a keyword.

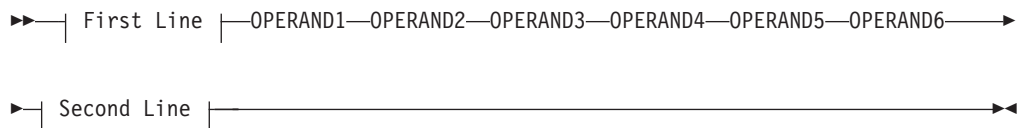
Variables are italicized, appear in lowercase letters, and represent names or values you supply. For example, a data set is a variable.

Syntax Examples

In the following example, the USER command is a keyword. The required variable parameter is *user_id*, and the optional variable parameter is *password*. Replace the variable parameters with your own values.



Longer than one line: If a diagram is longer than one line, the first line ends with a single arrowhead and the second line begins with a single arrowhead.



Required operands: Required operands and values appear on the main path line.



You must code required operands and values.

Choose one required item from a stack: If there is more than one mutually exclusive required operand or value to choose from, they are stacked vertically in alphanumeric order.



Optional values: Optional operands and values appear below the main path line.



You can choose not to code optional operands and values.

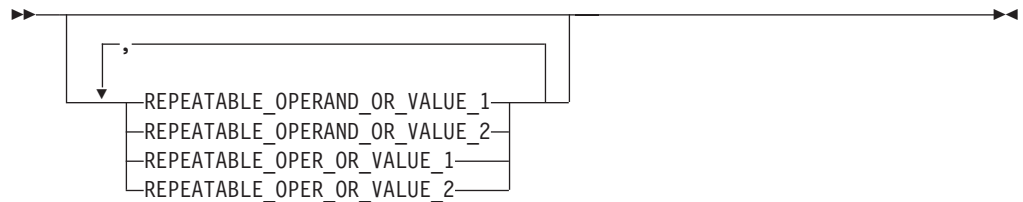
Choose one optional operand from a stack: If there is more than one mutually exclusive optional operand or value to choose from, they are stacked vertically in alphanumeric order below the main path line.



Repeating an operand: An arrow returning to the left above an operand or value on the main path line means that the operand or value can be repeated. The command means that each operand or value must be separated from the next by a comma.



Selecting more than one operand: An arrow returning to the left above a group of operands or values means more than one can be selected, or a single one can be repeated.



If an operand or value can be abbreviated, the abbreviation is described in the text associated with the syntax diagram.

Case Sensitivity: VTAM commands are not case sensitive. You can code them in uppercase or lowercase.

Nonalphanumeric characters: If a diagram shows a character that is not alphanumeric (such as parentheses, periods, commas, and equal signs), you must code the character as part of the syntax. In this example, you must code `OPERAND=(001,0.001)`.



Blank spaces in syntax diagrams: If a diagram shows a blank space, you must code the blank space as part of the syntax. In this example, you must code `OPERAND=(001 FIXED)`.



Default operands: Default operands and values appear above the main path line. VTAM uses the default if you omit the operand entirely.



Variables: A word in all lowercase italics is a *variable*. Where you see a variable in the syntax, you must replace it with one of its allowable names or values, as defined in the text.

▶▶ *variable* ◀◀

Syntax fragments: Some diagrams contain syntax fragments, which serve to break up diagrams that are too long, too complex, or too repetitious. Syntax fragment names are in mixed case and are shown in the diagram and in the heading of the fragment. The fragment is placed below the main diagram.

▶▶ | Reference to Syntax Fragment | ◀◀

Syntax Fragment:

|—1ST_OPERAND,2ND_OPERAND,3RD_OPERAND—|

References to syntax notes appear as numbers enclosed in parentheses above the line. Do not code the parentheses or the number.

▶▶ (1)
OPERAND ◀◀

Notes:

- 1 An example of a syntax note.

Appendix B. Flows for SNA over TCP/IP

This appendix describes line flows associated with SNA over TCP/IP. To help isolate the source of a problem, advanced programmers can compare dump and trace data to the flows described in this section and the trace formats described in *z/OS Communications Server: SNA Diagnosis V1 Techniques and Procedures*.

These line flows represent data frames that enter and leave SNA-over-TCP/IP nodes. All flows beginning with "MPTN_" are MPTN-defined formats and are described in *Multiprotocol Transport Networking: Formats* Refer to this publication and to *Multiprotocol Transport Networking: Technical Overview* for related information on MPTN architecture. MPTN-prefixed formats flow in the IP network as TCP data.

In these flows, the top of each figure indicates the type of flow from node to node. "TCP" indicates a TCP or UDP flow, "SNA" indicates an SNA flow. In each box, the information describes whether the node is the source LU, destination LU, DNS, transport gateway, DLUR, or DLUS.

Abbreviations Used in The Flow Diagrams

The flows in this section use the following abbreviations:

Abbreviation

for

DNS	Domain name server
IP@s	Source IP address
IP@d	Destination IP address
LL	Logical length
LUd	Logical unit of the destination node
LUs	Logical unit of the source node
MPTN_DG_OOB	Out-of-band datagram
DA_TERM	MPTN duplex abortive termination request
SON	Session outage notification
TP	Transaction program
SNA/IP	SNA over TCP/IP

Connection Setup: No Transport Gateway, New Session Required

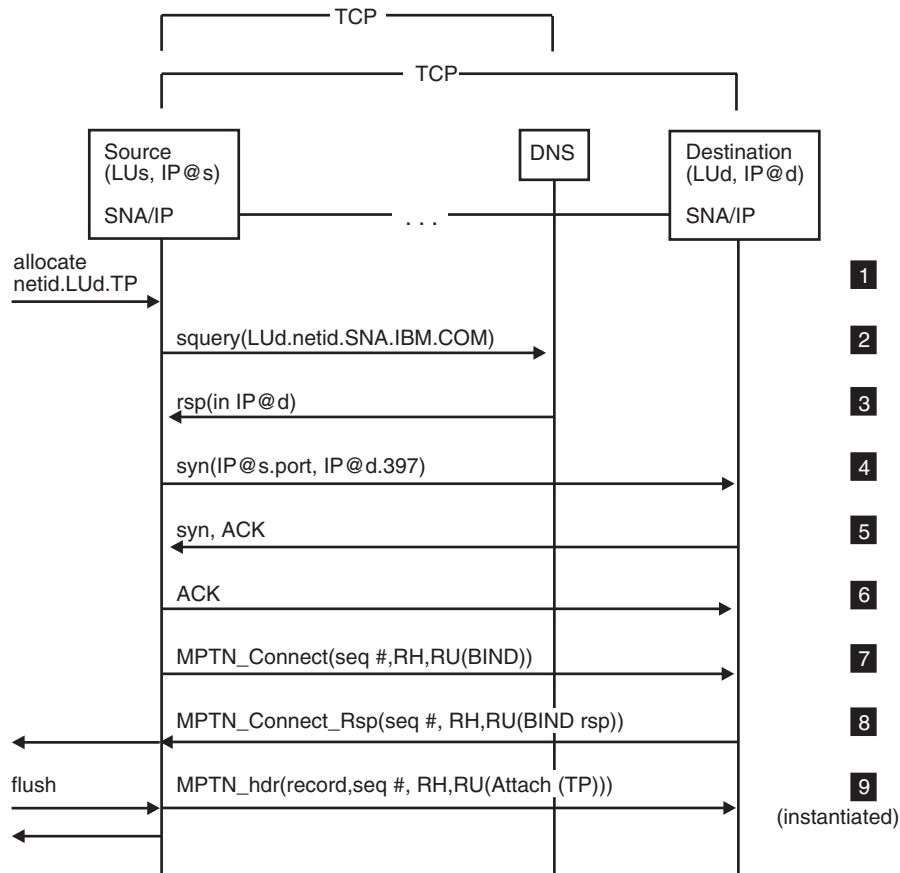


Figure 14. Connection Setup: No Transport Gateway, New Session Required

- 1 The application program requests a conversation with the destination transaction program (netid.LUd.TP). Currently, there are no sessions available between the source and destination logical units; the routing preference table indicates that nonnative (TCP/IP) routing should be tried initially.
- 2 Because the LU name-to-IP address mapping for LUd is not cached, the node converts the LU name into a domain name and then issues a gethostbyname() call. This results in an sqquery flow to the DNS.
- 3 The response provides the IP address for the destination node (netid.LUd).
- 4 This flow requests TCP connection establishment with the partner node using the destination IP address and the MPTN well-known port.
- 5 This flow acknowledges that the TCP connection establishment request has been received by the partner node.
- 6 This flow indicates that the TCP connection is established.
- 7 Next, an MPTN_Connect which contains the names of the logical units

(netid.LUs,netid.LUd) that are to be connected, the SNA BIND image, and other system information is forwarded to the partner over the TCP connection.

- 8** The MPTN component packages the BIND response in a positive MPTN_Connect response indicating that the connection is accepted and sends the MPTN_Connect response to the partner LU. This completes MPTN connection setup. The MPTN connection represents an SNA session.

Receipt of the BIND response causes the attach header to be placed in the send queue of the source LU (LUs).

- 9** When the application program issues a flush, the attach header is sent to the partner LU, as data on the MPTN connection. When the LU receives the data, it instantiates the transaction program.

Note: The flush could be issued immediately after the allocate or issued after some data has been sent. If data was sent prior to the flush, that data is included in the same MPTN record as the attach header.

Connection Setup: No Transport Gateway, Existing Session Available

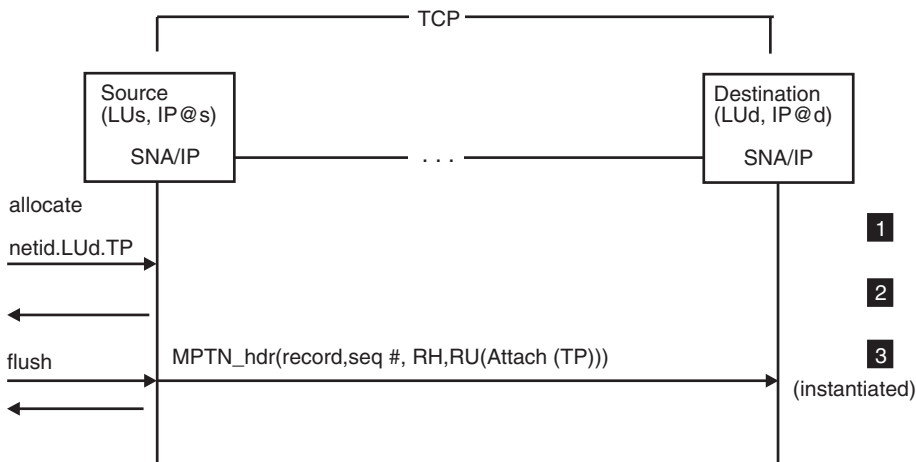


Figure 15. Connection Setup: No Transport Gateway, Existing Session Available

- 1** The application program requests a conversation with the destination logical unit (netid.LUd.TP). The source LU already has a session with the destination LU over an MPTN connection.
- 2** The conversation is assigned to that session. The attach header is placed in the send queue of the source LU (LUs).
- 3** When the application program issues a flush, the attach header is sent to the destination LU (LUd) as data on the MPTN connection. When the attach header is received, LUd instantiates the transaction program.

Connection Setup: From Native SNA Network, Existing Session Available

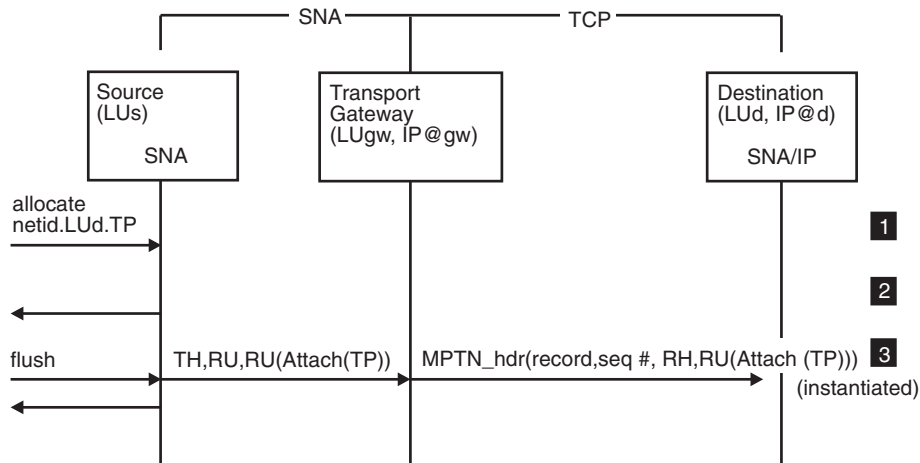


Figure 16. Connection Setup: From Native SNA Network, Existing Session Available

- 1** The application program requests a conversation with the destination transaction program (netid.LUd.TP). A session between the source and destination logical units already exists.
- 2** The conversation is assigned to the current session. The attach header is placed in the send queue of the source LU (LUs).
- 3** When the transaction program issues a flush, the attach header is sent on the session. The transport gateway handles the attach header as data and forwards it over the MPTN connection after packaging it. When the attach header is processed by LUd, LUd instantiates the transaction program.

Connection Setup: No Transport Gateway, Trying Native First, SNA Fails, IP Succeeds

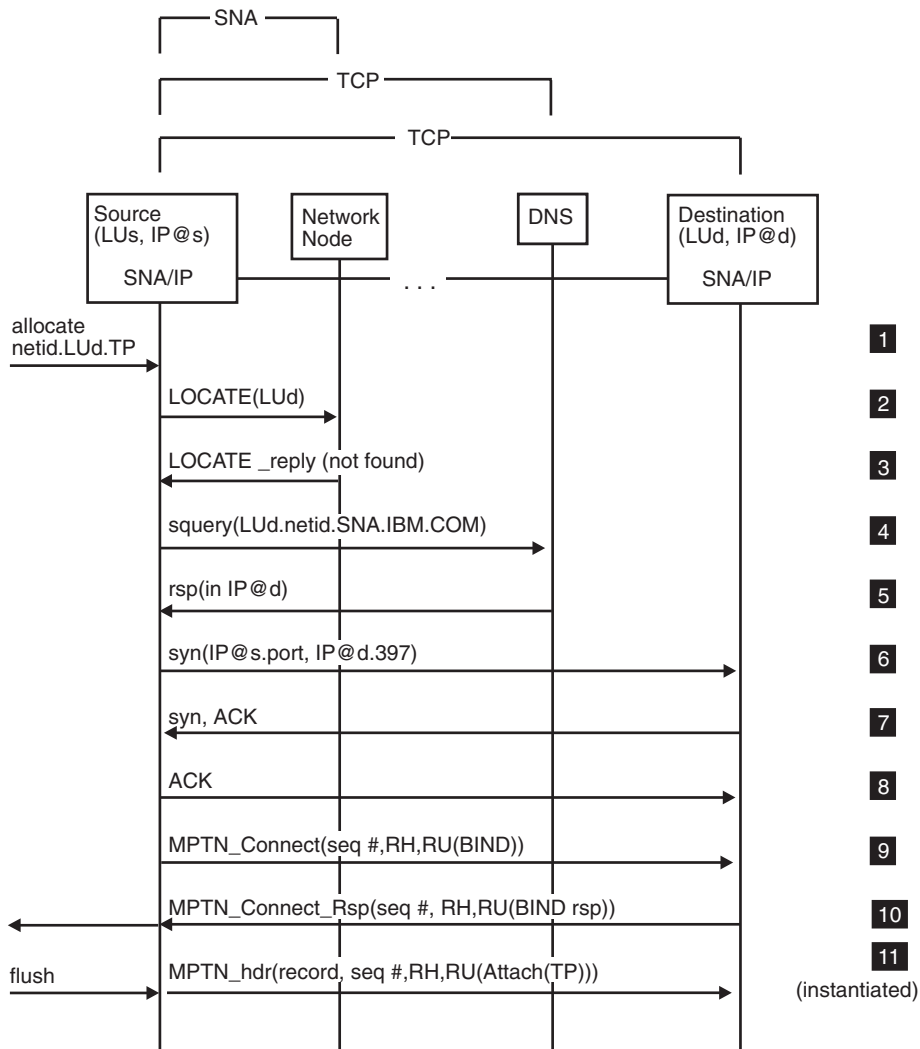


Figure 17. Connection Setup: No Transport Gateway, Trying Native First, SNA Fails, IP Succeeds

- 1 The application program requests a conversation with the destination transaction program (netid.LUd.TP). Currently, there are no sessions available between the source and destination logical units; the routing preference table indicates that native (SNA) routing should be tried initially.
- 2 The source node, an APPN end node, sends an APPN LOCATE for LUd to the serving network node.
- 3 After searching the network, the network node sends a LOCATE_reply indicating that it cannot find LUd because there is no SNA connectivity.
- 4 In this case, the routing preference table indicates to use nonnative

(TCP/IP) connection setup routing if native (SNA) connection setup fails. Therefore, the workstation tries to establish the connection nonnatively.

Because the LU name-to-IP address mapping for LUd is not cached, the node converts the LU name into a domain name and then issues a `gethostbyname()` call. This results in an squery flow to the DNS.

Note: This processing differs in MVS environments. If the host has defined LUd as located in the SNA network and the LOCATE fails (that is, the network node does not find LUd in its network), the host does not try to establish the connection nonnatively. Instead, the connection request fails and the LU fails the allocation request.

- 5** The response provides the IP address for the destination node (`netid.LUd`).
- 6** This flow requests TCP connection establishment with the partner node using the destination IP address and the MPTN well-known port.
- 7** This flow acknowledges that the TCP connection establishment request has been received by the partner node.
- 8** This flow indicates that the TCP connection is established.
- 9** Next, an MPTN_Connect which contains the names of the logical units (`netid.LUs,netid.LUd`) that are to be connected, the SNA BIND image, and other system information is forwarded to the partner over the TCP connection.
- 10** The MPTN component packages the BIND response in a positive MPTN_Connect response indicating that the connection is accepted. The MPTN component sends the MPTN_Connect response to the partner LU. This completes MPTN connection setup. The MPTN connection represents an SNA session.

Receipt of the BIND response causes the attach header to be placed in the send queue of the source LU (LUs).
- 11** When the application program issues a flush, the attach header is sent to LUd, as data on the MPTN connection, where LUd instantiates the transaction program.

Connection Setup: No Transport Gateway, Trying Nonnative First, IP Fails (Name Not in DNS), SNA Succeeds

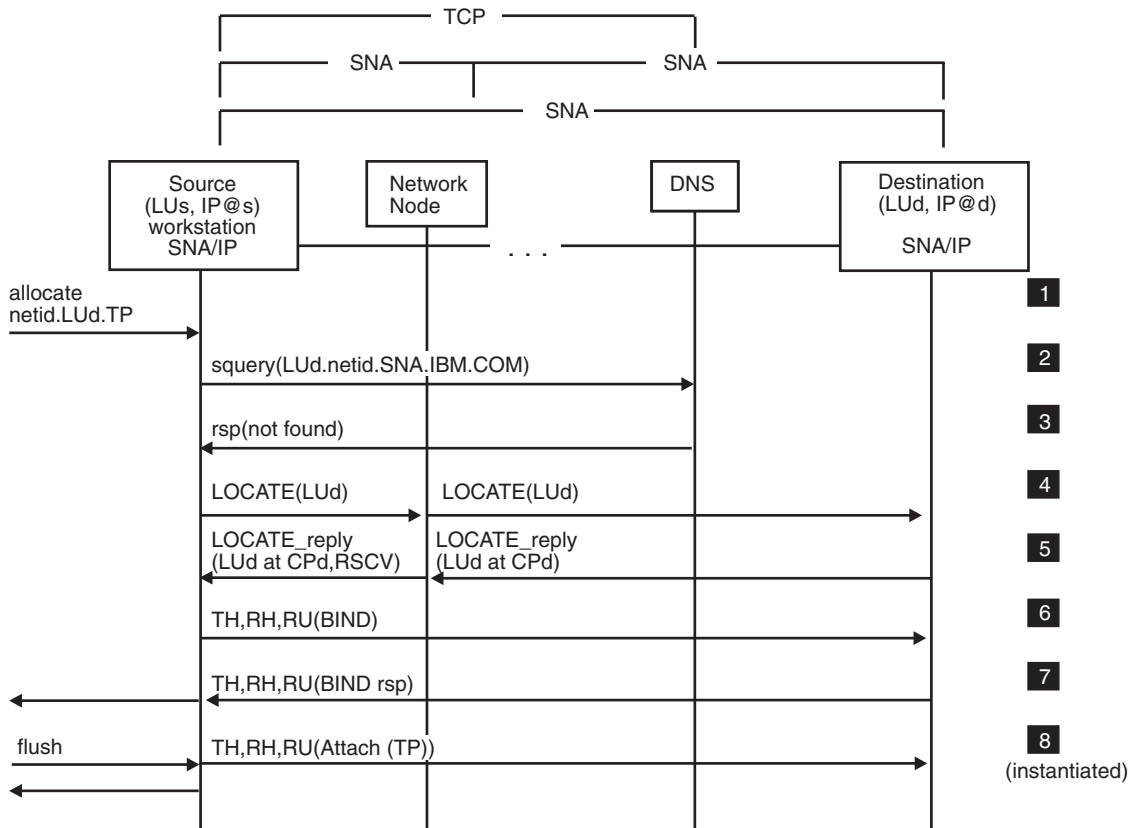


Figure 18. Connection Setup: No Transport Gateway, Trying Nonnative First, IP Fails (Name Not in DNS), SNA Succeeds

- 1 The application program requests a conversation with the destination transaction program (netid.LUd.TP). Currently, there are no sessions available between the source and destination logical units; the routing preference table indicates that nonnative (TCP/IP) routing should be tried initially.
- 2 Because the LU name-to-IP address mapping for LUd is not cached, the node converts the LU name into a domain name and then issues a gethostbyname() call. This results in an sqery flow to the DNS.
- 3 The DNS indicates that the domain name corresponding to LUd is not known. The connection attempt over TCP does not continue. In this case, the routing preference table indicates to use native connection setup routing if nonnative (SNA) connection setup fails. Therefore, the workstation tries to establish the connection natively.

Note: This processing differs in host environments. If the host has defined LUd as located in the IP network and the sqery fails, the host does

not try to establish the connection over SNA. Instead, the connection request fails and the LU fails the allocation request.

- 4** The source node sends a LOCATE request for the destination LU (LUd) to the serving network node. In this configuration, LUd exists on another end node that is served by the same network node that serves the source node. This LOCATE is forwarded to the destination node, where LUd is located.
- 5** The destination node sends a LOCATE_reply that indicates LUd is available on the destination node. When the network node receives this reply, it computes the route to be used and returns that information, in the RSCV, to the requesting workstation.
- 6** LUs sends a BIND to LUd.
- 7** LUd sends a positive BIND response; LUs places the attach header in the send queue.
- 8** When the transaction program issues a flush, LUs sends the attach header on the session. When LUd receives the attach header, it instantiates the transaction program.

Connection Setup: No Transport Gateway, Trying Nonnative First, DNS Not Found

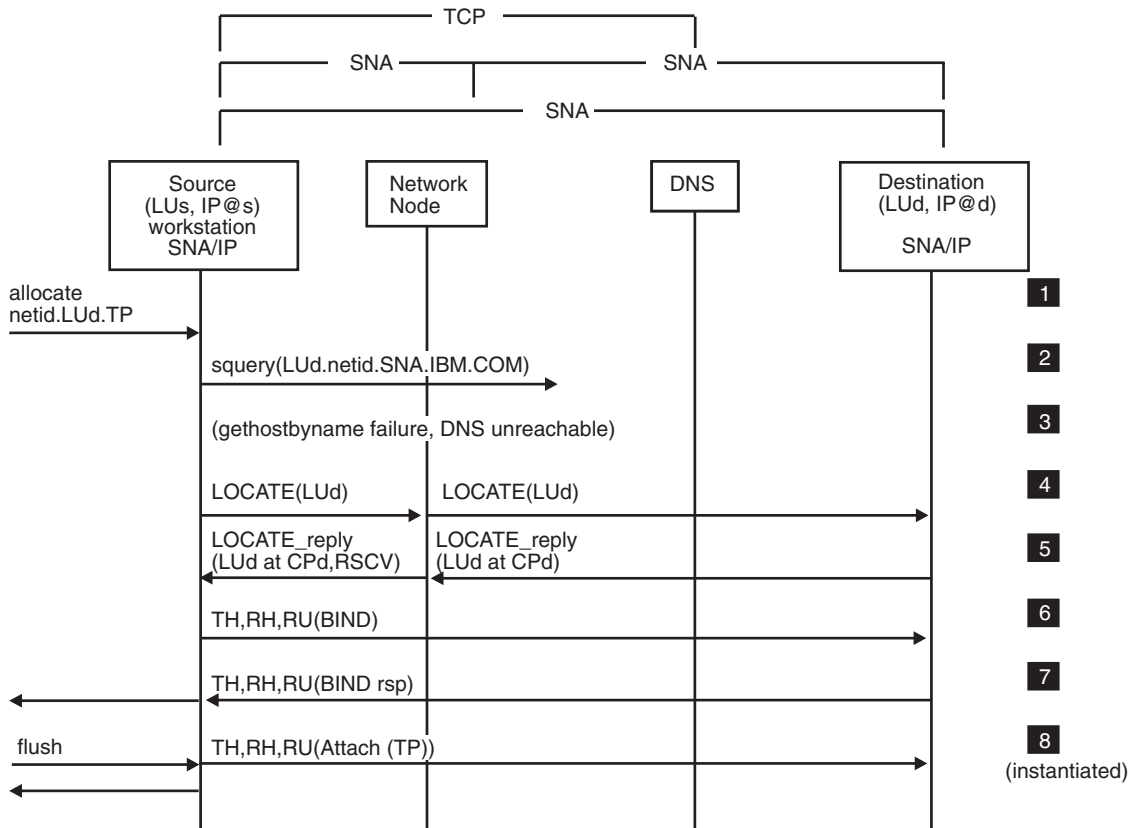


Figure 19. Connection Setup: No Transport Gateway, Trying Nonnative First, DNS Not Found

- 1** The application program requests a conversation with the destination transaction program (netid.LUd.TP). Currently, there are no sessions available between the source and destination logical units; the routing preference table indicates that nonnative (TCP/IP) routing should be tried initially.
- 2** Because the LU name-to-IP address mapping for LUd is not cached, the node converts the LU name into a domain name and then issues a gethostbyname() call. This results in an squery flow to the DNS.
- 3** When no response is received from the DNS, nonnative connection setup fails. In this case, the routing preference table indicates to use native (SNA) connection setup routing if nonnative (TCP/IP) connection setup fails. Therefore, the workstation tries to establish the connection natively.

Note: This processing differs in host environments. If the host has defined the destination node (LUd) as located in the IP network and the

squery fails, the host does not try to establish the connection over SNA. Instead, the connection request fails and the LU fails the allocation request.

- 4** A LOCATE request for LUd is sent to the serving network node. This LOCATE is forwarded to the destination node, where LUd is found.
- 5** The destination node sends a LOCATE_reply that indicates LUd is available on the destination node. When the network node receives this reply, it computes the route to be used and returns that information, in the RSCV, to the requesting workstation.
- 6** LUs sends a BIND to LUd.
- 7** LUd sends a positive BIND response; LUs places the attach header in the send queue.
- 8** When the transaction program issues a flush, LUs sends the attach header on the session. When LUd receives the attach header, it instantiates the transaction program.

Connection Setup: No Transport Gateway, Trying Nonnative First, No IP Connectivity

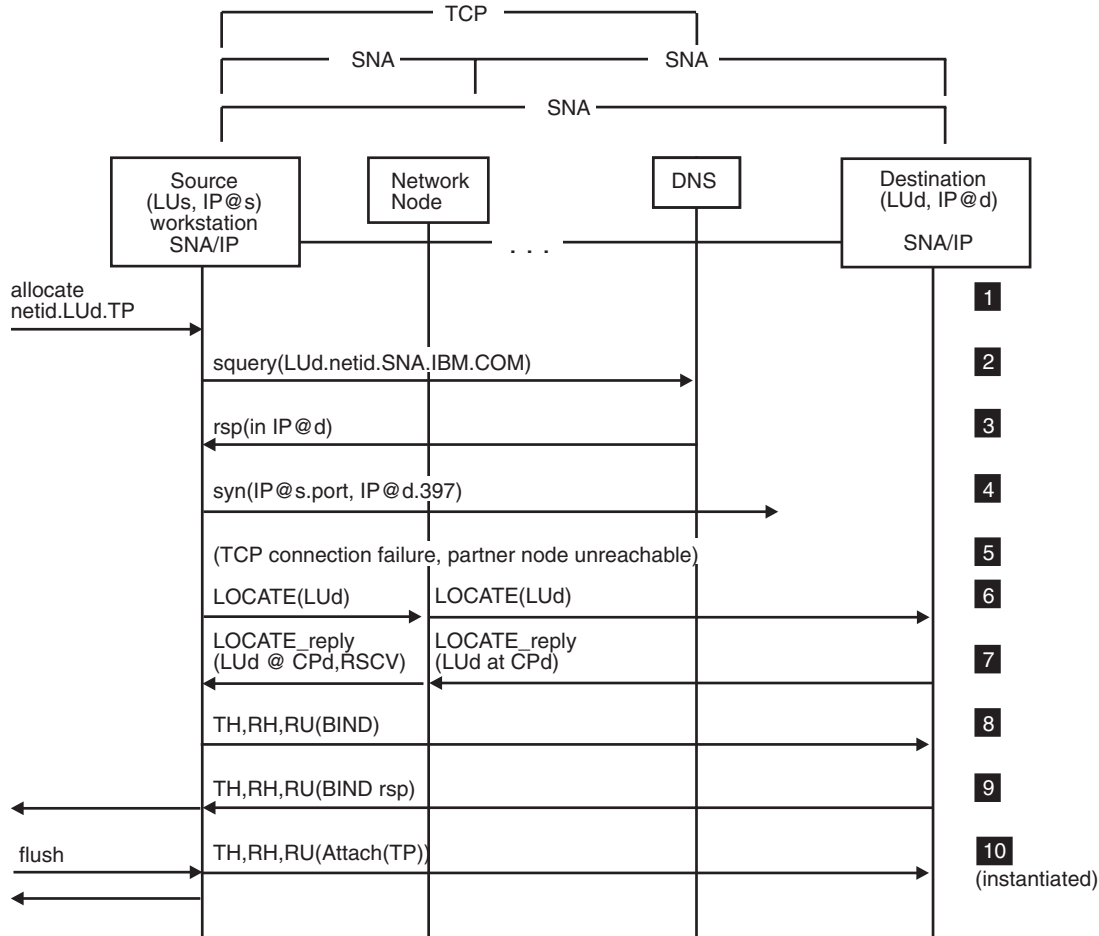


Figure 20. Connection Setup: No Transport Gateway, Trying Nonnative First, No IP Connectivity

- 1** The application program requests a conversation with the destination transaction program (netid.LUd.TP). Currently, there are no sessions available between the source and destination logical units; the routing preference table indicates that nonnative (TCP/IP) routing should be tried initially.
- 2** Because the LU name-to-IP address mapping for LUd is not cached, the node converts the LU name into a domain name and then issues a gethostbyname() call. This results in an squery flow to the DNS.
- 3** The response provides the IP address for the destination logical unit (netid.LUd).
- 4** This flow requests TCP connection establishment with the partner node using the destination IP address and the MPTN well-known port.
- 5** When no acknowledgement is received from the partner node, the nonnative connection fails. In this case, the routing preference table

indicates to use native (SNA) connection setup routing if nonnative (TCP/IP) connection setup fails. Therefore, the workstation tries to establish the connection natively.

Note: This processing differs in host environments. If the host has defined LUd as located in the IP network and the squery fails, the host does not try to establish the connection over SNA. Instead, the connection request fails and the LU fails the allocation request.

- 6** The source node sends a LOCATE request for LUd to the serving network node. This LOCATE is forwarded to the destination node, where the destination LU (LUd) is found.
- 7** The destination node sends a LOCATE_reply that indicates LUd is available on the destination node. When the network node receives this reply, it computes the route to be used and returns that information, in the RSCV, to the requesting workstation.
- 8** LUs sends a BIND to LUd.
- 9** LUd sends a positive BIND response; LUs places the attach header in the send queue.
- 10** When the transaction program issues a flush, LUs sends the attach header on the session. When LUd receives the attach header, it instantiates the transaction program.

Connection Setup: No Transport Gateway, BIND Rejected

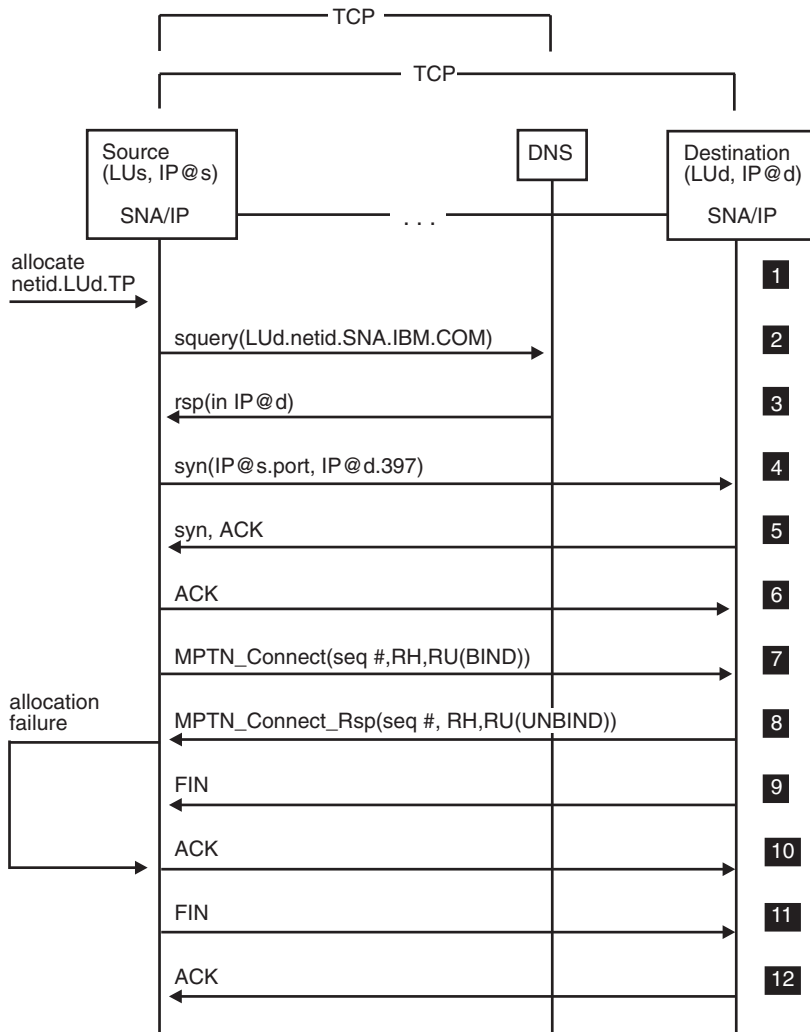


Figure 21. Connection Setup: No Transport Gateway, BIND Rejected

- 1** The application program requests a conversation with the destination transaction program (netid.LUd.TP). Currently, there are no sessions available between the source and destination logical units; the routing preference table indicates that nonnative (TCP/IP) routing should be tried initially.
- 2** Because the LU name-to-IP address mapping for LU_d is not cached, the node converts the LU name into a domain name and then issues a `gethostbyname()` call. This results in an `squery` flow to the DNS.
- 3** The response provides the IP address for the destination LU (netid.LUd).
- 4** This flow requests TCP connection establishment with the partner node using the destination IP address and the MPTN well-known port.
- 5** This flow acknowledges that the TCP connection establishment request has been received by the partner node.

- 6** This flow indicates that the TCP connection is established.
- 7** Next, an MPTN_Connect which contains the names of the logical units (netid.LUs,netid.LUd) that are to be connected, the SNA BIND image, and other system information is forwarded to the partner over the TCP connection.
- 8** The destination node (LUd) does not like the BIND that it receives and rejects it with an UNBIND. The UNBIND is packaged in a negative MPTN_Connect response, indicating that the connection is rejected. The MPTN_Connect response is sent to the partner LU. The UNBIND is forwarded to LUs, the session connection fails, and LUs rejects the allocation request.
- 9** When the MPTN connection request is rejected, the TCP connection to the partner is closed. TCP in the destination node sends a packet with the FIN bit set, thus taking down its half of the full-duplex connection.
- 10** TCP in the source node acknowledges this takedown by sending a packet with the ACK bit set.
- 11** After the negative MPTN_Connect response is received, and the UNBIND is forwarded to LUs, the MPTN component in the source node takes down the TCP connection to the destination node.
- 12** TCP in the destination node acknowledges this takedown by sending a packet with the ACK bit set. Connection takedown is now complete.

Connection Setup: From Native SNA Subarea Network, New Session Required

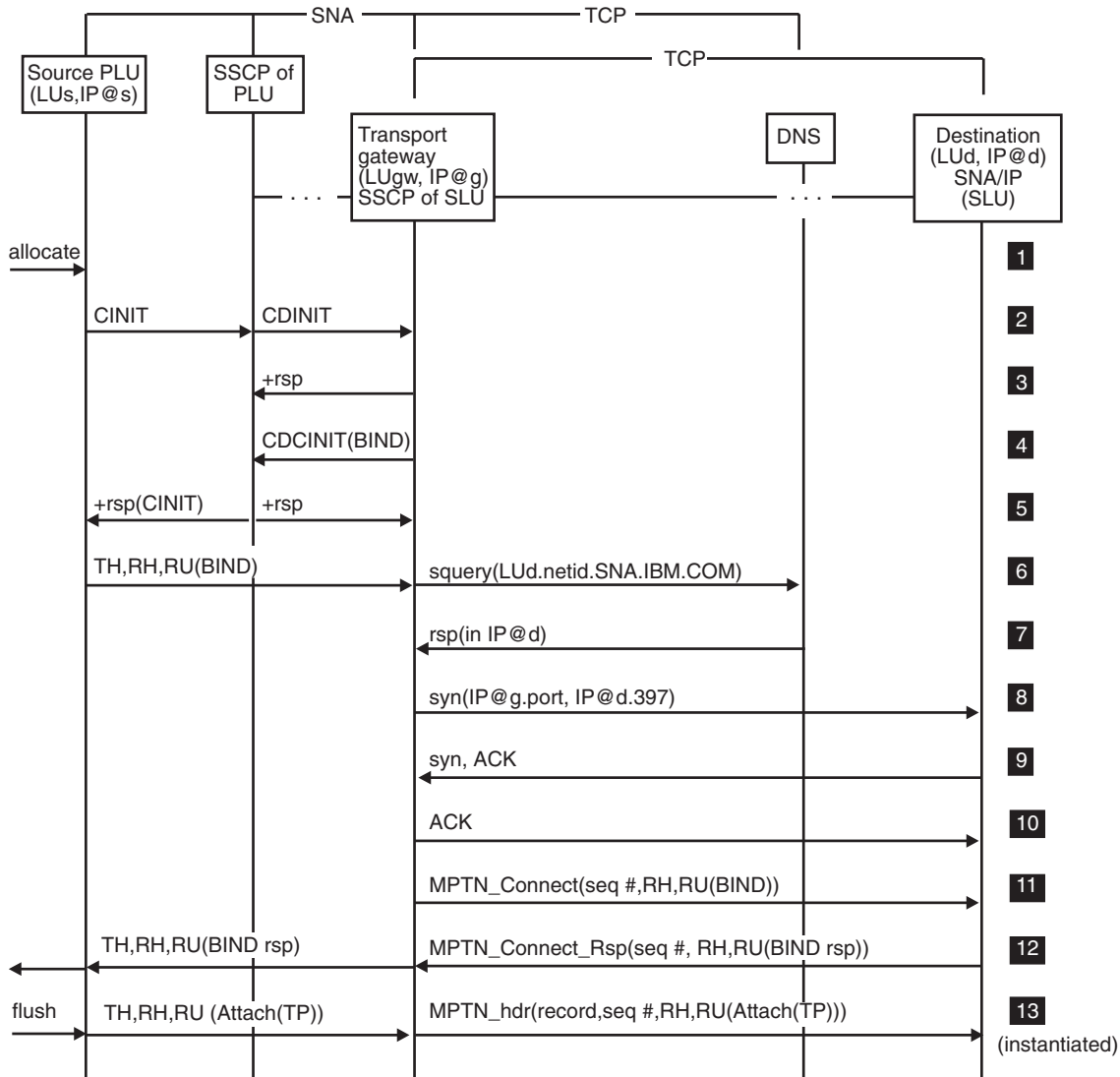


Figure 22. Connection Setup: From Native SNA Subarea Network, New Session Required

- 1** The application program requests a conversation with the destination transaction program (netid.LUd.TP). Currently, there are no sessions available and the source LU (LUs) initiates session establishment of an LU 6.2 session through the SSCP using SNA protocols.
- 2** The source LU (LUs) sends a CINIT to its SSCP. The SSCP sends a CDINIT to the SSCP of the SLU (the transport gateway). The gateway owns all logical units that are accessed through the IP network.
- 3** The gateway sends a response to the CDINIT flow.
- 4** The gateway sends a CDCINIT with the BIND image that is used by the PLU (LUs) to establish the session with the SLU (LUd).

- 5** The SSCP of the PLU:
 - Responds to the CINIT that was sent by the PLU (step **2**) by sending the BIND image that the PLU should use
 - Responds to the CDCINIT that was sent by the SSCP of the SLU (step **4**).
- 6** The source LU (LUs) sends the BIND to the transport gateway. The gateway recognizes the destination logical unit (netid.LUd) as part of the IP network and converts the LU name into a domain name and then issues a gethostbyname() call. This results in an squery flow to the DNS.
- 7** The response contains the IP address for the destination LU (netid.LUd).
- 8** This flow requests TCP connection establishment with the partner node using the destination IP address and the MPTN well-known port.
- 9** This flow acknowledges that the TCP connection establishment request has been received by the partner node.
- 10** This flow indicates that the TCP connection is established.
- 11** Next, an MPTN_Connect which contains the names of the logical units (netid.LUs,netid.LUd) that are to be connected, the SNA BIND image, and other system information is forwarded to the partner LU over the TCP connection.
- 12** The MPTN connection setup completes with the MPTN_Connect response. The MPTN_Connect response includes the BIND response. The transport gateway converts this BIND response into a native (SNA) BIND response. This native BIND response traverses the SNA network and is routed to the PLU (LUs). The MPTN connection represents an SNA session.

Receipt of the BIND response causes the attach header to be placed on the send queue of the source LU (LUs).
- 13** When the application program issues a flush, the attach header is sent on the SNA session to the transport gateway. The transport gateway handles the attach header as data and forwards it on the MPTN connection using MPTN headers. When the message arrives at the destination node, it is passed to LUd. LUd then instantiates the transaction program.

Connection Setup: From IP Network to native SNA APPN Network, New Session Required

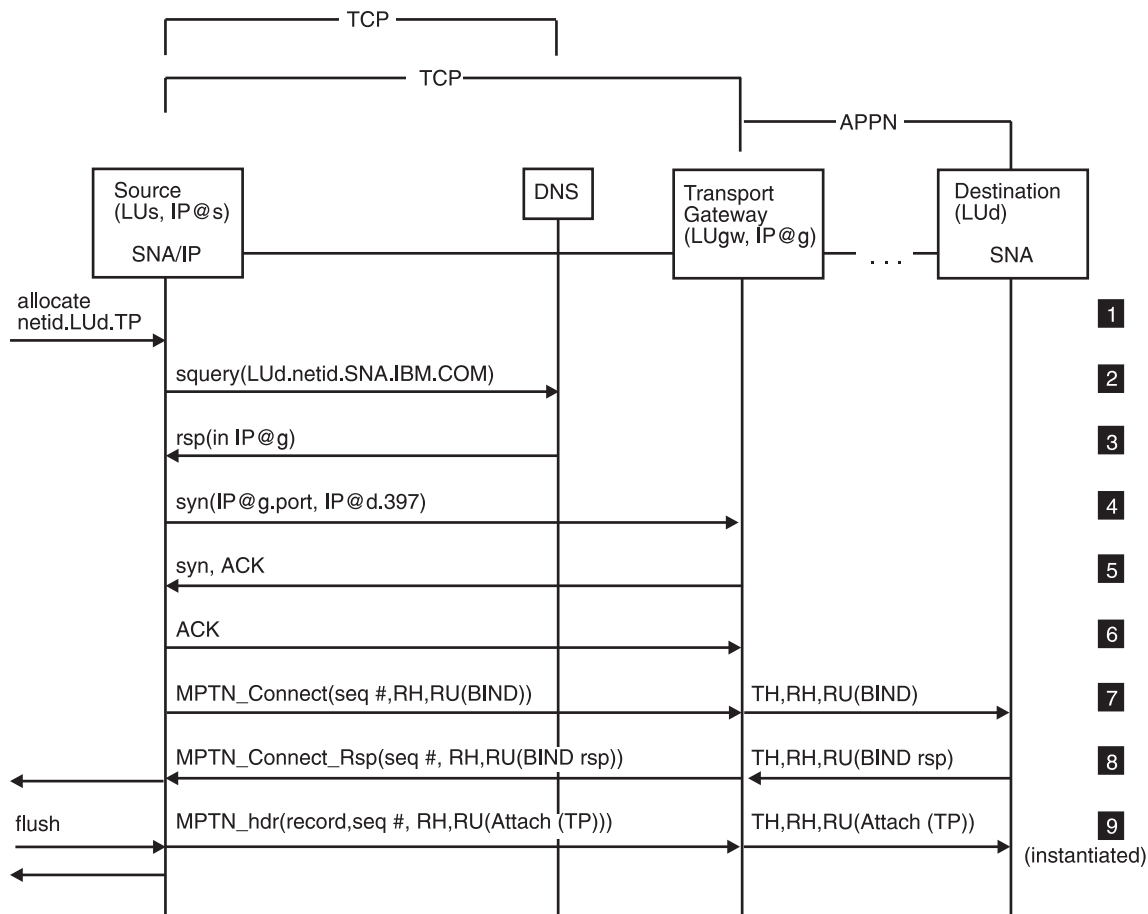


Figure 23. Connection Setup: From IP Network to native SNA APPN Network, New Session Required

- 1** The application program requests a conversation with the destination transaction program (`netid.LUd.TP`). Currently, there are no sessions available between the source and destination logical units. The routing preference table indicates that nonnative (TCP/IP) routing should be tried initially.
- 2** Because the LU name-to-IP address mapping for LU_d is not cached, the node converts the LU name into a domain name and then issues a `gethostbyname()` call. This results in an `squery` flow to the DNS.
- 3** The response provides the IP address for the transport gateway.
- 4** This flow requests TCP connection establishment with the transport gateway using the its IP address and the MPTN well-known port.
- 5** This flow acknowledges that the TCP connection establishment request has been received by the transport gateway.
- 6** This flow indicates that the TCP connection is established.
- 7** Next, an `MPTN_Connect` which contains the names of the logical units (`netid.LUs, netid.LUd`) that are to be connected, the SNA BIND image, and

other system information is forwarded to the transport gateway over the TCP connection. The transport gateway extracts the BIND, converts it, and sends it to LUd using SNA protocols.

- 8** The transport gateway receives the BIND response from LUd on the SNA network, converts it, and includes it in the MPTN_Connect response. The MPTN_Connect response is sent to the partner node over the TCP connection. This completes MPTN connection setup. The MPTN connection represents an SNA session.

Receipt of the BIND response causes the attach header to be placed in the send queue of the source LU (LUs).

- 9** When the transaction program issues a flush, the data that is ready to be sent in addition to the attach header is packaged as MPTN data. At the transport gateway, the attach header is unpackaged and sent on the SNA session to LUd. The destination LU instantiates the transaction program.

Connection Setup: From IP Network, Existing Session Available

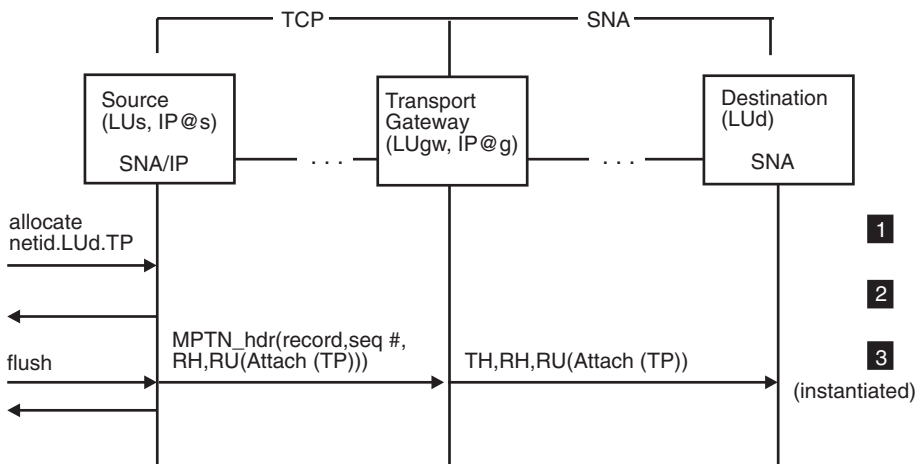


Figure 24. Connection Setup: From IP Network, Existing Session Available

- 1** The application program requests a conversation with the destination transaction program (netid.LUd.TP). The source LU already has a session with the destination LU over an MPTN connection.
- 2** The conversation is assigned to the current session. The attach header is placed in the send queue of the source LU (LUs).
- 3** When the transaction program issues a flush, the data that is ready to be sent, in addition to the attach header, is packaged as MPTN data. At the transport gateway, the attach header is unpackaged and sent on the SNA session to LUd. The destination LU (LUd) then instantiates the transaction program.

Send Data: Normal LU 6.2 Buffering

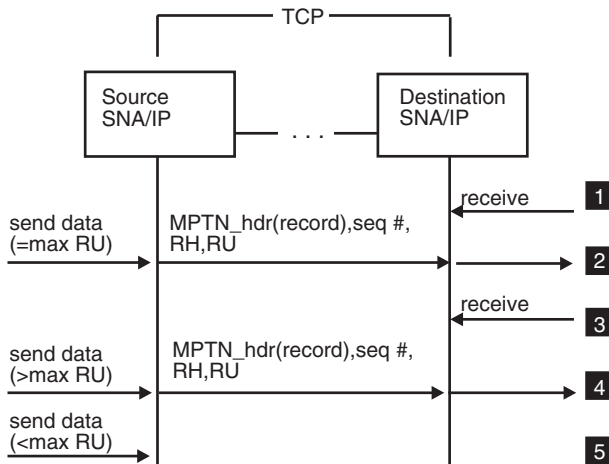


Figure 25. Send Data: Normal LU 6.2 Buffering

- 1** The destination transaction program is ready to receive data.
- 2** The application program sends an amount of data that is equal to one maximum RU.

Each RU is sent as an MPTN record. The data is passed to the destination LU. The destination LU recognizes the SNA headers and passes the data to the transaction program to fulfill the outstanding receive.
- Note:** Because TCP is a stream-oriented protocol, TCP does not recognize MPTN's packet boundaries and the MPTN record may be shipped in one or more TCP packets.
- 3** The destination transaction program now posts another receive.
- 4** The next time the application program sends data, the LU sends as many RUs as can be filled by the data. For example, if the maximum RU size is 100 bytes and the application program sends 250 bytes of data, the LU sends two RUs (100 bytes of data in each). The LU stores the remaining 50 bytes in the buffer internally.
- 5** The application program sends less data than one maximum RU and does not issue a flush, the LU stores the data in a buffer and does not send it. For example, if the application program sends 25 bytes of data; the LU adds it to the 50 bytes of data from step **4** and now has 75 bytes of data in the buffer. The data is stored in the buffer until the application program issues a flush, the next data send of at least 25 bytes, or the application program terminates.

Send Data: Sending Data with Flushing

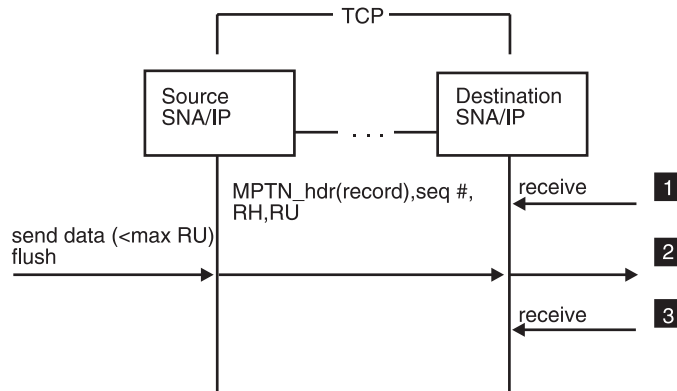


Figure 26. Send Data: Sending Data with Flushing

- 1** The destination transaction program is ready to receive data.
- 2** The application program sends less data than the maximum RU size. Because the application issued a flush as well, the LU packages and sends the data as an RU. The RU is sent on the MPTN connection as an MPTN record and passed to the LU. The LU extracts the data and passes it to the transaction program.
- 3** The destination transaction program now posts another receive.

Send Data: Segmented Data from Native SNA

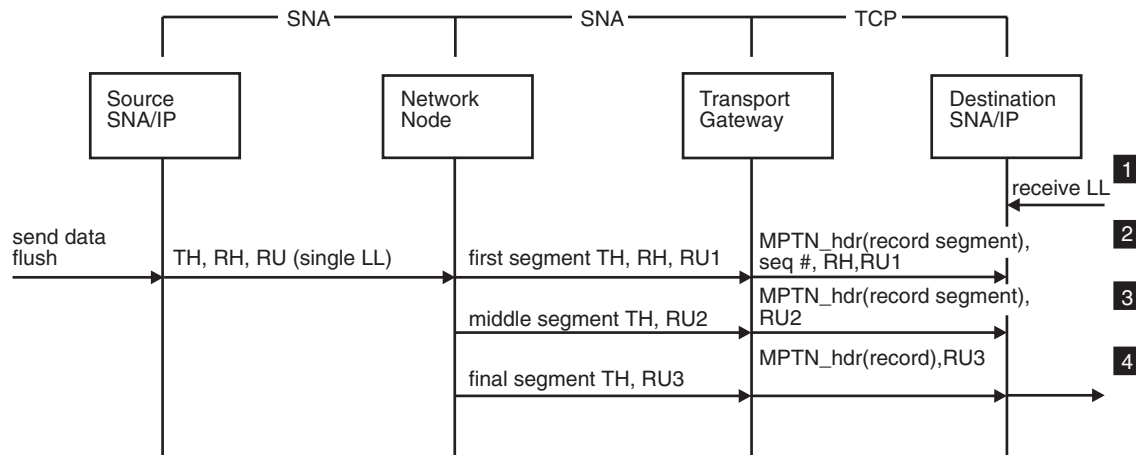


Figure 27. Send Data: Segmented Data from Native SNA

- 1** The destination transaction program posts a receive for a single logical length (LL) of data.
- 2** The source application program sends and flushes a record. This record is

packaged as a single LL. When the RU arrives at the network node, it is broken into three segments because of the outgoing link constraints. When the transport gateway receives the first segment, it sends the sequence number, the RH, and the RU as data on the MPTN connection. Because the RU that arrived was segmented, the transport gateway marks it as an MPTN record segment rather than a record. When the destination node receives the packet, it holds it for reassembly before being passing it to the application program.

- 3 When the transport gateway receives the second segment, it sends the RU segment (there is no RH) as a record segment on the MPTN connection. The destination node continues the reassembly, but again, does not pass it to the application.
- 4 When the transport gateway receives the third segment, it sends the segment as an MPTN record on the MPTN connection. When the destination node sees the record indication, it knows that reassembly is complete and the LU passes the next LL to the application program.

Send Data: Expedited Data, Datagram Copy Arrives First

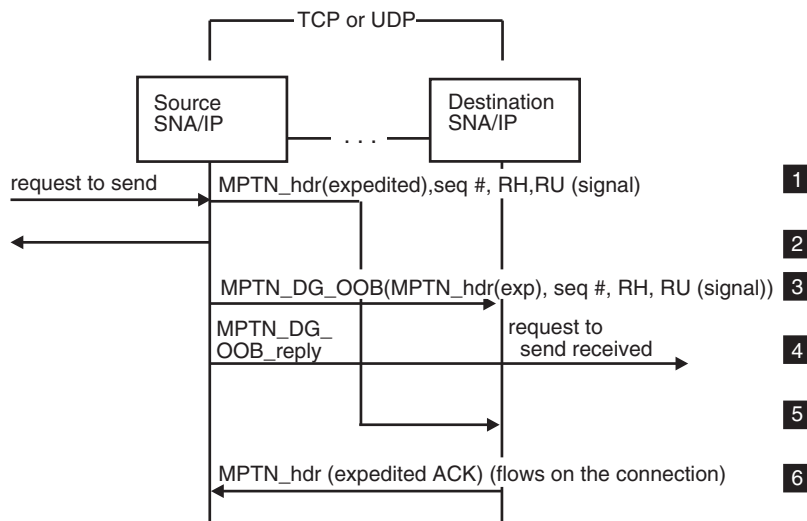


Figure 28. Send Data: Expedited Data, Datagram Copy Arrives First

- 1 When the source application program issues a request to send, the LU creates a signal command, which is an expedited flow. The RU is packaged as an MPTN expedited record and is sent on the TCP connection. At the same time, the MPTN component sets a timer to wait for an expedited acknowledgement on the connection.

Note: This timer is specified in OS/2 environments by the UNSENT_DG_RETRY_SECS environment variable; in MVS environments, this timer is specified on the EXTIMER operand on the TCP/IP major node. For some reason, such as congestion on the TCP connection, delivery of this message is delayed.

- 2 The source LU immediately responds to the request to send.

- 3** When the timer that is awaiting an acknowledgement pops at the source node, it sends out a UDP datagram that contains an MPTN out of band datagram using UDP. The data carried in the MPTN_DG_OOB is the same signal command as was sent in the expedited record. The MPTN_OOB datagram is received at the destination node.

The MPTN component also sets a datagram retransmission timer to resend the datagram if no reply is received.

Note: This timer is specified in OS/2 environments by the UNACKED_DG_RETRY_SECS environment variable; in MVS environments, this timer is specified on the DGTIMER operand on the TCP/IP major node.

- 4** When the MPTN component receives the MPTN_DG_OOB, it passes the expedited signal command to the LU. The MPTN component also replies to the MPTN_DG_OOB. When this arrives at the source node, the MPTN component recognizes that the expedited data has been delivered and cancels the datagram retransmission timer.

The signal command is passed to the LU. The LU notifies the application program of the arrival of the expedited data by indicating a request to send was received.

- 5** The expedited record arrives at the destination node on the TCP connection. By comparing the MPTN sequence number, the MPTN component recognizes this as a duplicate record and does not forward it to the LU.

- 6** The destination MPTN component does, however, return an MPTN expedited acknowledgement on the TCP connection. When the source node receives the acknowledgement, it recognizes it as a duplicate acknowledgement and ignores it.

Send Data: Expedited Data through a Transport Gateway

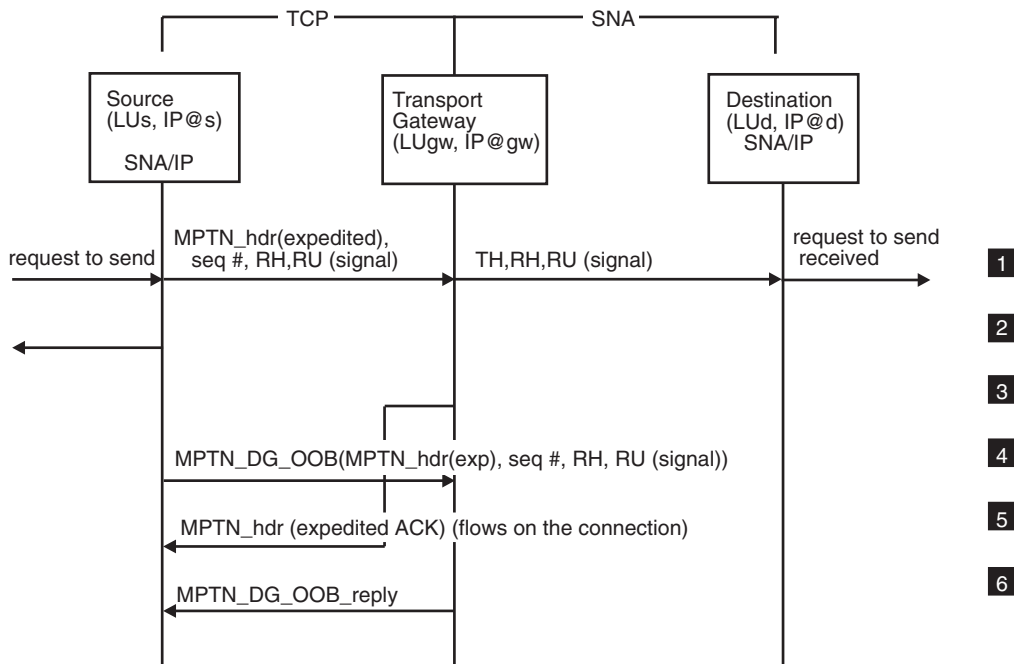


Figure 29. Send Data: Expedited Data through a Transport Gateway

- 1 When the source application program issues a request to send, the source LU (LUs) creates a signal command, which is an expedited flow. The RU is packaged as an MPTN expedited record and is sent on the TCP connection. At the same time, the MPTN component sets a timer to wait for an expedited acknowledgement.

Note: This timer is specified in OS/2 environments by the UNSENT_DG_RETRY_SECS environment variable; in MVS environments, this timer is specified on the EXTIMER operand on the TCP/IP major node. When the MPTN expedited record arrives at the transport gateway, it extracts the signal and sends it as expedited on the SNA session. The destination node (LUd) notifies the application of the arrival of the signal command by indicating that a request to send was received.

- 2 LUs immediately responds to the request to send.
- 3 When the transport gateway receives the MPTN expedited record on the TCP connection, it returns an expedited acknowledgement on the TCP connection.
- 4 Before the acknowledgement arrives at the source node, the timer awaiting the acknowledgement pops and the MPTN component sends out an MPTN out of band datagram using UDP. The data carried in the MPTN_DG_OOB is the same signal as was sent in the expedited record. The MPTN component also sets a datagram retransmission timer to resend the datagram if no reply is received.

The transport gateway receives the MPTN_DG_OOB. By comparing the MPTN sequence number, the gateway recognizes the data as duplicate information and does not forward the signal command across the SNA connection.

- 5 The MPTN expedited record acknowledgement arrives on the connection at the source node; the retransmission timer is ended.
- 6 When the transport gateway receives the MPTN_DG_OOB, it sends an MPTN_DG_OOB reply datagram to the source node. When the reply datagram arrives at the source node, the MPTN component recognizes it as a duplicate reply for the expedited data and discards it.

Send Data: Expedited Data, Connection Copy Acknowledgement Arrives Before Datagram Copy Sent

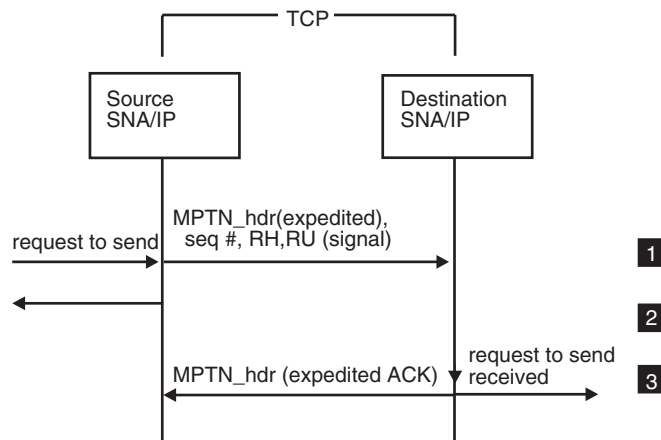


Figure 30. Send Data: Expedited Data, Connection Copy Acknowledgement Arrives Before Datagram Copy Sent

- 1 When the source application program issues a request to send, the LU creates a signal command, which is an expedited flow. The RU is packaged as an MPTN expedited record and is sent on the TCP connection. At the same time, the MPTN component sets a timer to wait for an expedited acknowledgement on the connection.

Note: This timer is specified in OS/2 environments by the UNSENT_DG_RETRY_SECS environment variable; in MVS environments, this timer is specified on the EXTIMER operand on the TCP/IP major node.

- 2 The source LU immediately responds to the request to send.
- 3 The destination node receives the MPTN expedited record and passes the signal command to the LU. It also returns an MPTN expedited acknowledgement on the TCP connection. The destination LU notifies the application that the expedited data has arrived by indicating that a request to send was received. When the expedited acknowledgement arrives at the source node, the timer is cleared.

Connection Takedown: Deallocate, No Transport Gateway

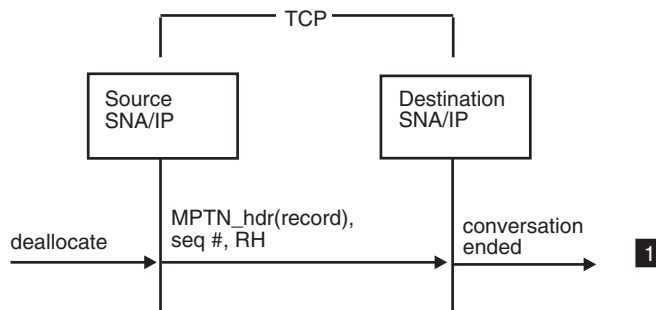


Figure 31. Connection Takedown: Deallocate, No Transport Gateway

- 1** When an APPC application program issues a deallocate, the LU creates an RU with no data and sets the conditional end bracket (CEB) bit in the RH. This RU is packaged as an MPTN record and is sent to the destination node. The destination node passes the RU to the LU; this indicates to the transaction program that the conversation has ended. The MPTN connection remains intact.

Connection Takedown: UNBIND, No Transport Gateway, AnyNet Environments

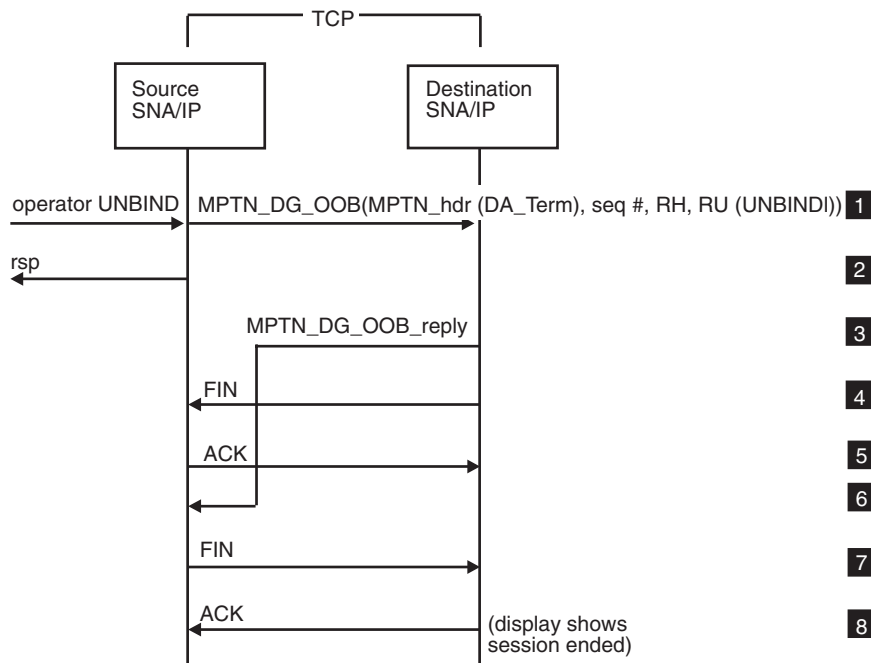


Figure 32. Connection Takedown: UNBIND, No Transport Gateway, AnyNet Environments

- 1** When the operator requests that an UNBIND for a session be issued, the MPTN component packages the UNBIND with an MPTN duplex-abortive termination header. This header indicates that both halves of the connection are to come down and that the connection is to be terminated without waiting for all data to be delivered. The UNBIND and the header are sent as an MPTN out of band datagram using UDP. As with any OOB datagram, a retry timer is set to await the reply and, if the reply is not received before the timer pops, the datagram is re-sent.
- 2** The MPTN component immediately returns an UNBIND response to the source LU.
- 3** When the destination node receives the OOB datagram, it immediately responds with an MPTN_DG_OOB reply over UDP.
- 4** After sending the MPTN_DG_OOB reply, the MPTN component in the destination node closes the TCP connection to the partner. TCP in the destination node sends a packet with the FIN bit set, thus taking down its half of the full-duplex connection.
- 5** TCP in the source node acknowledges this takedown by sending a packet with the ACK bit set.
- 6** When the source node receives the reply, it cancels the retransmission timer and begins to take down the underlying TCP connection.
- 7** The MPTN component in the source node takes down the TCP connection to the destination node.

- 8** TCP in the destination node acknowledges this takedown by sending a packet with the ACK bit set. Connection takedown is now complete.

Connection Takedown: UNBIND, No Transport Gateway, AnyNet/2 Environments

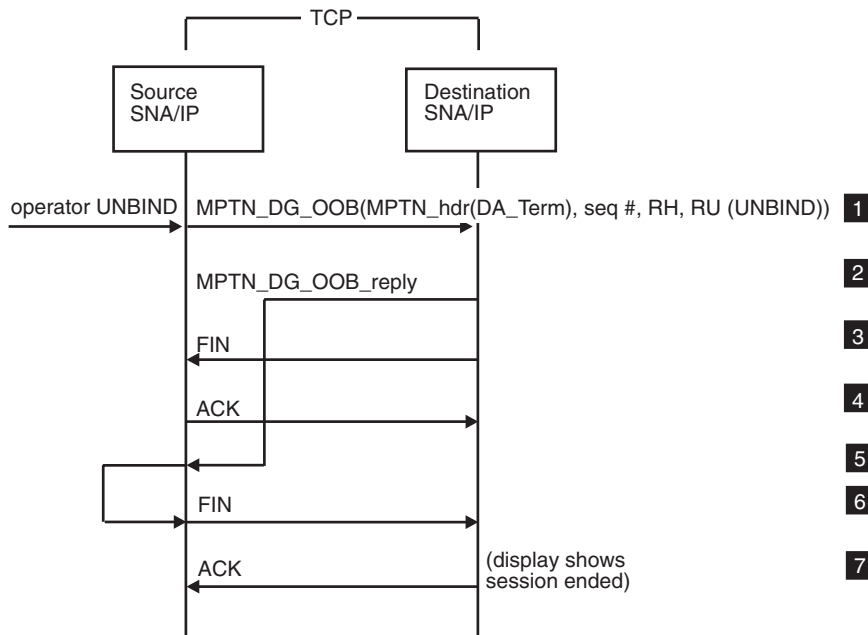


Figure 33. Connection Takedown: UNBIND, No Transport Gateway, AnyNet/2 Environments

- 1** When the operator requests that an UNBIND for a session be issued, the MPTN component packages the UNBIND with an MPTN duplex-abortive termination header. This header indicates that both halves of the connection are to come down and that the connection is to be terminated without waiting for all data to be delivered. The UNBIND and the header are sent as an MPTN out of band datagram using UDP. As with any OOB datagram, a retry timer is set to await the reply and, if the reply is not received, the OOB datagram is re-sent.
- 2** When the destination node receives the MPTN_DG_OOB datagram, the MPTN component performs internal cleanup and passes the UNBIND to the destination LU. After sending the UNBIND to the LU, the MPTN component sends a response to the OOB datagram over UDP.
- 3** After sending the MPTN_DG_OOB reply, the MPTN component in the destination node closes the TCP connection to the partner. TCP in the destination node sends a packet with the FIN bit set, thus taking down its half of the full-duplex connection.
- 4** TCP in the source node acknowledges this takedown by sending a packet with the ACK bit set.
- 5** When the source node receives the reply, it cancels the retransmission timer and informs the source LU that the partner has taken the session down and then begins to take down the underlying TCP connection.

- 6** The MPTN component in the source node takes down the TCP connection to the destination node.
- 7** TCP in the destination node acknowledges this takedown by sending a packet with the ACK bit set. Connection takedown is now complete.

Connection Takedown: UNBIND through a Transport Gateway

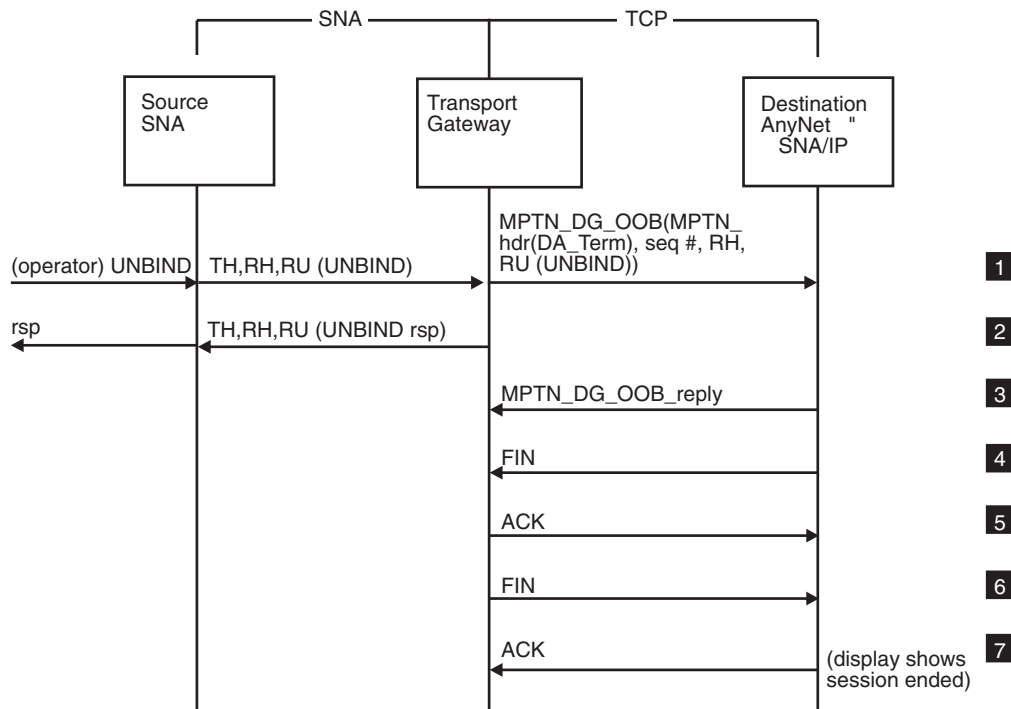


Figure 34. Connection Takedown: UNBIND through a Transport Gateway

- 1** When the operator requests that an UNBIND be issued for a session, the LU creates an UNBIND and sends it out as an expedited message on the SNA connection. When the transport gateway receives the UNBIND, it packages the UNBIND with an MPTN duplex-abortive termination header. This header indicates that both halves of the connection are to come down and that the connection is to be terminated without waiting for all data to be delivered. The UNBIND and the header are sent as an MPTN out of band datagram using UDP. As with any OOB datagram, a retry timer is set to await the reply and, if the reply is not received, resends the OOB datagram.
- 2** The transport gateway immediately returns an UNBIND response. When the source node receives this response, it completes the takedown process from its point of view.
- 3** When the destination node receives the OOB datagram, it immediately replies with a datagram reply over UDP. When the transport gateway receives the reply, it cancels the retransmission mechanism and begins to take down the underlying TCP connection.
- 4** After sending the MPTN_DG_OOB reply, the MPTN component in the destination node closes the TCP connection to the transport gateway. TCP

in the destination node sends a packet with the FIN bit set, thus taking down its half of the full-duplex connection.

- 5** TCP in the transport gateway acknowledges this takedown by sending a packet with the ACK bit set.
- 6** When the MPTN_DG_OOB reply is received, the MPTN component in the transport gateway takes down the TCP connection to the destination node.
- 7** TCP in the destination node acknowledges this takedown by sending a packet with the ACK bit set. Connection takedown is now complete.

Connection Takedown: Outage in the SNA Network

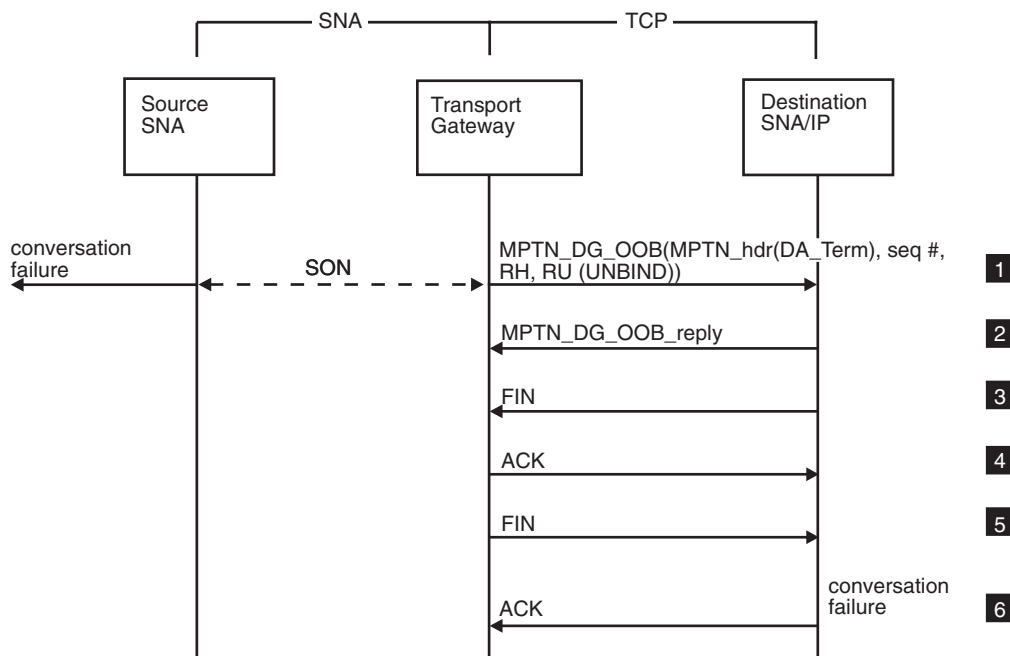


Figure 35. Connection Takedown: Outage in the SNA Network

- 1** When a session fails with the SNA network, session outage notification is delivered to both the source node and the transport gateway. At the source node, the LU reflects the failure to the application program as a conversation failure. At the transport gateway, either an UNBIND arrives or the notification triggers the creation of an appropriate UNBIND. This UNBIND is packaged with an MPTN duplex-abortive termination header. This header indicates that both halves of the connection are to come down and that the connection is to be terminated without waiting for all data to be delivered. The UNBIND and the header are sent as an MPTN out of band datagram using UDP. As with any OOB datagram, a retry timer is set to await the reply and, if the reply is not received, resends the OOB datagram.
- 2** When the destination node receives the OOB datagram, it immediately replies with a datagram reply over UDP. When the transport gateway receives the reply, it cancels the retransmission mechanism and begins to take down the underlying TCP connection.

- 3 After sending the MPTN_DG_OOB reply, the MPTN component in the destination node closes the TCP connection to the partner. TCP in the destination node sends a packet with the FIN bit set, thus taking down its half of the full-duplex connection.
- 4 TCP in the source node acknowledges this takedown by sending a packet with the ACK bit set.
- 5 When the MPTN_DG_OOB reply is received, the MPTN component in the transport gateway takes down the TCP connection to the destination node.
- 6 TCP in the destination node acknowledges this takedown by sending a packet with the ACK bit set. Connection takedown is now complete.

Connection Takedown: Deallocate through a Transport Gateway

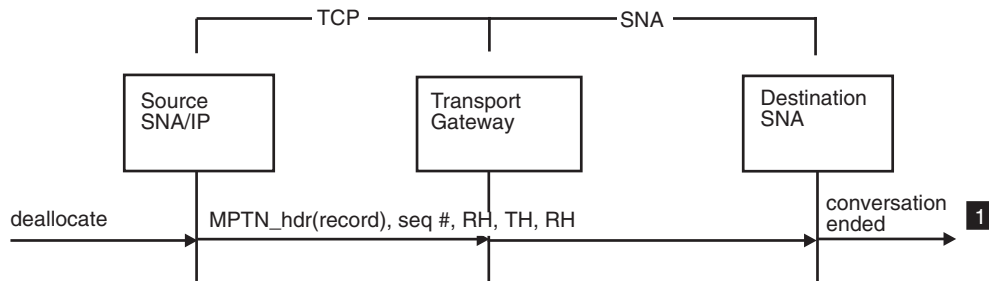


Figure 36. Connection Takedown: Deallocate through a Transport Gateway

- 1 When an APPC application program issues a deallocate, the LU creates an RU with no data and sets the conditional end bracket (CEB) bit in the RH. This RU is packaged as an MPTN record and is sent to the destination node. When the MPTN record arrives at the transport gateway, the RU with the CEB set is extracted and sent over the SNA connection to the destination node. When the LU receive this RU, it indicates to the transaction program that the conversation has ended. Both the MPTN connection and the SNA session remain intact.

Connection Takedown: Outage in an IP Network

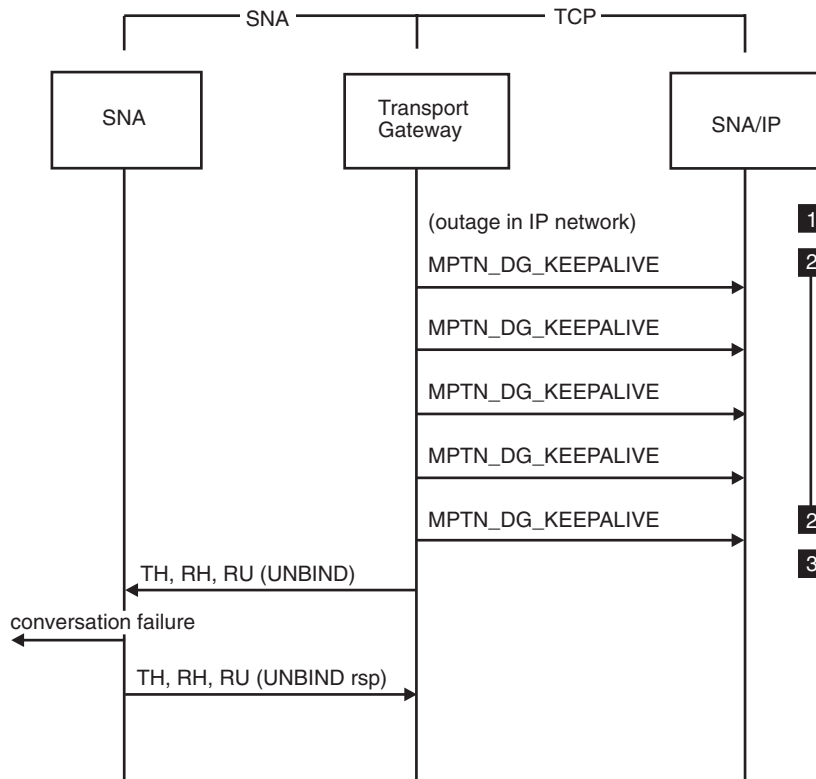


Figure 37. Connection Takedown: Outage in an IP Network

APPC over TCP/IP Users: This flow does not apply to APPC over TCP/IP.

- 1** There is an outage in the IP network. This could be because:
 - SNA over TCP/IP failure
 - An intermediate IP router failure
 - Link failure, or any other problem that would prevent traffic from flowing between the SNA over TCP/IP node and the transport gateway.
- 2** The transport gateway detects inactivity on one or more connections between itself and the SNA over TCP/IP node and starts sending KEEPALIVE datagrams. After 5 datagrams are sent and no response is received, the transport gateway concludes that the partner cannot be reached.
- 3** For each connection (session) to that partner, the transport gateway sends an UNBIND to the SNA node to terminate the session.

DLUR-Initiated CPSVRMGR Pipe Activation

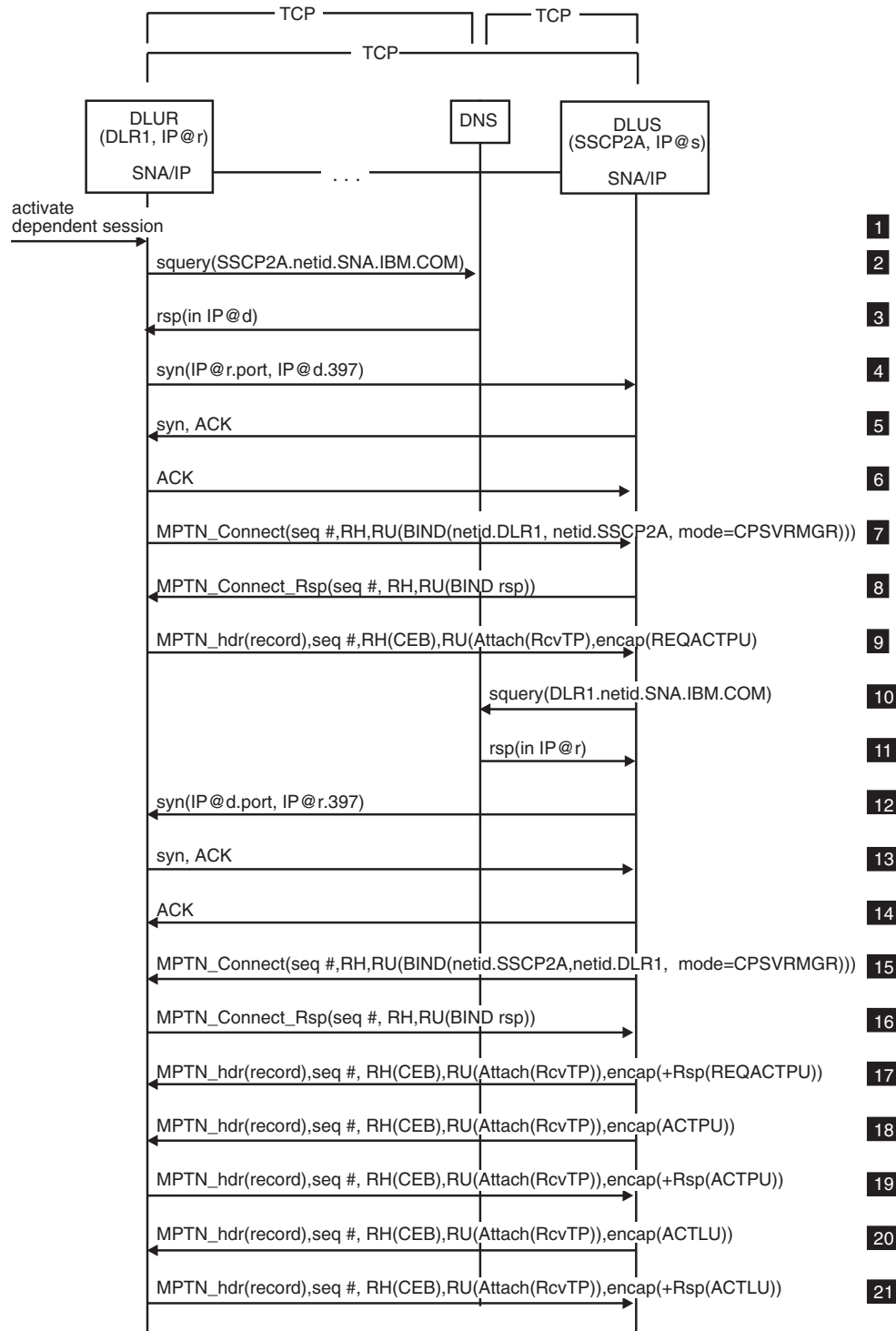


Figure 38. DLUR-Initiated CPSVRMGR Pipe Activation

Note: Communication between the DLUR and DLUS components is on two sessions; each session carries traffic in one direction. The session being setup in this flow scenario is used to send data from the DLUR to the DLUS. Messages sent on SSCP-PU and SSCP-LU sessions in subarea SNA are sent as encapsulated data on the DLUS-DLUR sessions. Each message is sent on its own conversation; they include an attach header, the encapsulated data, and a deallocation indication (CEB, conditional end bracket). These messages all appear as data on the MPTN connection.

The DLUR node initiates the normal MPTN flow sequence to bring up a session that uses the CPSVRMGR mode.

- 1** Upon receipt of some indication that a dependent LU session is needed, the DLUR node determines the DLUS node to be contacted and requests a session to the DLUS node.
- 2** Because the LU name-to-IP address mapping for SSCP2A is not cached, the node converts the LU name into a domain name and then issues a `gethostbyname()` call. This results in an squery flow to the DNS.
- 3** The response provides the IP address for the DLUS (`netid.SSCP2A`).

Note: If the LU name-to-IP address mapping for SSCP2A is cached, step **2** and step **3** are omitted.

- 4** This flow requests TCP connection establishment with the partner node using the destination IP address and the MPTN well-known port.
- 5** This flow acknowledges that the TCP connection establishment request has been received by the partner node.
- 6** This flow indicates that the TCP connection is established.
- 7** The MPTN connect packet contains the names of the DLUR and DLUS that are to be connected, the SNA BIND image, and other system information. The BIND contains the mode name of CPSVRMGR, a mode that is reserved for DLUR-DLUS communication. The MPTN connection represents an SNA session. Within this flow, the BIND image is forwarded to the LU of the DLUS for processing.
- 8** This BIND response, forwarded to the LU of the DLUR for processing, completes the MPTN connection.
- 9** The DLUR sends a REQACTPU to the DLUS.
- 10** Upon receipt of the REQACTPU, the DLUS prepares to send a positive response to the REQACTPU. The DLUS requests a session to send data from the DLUS to the DLUR. The DLUS node initiates the normal MPTN sequence to bring up a CPSVRMGR-mode session.

Because the LU name-to-IP address mapping for DLR1 is not cached, the node converts the LU name into a domain name and then issues a `gethostbyname()` call. This results in an squery flow to the DNS.

- 11** The response provides the IP address for the LU of the DLUR (`netid.DLR1`).
- 12** This flow requests TCP connection with partner node using the destination IP address and the MPTN well-known port.
- 13** This flow acknowledges that the TCP connection establishment request has been received by the partner node.
- 14** This flow indicates that the TCP connection is established.

- 15** Next, an MPTN_Connect which contains the names of the DLUR and DLUS that are to be connected, the SNA BIND image, and other system information is forwarded to the LU of the DLUR. The MPTN_Connect also indicates that the mode name is CPSVRMGR.
- 16** The BIND response, provided by the DLUR node, is forwarded to the LU of the DLUS for processing. This flow completes the MPTN connection.
- 17** The DLUS sends a +Rsp(REQACTPU) to the DLUR.
- 18** The DLUS sends an ACTPU to the DLUR.
- 19** The DLUR sends a positive response to the ACTPU request to the DLUS. When the DLUS receives the positive response, the logical connection between the DLUR and DLUS is completed.
- 20** In the above flow scenario, assume there is only one dependent LU to be activated. The DLUS sends an ACTLU to the dependent LU to activate it. When the DLUR receives the ACTLU, it passes the information to the dependent LU.
- 21** The dependent LU creates a positive response to the ACTLU request. The DLUR sends the response to the DLUS. When the DLUS receives the positive response, activation of the LU is complete and sessions for that LU can now be processed.

DLUS-Initiated CPSVRMGR Pipe Activation



Figure 39. DLUS-Initiated CPSVRMGR Pipe Activation

Note: Communication between the DLUS and DLUR components is on two sessions; each session carries traffic in one direction. The session being setup in this flow scenario is used to send data from the DLUS to the DLUR. Messages sent on SSCP-PU and SSCP-LU sessions in subarea SNA are

sent as encapsulated data on the DLUS-DLUR sessions. Each message is sent on its own conversation; they include an attach header, the encapsulated data, and a deallocation indication (CEB, conditional end bracket). These messages all appear as data on the MPTN connection.

The DLUS node initiates the normal MPTN flow sequence to bring up a session that uses the CPSVRMGR mode.

- 1** When the DLUS node needs to activate a dependent LU, it requests a session to the DLUR node.

Because the LU name-to-IP address mapping for LUs is not cached, the node converts the LU name into a domain name and then issues a `gethostbyname()` call. This results in an `squery` flow to the DNS.
- 2** The response provides the IP address for the DLUR (`netid.DLR1`).
- 3** This flow requests TCP connection establishment with the partner node using the destination IP address and the MPTN well-known port.
- 4** This flow acknowledges that the TCP connection establishment request has been received by the partner node.
- 5** This flow indicates that the TCP connection is established.
- 6** The MPTN connect packet indicates the names of the logical units (`netid.SSCP2A,netid.DLR1`) that are connected by this MPTN connection. The flow also indicates that the mode name is CPSVRMGR. The MPTN connection represents an SNA session. Within this flow, the BIND image is forwarded to the LU for processing.
- 7** This BIND response, forwarded to the LU for processing, completes the MPTN connection.
- 8** The DLUS sends an ACTPU to the DLUR.
- 9** The DLUR sends a `+Rsp(ACTPU)` to the DLUS.

Because the LU name-to-IP address mapping for SSCP2A is not cached, the node converts the LU name into a domain name and then issues a `gethostbyname()` call. This results in an `squery` flow to the DNS.
- 10** The response provides the IP address for the DLUS node (`netid.SSCP2A`).

Note: If the LU name-to-IP address mapping for SSCP2A is cached, step **9** and step **10** are omitted.
- 11** This flow requests TCP connection establishment with the partner node using the destination IP address and the MPTN well-known port.
- 12** This flow acknowledges that the TCP connection establishment request has been received by the partner node.
- 13** This flow indicates that the TCP connection is established.
- 14** Next, an MPTN_Connect which contains the names of the DLUR and DLUS that are to be connected, the SNA BIND image, and other system information is forwarded to the LU of the DLUS (SSCP2A) for processing. The MPTN_Connect also indicates that the mode name is CPSVRMGR.
- 15** The BIND response, provided by the DLUS node, is forwarded to the LU of the DLUR for processing. This flow completes the MPTN connection.

- 16** The DLUR sends a positive response to the ACTPU to the DLUS. When the DLUS receives the response, the logical connection between the DLUR and DLUS is completed.
- 17** This flow scenario assumes that there is only one dependent LU to be activated. The DLUS sends an ACTLU to the dependent LU to activate it. When the DLUR receives the ACTLU, it passes the information to the dependent LU.
- 18** The dependent LU creates a positive response to the ACTLU request. The DLUR sends the positive response to the DLUS. When the DLUS receives the positive response, activation of the LU is complete and sessions for that LU can now be processed.

SLU-Initiated LU-LU Session to PLU LUp

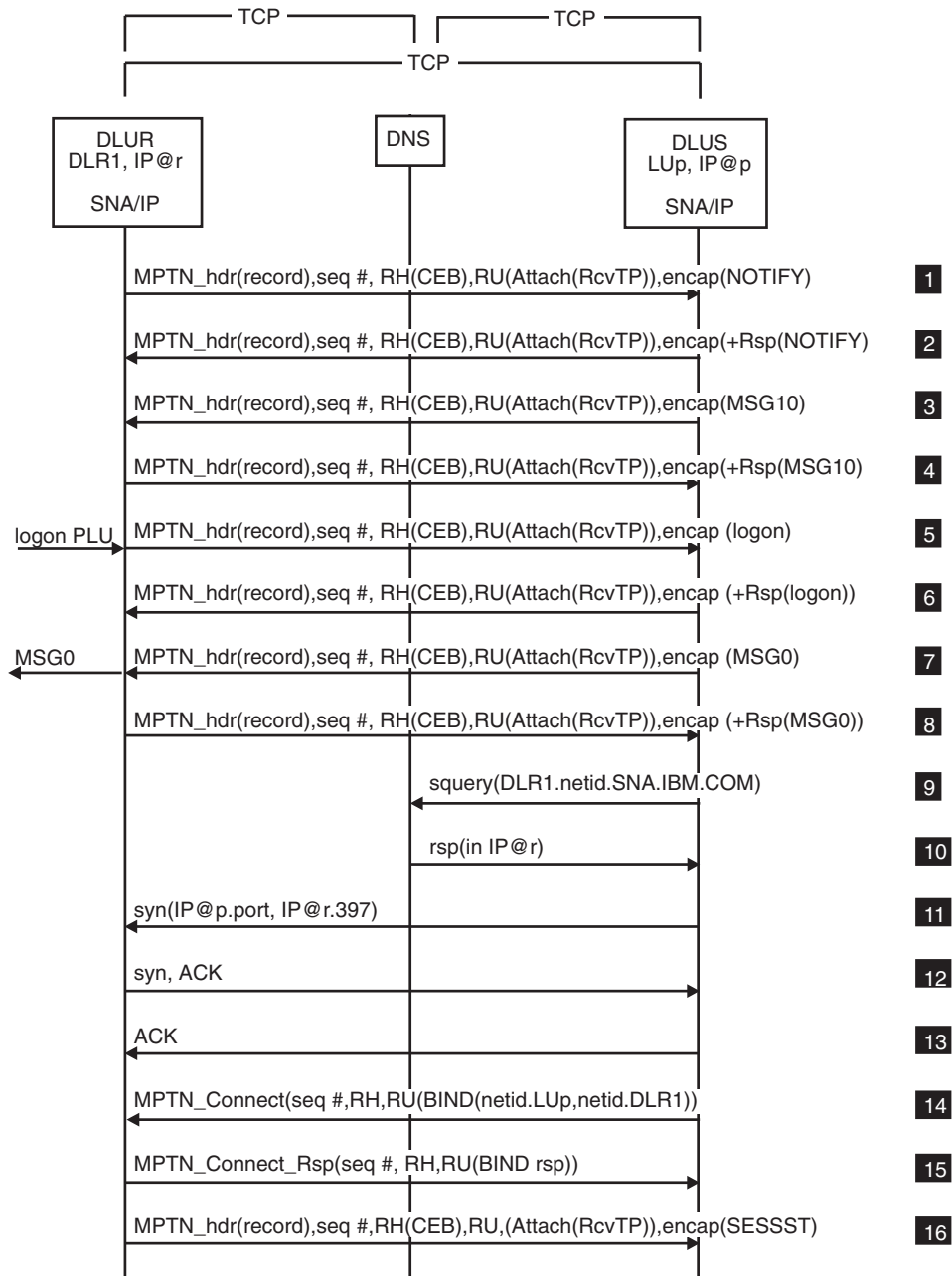


Figure 40. SLU-Initiated LU-LU Session to PLU LUp

In this flow scenario, the communication pipe between the DLUR and DLUS is already active. Messages sent on SSCP-PU and SSCP-LU sessions in subarea SNA are sent as encapsulated data on the DLUS-DLUR sessions. Each message is sent on its own conversation; they include an attach header, the encapsulated data, and a deallocation indication (CEB, conditional end bracket). These messages all appear as data on the MPTN connection.

- 1** The dependent LU indicates it is enabled by sending a NOTIFY to the DLUS node.
- 2** The DLUS sends the dependent LU a positive response to the NOTIFY.
- 3** After detecting that the dependent LU in DLUR node is enabled, the DLUS node sends a USS MSG10 (logon screen) to the DLUR.
- 4** The dependent LU generates an acknowledgment and the DLUR node sends a positive response to the DLUS.
- 5** The logon PLU command is sent to the DLUS.
- 6** The DLUS receives the logon LUP command and issues a positive response to the DLUR.
- 7** The DLUS then follows the positive response to the logon PLU command with a USS MSG0 (logon in progress) message to the DLUR.
- 8** The DLUR sends a positive response to the DLUS.
- 9** The PLU constructs a BIND image and prepares to send it to the dependent LU (SLU).

Note: Because the LU name-to-IP address mapping for LUs is not cached, the node converts the LU name into a domain name and then issues a `gethostbyname()` call. This results in an squery flow to the DNS.

- 10** The response provides the IP address for DLR1.
- 11** This flow requests TCP connection establishment with the partner node using the destination IP address and the MPTN well-known port.
- 12** This flow acknowledges that the TCP connection establishment request has been received by the partner node.
- 13** This flow indicates that the TCP connection is established.
- 14** Next, an MPTN_Connect which contains the names of the logical units that are to be connected, the SNA BIND image, and other system information is forwarded to the dependent LU and DLR1 for processing.
- 15** The dependent LU generates a positive response to the BIND. This response flows along the data path to the PLU. The LU-LU session is now established.
- 16** When the DLUR node receives the positive response to the BIND, it issues as SESSST RU to notify the DLUS that the session has started.

SLU-Initiated LU-LU Session Deactivation

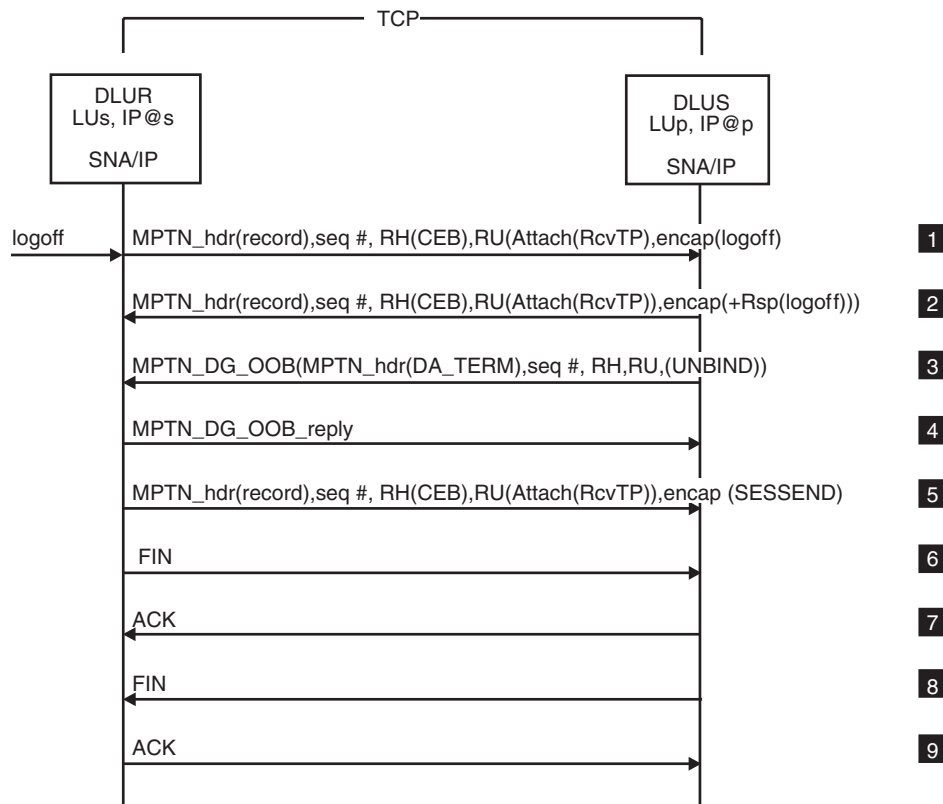


Figure 41. SLU-Initiated LU-LU Session Deactivation

Communication between the DLUR and DLUS components is on two sessions; each session carries traffic in one direction. In this flow scenario, the communication pipe between the DLUR and DLUS is already active. Messages sent on SSCP-PU and SSCP-LU sessions in subarea SNA are sent as encapsulated data on the DLUS-DLUR sessions. Each message is sent on its own conversation; they include an attach header, the encapsulated data, and a deallocation indication (CEB, conditional end bracket). These messages all appear as data on the MPTN connection.

- 1** The user enters a logoff at the terminal in order to logoff from the PLU. The DLUR node sends the logoff to the DLUS.
- 2** The DLUS receives the logoff command and issues a positive response to the DLUR.
- 3** The primary LU (PLU) learns the intention of the user from the DLUS and generates an UNBIND. The UNBIND is then packaged with an MPTN duplex-abortive termination header. The UNBIND and the header are sent as an MPTN out of band datagram across UDP.
- 4** The UNBIND is forwarded to the LU and the DLUR node responds to the OOB datagram with a datagram over UDP.

- 5** When the DLUR node recognizes that the LU-LU session has been deactivated, it issues a SESSEND RU to notify the DLUS that the session has ended.
- 6** After sending the MPTN_DG_OOB reply, the MPTN component in the dependent LU node closes the TCP connection for the LU-LU session. TCP in the dependent LU node does so by sending a packet with the FIN bit set, thus taking down its half of the full-duplex connection.
- 7** TCP in the primary LU node acknowledges this takedown by sending a packet with the ACK bit set.
- 8** When the MPTN_DG_OOB reply is received, the MPTN component in the primary LU node takes down the other half of the TCP connection for the LU-LU session.
- 9** TCP in the dependent LU node acknowledges this takedown by sending a packet with the ACK bit set. Connection takedown for the LU-LU session is now complete.

DLUR-Initiated CPSVRMGR Pipe Deactivation

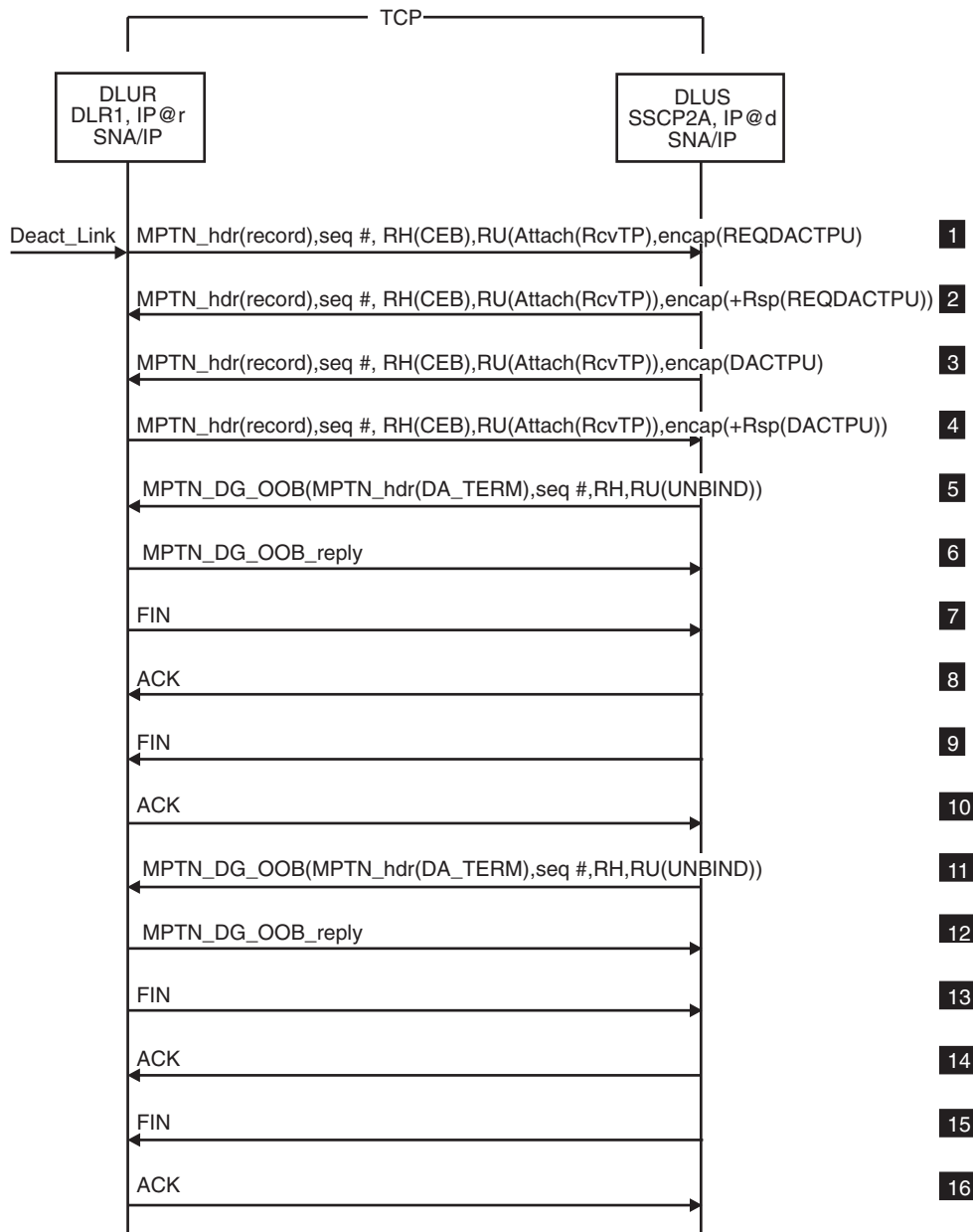


Figure 42. DLUR-Initiated CPSVRMGR Pipe Deactivation

Communication between the DLUR and DLUS components is on two sessions; each session carries traffic in one direction. In this flow scenario, the communication pipe between the DLUR and DLUS is already active. Messages sent on SSCP-PU and SSCP-LU sessions in subarea SNA are sent as encapsulated data on the DLUS-DLUR sessions. Each message is sent on its own

conversation; they include an attach header, the encapsulated data, and a deallocation indication (CEB, conditional end bracket). These messages all appear as data on the MPTN connection.

- 1** The DLUR receives an indication that the connection to the DLUS be deactivated and sends a REQDACTPU to the DLUS.
- 2** The DLUS sends a positive response to the REQDACTPU to the DLUR.
- 3** The DLUS sends a DACTPU to the DLUR node.
- 4** The DLUR sends a positive response to the DACTPU to the DLUS.
- 5** When the DLUS determines that the CPSVRMGR pipe to the DLUR node is no longer needed, it sends UNBIND to the DLUR node for its conwinner session (DLUS to DLUR communication) The UNBIND is packaged with an MPTN duplex-abortive termination header and sent as an MPTN out of band datagram across UDP.
- 6** The MPTN component at the DLUR replies to the OOB datagram by sending a datagram reply over UDP.
- 7** After sending the MPTN_DG_OOB reply, the MPTN component in the DLUR closes the TCP connection for this session. TCP in the DLUR node sends a packet with the FIN bit set, thus taking down its half of the full-duplex connection.
- 8** TCP in the DLUS node acknowledges this takedown by sending a packet with the ACK bit set.
- 9** When the MPTN_DG_OOB reply is received, the MPTN component in the DLUS node takes down the other half of the TCP connection.
- 10** TCP in the DLUR node acknowledges this takedown by sending a packet with the ACK bit set.
- 11** The DLUS node sends an UNBIND to the DLUR node for its conloser session (DLUR to DLUS communication). This serves to deactivate the other half of the pipe. The UNBIND is packaged with an MPTN duplex-abortive termination header and sent as an MPTN out of band datagram across UDP.
- 12** The MPTN component at the DLUR replies to the OOB datagram by sending a datagram reply over UDP.
- 13** After sending the MPTN_DG_OOB reply, the MPTN component in the DLUR closes the TCP connection for this session. TCP in the DLUR node sends a packet with the FIN bit set, thus taking down its half of the full-duplex connection.
- 14** TCP in the DLUS node acknowledges this takedown by sending a packet with the ACK bit set.
- 15** When the MPTN_DG_OOB reply is received, the MPTN component in the DLUS node takes down the other half of the TCP connection.
- 16** TCP in the DLUR node acknowledges this takedown by sending a packet with the ACK bit set. The logical connection between the DLUR and DLUS components is now ended.

Appendix C. Information Apars

This appendix lists information apars for IP and SNA books.

Notes:

1. Information apars contain updates to previous editions of the manuals listed below. Books updated for V1R2 are complete except for the updates contained in the information apars that may be issued after V1R2 books went to press.
2. Information apars are predefined for z/OS V1R2 Communications Server and may not contain updates.

IP Information Apars

Table 2 lists information apars for IP books.

Table 2. IP Information Apars

Title	z/OS CS V1R2	CS for OS/390 2.10 and z/OS CS V1R1	CS for OS/390 2.8	CS for OS/990 2.7	CS for OS/390 2.6	CS for OS/390 2.5
IP API Guide	ii12861	ii12371	ii11635	ii11558	ii11405	ii11144
IP CICS Sockets Guide	ii12862		ii11626	ii11559	ii11406	ii11145
IP Configuration			ii11620 ii12068 ii12353 ii12649	ii11555 ii11637 ii11995 ii12325	ii11402 ii11619 ii12066 ii12455	ii11159 ii11979 ii12315
IP Configuration Guide	ii12498	ii12362 ii12493				
IP Configuration Reference	ii12499	ii12363 ii12494 ii12712				
IP Diagnosis	ii12503	ii12366 ii12495	ii11628	ii11565	ii11411	ii11160 ii11414
IP Messages Volume 1	ii12857	ii12367	ii11630	ii11562	ii11408	ii11636
IP Messages Volume 2	ii12858	ii12368	ii11631	ii11563	ii11409	ii11281
IP Messages Volume 3	ii12859	ii12369	ii11632 ii12883	ii11564 ii12884	ii11410 ii12885	ii11158
IP Messages Volume 4	ii12860					
IP Migration	ii12497	ii12361	ii11618	ii11554	ii11401	ii11204
IP Network Print Facility	ii12864		ii11627	ii11561	ii11407	ii11150
IP Programmer's Reference	ii12505		ii11634	ii11557	ii11404	ii12496

Table 2. IP Information Apars (continued)

Title	z/OS CS V1R2	CS for OS/390 2.10 and z/OS CS V1R1	CS for OS/390 2.8	CS for OS/990 2.7	CS for OS/390 2.6	CS for OS/390 2.5
IP and SNA Codes	ii12504	ii12370	ii11917	Added TCP/IP codes to VTAM codes V2R6 ii11611	ii11361	ii11146 ii11097
IP User's Guide		ii12365	ii11625	ii11556	ii11403	ii11143
IP User's Guide and Commands	ii12501					
IP System Admin Guide	ii12502					
Quick Reference	ii12500	ii12364				

SNA Information Apars

Table 3 lists information apars for SNA books.

Table 3. SNA Information Apars

Title	z/OS CS V1R2	CS for OS/390 2.10 and z/OS CS V1R1	CS for OS/390 2.8	CS for OS/390 2.7	CS for OS/390 2.6	CS for OS/390 2.5
Anynet SNA over TCP/IP			ii11922	ii11633	ii11624	ii11623
Anynet Sockets over SNA			ii11921	ii11622	ii11519	ii11518
CSM Guide						
IP and SNA Codes		ii12370	ii11917	ii11611	ii11361	ii11097
SNA Customization	ii12872	ii12388	ii11923	ii11925 ii12008	ii11924 ii12007	ii11092 ii11621 ii12006
SNA Diagnosis	ii12490	ii12389	ii11915	ii11615	ii11357	ii11585
SNA Messages	ii12491	ii12382	ii11916	ii11610	ii11358	ii11096
SNA Network Implementation Guide	ii12487	ii12381	ii11911	ii11609 ii12683	ii11353 ii11493	ii11095
SNA Operation	ii12489	ii12384	ii11914	ii11612	ii11355	ii11098
SNA Migration	ii12486	ii12386	ii11910	ii11614	ii11359	ii11100
SNA Programming		ii12385	ii11920	ii11613	ii11360	ii11099
Quick Reference	ii12500	ii12364	ii11913	ii11616	ii11356	
SNA Resource Definition Reference	ii12488	ii12380 ii12567	ii11912 ii12568	ii11608 ii12569	ii11354 ii12259 ii12570	ii11094 ii11151 ii12260 ii12571
SNA Resource Definition Samples						

Appendix D. Notices

IBM may not offer all of the products, services, or features discussed in this document. Consult your local IBM representative for information on the products and services currently available in your area. 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 IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

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
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

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

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs

and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

Site Counsel
IBM Corporation
P.O.Box 12195
3039 Cornwallis Road
Research Triangle Park, North Carolina 27709-2195
U.S.A

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 information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurement may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

All IBM prices shown are IBM's suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

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

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrates programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly

tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to IBM's application programming interfaces.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

© (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. _enter the year or years_. All rights reserved.

This product includes cryptographic software written by Eric Young.

If you are viewing this information softcopy, photographs and color illustrations may not appear.

You can obtain softcopy from the z/OS Collection (SK3T-4269), which contains BookManager and PDF formats of unlicensed books and the z/OS Licensed Product Library (LK3T-4307), which contains BookManager and PDF formats of licensed books.

Programming Interfaces

This publication documents information NOT intended to be used as a Programming Interface of z/OS Communications Server.

Trademarks

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

ACF/VTAM	Micro Channel
Advanced Peer-to-Peer Networking	MVS
AFP	MVS/DFP
AD/Cycle	MVS/ESA
AIX	MVS/SP
AIX/ESA	MVS/XA
AnyNet	MQ
APL2	Natural
APPN	NetView
AS/400	Network Station
AT	Nways
BookManager	Notes
BookMaster	NTune
CBPDO	NTuneNCP
C/370	OfficeVision/MVS
CICS	OfficeVision/VM
CICS/ESA	Open Class
C/MVS	OpenEdition
Common User Access	OS/2
C Set ++	OS/390
CT	OS/400
CUA	Parallel Sysplex
DATABASE 2	Personal System/2
DatagLANce	PR/SM
DB2	PROFS
DFSMS	PS/2
DFSMSdfp	RACF
DFSMSHsm	Resource Link
DFSMS/MVS	Resource Measurement Facility
DPI	RETAIN
Domino	RFM
DRDA	RISC System/6000
eNetwork	RMF
Enterprise Systems Architecture/370	RS/6000
ESA/390	S/370
ESCON	S/390
@server	SAA
ES/3090	SecureWay
ES/9000	Slate
ES/9370	SP
EtherStreamer	SP2
Extended Services	SQL/DS
FAA	System/360

FFST	System/370
FFST/2	System/390
FFST/MVS	SystemView
First Failure Support Technology	Tivoli
GDDM	TURBOWAYS
Hardware Configuration Definition	UNIX System Services
IBM	Virtual Machine/Extended Architecture
IBMLink	VM/ESA
IBMLINK	VM/XA
IMS	VSE/ESA
IMS/ESA	VTAM
InfoPrint	WebSphere
Language Environment	XT
LANStreamer	z/Architecture
Library Reader	z/OS
LPDA	zSeries
MCS	400
	3090
	3890

Lotus, Freelance, and Word Pro are trademarks of Lotus Development Corporation in the United States, or other countries, or both.

Tivoli and NetView are trademarks of Tivoli Systems Inc. in the United States, or other countries, or both.

DB2 and NetView are registered trademarks of International Business Machines Corporation or Tivoli Systems Inc. in the U.S., other countries, or both.

The following terms are trademarks of other companies:

ATM is a trademark of Adobe Systems, Incorporated.

BSC is a trademark of BusiSoft Corporation.

CSA is a trademark of Canadian Standards Association.

DCE is a trademark of The Open Software Foundation.

HYPERchannel is a trademark of Network Systems Corporation.

UNIX is a registered trademark in the United States, other countries, or both and is licensed exclusively through X/Open Company Limited.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

ActionMedia, LANDesk, MMX, Pentium, and ProShare are trademarks of Intel Corporation in the United States, other countries, or both. For a complete list of Intel trademarks, see <http://www.intel.com/tradmarx.htm>.

Other company, product, and service names may be trademarks or service marks of others.

Index

A

- activating the network 35
- adjacent link station (ALS) 19
- ALS (adjacent link station) 19
- ALSLIST operand 20
- APIs (application program interfaces), supported 2
- application program interfaces (APIs), supported 2
- attachments, supported 3

C

- CDRSC (cross-domain resource), defining for SNA over TCP/IP 22
- chaining MPTN transport gateways 28
- coding TCP/IP major node
 - GROUP definition statement 10
 - LINE definition statement 10
 - PU definition statement 10
 - VBUILD definition statement 10
- configurations
 - examples of installation requirements 3, 6
 - requirements 31
 - SNA over TCP/IP (MVS/ESA) 24, 29
 - SNA over TCP/IP (OS/2) 24, 29
- CONTIMER operand 15, 33
- CRITICAL_WS 6
- cross-domain resource (CDRSC), defining for SNA over TCP/IP 22
- cross-network support 30

D

- data transmission with SNA over TCP/IP 1
- datagrams
 - for expedited data 17
 - retries 15
- defining
 - ALS name 19
 - CDRSC for SNA over TCP/IP 22
 - IP network to VTAM 9
 - LU names to IP addresses 31
 - LUs dynamically 22
 - network ID 30
 - TCP/IP major node 9
- defining minor nodes 19
- definition requirements
 - dependent logical units 23
 - independent logical units 22
 - overview 9
 - TCP/IP major node 9
- definition statements
 - GROUP 10
 - LINE 10
 - PU 10
 - VBUILD 10
- dependent logical units
 - and configuration requirements 23
 - and cross-network support 30

- dependent logical units (*continued*)
 - defining for IP environment 23
 - flow diagrams 78, 91
 - sample configurations using 29
- DGTIMER operand on VBUILD definition statement 15, 34
- diagnosing SNA over TCP/IP
 - AnyNet 39
 - flow diagrams 47, 91
 - VTAM sense codes 40
- DISPLAY ID command
 - general description 35
 - sample display 35
 - syntax of 35
- DISPLAY SESSIONS command
 - general description 36
 - SID operand 36
 - syntax of 36
- DISPLAY STATS command
 - general description 36
 - sample display 37
 - syntax of 37
- DNS (domain name server)
 - coding (MVS/ESA) 16
 - DNSUFFIX operand 16
 - in network configurations 24, 29
 - mapping LU names to IP addresses 30
- DNSUFFIX operand
 - description of 15, 31
- domain name server (DNS)
 - coding (MVS/ESA) 16
 - DNSUFFIX operand 16
 - in network configurations 24, 29
 - mapping LU names to IP addresses 30
- dynamic definition of LUs 22

E

- environment variable 6
- expedited data 17
- EXTIMER operand 17, 34

F

- flows
 - abbreviations 47
 - deallocate, no transport gateway, connection takedown 71
 - deallocate through a transport gateway, connection takedown 76
 - description 47
 - DLUR-initiated CPSVRMGR pipe activation 78
 - DLUR-initiated CPSVRMGR pipe deactivation 88
 - DLUS-initiated CPSVRMGR pipe activation 81
 - expedited data, connection copy acknowledgement arrives before datagram copy sent, send data 70
 - expedited data, datagram copy arrives first, send data 67

flows (*continued*)

- expedited data through a transport gateway, send data 69
- From IP Network
 - Existing Session Available 64
 - to native SNA APPN Network, New Session Required, setup 63
- From Native SNA Network, Existing Session Available, setup 50
- From Native SNA Subarea Network, New Session Required, setup 61
- No Transport Gateway
 - BIND Rejected, setup 59
 - Existing Session Available, setup 49
 - New Session Required, setup 48
 - Trying Native First, SNA Fails, IP Succeeds, setup 51
 - Trying Nonnative First, DNS Not Found, setup 55
 - Trying Nonnative First, IP Fails, SNA Succeeds, setup 53
 - Trying Nonnative First, No IP Connectivity, setup 57
- Normal LU 6.2 Buffering, send data 65
- outage in an IP network, connection takedown 77
- Outage in the SNA Network, connection takedown 75
- Segmented Data from Native SNA, send data 66
- SLU-initiated LU-LU session deactivation 86
- SLU-initiated LU-LU session to PLU LUp 84
- UNBIND, No Transport Gateway, AnyNet/2 Environments, Connection Takedown 73
- UNBIND, No Transport Gateway, AnyNet Environments, Connection Takedown 72
- UNBIND Through a Transport Gateway, Connection Takedown 74

G

gateway node, MPTN

- chaining 28
- description of 24
- example of 26

GROUP definition statement

- description of 10
- ISTATUS operand 19
- name operand 19
- SPAN operand 22
- syntax of 13

H

host attachments, supported by

- MVS/ESA configurations 3
- OS/2 configurations 3

host-based configurations

- applications supported by 2
- attachments supported by 3
- coding requirements 9
- configuration requirements 31
- examples 24, 29

host-based configurations (*continued*)

- naming restrictions 31
- software requirements 2
- support services used by 2
- tuning recommendations 33
- use of SNA layers 1

HOSTS.LOCAL file, mapping LU names to IP addresses 31

I

I/O tracing 39

IATIMER operand 18, 34

ILU (independent logical unit)

- defined in IP network 23
- rules for defining 23

independent logical unit (ILU)

- defined in IP network 23
- rules for defining 23

initializing SNA over TCP/IP 35

installation

- examples of configurations 3, 6
- software requirements (MVS/ESA) 2

internet protocol (IP) network

- configuration examples 3, 6
- defining to VTAM 9

IP (internet protocol) network

- configuration examples 3, 6
- defining to VTAM 9

IP address, determining 36, 37

IP address, related to LU names 30

IST message format 37

ISTATUS

- on LINE definition statement 19
- on PU definition statement 19
- overriding coded value 19

L

LINE definition statement

- description of 10
- ISTATUS operand 19
- name operand 19
- SPAN operand 22
- syntax of 13

logical unit (LU)

- and cross-network support 30
- CDRSC definition guidelines 23
- defining independent logical units 22
- dynamic definition of 22
- mapping LU names to IP addresses 30

logical units, dependent

- and configuration requirements 23
- and cross-network support 30
- defining for IP environment 23
- flow diagrams 78, 91
- sample configurations using 29

LU (logical unit)

- and cross-network support 30
- CDRSC definition guidelines 23
- defining independent logical units 22
- dynamic definition of 22

LU (logical unit) *(continued)*
mapping LU names to IP addresses 30
LU names, converting to IP 30

M

major node
defining TCP/IP 9
timers 33
mapping LU names to IP addresses
guidelines 30
MVS/ESA
DNS 31
DNSUFFIX operand 16
HOSTS.LOCAL file 31
overview 30
MODIFY TRACE command, syntax of 39, 40
monitoring SNA over TCP/IP (MVS/ESA) 35
MPTN transport gateway node
chaining 28
description of 24
example of 26
MVS/ESA-based configurations
applications supported by 2
configuration requirements 31
defining 9
diagnosing 39
initializing 35
monitoring 35
software requirements 2
support services used by 2
TCP/IP major node 9
tuning recommendations 33

N

name operand
on GROUP definition statement 19
on LINE definition statement 19
on PU definition statement 19
on VBUILD definition statement 19
naming conventions, restrictions 31
NETID operand 20
NetView
and SPAN operand 22
managing complex networks 7
managing SNA over TCP/IP 6
managing TCP/IP 7
network, activating 35
network configurations
defining TCP/IP major node 9
examples 3, 6, 24, 29
guidelines for defining 24
network ID 30
network management 6

O

operating SNA over TCP/IP 35
operator commands
VTAM
DISPLAY ID 35

operator commands *(continued)*
VTAM *(continued)*
DISPLAY SESSIONS 36
DISPLAY STATS 36
MODIFY TRACE 40
VARY ACT 19

optimizing performance
CONTIMER 33
DGTIMER 34
EXTIMER 34
IATIMER 34
TCP/IP parameters 33
VTAM parameters 33
OS/2-to-MVS communications, configuration 31
OS/2-to-VTAM communication
intermediate VTAM node, through 4
network configuration 3

P

performance
CONTIMER 33
DGTIMER 34
EXTIMER 34
IATIMER 34
TCP/IP parameters 33
VTAM parameters 33
port number, determining 36
PORT operand 20
problem determination
AnyNet 39
flow diagrams 47, 91
VTAM sense codes 40
protocol port, specifying 20
protocol support 2
protocols, use of SNA layers 1
PU definition statement
description of 10
ISTATUS operand 19
name operand 19
NETID operand 20
SPAN operand 22
syntax of 14

R

refid operand, DISPLAY ID 35
requirements
coding (MVS/ESA) 9
installation, examples of 3, 6
software (MVS/ESA) 2
resource record (RR) 31
restrictions for complex networks 30

S

service information 42
Sid operand, DISPLAY SESSIONS 36
SNA expedited data 17
SNA flows, tracing 39
SNA over TCP/IP
data transmission 1

SNA over TCP/IP (continued)

MVS/ESA

- defining 9
- description of 1
- diagnosing 39
- host attachments supported by 3
- I/O tracing 39
- initializing 35
- managing 6
- monitoring 35
- node structure 2
- operating 35
- starting 35
- support services used by 2
- TCP/IP major node 9
- tuning recommendations 33
- use of SNA layers 1
- VTAM internal tracing 39
- VTAM messages related to 37

OS/2

- description of 1
- host attachments supported by 3
- managing 6
- performance, optimizing 33
- protocol usage 1
- sense codes 40
- structure, examples of 2

SNA protocols, use of layers 1

software requirements, MVS/ESA 2

software supported 2

SPAN operand

- on GROUP definition statement 22
- on LINE definition statement 22
- on PU definition statement 22
- use with NetView 22

specifying protocol ports 20

START options 39

structure of host-based node 2

suffix, domain name 16

supported APIs 2

syntax diagram, reading 43

syntax diagrams 43

syntax of

- DISPLAY ID 35
- DISPLAY SESSIONS 36
- DISPLAY STATS 36
- GROUP 10
- LINE 13
- MODIFY TRACE 39
- PU 14
- VBUILD 10

system performance

- CONTIMER 33
- DGTIMER 34
- EXTIMER 34
- IATIMER 34
- TCP/IP parameters 33
- VTAM parameters 33

T

task control block, specifying number 21

TCB operand 21

TCP/IP, module name in VIT 40

TCP/IP major node

- defining 9
- defining minor nodes 19
- determining status 35
- monitoring 35

TCP/IP support services 2

TCP protocol port, specifying 20

TCPIPJOB 22

tracing

- MPTN headers 39
- socket API 39
- using MODIFY TRACE 39, 40
- using VTAM START 39, 40
- VTAM internal trace 39

troubleshooting

- AnyNet 39
- flow diagrams 47, 91
- VTAM sense codes 40

tuning recommendations

- CONTIMER 33
- DGTIMER 34
- EXTIMER 34
- IATIMER 34
- TCP/IP parameters 33
- VTAM parameters 33

TYPE=TCP 22

V

VBUILD definition statement

- description of 10
- DGTIMER operand 15
- displaying TCP/IP major node parameters 35
- DNSUFFIX operand 16
- EXTIMER operand 17
- name operand 19
- PORT operand 20
- syntax of 10
- TCB operand 21

VBUILD parameters, displaying 35

VIT (VTAM internal trace)

- NRM option 40
- PSS option 39
- TCP option 39

VTAM

- as chained MPTN transport gateway 28
- as MPTN transport gateway 26
- internal tracing 39
- messages 37
- sense codes 40

VTAM definition requirements

- dependent logical units 23
- independent logical units 22
- overview 9
- TCP/IP major node 9

VTAM internal trace (VIT)

- NRM option 40

VTAM internal trace (VIT) *(continued)*

PSS option 39

TCP option 39

VTAM-to-Communications Manager

network configuration 24, 25

single IP network 25

VTAM-to-OS/2 communication

intermediate VTAM node, through 4

network configuration 3

VTAM-to-VTAM communication

intermediate IP network 5

network configuration 3

single IP network 24

Readers' Comments — We'd Like to Hear from You

**z/OS Communications Server
AnyNet SNA over TCP/IP
Version 1 Release 2**

Publication No. SC31-8832-00

Overall, how satisfied are you with the information in this book?

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Overall satisfaction	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

How satisfied are you that the information in this book is:

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Accurate	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Complete	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Easy to find	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Easy to understand	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Well organized	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Applicable to your tasks	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Please tell us how we can improve this book:

Thank you for your responses. May we contact you? Yes No

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.

Name

Address

Company or Organization

Phone No.



Fold and Tape

Please do not staple

Fold and Tape



NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

IBM Corporation
Software Reengineering
Department G71A/ Bldg 503
Research Triangle Park, NC
27709-9990



Fold and Tape

Please do not staple

Fold and Tape



Program Number: 5694-A01



Printed in the United States of America
on recycled paper containing 10%
recovered post-consumer fiber.

SC31-8832-00



Spine information:



z/OS Communications Server

z/OS VIR2.0 CS: AnyNet SNA over TCP/IP

Version 1
Release 2