Communications Server
for Windows®

Network Administration Guide

Version 6.1.3

SC31-8656-03

Communications Server
for Windows®

Network Administration Guide

Version 6.1.3

SC31-8656-03

Note

Before using this information and the product it supports, be sure to read the information under Appendix B, "Notices," on page 179.

Fourth Edition (October 2007)

This edition applies to Version 6.1.3 of Communications Server for Windows and to all subsequent releases and modifications until otherwise indicated in new editions or technical newsletters.

Contents

Figures

Tables

About this document

IBM® Communications Server for Windows® is a communications services platform. This platform provides a wide range of services for Windows workstations that communicate with host computers and with other workstations. Communications Server users can choose from among a variety of remote connectivity options.

This document provides an overview of Communications Server functions and the steps required to create the Communications Server configurations for the workstations you support. The term *Communications Server* refers to the Windows version of the product, unless otherwise specified.

Who should read this document

The Network Administration Guide is provided as a planning and administration source for the person using or planning to use Communications Server. Administrative personnel, technical and service support personnel, service coordinators, and IBM personnel, as well as anyone who has responsibility for making decisions on information systems will find this document helpful.

How this document is organized

If you want to...	Refer to...
Read about Communications Server.	Chapter 1, "Introducing Communications Server functions," on page 1
Read about SNA network functions provided by Communications Server.	Chapter 2, "Communications Server and SNA," on page 17
Read about implementing AnyNet® support.	Chapter 3, "Planning for AnyNet support," on page 61
Read about implementing Client/Server communication.	Chapter 4, "Planning for Client/Server communication," on page 75
Read about implementing high performance routing (HPR) support.	Chapter 5, "Planning for HPR," on page 79
Read about implementing SNA gateway support.	Chapter 6, "Planning for SNA Gateway," on page 85
Read about implementing the TN3270E server function.	Chapter 7, "Planning for TN3270E Server," on page 93
Read about implementing the TN5250 server function.	Chapter 8, "Planning for TN5250 Server," on page 101
Read about implementing Secure Sockets Layer-based security.	Chapter 9, "Planning for Secure Sockets Layer-based security," on page 107
Read about load balancing for dependent LU and LU6.2 sessions.	Chapter 10, "Planning for load balancing," on page 115
Read about backup host connection options.	Chapter 11, "Planning for backup host connections," on page 119
Read about X.25 planning considerations.	Chapter 12, "Planning X.25," on page 127
Read about how to configure host connectivity.	Chapter 13, "Configuring host connectivity," on page 129

If you want to...	Refer to...
Read about installing and configuring the software for your network.	Chapter 14, "Implementing Communications Server configurations," on page 151
Read an overview of Communication Server system management facilities.	Chapter 15, "System management facilities," on page 169
Read about what you need to know for planning your Communications Server environment.	Appendix A, "Planning for Communications Server," on page 173

How to contact IBM service

For immediate assistance, go to the following Web site: http://www.ibm.com/software/network/commserver/windows/support

Most problems can be resolved at this Web site, where you can submit questions and problem reports electronically, as well as access a variety of diagnosis information.

For telephone assistance in problem diagnosis and resolution (in the United States or Puerto Rico), call the IBM Software Support Center anytime (1-800-IBM-SERV). 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.

Conventions and terminology used in this document

Knowing the conventions used in this book will help you use it more efficiently.

- **Boldface type** indicates:
  - Verbs, functions, and parameters that you can use in a program or at a command prompt. These values are case sensitive and should be entered exactly as they appear in the text.
  - The names of window controls, such as lists, check boxes, entry fields, push buttons, and menu choices.
- *Italics type* indicates:
  - A variable for which you supply a value
  - Book titles
  - A letter that is being used as a letter or a word that is being used as a word. For example, when you see an *a*, make sure it is not supposed to be an *an*.
- ***Bold italics type*** is used to emphasize a word.
- UPPERCASE indicates constants, file names, keywords, and options that you can use in a program or at a command prompt. You can enter these values in uppercase or lowercase.
- `Example type` indicates information that you are instructed to type at a command prompt or in a window.
- Double quotation marks indicate messages you see in a window. An example of this would be the messages that appear in the operator information area (OIA) of an emulator session.

- Binary numbers are represented as B'xxxx xxxx' or B'x' except in certain instances where they are represented with text ("A value of binary xxxx xxxx is...").
- Bit positions start with 0 at the right position (least significant bit).
- Decimal numbers over 4 digits are represented in metric style. A space is used rather than a comma to separate groups of 3 digits. For example, the number sixteen thousand, one hundred forty-seven is written 16 147.
- Hexadecimal numbers are represented in text as hex xxxx or X'xxxx' ("The address of the adjacent node is hex 5D, which is specified as X'5D'.")

This book also uses *icons* (pictures) in the text to help you find different types of information.

This icon represents a note, important information that can affect the operation of Communications Server or the completion of a task.

This icon represents a hint or additional information that can help you complete a task.

For definitions of the terms and abbreviations used in this document, you can view the latest IBM terminology at the IBM Terminology Web site.

Prerequisite and related information

This section describes the sources of information that can be useful when you are using Communications Server or related products.

Required information

Communications Server Library

The following documents are provided as Adobe Acrobat Portable Document Format (PDF) files on the Communications Server for Windows CD-ROM:

Table 1. Communications Server for Windows Library

Order Number	Title	Description
GC31-8424	*Quick Beginnings*	Provides an overview of Communications Server and the access features and emulator provided with it. Also gives installation and configuration information.
SC31-8425	*Client/Server Communications Programming*	Describes the advanced program-to-program communications (APPC) application programming interface (API) support provided by Communications Server.

Table 1. Communications Server for Windows Library (continued)

Order Number	Title	Description
SC31-8655	*Configuration File Reference*	Describes how to use an ASCII editor to copy or change configuration files. This book lists all of the configuration file keywords and their valid values.
SC31-8656	*Network Administration Guide*	Provides information for the Network Administrator's use in planning and maintaining networks that use Communications Server.
SC31-8426	*System Management Programming*	Describes the usage of the Node Operator Facility (NOF) API messages for developing programs using Communications Server.

For information about ordering publications listed in this document, contact your IBM authorized dealer or marketing representative. In the United States, you can also order publications by dialing **1-800-879-2755**. In Canada, you can also order publications by dialing **1-800-IBM-4YOU** (**426-4968**).

You can view the books directly from the CD-ROM. The .PDF files are located as follows:

Client/Server Communications Programming

auxiliaryfiles\sdk\pubs\<*lang*>\cscsp.pdf

where <***lang***> is one of the following languages:

- chs
- cht
- deu
- enu
- esp
- fra
- ita
- jpn
- kor
- ptb

Configuration File Reference

auxiliaryfiles\doc\<*lang*>\cscfr.pdf

where <***lang***> is one of the following languages:

- chs
- cht
- deu
- enu
- esp
- fra
- ita
- jpn
- kor
- ptb

Network Administration Guide

auxiliaryfiles\doc\<*lang***\csnag.pdf**

where <***lang***> is one of the following languages:

- chs
- cht
- deu
- enu
- esp
- fra
- ita
- jpn
- kor
- ptb

Quick Beginnings

auxiliaryfiles\doc\<*lang***\csqkb.pdf**

where <***lang***> is one of the following languages:

- chs
- cht
- deu
- enu
- esp
- fra
- ita
- jpn
- kor
- ptb

SNA Management Services formats

auxiliaryfiles\sdk\pubs\snams.pdf

SNA Format Reference

auxiliaryfiles\sdk\pubs\snafmt.pdf

SNA Format Reference

auxiliaryfiles\doc\snafmt.pdf

System Management Programming

auxiliaryfiles\sdk\pubs\<*lang***>\cssmp.pdf**

where <***lang***> is one of the following languages:

- chs
- cht
- deu
- enu
- esp
- fra
- ita
- jpn
- kor
- ptb

CPIC Reference

auxiliaryfiles\sdk\pubs\cpicref.pdf

Related Information

Additional information is contained in the following related publications:

- *Common Programming Interface Communications Reference* SC26–4399

  Describes CPI Communications. It is intended for programmers who want to write applications that use communications products supporting CPI Communications. Available on the Communications Server CD-ROM in HTML and PDF formats.

- *IBM Dictionary of Computing*

  Defines technical terms used in the documentation for IBM products. Available on the Web at http://www.ibm.com/networking/nsg/nsgmain.htm

- *Systems Network Architecture Formats* GA27–3136

  Describes SNA formats used between subarea nodes and peripheral nodes, and between nodes implementing APPN or low-entry networking (LEN) protocols. Available on the Communications Server CD-ROM in PDF format.

- *Systems Network Management Services Reference* GC31–8302

  Describes SNA Management Services formats used between nodes in subarea networks, and between nodes implementing APPN or low-entry networking (LEN) protocols, or both. Available on the Communications Server CD-ROM in PDF format.

How to send your comments

Your feedback is important in helping to provide the most accurate and high-quality information. If you have any comments about this document, do one of the following:

- Send your comments by e-mail to comsvrcf@us.ibm.com.
- If you prefer to send comments by post, use this address:

```
International Business Machines Corporation
Attn: IBM Communications Server for Windows Information Development
P.O. Box 12195, 3039 Cornwallis Road
Department AKCA, Building 501
Research Triangle Park, North Carolina 27709-2195
```

Be sure to include the name of the document, the part number of the document, the version of Communications Server for Windows, and, if applicable, the specific location of the text you are commenting on (for example, a page number or table number).

Summary of changes

Summary of changes
for SC31-8656-03
V6.1.3

This document contains information previously presented in SC31-8656-02, which supports V6.1.

New information

- Server and Remote API Client operating system support on the following platforms:
  - Windows® Vista (32-bit only)
- Windows NT® is no longer supported on the server
- SNA API Client for the following operating systems:
  - Windows Server 2003 (64-bit)
  - Windows Vista (32-bit and 64-bit)
- Enterprise Extender supports IPv6
- Easy installation using Microsoft Software Installer (MSI)
- Connection Network Reachability Awareness support
- RTP tuning parameters to improve HPR performance
- LU 6.2 session level timeout; see "LU 6.2 session level timeout" on page 33
- CPI-C Communication Server compatibility tool for easier migration
- Synchronous SLI Close for Microsoft compatibility allows you to migrate applications from a Microsoft HIS Server to Communications Server for Windows.

Changed information

- Tivoli® Plus support is replaced by IBM Tivoli License Manager (ITLM).

Deleted information

- Windows NT operating system support
- IBM AnyNet Sockets over SNA
- AS/400® shared folders
- Web Administration
- Communication port support
- Twinax support for AS/400 connections
- SDLC and X.25 WAC support

Chapter 1. Introducing Communications Server functions

This chapter provides an overview of the communications functions supported by Communications Server and of the methods that you can use to configure your systems to accomplish these functions. Later chapters in this book describe the functions and the configuration methods in more detail.

Supported functions

This section describes the following Communications Server functions:

- SNA communication support
- Advanced Program-to-Program Communications (APPC), which includes Advanced Peer-to-Peer Networking® (APPN)
- 32-bit application programming interfaces (APIs)
- High performance routing (HPR)
- SNA data compression
- Discovery of service providers
- Conventional LU application (LUA)
- Dependent logical unit requester (DLUR)
- SNA gateway v AnyNet support
- AnyNet support
  - AnyNet SNA over TCP/IP (access node and gateway)
- TN3270E server
- TN5250 server
- Legacy data access
  - iSeries™ OLE DB provider
- SNA API client support
- Configuration and administration support
- Load balancing
- Data security
- Adapter support
- Data Link Control (DLC) support
- APPN MIB support
- Hot standby

SNA communications support

Communications Server can act as a Systems Network Architecture (SNA) type 2.0 and SNA type 2.1 node. This support lets you write programs to communicate with many other IBM SNA products.

APPC/APPN

Communications Server provides Advanced Peer-to-Peer Networking (APPN) end node and network node support for workstations, enabling them to communicate more flexibly with other systems in the network. Additionally, a branch extender function enables you to isolate branches to avoid unnecessary CP-CP traffic.

Communications Server provides advanced program-to-program communications (APPC) to support communications between distributed processing programs, called transaction programs (TPs). The TPs can be located at any node in the network that provides APPC. APPC uses the LU 6.2 protocol for exchanging data between programs located at different logical units (LUs). In addition, APPC

supports multiple concurrent links and parallel sessions. Conversation or session security between the communicating programs is also supported through APPC.

Communications Server provides APPC throughput in performance-critical LAN environments. Communications Server provides support for the following connections:

Table 2. Supported APPC connections

Protocols	Connection	Connection type
Channel	• CDLC (OEM) • MPC	
LAN (802.2)	• Token-Ring • Ethernet • Frame Relay • ATM Lan Emulation	
SDLC	• Synchronous • Autosynchronous • Modems	• Autodialed, DTR dialed, Permanent • Autodialed • Autodialed, Permanent
X.25	• Synchronous • Hayes AutoSync • X.25 network	• Autodialed, Semipermanent • Autodialed, Permanent • Autodialed
IP	SNA over IP	IP-provided connections
IP	HPR over IP	IP-provided connections

Note: APPC/APPN also has data compression capability. Refer to "Data compression" on page 50 for more information on data compression. Also, refer to "SNA session-level encryption" on page 53 for information on encryption.

SNA LU Type 6.2 Support

LU 6.2 is an architecture for program-to-program communications. Communications Server supports the following optional SNA LU 6.2 features:

- Basic and mapped conversations
- Synchronization level of confirmation
- Security support at session and conversation levels
- Multiple LUs
- Parallel sessions, including the ability to use a remote system to change the number of sessions
- Transactions in either a one-way or two-way simultaneous mode
- Concurrent multiple links with SSCP-PU sessions, each of which can be connected to a subarea network

Refer to *Communications Server Programming Guide and Reference* for a complete list of features.

APPC configuration enhancements

Basic end node configuration requires as few as four parameters: network ID, local node name, link type, and destination address. System definition is reduced by:

- Modes supplied by IBM
- Implicit creation of partner LU and mode definitions when a conversation is allocated

- Implicit initialization of session limits (the CNOS function), which removes the requirement to explicitly initialize the session limits before conversations can be allocated
- Removal of session limits for local and partner LUs, which enables greater freedom in configuring and initializing mode session limits
- Defaults for transaction program names, operation, and type, which remove the requirement to configure transaction program definitions for applicable programs
- Dynamic configuration updates
- SNA gateway implicit workstation definitions
- Ability to edit an .ACG file containing configuration parameters
- Ability to find system definition information with Discovery

Because the SNA configuration is stored as a text file, you can quickly and easily modify the file by using an editor or a program written by a user. You can then verify and dynamically update an active configuration (without stopping Communications Server).

You can configure connections to multiple hosts, and multiple host connections can be active at the same time. Connections can be set to start on demand, or as a result of a hot standby failure.

Communications Server support of Discovery enables a node to dynamically find the control point name, medium access control (MAC) address and service access point (SAP) address of another Communications Server network node server on a Token-Ring or Ethernet LAN. This means the user does not have to know the control point name, MAC and SAP address of a partner machine before being able to define a connection to it. Currently, iSeries Client Access, Host On-Demand, Communications Server for Windows end node and Personal Communications users can use this facility.

Fundamental APPC concepts

This section introduces APPC concepts and terms.

Transaction Programs

A *transaction program* (TP) is a program, or part of an application program, that uses APPC communications functions. Application programs use these functions to communicate with application programs on other systems that support APPC.

Communications Server provides the APPC API and supports the IBM Systems Application Architecture® (SAA®) Common Programming Interface for Communications (CPI-C) calls for transaction programs.

Transaction programs issue APPC parameters to invoke APPC functions. A *parameter* is a formatted request that a transaction program issues and APPC executes. A program uses APPC parameter sequences to communicate with another program. Two programs that communicate with each other can be located at different systems or on the same system. The APPC API is the same in both cases.

When a transaction program exchanges data with another transaction program, the other transaction program is called a *partner* transaction program.

Transaction programs can issue CPI-C calls. These calls let application programs take advantage of the consistency that SAA provides.

Logical Units

Every transaction program gains access to an SNA network through a *logical unit* (LU). An LU is SNA software that accepts parameters from your programs and acts on those parameters. A transaction program issues APPC parameters to its LU. These parameters cause commands and data to flow across the network to a partner LU. An LU also acts as an intermediary between the transaction programs and the network to manage the exchange of data between transaction programs. A single LU can provide services for multiple transaction programs. Multiple LUs can be active in the node simultaneously.

LU types

Communications Server supports LU types 0, 1, 2, 3, and 6.2. LU types 0, 1, 2, and 3 support communication between host application programs and different kinds of devices, such as terminals and printers.

LU 6.2 supports communications between two programs located at type 5 subarea nodes, type 2.1 peripheral nodes, or both, and between programs and devices. APPC is an implementation of the LU 6.2 architecture.

LU sessions

Before transaction programs can communicate with each other, their LUs must be connected in a mutual relationship called a session. A session connects two LUs, so it is called an LU-LU session. Figure 1 illustrates this communication relationship.

Figure 1. A session between two LUs (LU-LU)

Sessions act as conduits that manage the movement of data between a pair of LUs in an SNA network. Specifically, sessions deal with things such as the quantity of data transmitted, data security, network routing, and traffic congestion.

Sessions are maintained by LUs. Normally, transaction programs do not work with session characteristics. You define session characteristics when you:

- Configure your system
- Use **SNA Node Operations**
- Use the management parameters

Transaction program conversations

The communication between transaction programs is called a *conversation*. Like a telephone conversation, one transaction program calls the other, and they "converse", one transaction program talking at a time, until a transaction program ends the conversation. A conversation starts when a transaction program issues an APPC parameter or CPI-C call that allocates a conversation. Conversations occur across LU-LU sessions.

Allocating a conversation to a session establishes a send-receive relationship between the transaction programs connected to the conversation. One transaction program issues parameters to send data. The other transaction program issues parameters to receive data. When the sending transaction program finishes sending data, it can transfer send control of the conversation to the receiving transaction program. Conversations can exchange control information and data.

Figure 2 shows a conversation between two transaction programs occurring over a session.

Figure 2. Transaction program conversation occurring over a session

A session can support only one conversation at a time, but one session can support many conversations in sequence. Because multiple conversations can reuse sessions, a session is a long-lived connection compared to a conversation. When a program allocates a conversation and all applicable sessions are in use, the LU puts the incoming attach (allocation request) on a queue. It completes the allocation when a session becomes available.

Two LUs can also establish parallel sessions with each other to support multiple concurrent conversations. A parallel session occurs when either transaction program allocates a conversation and a session exists, but is being used by a conversation. The LU can request a new session to satisfy the allocation.

Figure 3 shows three parallel sessions between two LUs; each session carries a conversation.

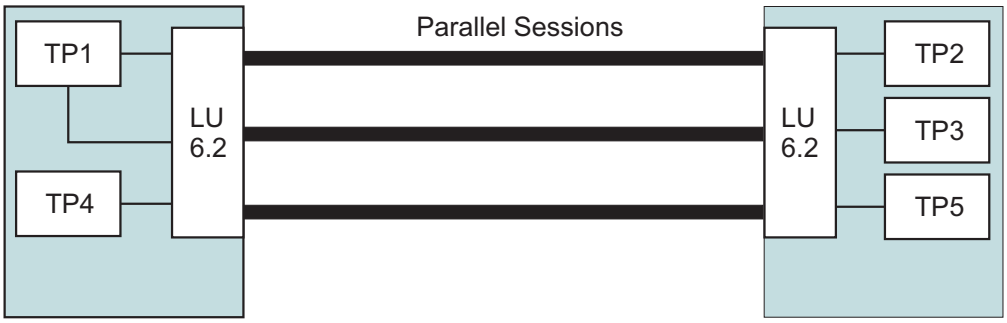

Figure 3. Parallel sessions between LUs

Advanced Peer-to-Peer Networking (APPN)

Advanced peer-to-peer networking (APPN) is a set of functions, formats, and protocols that greatly enhances the managing of an SNA network and the usability

of APPC applications running in the network. APPN does this through reduced configuration requirements, dynamic directory searches, route calculation capabilities, and intermediate session routing.

With APPN, you can write programs without knowing the details of the underlying network. All you need to know is the name of the partner LU; you do not need to know its location. SNA determines the partner LU location and the best path for routing data. A change to the underlying network, such as a physical address change, the addition of a new adapter, or the relocation of a machine, does not affect APPC programs.

Communications Server provides APPN end node and network node support for workstations, enabling them to communicate more flexibly with other systems in the network. Additionally, a branch extender function enables you to isolate branches to avoid unnecessary CP-CP traffic.

- End node capability provides the following services:
  - Connection of the local node to a serving network node for directory and routing services
  - Automatic reactivation of CP-CP sessions when an end node loses its network node server.
- Network node capability provides the following services:
  - Route selection services that calculate the best route for a session through an APPN network
  - Intermediate session routing that enables two LUs in nonadjacent nodes to be in session with each other, where the session traverses one or more intermediate nodes
  - Directory services that provide the capability to learn partner LU locations dynamically
  - Connection network services for LAN or EEDLC that provide the capability to learn destination addresses without having to configure them

    This capability enables a node to establish a link connection directly to another node with no destination address configured.
  - Branch extender support that enables you to efficiently interconnect a branch office with LANs, end nodes, and low end network nodes with dependent and independent LUs, and PUs such as teller machines, to one or several WANs. This enables more network nodes to function as gateways to branch offices. A LAN at each branch office can be configured as a connection network.

Programming Interfaces

Communications Server supports a wide range of 32-bit application programming interfaces (APIs) on the server for the application program developer. These APIs provide convenient ways for application programs to access Communications Server functions and allow applications to address the communication needs of connections to both IBM and other computers. In addition, the provided interfaces support SNA protocols so that standardization is ensured.

The APIs supported include:

- Advanced program-to-program communications (APPC)
- Common Programming Interface for Communications (CPI-C)
- Conventional LU Application Interface (LUA) RUI and SLI
- Host Access Class Libraries (HACL)
- Java™ CPI-C (JCPI-C)

- Network Operator Facility
- Management Services
- Common Services

On the clients, the Enhanced APPC (EHNAPPC) API and Host Access Class Libraries (HACL) are also provided.

The Communications Server Software Developers Tool Kit (which can be separately installed from the Communications Server CD-ROM) is also available for application developers to use. This tool kit contains samples, header files, library files, and online manuals for each of the APIs.

For more information on Communications Server programming interfaces, refer to *Client/Server Communications Programming and System Management Programming*.

High Performance Routing

High performance routing (HPR) is an enhancement to APPN that increases data routing performance and reliability and establishes a virtual link between rapid transport protocol (RTP) nodes. HPR replaces intermediate session routing, which is the routing technique used in APPN.

HPR provides faster transmission at intermediate nodes, non-disruptively reroutes sessions around failed nodes and links, and regulates traffic flow by predicting and reducing congestion in the network.

Communications Server supports HPR connections over Enterprise Extender (IPv4 and IPv6), synchronous data link control (SDLC), LAN, WAN, channel, Multi-Path Channel (MPC), and X.25 connections.

SNA data compression

Data compression at the session level increases throughput for large amounts of data across communication links, resulting in the following benefits:

- Enhanced data throughput on low-speed lines
- Reduced costs on high-cost lines
- Faster response times, resulting in productivity improvements

SNA data compression is compatible with the zSeries® and iSeries implementations and can be used with all LU types.

Discovery of service providers

Discovery is a LAN address resolution protocol that can be used by a node on the LAN to find another node that matches given search criteria. By adjusting the search parameter, a node can search for APPN network nodes, nodes that provide SNA boundary function, iSeries, SNA gateways, or user-defined classes of server. A Communications Server for Windows can respond to requests from clients as a network node server, a PU 2.0 gateway, or as a user-defined class of server. A Communications Server can also use discovery to find APPN nodes and SNA gateways.

Dependant LU Requester (DLUR)

Communications Server provides dependent LU requester (DLUR) end node and network node support for workstations, enabling them to take advantage of the enhanced system services control point (SSCP) support provided by a dependent

LU server (DLUS). DLUS is supported by VTAM® V4R2 and later. With this support, traditional SNA dependent LUs, such as emulators and even printers, can gain the many advantages of an APPN network.

Some of these benefits include:

- Enabling dependent LUs to reside on nodes that are nonadjacent to the host
- Using APPN search logic to provide the best path for LU-LU sessions

See "Dependent Logical Unit Requester support" on page 47 for more information about DLUR.

SNA Gateway

A gateway permits communication between hosts that support PU 2.0 workstations and workstations that use different DLC types. An SNA gateway can do the following things:

- Enable workstations to share the same host link
- Provide multiple host connectivity to workstations that do not have multiple PU capability
- Act as a PU pool for dedicated PU downstream workstations
- Act as a protocol converter between workstations that use DLCs on their links that are different from the DLC that is used on the host link. For example, workstations can be LAN-attached while the host connections use DLUR over EEDLC.
- Reduce the amount of system definition at the host and workstations
- Enable you to dynamically change network definitions and add workstations
- Reduce host resources and reduce the number of host connections through the use of pooled LUs and by automatic logoff of sessions that are unused for a user-specified length of time.
- Improve reliability by providing a host backup link

The SNA gateway enables an zSeries family host to support workstations that implement LU 0, 1, 2, 3, or dependent LU 6.2 (APPC). The SNA gateway also supports LU 0, 1, 2, or 3 to an iSeries host. The iSeries host passes the data through to an zSeries family host.

Each host views the SNA gateway as an SNA PU 2.0 node, supporting one or more LUs per workstation. As far as the host is concerned, all LUs belong to the SNA gateway PU. The SNA gateway can have multiple host connections simultaneously and direct different workstation sessions to specified hosts. However, only one host (and it must be on a link with a CP PU) can act as the focal point, and the control point name is appended to all network management vector transports (NMVTs) routed through the gateway.

To the supported workstations, the SNA gateway looks like an SNA PU 4 communications controller and forwards such host messages as BIND and UNBIND. The network LUs are not aware of the SNA gateway. The SNA gateway, however, is aware of all LUs at the workstations.

In reality, the SNA gateway is a special type of PU 2.0. As long as a dependent workstation is inactive, the SNA gateway implements the LU functions for the workstation, just as a real PU 2.0 would. However, as soon as a workstation is online to the host, the SNA gateway enables the workstation to implement LU functions and merely passes data between workstations and the host.

An SNA gateway enables supported workstation applications to access remote supported applications on a subarea network without requiring a separate direct connection to each host in each workstation. From a host point-of-view, the host has a single connection to the gateway.

See Chapter 6, “Planning for SNA Gateway,” on page 85 for more information about using an SNA gateway.

Figure 4 shows an example of a connection using an SNA gateway.

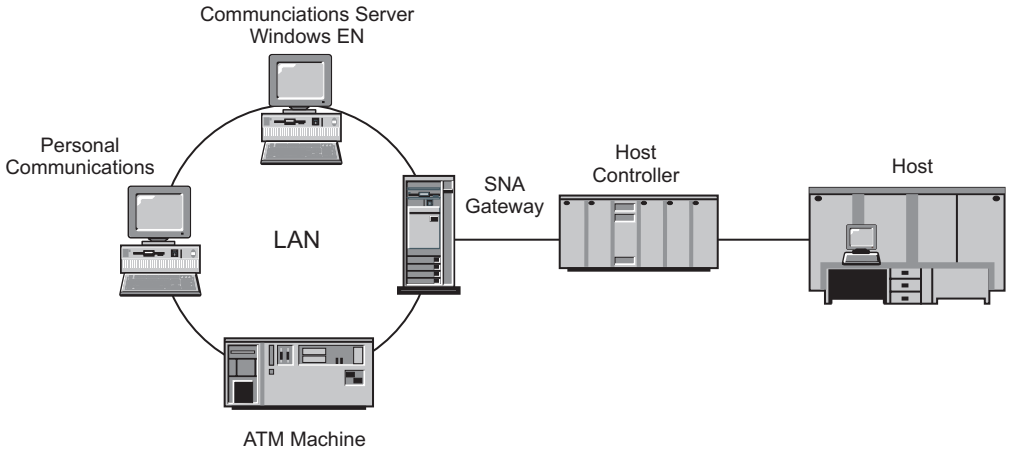

Figure 4. Example of SNA gateway connections

AnyNet support

Communications Server incorporates the SNA over TCP/IP function from the AnyNet product family. This support enables you to extend and simplify your network by allowing SNA applications to communicate across a TCP/IP network without changes to the applications.

SNA over TCP/IP

The SNA over TCP/IP access node function allows SNA applications residing on an IP network to communicate. This function supports independent LU6.2 and dependent LU 0, 1, 2, 3, or 6.2 either with or without dependent LU requester (DLUR). In addition, the SNA over TCP/IP access node can be used in conjunction with SNA gateway to enable SNA gateway sessions over TCP/IP.

The SNA over TCP/IP gateway function extends the reach of SNA applications by allowing SNA applications in an SNA network to communicate with SNA applications in an IP network. The SNA over TCP/IP gateway supports independent LU 6.2 sessions.

Figure 5 on page 10 shows SNA applications communicating through an SNA over TCP/IP Gateway across IP and SNA networks.

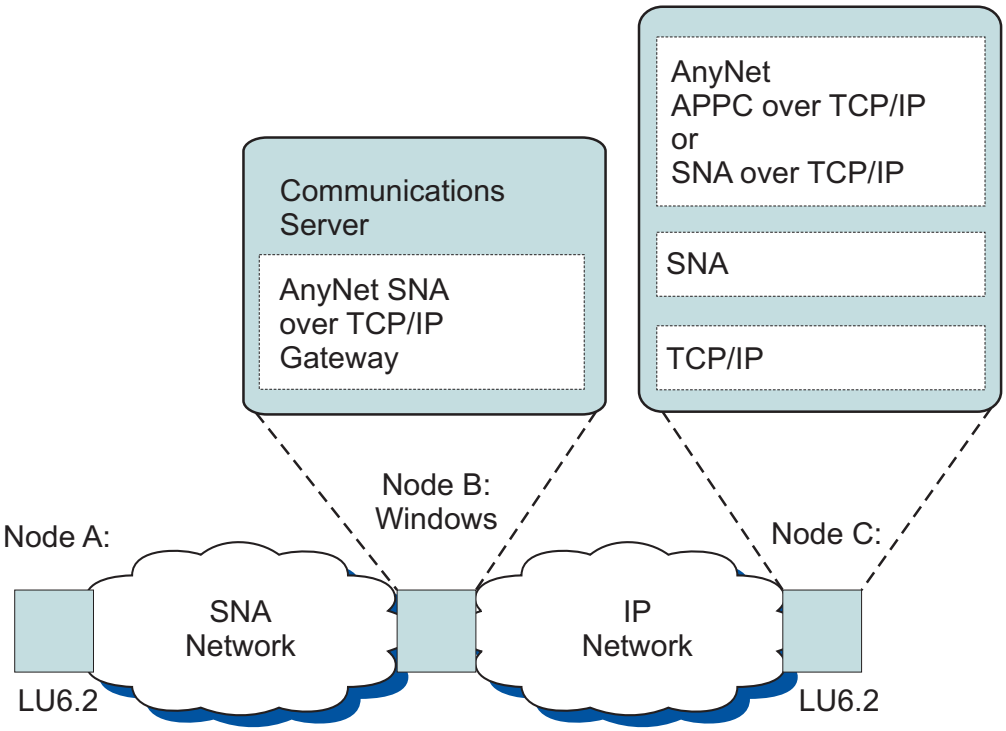

Figure 5. SNA over TCP/IP gateway

TN3270E Server

The TN3270E server function enables TCP/IP users to access applications on a host machine in an SNA network. Any industry-standard TN3270 or TN3270E client workstation can connect to the TN3270E server for access to SNA networks. The TN3270E server supports ATTN and SYSREQ key handling and enables users to print from host applications to printers attached to their workstation. These printers may be locally-attached or network-attached.

Communications Server supports load balancing for client connections of a TN3270E server that connects to the same host resources, if the client is enabled for load balancing.

TN3270E server supports IP and hostname filtering that allows controlled access to LUs without modifying client configurations.

TN3270E server also supports Secure Sockets Layer (SSL) authentication and encryption, providing secure access across the TCP/IP network. If you specify security, the server must have an authenticated certificate provided by a certificate authority such as IBM Vault Registry or Verisign. Communications Server provides a utility that generates and manages keys and certificates used by SSL Version 3. For more information on using SSL authentication and encryption, refer to Chapter 10, Planning for Secure Sockets Layer-based Security.

Figure 6 on page 11 shows an example of TN3270E server connections.

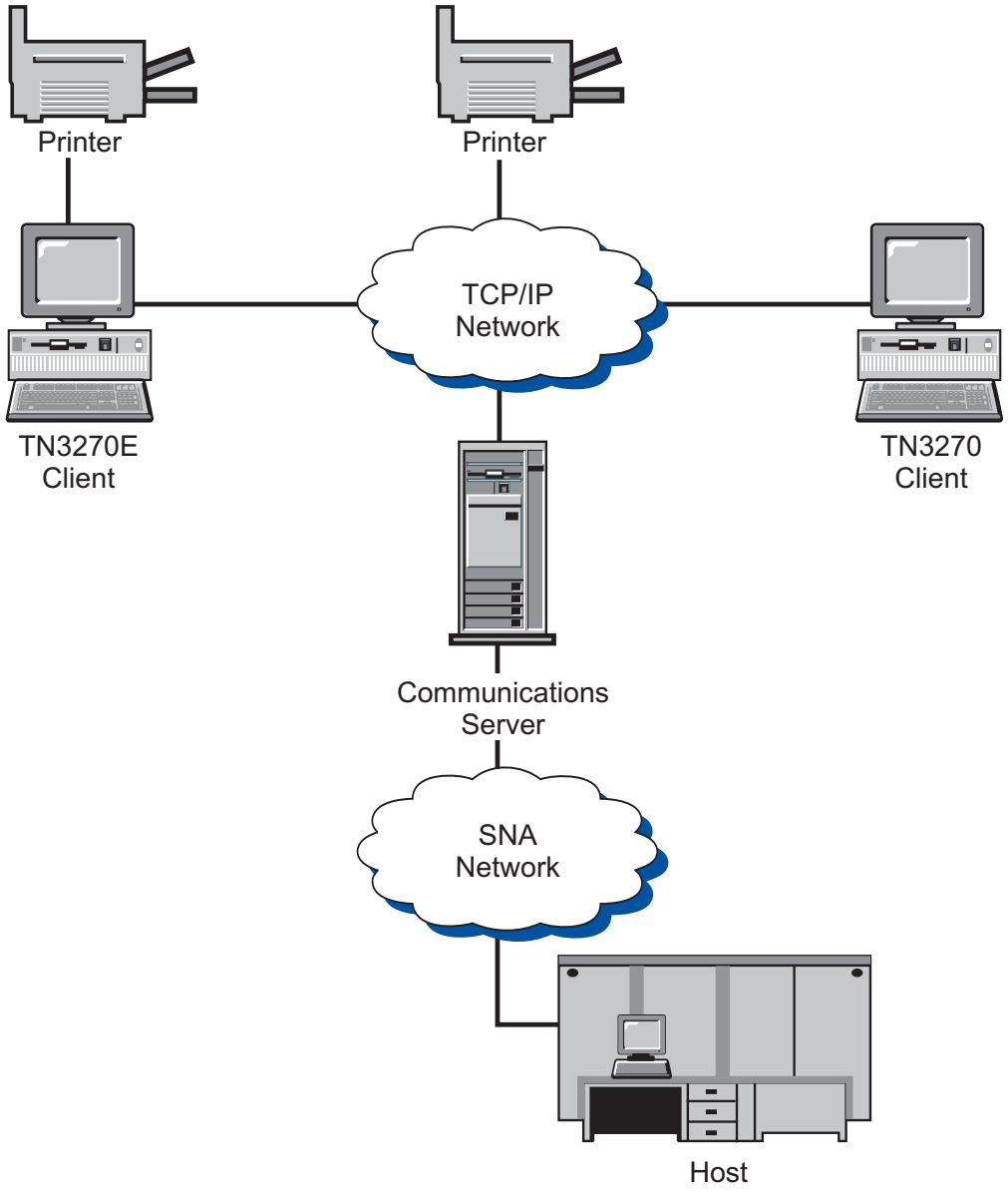

Figure 6. TN3270E Server Connections

TN5250 Server

The TN5250 server function enables TCP/IP users to access applications on an iSeries in an SNA network. Any industry-standard TN5250 client workstation can connect to the TN5250 server for access to SNA networks.

Communications Server supports load balancing for client connections of TN5250 servers that connect to the same iSeries, if the client is enabled for load balancing.

TN5250 server supports IP and hostname filtering that allows central administration of client access to the server, as well as directing clients to specific iSeries.

TN5250 server also supports Secure Sockets Layer (SSL) authentication and encryption, providing secure access across the TCP/IP network. If you specify security, the server must have an authenticated certificate provided by a certificate authority such as IBM Vault Registry or Verisign. Communications Server provides a utility that generates and manages keys and certificates used by SSL Version 3. For more information on using SSL authentication and encryption, refer to Chapter 10, Planning for Secure Sockets Layer-based Security.

Figure 7 shows an example of TN5250 server connections.

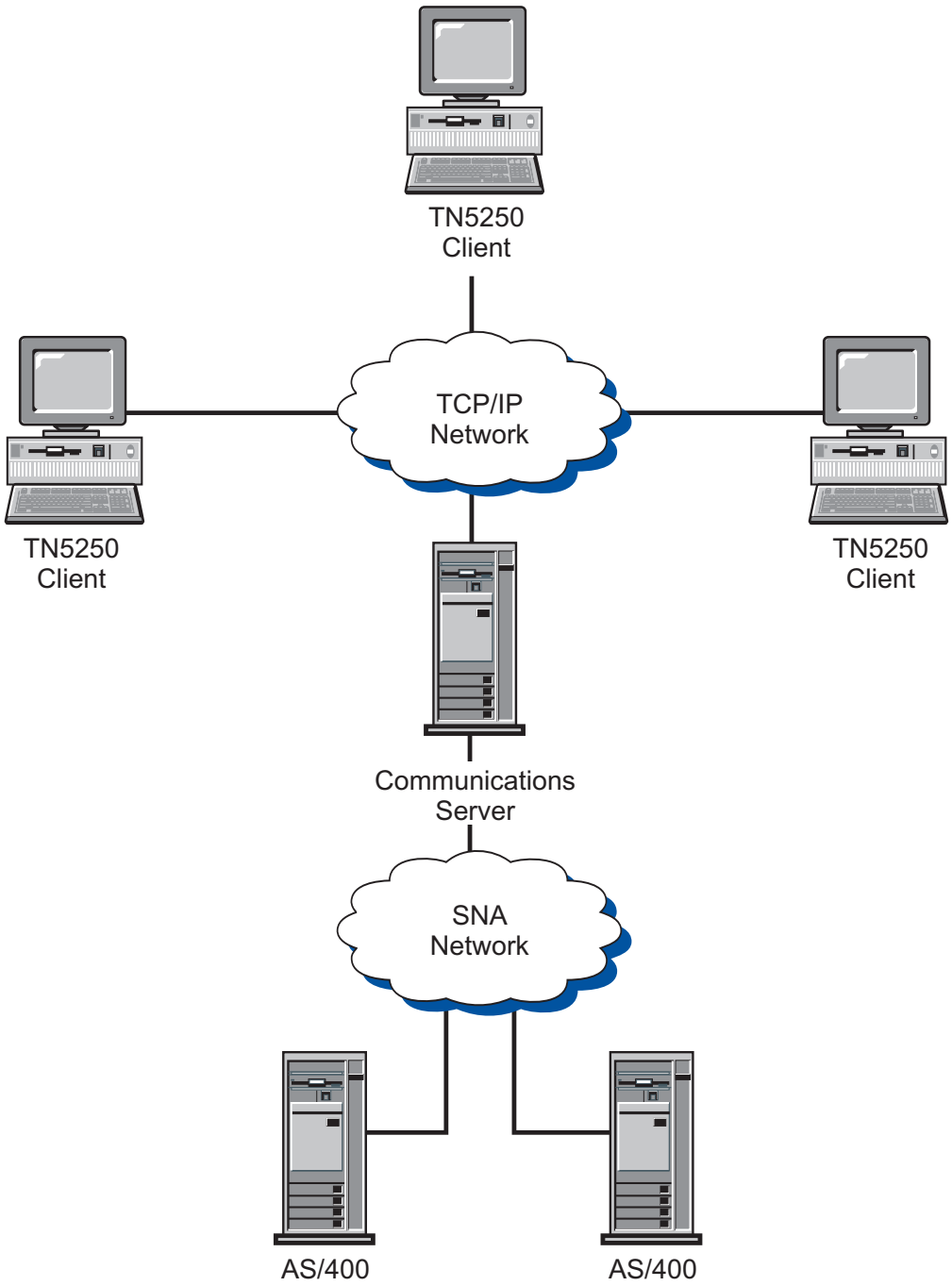

Figure 7. TN5250 Server Connections

Communications Server Client support

Communications Server provides support for SNA API clients (available on the CD-ROM).

SNA API Client support

The Communications Server SNA API client support allows TCP/IP-attached clients to access SNA APIs without requiring SNA protocols to flow between the clients and the server. This allows most SNA configuration to take place at the central server.

Communications Server supports SNA API clients on Windows 2000, Windows XP, Windows Server 2003 and Windows Vista.

The SNA clients provide support for CPI-C, APPC, EHNAPPC, LUA SL1, LUA RUI, and JCPI-C API interfaces, while providing the actual SNA processing at the server. These clients are delivered as part of the server but are actually installed and configured at the client.

The 32-bit Windows clients have additional enhancements:

- Encryption between the client and server
- More detailed trace formatting that can help you debug problems faster
- Ability to start and stop tracing while the client is running
- Ability to reinstall the client without uninstalling

The Windows 2000, Windows XP, Windows Server 2003 and Windows Vista run from the same executable. This executable can be installed on a shared drive; any fixes apply to all clients. The 32-bit Windows client communicates with Communications Servers for Windows.

For more information about the API client in Communications Server, see Chapter 4, “Planning for Client/Server communication,” on page 75.

Load balancing

Communications Server supports load balancing for all client types. Load balancing enables you to distribute LU 0 to 3 and LU 6.2 sessions across Communications Server. The server advertises services including load factors, which the clients or servers can gather and organize to select a server.

Directory exploitation

You can configure multiple clients from a central location using the Lightweight Directory Access Protocol (LDAP) to simplify the configuration process.

For information about using directory exploitation, refer to Chapter 4, “Planning for Client/Server communication,” on page 75.

Configuration and administration support

Communications Server provides facilities for the configuration and administration of resources.

Communications Server configuration

This section provides an overview of the Communications Server configuration components and the methods used to create or change them. The configuration is composed of a single file (ACG) stored in the PRIVATE subdirectory of the directory where you installed the product (for example, C:\Program Files\IBM\Communications Server\PRIVATE). The ACG file can be created using

SNA Node Configuration. You can modify the ACG file using **SNA Node Configuration**, or by using an ASCII editor. A verification (vacgcon/vacgwin) program is available to check the validity of the ACG file prior to use.

You can use the following methods to create or change a Communications Server configuration:

SNA Node configuration
Communications Server provides the SNA Node Configuration application (PCSCFG) that enables you to configure the Communications Server functions using a graphical interface and supplies defaults so you can configure them easily using a minimum number of parameters. During configuration, each definition you create is checked to ensure that it is valid. When you save the configuration, the required configuration files are created.

Local configuration is supported at both the client and server level. Remote configuration of the server is supported from Windows 2000, Windows XP, Windows Server 2003 and Windows Vista clients.

Most configurations can be created using **SNA Node Configuration**. However, a few keywords and some keyword parameters are not supported by **SNA Node Configuration**.

SNA Node operations
SNA Node operations provides the capability to create and modify select resources.

Communications Server administration

Communications Server provides the following facilities for administration of resources.

- **SNA Node Operations**
- Command line utilities

For more information about the capabilities of these facilities, refer to Chapter 15, "System management facilities," on page 169.

Load balancing

Load balancing is a function of Communications Server that dynamically balances dependent LU (host-to-workstation) sessions and independent LU 6.2 sessions by distributing them to the communications server with the smallest load. Communications Server performs load balancing for Communications Server API programs and third-party 3270 emulators that connect over TCP/IP protocols, or third-party TN3270 and TN5250 emulators. The resources across which balancing occurs depend on the session type:

- For dependent LU sessions, the load is distributed across servers within a named LU pool and a named scope.
- For LU 6.2 sessions, the load is distributed across all available servers in a named scope or across selected servers within a named server list.

The load balancing capabilities of Communications Server are built into the SNA client APIs. Load balancing is configured for the clients using **SNA Client Configuration.**

For dependent LU sessions, emulators that use SNA client APIs can participate in load balancing. Otherwise, you must purchase third-party 3270, TN3270, or TN5250 emulator software that supports load balancing.

For LU 6.2 sessions, the initial connection established by an SNA API client determines which server manages all subsequent LU 6.2 sessions.

For more information on load balancing, refer to Chapter 10, "Planning for load balancing," on page 115.

Data security

Communications Server provides basic and enhanced security support at session and conversation levels. There is security in limiting which Windows NT or Windows 2000 users may access SNA resources through the SNA API clients. Conversation security includes support for password substitution. There is also enhanced LU-LU security.

Communications Server provides support for Secure Sockets Layer-based (SSL-based) security on connections between TN clients and the TN3270E server or TN5250 server. This security uses SSL Version 3 to provide data encryption and server authentication using signed certificates.

Adapter support

Communications Server provides an open interface for adapter manufacturers to build connectivity solutions. A shallow (nonprogrammable) adapter interface is provided for adapter manufacturers to work with Communications Server's SDLC and X.25 protocol stack. A deep (programmable) adapter interface is provided for adapter manufacturers to build connectivity solutions using manufacturer-supplied data link controls.

Data Link Control

Communications Server enables communications over the following DLCs:

- AnyNet (SNA over TCP/IP)
- LAN (Any NDIS compliant network adapter)
- X.25 SDLC (synchronous, asynchronous, and AutoSync)
- OEM (third-party adapter support)
- Channel (upstream only)
- Multi-Path Channel (HPR only)
- Enterprise Extender

AnyNet (SNA over TCP/IP) DLC

For more information about the AnyNet SNA over TCP/IP DLC, refer "SNA over TCP/IP" on page 9.

IBM Multi-Path Channel DLC

The Multi-Path Channel (MPC) DLC provides high-capacity, high-availability fiber connections to one or more zSeries MPC-capable hosts over the ESCON channel adapter card (P/N 9663 001). MPC connections provide high data transmission rates with transparent backup when physical connections break or become temporarily unavailable. This channel-to-channel connection enables you to provide LAN clients ready access to zSeries resources and services.

Enterprise Extender DLC

Communications Server provides HPR connections on IPv4 and IPv6 networks, using UDP/IP packets. To the HPR network, the IP backbone appears to be a logical link. To the IP network, the SNA traffic appears to be UDP datagrams. These datagrams are routed without changes to the IP backbone. Because there is

no protocol transformation and because packaging takes place at the routing layer without the overhead of additional transport layers, this results in efficient use of the intranet infrastructure for IP clients that access SNA-based data (TN3270 clients or Web browsers using IBM Host on Demand, for example), as well as for SNA clients.

APPN MIB support

Communications Server supports simple network management protocol (SNMP) requests for APPN management information from any SNMP management system.

Hot Standby

In Communications Server, you can configure certain host links to activate automatically if a specified critical server fails. Configured connections to a host can continue to function by activation of alternative connections on a backup server. This function is known as hot standby.

The connections named in a critical server configuration on the backup server are activated when the backup server detects a loss of contact with the critical server and licensing charges for the critical server are managed on the backup server.

Note: The hot standby feature only provides for activation of host connections on a backup server and depends on the use of emulator software that supports alternate routing to the backup server when a critical server becomes inactive.

For more information about using hot standby for backup connections, refer to Chapter 11, “Planning for backup host connections,” on page 119 Chapter 12, Planning for Backup Host Connections.

Chapter 2. Communications Server and SNA

This chapter discusses the SNA network functions provided by Communications Server and includes the following topics:

- Overview of SNA functions
- Data link control (DLC) definitions
- Advanced Peer-to-Peer Networking (APPN)
- High Performance Routing (HPR)
- LU support
- Data compression
- Encryption
- Management services (MS)
- Flow control
- SNA gateway support

Overview of SNA functions

This section provides an overview of how Communications Server implements SNA on a workstation. It is not a comprehensive discussion of the SNA functions. For more detailed information on SNA, refer to the following books:

- *Systems Network Architecture Concepts and Products*
- *Systems Network Architecture Technical Overview*
- *Systems Network Architecture Network Product Formats* (available online in the Communications Server documentation list)
- *Systems Network Architecture Format and Protocol Reference Manual: Architecture Logic for LU Type 6.2*

SNA defines the standards, protocols, and functions used by devices in the network (from mainframes to terminals) to communicate with one another. This communication enables these devices to transparently share information and process resources. In other words, a user at a workstation does not have to know what happens in the background to access information at a host or to communicate with another user.

An SNA network is organized as a system of nodes and links. It is important to remember that this organization is logical. SNA classifies the nodes according to their capabilities and the amount of control that they have over other nodes in the network. The node type is not necessarily associated with a specific type of hardware. The capabilities of a node can be performed by different devices. A workstation acting as a gateway can perform the same functions as a communications controller. It is even possible for a single device to contain more than one node.

The SNA network is responsible for moving data between two end users in an efficient, orderly, and reliable manner. For example, when a user at a workstation sends a message to another workstation, SNA does the following things:

- Packages the message in a form that can be used on the link type
- Addresses the package

- Selects a route for the message
- Monitors the communication to ensure that the message arrives at its destination
- Converts the message back to a form that can be used at the destination

These tasks are defined in SNA as separate functional layers. These layers are not part of the discussion here, but it is important to remember that the SNA layers are all part of a logical link.

As mentioned previously, the SNA nodes are classified according to their communications capabilities and the amount of control the node has over other nodes in the network. The SNA nodes are broadly classified as subarea nodes and peripheral nodes. The subarea nodes are like hubs and can communicate with the peripheral nodes and with other subarea nodes. The subarea nodes activate and control resources at the peripheral nodes. Subarea nodes are also classified as type 4 or type 5 nodes. Type 5 nodes contain a system service control point (SSCP) that provides a central control point for the type 4 nodes attached to it. A type 5 node is sometimes referred to as a host node. A peripheral node can only communicate directly with the subarea node that it is attached to. However, a peripheral node can control devices to which it is attached. For example, a cluster controller acting as a peripheral node can support the terminals that are attached to it. Peripheral nodes are also referred to as type 2 or type 2.1 nodes.

Each node contains network accessible units (NAUs) that perform control and communication functions. One of these NAUs is a physical unit (PU). The PU manages the physical resources of the node. Other NAUs called logical units (LUs) provide logical access points to the network that enable communication between users and applications at each node. The communication between the logical units is referred to as a session. Sessions not only support communication between users and applications, but also support communication between applications in order to share processing resources. Communication between applications is known as advanced program-to-program communication (APPC). APPC is a set of programming conventions and protocols that implement LU 6.2. (APPC is the name given to the LU 6.2 capability in products that implement this LU type.)

Data Link Control profiles

Data link control (DLC) enables orderly exchanges of data between two nodes through a logical link. The DLC provides the protocols necessary for reliable delivery of basic transmission units (BTUs) between a pair of nodes in the SNA network. You must configure the appropriate Communications Server DLC profiles for a workstation to access an SNA network.

Advanced Peer-to-Peer Networking

Advanced Peer-to-Peer Networking (APPN) is an extension of SNA that adds communications functions beyond those described in the previous sections. Its basic components include:

- APPN node types
- Control points
- Branch extender
- Data link control
- Logical links
- Directory services
- Topology and route-selection services

APPN node types

This section discusses the three types of nodes implemented by Communications Server that can participate in an APPN network:

- Network node
- End node
- Low entry networking (LEN) node

In addition, branch extender is an extension to a network node. Though it provides services to the end nodes as a network node, it appears in the network like an end node connected to other network nodes. This type of node reduces the amount of data that must flow in the network.

Each node is distinguished from other nodes in the network by a unique name consisting of two parts—a network ID and a local node name (also known as a control point [CP] name). The name identifies each node to all other nodes in the network. Also, the node can have multiple PU names for simultaneous access to multiple PU T4/5 hosts.

A node can be configured to be an end node or a network node, but when an end node does not have CP-CP sessions (see "CP-CP sessions" on page 22) to an APPN network node, it acts as a LEN node. A LEN node does not support APPN functions.

The node types are described in more detail in the following sections. Figure 8 on page 20 illustrates a sample APPN network that includes all of these node types.

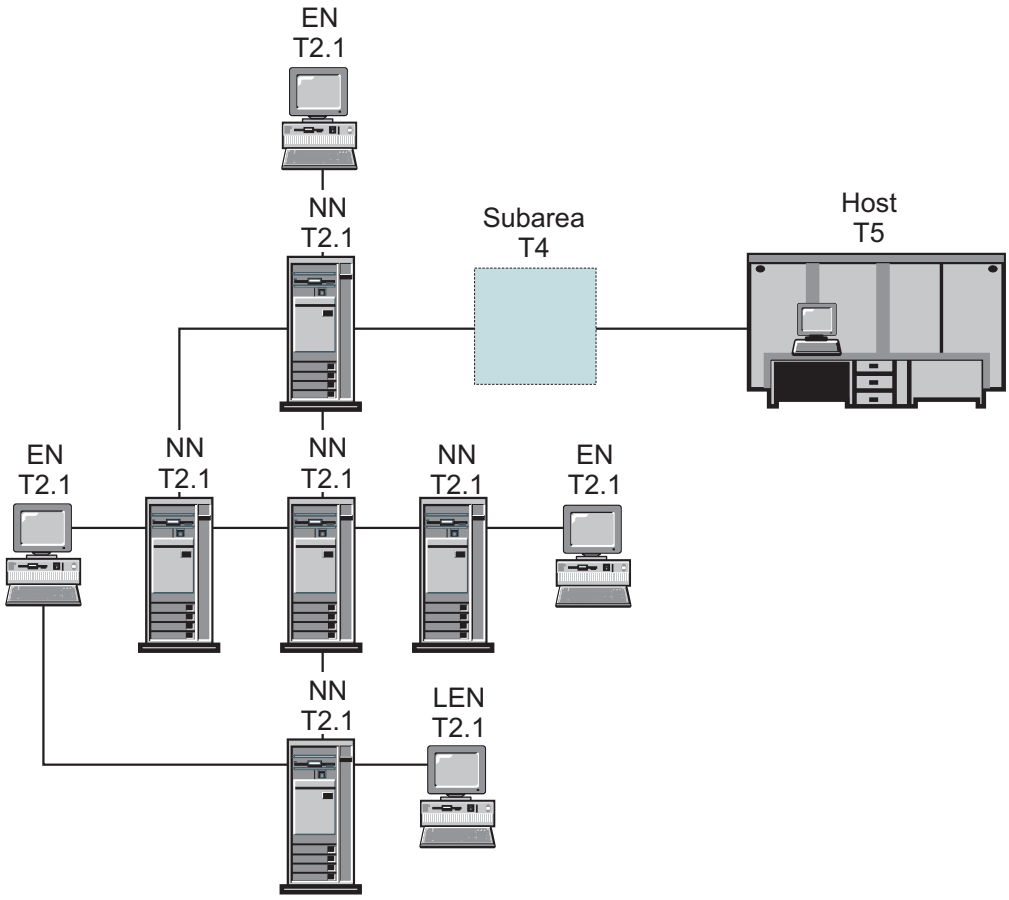

Figure 8. A Portion of a Sample APPN Network. This view of network topology shows five network nodes (NNs). Three end nodes (ENs) are connected, as well as a LEN node and subarea. APPC application programs written for any node in this network can communicate with any other.

Network Node

A network node supports its own end users; provides directory, route selection, and management services to end nodes; and performs intermediate routing of data on sessions that traverse it. The network node performs distributed searches of the network to locate partner LUs and calculates the best route from origin node to destination node based on user-specified criteria.

A network node server refers to the role of a network node in acting as the network entry point for specific end nodes attached to it. These end nodes are defined as being in its domain. For example, all directory requests regarding resources (such as LUs) in these end nodes (as well as its own resources) pass through directory services in the network node server. The network nodes are able to collect and control directory information that passes into the APPN network.

A network node provides the following:

- LU-LU session services for its locally resident LUs
- Intermediate session routing
- Network server functions (to perform network directory searches and route selection) for attached end nodes or LEN nodes (as well as for its own locally resident LUs)

- Management services (MS) routing capability for forwarding management services data (such as alerts) between a served end node and a management services focal point (FP).
- Supports configuration of a port to allow only a specific number of implicit inbound links for LAN connections.

End Node

An end node operates in a peer environment for LU-LU sessions (using LU 6.2 protocols) while providing additional APPN functions. An end node provides APPN functions such as directory services and route selection services to end users at its own node. It can participate in the APPN network by using the services of an attached network node server for session requests that involve nodes not directly connected; it does this by exchanging requests and replies for directory services with an adjacent network node (its server) using CP-CP sessions.

APPN end nodes can register their local LUs with their network node server. By registering the end nodes, the network operator at the network node server does not need to predefine the LU names for the LUs in all the attached end nodes for which the network node provides services.

An APPN end node can be attached to multiple network nodes, but it can only have CP-CP sessions active with one network node at a time—its network node server. The other network nodes can be used to provide intermediate session routing for the end node, or as a substitute network node server if the main (preferred) network node server becomes unavailable. CP-CP sessions are never established between two end nodes.

LEN Node

A LEN node is a node that implements the basic T2.1 protocols without the APPN enhancements. In a LEN node, all potential connections with partner LUs are predefined before initiating sessions to them. A LEN node, connected to an adjacent APPN network node, uses the advanced functions of APPN by predefining potential connections with partner LUs as if they existed at that network node. The network node, in turn, can automatically act as the LEN node's network node server and locate the actual destination of the partner LU and select the best route to it. By going through a network node, the LEN node can participate in an APPN network without requiring direct connections to all nodes.

Control Points

The control point (CP) is responsible for managing the node and its resources. To obtain APPN network services, the control point in an APPN end node must communicate with the control point in an adjacent network node. Also, to manage the network, the control point in an APPN network node must communicate with the control points in adjacent network nodes. The control point directs functions such as the following:

- Adapter activation and deactivation
- Link activation and deactivation
- Assists LUs in session initiation and termination.

When setting up a workstation, you must define the control point name (also known as the local node name). The control point is also an LU, and you can choose to have the control point LU be the only LU defined in your workstation.

CP-CP sessions

To perform directory services and topology and route-selection services, adjacent nodes throughout the APPN network use a pair of parallel CP-CP sessions to exchange network information. Network nodes use CP-CP sessions to monitor nodes in a network link, as well as to track directory and session services. A network node establishes two parallel sessions with each adjacent network node and with each served end node. An APPN end node establishes two parallel sessions with a single adjacent network node acting as its current server. LEN nodes do not support CP-CP sessions.

After a connection has been established, the nodes exchange identification information (XID). Then, CP-CP sessions are started between the control points in the directly attached nodes. The CP-CP sessions use LU 6.2 protocols and both sessions of a given pair must be active for the partner control points to begin and sustain their interactions. All CP-CP sessions are used to conduct directory searches.

After the CP-CP sessions are established, the two nodes exchange control point capability messages that inform each node of the other's capabilities. When both nodes are network nodes, they exchange topology database update (TDU) messages. The TDU messages contain identifying information, node and link characteristics, and resource sequence numbers to identify the most recent updates for each of the resources described in the TDU.

CP-CP connection activation

When Communications Server is started, it begins activating all connections defined as *Activate at startup*. When a connection to a NN server becomes active, Communications Server will activate the CP-CP sessions on that connection. If a connection to a preferred NN server eventually becomes active, Communications Server will switch the CP-CP sessions to the preferred NN server connection.

Note: If the connection was deactivated by operator request from the local node, CP-CP sessions are not redriven. If the connection was deactivated by operator request from the remote node, CP-CP sessions are redriven at the local node. For links between NN nodes, only demand-activated links (links with adjacent CP name specified and not defined as *Activate at startup*) links are activated.

CP-CP connection reactivation

Communications Server provides support for CP-CP connection reactivation. Loss of CP-CP sessions between an end node and its network node server and between adjacent network nodes can interfere with the operation of an APPN network. CP-CP connection reactivation support improves reliability of an APPN network by reestablishing these important sessions when they are terminated due to failure or connection inactivation.

A CP-CP connection reactivation attempt is initiated by a CP-CP link activation, CP-CP session failure, or by a CP-CP retry timer expiring. CP-CP sessions are initiated by Communications Server with the first of the following:

1. The preferred server CP-CP (if it has not previously been attempted).
2. If DLUR is configured, use an adjacent CP that supports DLUR registration.
3. The last (most recent) activation of NNs that has not been attempted, to which an active connection exists.
4. The first of any other uplevel NNs for branch extender.

Note: If the CP-CP connections were terminated due to a link failure, Communications Server does not reactivate the link. You can configure a connection as *auto-reactivate* (infinite retry) to keep important connections active.

Branch Extender

The branch extender is a border node subset that is designed to interconnect a branch office to an APPN WAN backbone network. The interconnected networks can be native (that is, they have the same network ID) or nonnative. A node that supports the branch extender is a branch network node that typically has LAN and WAN interfaces, and can also include DLUR and HPR.

Links at a node that support branch extender are defined *branch uplinks* or *branch downlinks*. Figure 9 provides an example of the way that a branch network node works in a network. In this figure, the node in the center is a branch network node. Usually, the adjacent CP (branch uplink node) will be the network node server (NNS) for the branch network node, which looks like an end node to the branch uplink node.

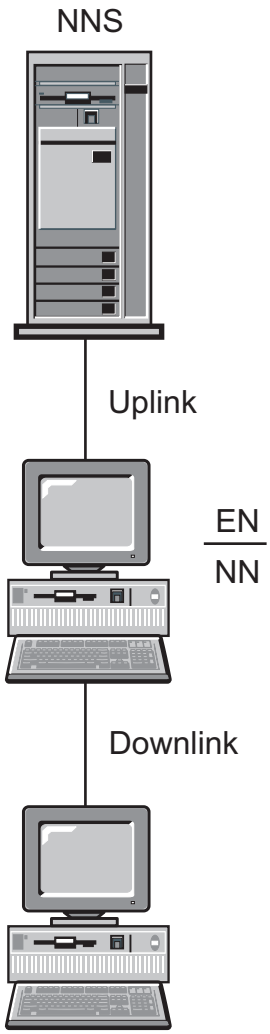

Figure 9. Conceptual Overview of Branch Uplinks and Branch Downlinks

Branch uplinks are defined at the branch network node as upstream to the backbone network. You can consider a node with an uplink to be peripherally attached to the backbone network.

Branch downlinks are defined from the branch network node as downstream. The node sees downlinks as connections to end nodes (control points) in the domain. Branch downlinks are typically LAN links (but are not required to be). You can consider end nodes attached through branch downlinks to be local resources. The branch network node is the network node server for these end nodes. On branch downlinks, it provides network node services for domain end nodes, LEN end nodes, dependent T2.0 nodes and T2.1 nodes, and local LUs and PUs.

A branch network node works as a network node server for its domain. It maintains topological information about all of its branch downlink nodes, but does not maintain complete information about the entire uplink network. If the information the node has is not sufficient, it passes the LOCATE requests to its uplink network node server, which may be another branch network node or an APPN network node.

The branch extender optimizes the peer-to-peer communication environment for administrators who want to connect LAN-based branches to one large WAN primarily based on a switched network. The branch extender enhances performance in large APPN networks. Specifically, it:

- Reduces the number of network nodes in large APPN networks, enabling you to add additional branch networks
- Hides branch topology information
- Enables peer-to-peer communication between branches connected to the same APPN connection network
- Enables coexistence with PU gateway servers
- Reduces uplink CP-CP session traffic (WAN traffic)
- Isolates the branch network from backbone WAN traffic overhead

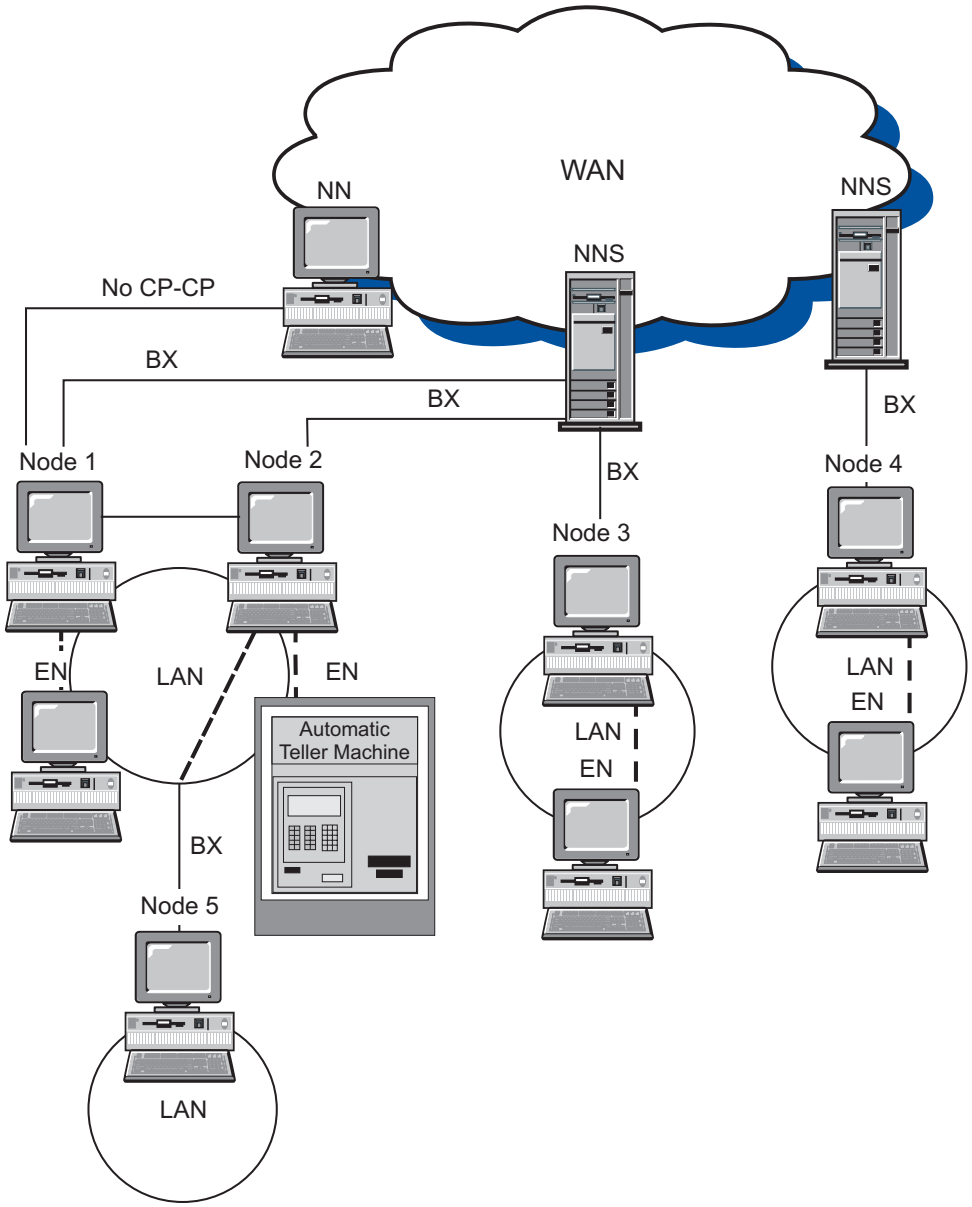

Figure 10. The Branch Extender in a Network

Figure 10 shows how branch network nodes work in a network. Dashed lines represent logical links. BX refers to the branch extender uplinks. In the figure, nodes 1, 2, 3, 4 and 5 are configured to support the branch extender and function as both end nodes and network nodes. They hide their downlink topology from the WAN network. To the upstream network node servers, they appear as end nodes. To the nodes on the LANs downstream, the nodes function as network node servers. Node 5 appears as an end node to Node 2, but is a network node server (NNS) for other end nodes on its LAN. To the NNS, Node 5 appears to be an LU on Node 2.

When an end node served by Node 1 attempts to establish a session to an end node across the WAN, Node 1 can send a Locate (Send) request on its branch extender link to its network node server. If the target CP is found, the network

node server determines a route from Node 1 to the target CP. Node 1 will modify the route before returning it to the source end node. The source end node uses this route for its session.

Branch Extender restrictions

The following restrictions apply to networks configured to use the branch extender:

- A branch network node can only connect to a network node over a branch uplink.
- To prevent search looping, downlink end nodes must be real end nodes, not nodes presenting an end node image.
- A node cannot have CP-CP sessions with the branch network node through both branch uplinks and branch downlinks simultaneously.
- The branch network node can have CP-CP sessions over only one branch uplink at a time; that is, it can have CP-CP sessions with at most one network node server.
- A branch extender node cannot have DLUR sessions downstream.

Branch Extender configuration

To configure the branch extender, you must first configure a branch network node. Then you must configure either a DLC (for an implicit link) or a branch network node. A link configured to support the branch extender is a branch uplink. On a branch network node, any links that are not configured to support the feature are branch downlinks.

If branch network nodes have links defined between one another, they must be defined to be peer connections, which give them a link type of **LEARN** in the ACG file. Alternatively, you can define the link as **ACTIVATE_AT_STARTUP=1** so the link is always active. When branch nodes connect to each other, loops in the topology occur. This is acceptable as long as the links are always active or the links are learned as they are activated. If links are defined between branch network nodes as **END_NODE** or **NETWORK_NODE** links, the topology reported upstream might interpret the links inappropriately and cause allocation failures.

You can also use an ACG file to configure branch extender.

Branch Extender administration

The following sections describe how to verify a configuration and restrictions on how you configure your network.

Verifying the configuration

You can use **SNA Node Operations** at a node that supports the branch extender to determine whether a local branch has been configured successfully. At run time, a display of the topology from the node should never have more than two network nodes, itself and the uplink network node server.

Note: Only one uplink is available for CP-CP. Each workstation should be configured as an end node with the branch network node defined as its preferred network node server.

You can use **SNA Node Operations** to verify whether the DLC or link has been configured to support branch extender correctly. You can also use **SNA Node**

Operations to determine whether an active link is a branch uplink or branch downlink. Downstream end nodes registered using AnyNet will not register their resources.

Supported functions

Communications Server supports all the APPN Version 2 base functions (both end node and network node). Additionally, the following options are supported:

- End Node

Function Set	Option
162	MS_CAPS Be Sphere of Control (SOC) EN
171	Problem Diagnosis Data in Alert
177	LAN Alert
178	SDLC/LAN LLC Alert
181	X.25 Alert
182	Held Alert for CPMS
1002	Adjacent Link Station Name
1007	Parallel TGs
1011	Multiple Local Independent LUs
1012	CP=LU
1018	Delete EN Resources Before Registering
1067	Dependent LU Requester
1070	Session Cryptography
1107	Central Resource Registration (of LUs)
1116	DLUS-Served LU Registration
1200	Tree and TG Caching
1400	HPR Base (ANR)
1401	Rapid Transport Protocol
1402	Control Flows over RTP
1520	Common Operations Services
2355E	Contention Resolution
N/A	Contention Network Reachability Awareness

- Network Node

Function Set	Option
164	MS_CAPS Have a Subarea Focal Point
171	Problem Diagnosis Data in Alert
177	LAN Alert
178	SDLC/LAN LLC Alert
181	X.25 Alert
182	Held Alert for CPMS
1002	Adjacent Link Station Name
1007	Parallel TGs
1011	Multiple Local Independent LUs
1012	CP=LU
1018	Delete EN Resources Before Registering
1067	Dependent LU Requester
1070	Session Cryptography
1100	Safe-Store of Directory Cache
1101	Preload Directory Cache
1107	Central Resource Registration (of LUs)
1116	DLUS-Served LU Registration
1118	EN TG Vector Registration
1121	Branch Extender Function
1200	Tree and TG Caching
1203	Detection and Elimination of TDU Wars
1301	Nonpaced Intermediate Session Traffic

1400	HPR Base (ANR)
1401	Rapid Transport Protocol
1402	Control Flows over RTP
1520	Common Operations Services
2355E	Contention Resolution

Data Link Control

The DLC provides the protocols necessary for reliable delivery of basic transmission units (BTUs) between a pair of nodes in the APPN network and maintaining the logical connections between nodes.

Connections

A connection links a pair of adjacent nodes across the underlying DLC.

Parallel links

Your local node can have multiple links to an adjacent node. This association is referred to as *parallel links*. Parallel links are each assigned a unique number (transmission group number) and can have different link characteristics assigned to them. To have two parallel links between two nodes, the link stations for the links can be on a single adapter in one node but must be on separate adapters in the other node; that is, the combination of adapter number and adjacent (or destination) link station address must be unique for each link.

Link activation

The message unit that is used to convey node and link characteristics to an adjacent node is referred to as an *Exchange Identification* (XID). If **USE_PU_NAME_IN_XID=1**, the PU name is used in the CP name field of the XID. This can only be specified on LEN links that do not support CP-CP sessions. Otherwise, the control point name is used in that field. XIDs are exchanged between nodes before and during link activation to establish and negotiate link and node characteristics, and after link activation to communicate changes in these characteristics.

APPN nodes exchange XID format 3 (XID3) with other T2.1 or boundary nodes to perform role negotiation. For PU 2.0 connections, you use the **LINK_STATION** keyword to specify a PU name and node ID that are exchanged on XID3. If USE_PU_NAME_IN_XID=1, the PU name is used in the name field of the XID. Otherwise, the control point name is used in that field. Information about the sending node's characteristics is contained in the XID3, including link station role (primary, secondary, or negotiable), TG number, node type, logical link number, the maximum basic transmission unit size that can be received, node ID, and PU name. The PU name is normally the control point name, but alternate PU name and node ID can be specified on the **LINK_STATION** keyword to support simultaneous PU 2.0 attachments.

Link types

The following five types of links are typically defined in Communication Server nodes:

- Primary network access, initially activated (APPN or host)
- Primary network access, host link
- Primary network access, in use full time (use when required)
- Secondary network access, inbound link (predefined link characteristics)
- Secondary network access, limited use (also referred to as limited resource link)

Communications Server configuration provides a way to define and control the use of these link types. This section describes the node, DLC, and link configuration parameters used as well as the relationships and dependencies that exist between these parameters. The following categories are described:

- Link definition and activation parameters
- Link deactivation parameters
- Other link parameters

The following parameters are described:

- **Definition and activation parameters**
  - Activate at startup
  - Activate on demand
  - Automatic link retry
  - Maximum activation attempts
  - User requested reactivation
- **Deactivation parameters**
  - Inactivity timeout
  - Limited resource
- **Other link parameters**
  - Connection network
  - CP-CP session support
  - High performance routing (HPR) support
  - Adjacent node type
  - Preferred network node server
  - Solicit SSCP sessions
  - PU name (when solicit SSCP sessions is **1**)

Note: These parameters are in the ACG file and might not be available on the panels (through **SNA Node Configuration**).

Link definition and activation parameters

This section describes definition and activation parameters.

Activate at startup

An *activate at startup* link is typically used for primary network access links that are initially activated when you start Communications Server. The link is activated when Communications Server is started at your machine and stays active as long as Communications Server is running.

To define a link to activate at startup, specify **ACTIVATE_AT_STARTUP=1** on the **LINK_STATION** keyword of the ACG File. Links are generally configured to activate at startup when they are important for network connectivity. An important link can also be configured for automatic link retry (see "Automatic link retry" on page 30).

The link from an APPN end node (EN) to its preferred network node (NN) server is an example of this type of link.

Examples of a non-activate at startup links, coded as **ACTIVATE_AT_STARTUP=0**, could be a host link that is not needed immediately when Communications Server is started, an inbound link that a node would like to control the link characteristics of, or a link that may be too costly to have active all the time. These links are defined as non-activate at startup and are activated when the link resources are requested by an application (see "Activate on demand" or the partner.

Activate on demand

An *activate on demand* (also referred to as "auto-activate") link is typically used for access to a partner LU which requires dynamic activation of the link. When Communications Server is started, the link remains inactive. However, it is placed into the topology as an available link if an adjacent CP name is specified. The link is activated when a transaction program (TP) requests a connection to a remote LU that requires the link to be active. Communications Server uses the fully qualified partner LU defined to activate the link.

To define a link as activate on demand, **ACTIVATE_AT_STARTUP=0** and **FQ_ADJACENT_CP_NAME=(netid.cpname)** must be configured on the **LINK_STATION** keyword in the ACG file of the originating node. If the partner is not the **FQ_ADJACENT_CP_NAME**, configure the **PARTNER_LU** keyword.

Frequently, an activate on demand link is also configured as either a limited resource (see "Limited resource" on page 32) or with an inactivity timeout (see "Limited resource" on page 32) so that the link will be deactivated when it is no longer required.

An example of an activate on demand link is a link defining a connection to a partner which needs to be active for a limited amount of time. The link may cost more than you are willing to pay to keep active at all times. For example, you might have a collection of one or more computers communicating on a regular basis. At the end of each day, one of the machines is required to activate a link to some remote machine, in order to send the daily results, or to make a backup of the data.

Another example might be when you have connections to a data server or a print server. The connection requires resources at the server. To avoid limitations on the maximum number of link stations and sessions at the server, configure an activate on demand link to free the resources at the data server after the requests over the activate on demand link are complete.

An activate on demand link is not necessarily a limited resource link, but it might be defined as one by including the **LIMITED_RESOURCE=1** parameter on the **LINK_STATION** keyword (see "Limited resource" on page 32).

Automatic link retry

Automatic link retry is an error recovery function that enhances the availability of a failed link. If a link needs to be reactivated (without user intervention) after a failure, automatic link retry can be used. Automatic link retry causes automatic reactivation attempts of the link if the following parameters have been specified on either the **LINK_STATION** or **PORT** keywords:

- **DELAY_APPLICATION_RETRIES**
- **RETRY_LINK_ON_DISCONNECT**
- **RETRY_LINK_ON_FAILED_START**
- **RETRY_LINK_ON_FAILURE**

If one of these parameters is specified on the **PORT** keyword, the values are used by the **LINK_STATION** keyword if the **INHERIT_PORT_RETRY_PARMS** parameter has been specified.

After a successful activation, the interval timer is reset to 0.

It might be beneficial to use automatic link retry on any of the following kinds of links:

- A link to a NN that is activate at startup. If the link fails, only a network node server (NNs) link would be reactivated.
- Host link
- Primary link that has a backup (also normally defined as an activate at startup link)
- Any link that needs to remain active at all times.

Maximum activation attempts

Maximum activation attempts is a link activation parameter that provides a mechanism to prevent dependent LU host traffic (for example, LUA, 3270, an LU 2 gateway and/or DLUR trying to activate a host link for a downstream application) from indefinitely retrying link activation. The **INHERIT_PORT_RETRY_PARMS** maximum activation attempts parameter represents the number of times an activate link request is attempted. After this number of attempts is reached, subsequent requests are rejected until the number of attempts is reset. Dependent LU link activation requests issued after maximum activation attempts is reached are immediately rejected without an actual attempt to activate the link. In this case, a primary return code of X'0003' and a secondary code of X'00000005', DLC retry, with a sense code of X'00000000' is returned. Independent LU link activation requests issued after maximum activation attempts is reached are attempted, but along with the primary X'0003' and secondary X'00000005' return code, a sense code of 081C0001 is returned to indicate the maximum activation attempts limit has been reached.

Note: If a TP, such as an LUA application, is in a loop trying to activate a session to the same host as the dependent LU application, the TP causes the maximum activation attempts number to be exceeded before the dependent application can make its first request.

The maximum activation attempts number is reset as follows:

- When an independent TP activates the link. It is the TPs responsibility to check the sense code and implement a reasonable protocol for link activation that does not infinitely retry the activation.
- When the user activates the link using **SNA Node Operations**.
- When the link is activated from the other direction (for example, by the partner).

If **SNA Node Operations** is attempting a link activation, the maximum activation attempts number is decremented by 1, and the result is ignored. If the limit has been exceeded on a gateway host link, activation of the link will be attempted after 30 minutes if a workstation link has become active. This enables retries to cease, although they are restarted later when the workstations are restarted. When gateway, LUA, or **SNA Node Operations** successfully activates a link, the maximum activation attempts number is set to 0.

To configure maximum activation attempts on a link, the **MAX_ACTIVATION_ATTEMPTS=** n parameter is configured on the

LINK_STATION keyword, where n is the number of attempts from -1–127. A -1 indicates that the value on the **PORT** keyword should be used, and 0 indicates infinite retry.

User requested reactivation

User requested reactivation is when the user requests to reactivate a link from either **SNA Node Operations** or from the command line using, for example, Communications Server.

Link deactivation parameters

The following sections describe deactivation parameters.

Inactivity timeout

Inactivity timeout is an SDLC link deactivation parameter that controls when links are deactivated. The inactivity timeout value specifies the time (in seconds) that a link may be idle before it is deactivated. This is similar to the **LINK_DEACT_TIMER** on the **LINK_STATION** keyword. The difference is the **LINK_DEACT_TIMER** waits for all sessions to end (session count reaches 0) before it triggers the link deactivation. The **INACTIVITY_TIMER** ignores the session count and triggers deactivation after the link has been idle for the specified time.

The inactivity timeout function was implemented to handle the situation where an emulator session, LUA, 3270, or a LEN connection, was accidentally left active for long periods of time. When the node detects no activity over this type of connection for the **INACTIVITY_TIMER** duration, the link is automatically deactivated, regardless of whether or not sessions and conversations exist on the link. By definition, the **LINK_DEACT_TIMER** is considered non-disruptive, but the **INACTIVITY_TIMER** is considered disruptive.

Notes:

1. Inactivity timeout can be used on limited resource or non-limited resource links. If a link is defined as limited resource and conversations remain active, limited resource timeout will not expire, and the link will be deactivated when the inactivity timeout expires. If a link is defined as non-limited resource, inactivity timeout is used to deactivate the link to free resources at the remote end.
2. Currently with HPR, the inactivity timeout is ignored. This is because HPR cannot tell what type of traffic is on the link, and the HPR keep alive protocol generates enough traffic so that the link will never be idle.

To configure inactivity timeout on an SDLC connection, the **INACTIVITY_TIMER**= n parameter is coded on the **LINK_STATION_SDLC_SPECIFIC_DATA** parameter of the **LINK_STATION** keyword in the ACG file, where n is 40–160. To configure inactivity timeout on a link, the **LINK_DEACT_TIMER**= n parameter is coded on the **LINK_STATION** keyword of the ACG file, where n is 0–1000. The 0 indicates no timeout (the link will stay active). The SDLC default is 80 and the default for the link is 10.

Limited resource

A limited resource link is typically used for limited use primary network access links and secondary network access links. A limited resource link is a link that is automatically deactivated when its session count reaches 0. The limited resource

link can be defined as **ACTIVATE_AT_STARTUP=1** or **ACTIVATE_AT_STARTUP=0**. If the link is **ACTIVATE_AT_STARTUP=1**, it is started when Communications Server is started. If the link is **ACTIVATE_AT_STARTUP=0**, it is placed in the topology when Communications Server is started, if the adjacent CP name is specified, and activated when services are requested.

Note: Activate at startup links are not placed into the topology unless they are active.

To configure a limited resource link, **LIMITED_RESOURCE=1** is specified on the **LINK_STATION** keyword of the ACG file. The **LINK_DEACT_TIMER= n** is specified on the **LINK_STATION** keyword and the **ADJACENT_NODE_TYPE=LEARN** parameter must be specified.

Note: If **CP_CP_SESSION_SUPPORT=1**, the link is not a limited resource link. Configuration verification will flag this as a warning. Active CP-CP sessions will keep the link from deactivating.

LU 6.2 session level timeout

LU 6.2 session level timeout based on LU, partner LU or mode allows you to terminate the LU 6.2 session on completion of the LU 6.2 conversation. The feature is configurable in the Communications Server for Windows ASCII configuration file. The setting is global on all LU 6.2 sessions with the exception of the IBM Service TPs such as the CPSVCMGR session. Refer to the *Configuration Reference* for more information.

Connection networks

Connection networks enable APPN nodes in a LAN or EEDLC to have direct links with each other without requiring logical link definitions at each node. This feature greatly reduces system definition without adding the performance burden of routing all sessions through a network node. It also enables new nodes that are added to the protocol to fully participate in APPC conversations without requiring definition changes at every other node. Note that multiple protocols cannot be in the same connection network. LAN, EEDLC, IPv4 and EDLC IPv6 must each have a separate connection network.

A network node in the connection network assumes that all the nodes in one connection network can have links directly between one another. When calculating the route for a session, the network node considers the direct link and normally selects the direct link as the optimal route. Having calculated the direct route, the network node simply sends the end node the address of the partner to use for activating the link.

The connection network route might not be taken when connection network security is less than required. If the connection network DLC is not secure and a mode like #BATCHSC is used on the **MODE_NAME** parameter, the network node attempts to find a secure route, ignoring the connection network.

If LAN bridges are being used, APPN views the entire bridged LAN as a single logical network. Because links can be activated between any two systems on the LAN, only one connection network is needed. The connection network should be defined at all the APPN systems on the LAN. Similarly, if EEDLC connection networks are being used the entire IP network is considered a single logical network.

A network node learns connection network information during EN registration and APPN directory searches. The network node server then has enough information to calculate a direct connection between the session endpoint nodes without routing through intermediate nodes. Nodes that are being connected must have CP-CP sessions with a network node that has CP-CP connectivity with the other node's network node server.

Only end nodes and network nodes can take advantage of the connection network; links to LEN nodes must still be explicitly defined.

Figure 11 illustrates a sample connection network. This view of a LAN shows a connection network given a name of LOCALNET.IBMLAN. With this type of definition, any EN can connect directly to any other EN as long as NN1 is the active network node server for all the end nodes.

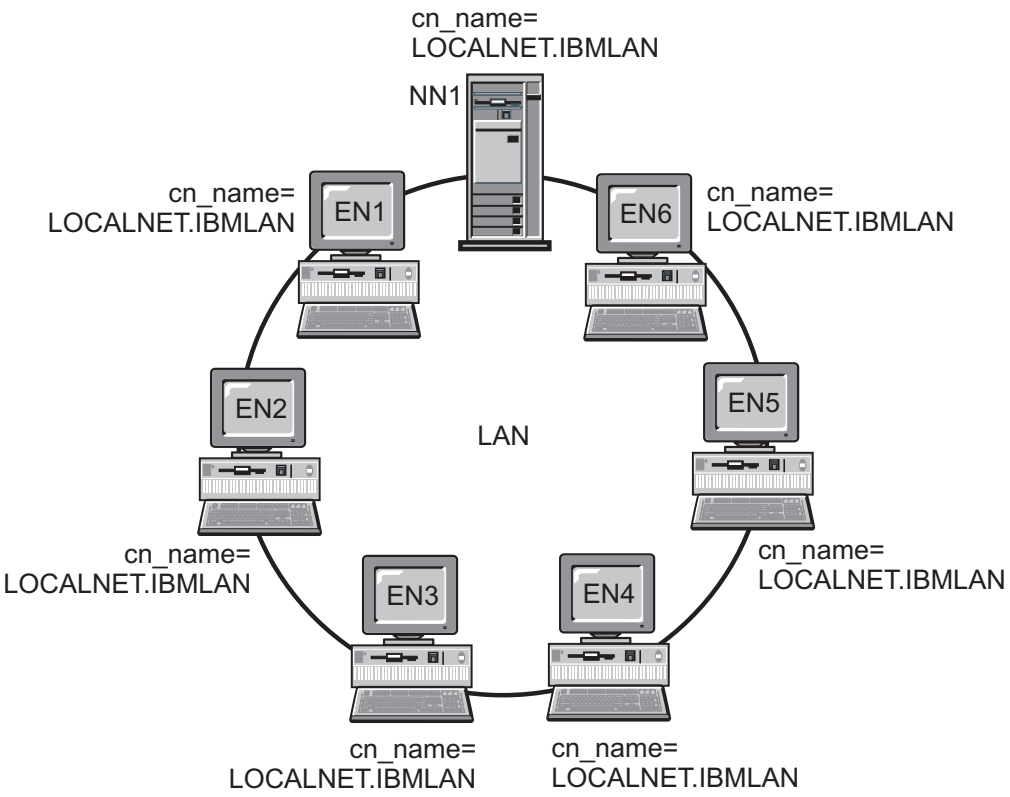

Figure 11. A sample connection network

Other link parameters

Other **LINK_STATION** keyword parameters that are mentioned in the preceding sections, but may not be obvious or implicitly defined are:

- Adjacent node type
- Preferred network node server
- Solicit SSCP sessions

This section contains a brief definition of each.

Adjacent node type

The adjacent node type specifies the type of node that is adjacent to the node defining the link. Valid types include:

- DSPU_NOXID
- DSPU_XID
- END_NODE
- HOST_DEP_LU_ONLY
- HOST_XID0
- LEARN
- NETWORK_NODE
- SUBAREA_LEN

See the *Configuration File Reference* for further details.

Preferred network node server

The preferred network node server specifies whether the adjacent network node is to be used as the network node server over the link being defined.

Solicit SSCP sessions

Solicit SSCP sessions specifies whether or not SSCP-PU sessions are requested from the host over the link being defined.

Directory services

A network node provides directory services to the LUs located in the network node and to the LUs in the end nodes that the network node serves. The network node also assists in the directory services provided by the other network nodes in the network by responding positively to received directory search requests when the resource named is found in the local directory. The local directory maps an LU name to the control point name of the node where that LU is located. If the destination control point is a LEN or end node, the directory includes the name of the serving network node.

The directory services component resides in every node; however, its scope and functions vary depending on the level of directory support in the node.

An end node maintains a local directory containing entries for locally resident LUs. In addition, the end node maintains directory entries for LUs in adjacent nodes with which the end node has been in session. For an LU-LU session with an adjacent peer node, a search of the local directory returns the appropriate destination control point associated with the LU searched for, permitting the proper logical link to be selected.

In a LEN node, all partner LUs are entered in the directory, as shown in Figure 12 on page 36. Those not in an adjacent peer end node but out in the APPN network are associated in its directory with its designated network node server.

The LEN node sends an LU-LU session activation (BIND) request to its network node server for any LU associated in its directory with its server; the server automatically locates the destination LU for it and forwards the BIND appropriately. The network node can send a Locate search, wait for a response, and then send the BIND.

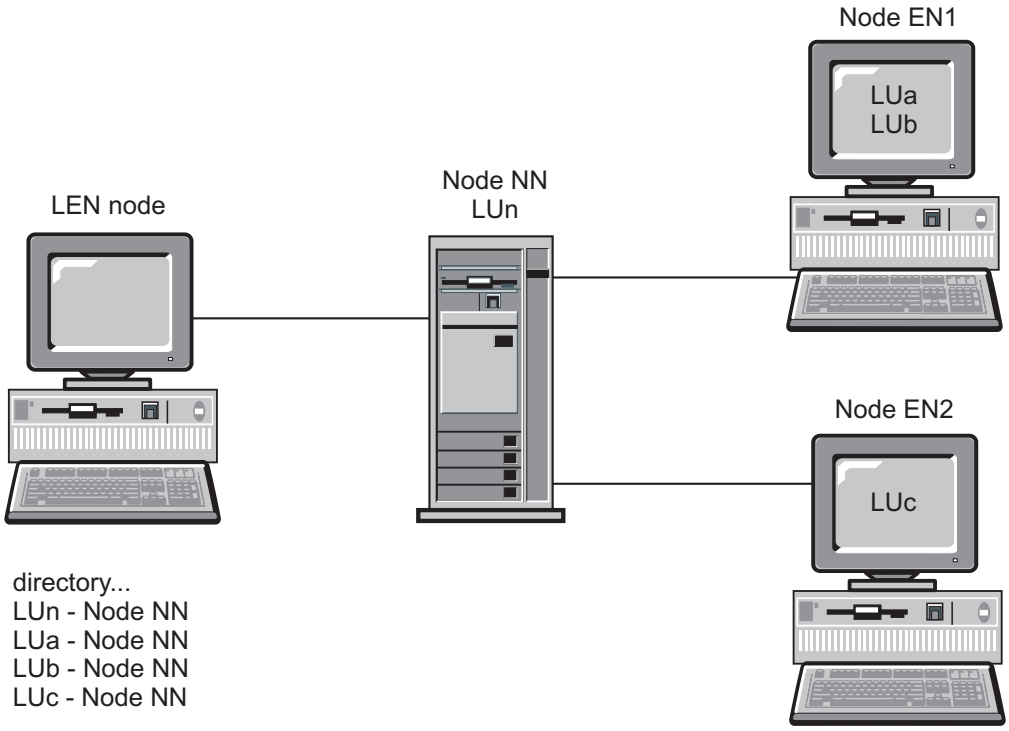

Figure 12. LEN Node Directory

When an LU is not represented in an end node directory, the end node initiates a Locate search to find the desired LU. To activate the search, the end node invokes the services of its network node server. Figure 13 on page 37 shows an example of an end node directory.

The end node (EN) uses the services of its network node server to find the location of the LUs. None of the LUs in the APPN network need to be defined in the end node. The adjacent LEN node LU, however, must be defined because it is not connected to the network node and is not part of the APPN network.

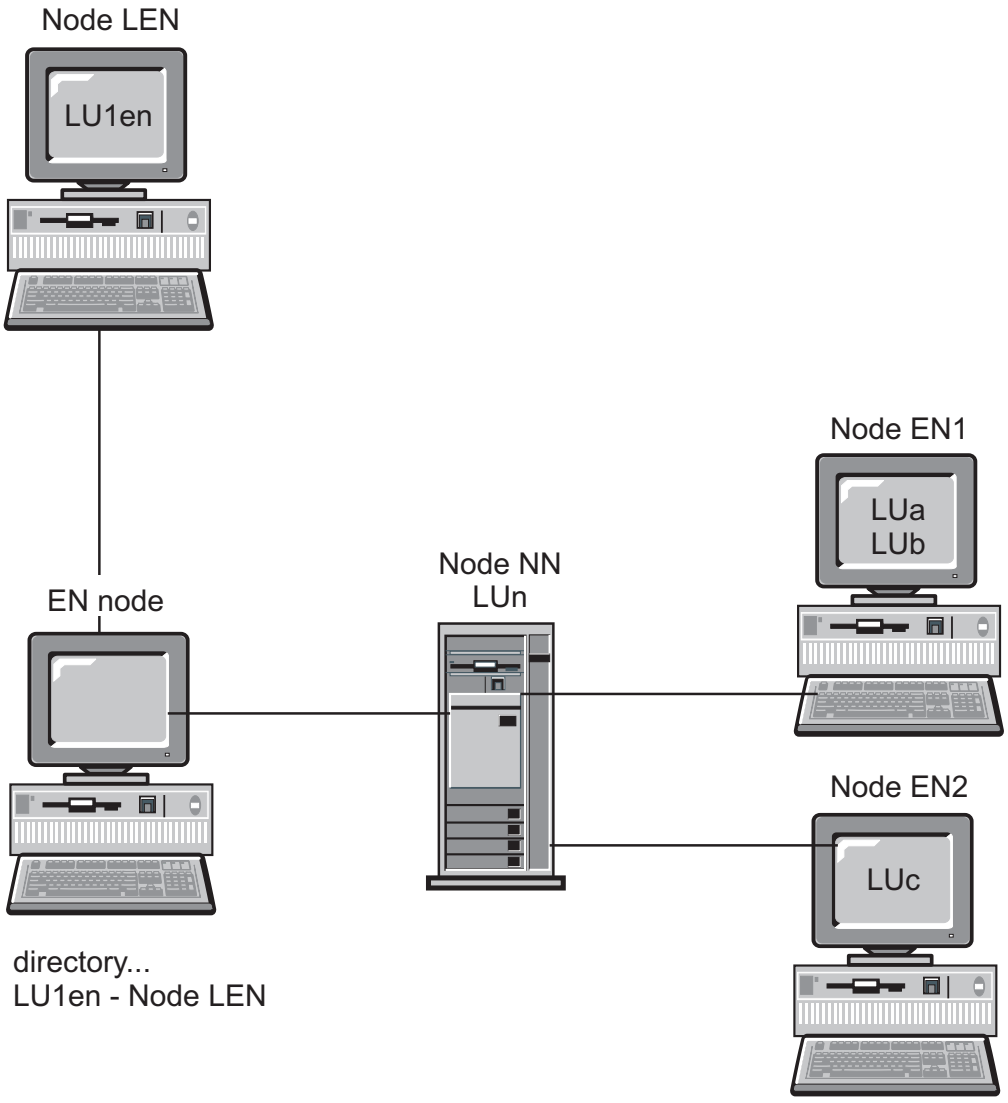

Figure 13. End Node Directory

A network node provides distributed directory services to its served end nodes in cooperation with all other network nodes in the APPN network. The origin network node receives the name of a destination LU in a Locate search request from a served end node, or the name of a secondary LU in a BIND from a LEN node. The network node verifies the current location of the LU if it is represented in the network node's directory (but is not in the network node itself). The verification is done by sending a directed search to the destination network node server.

If the LU is not in the origin network node's directory, the network node initiates a search of the network. The search is initiated by sending a broadcast search to every adjacent network node, each of which in turn propagates the broadcast and returns replies indicating success or failure. For its future needs, a network node caches information obtained from successful broadcast searches.

An APPN end node can also receive (and respond to) Locate search requests from its network node server to search for, or ensure the continued presence of, specific LUs in the end node.

Each end node can register its LUs with its network node server by sending the network node a registration message. If the end node is registered with the network node server, the network node maintains current directory information pertaining to the end nodes in its domain.

Figure 14 illustrates a network node directory.

The network node (NN) directory contains all the LUs it serves. The end nodes (ENs) register their LUs; the LEN node LU must be configured.

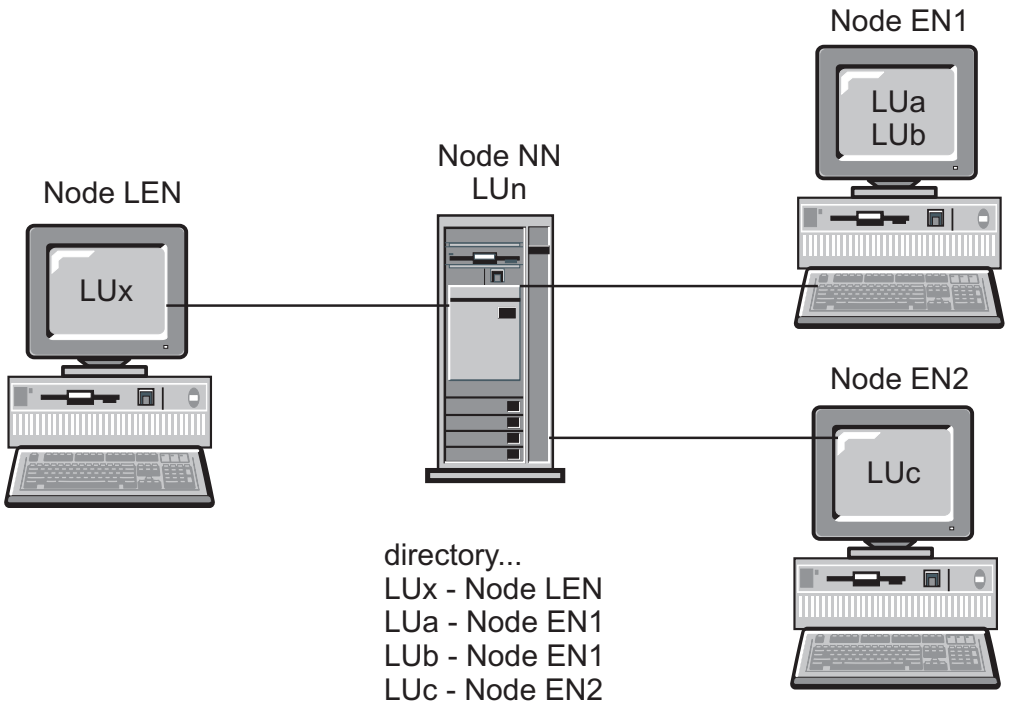

Figure 14. Network node directory

Topology and route-selection services

A network node provides route selection services to itself and to the end nodes it serves. It maintains an internal network topology database that has complete and current topology information about the network. This topology information consists of the characteristics of all network nodes in the network and of all links between network nodes. All network nodes contain a copy of the topology database.

A network node uses the network topology database to compute routes for sessions that originate at the LUs in it and at the end nodes that it serves. Each route that a network node computes is the current least-weight route from the node containing the origin LU to the node containing the destination LU. To provide an appropriate path through the network, the algorithm used to select the route first assigns weights to links and nodes. Based on the relative significance of the characteristics for the requested class of service, the weighting algorithm computes a scalar value for each node and logical link.

Topology database

The network topology database in a network node contains information about all network nodes and all transmission groups interconnecting them. It is a fully

replicated database that is shared among all network nodes in the network and used for route selection. The maintenance of the database requires broadcast updates among all network nodes. The updates are accomplished through topology database update (TDU) messages, which contain node-identifying information, node and link characteristics, and update-sequence numbers to identify the most recent changes for each of the resources described in a TDU.

A local topology database in an end node contains information about itself and directly attached nodes only.

The topology and routing services component uses the CP-CP sessions between network nodes to exchange information to build and maintain a topology database. This topology database in network nodes is kept current using updates that are transmitted among all network nodes whenever a resource (node or link) is activated or deactivated, or the characteristics of an existing resource change.

A local configuration database and a network topology database are maintained at each network node as illustrated in Figure 15 on page 40. The local configuration database is unique to the node, while the network topology database is replicated at all network nodes.

Table 3 shows the information contained in the configuration database at the local network node.

Table 3. Local NN configuration database

Node	Links	Connection
NN5	e	NN5-EN1
	a	NN5-NN7
	b	NN5-NN6
NN7	a	NN7-NN5
	d	NN7-NN8
NN6	b	NN6-NN5
	f	NN6-EN2
	c	NN6-NN8
	g	NN6-EN3
NN8	c	NN8-NN6
	d	NN8-NN7
	j	NN8-EN3
	h	NN8-EN4

Table 4 on page 40 shows the information contained in the network topology database at the local network node.

Table 4. Local NN network topology database

Node	Links	Connection
NN5, NN6, NN7, NN8	a	NN5-NN7
	a	NN7-NN5
	b	NN5-NN6
	b	NN6-NN5
	c	NN6-NN8
	c	NN8-NN6
	d	NN7-NN8
	d	NN8-NN7

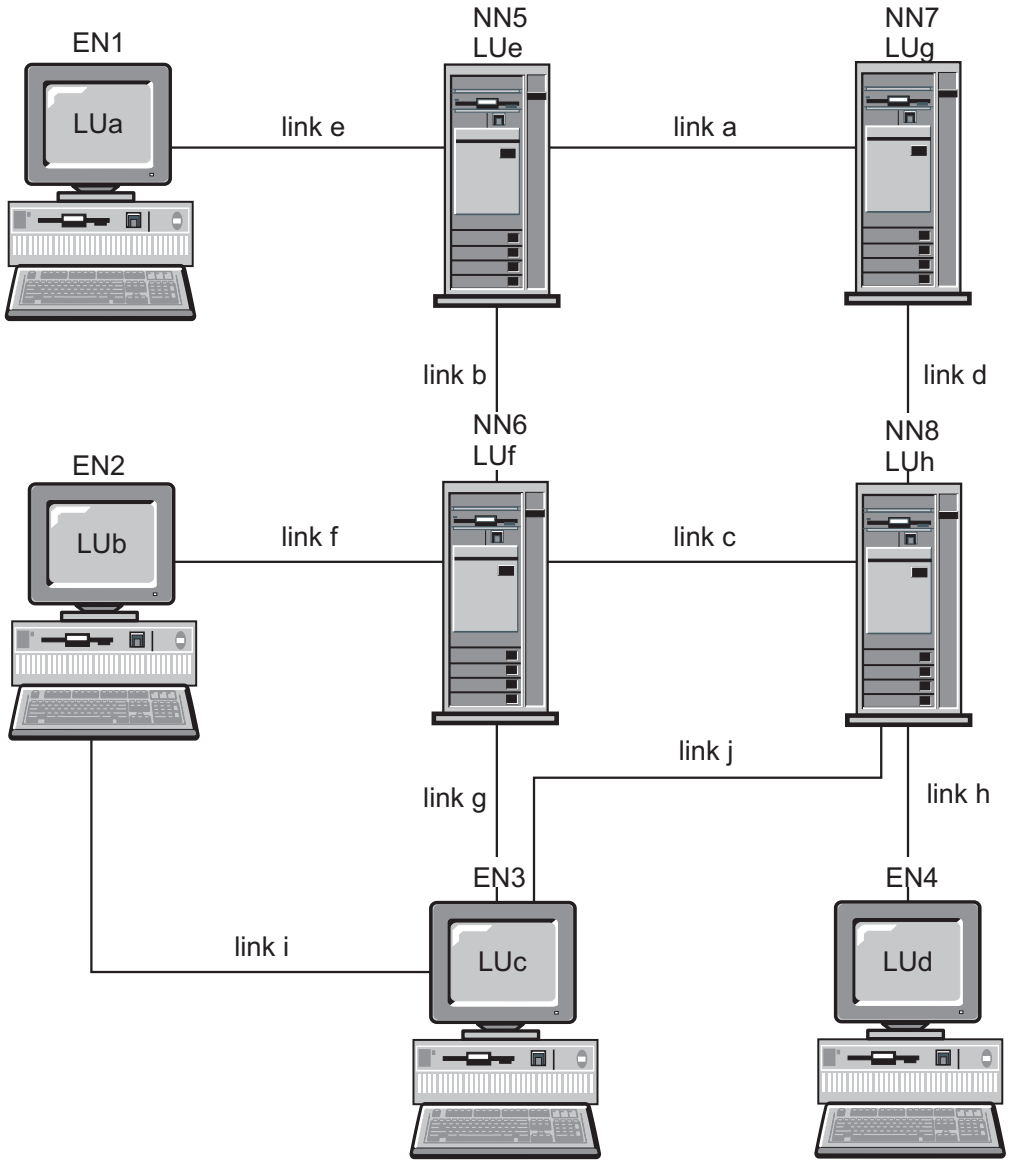

Figure 15. Local configuration database and network topology database in network nodes

Modes

The mode determines the values for the session characteristics and number of sessions between session partners. For example, the size of the largest request unit (RU) to be exchanged on a session (that is, the *maximum RU size*) is one of the characteristics of a mode. The mode also specifies a class of service, which is used to select the route for the session.

Class of Service

At session initiation time, the BIND specifies a mode name. This mode name is associated with a class-of-service (COS) definition that is used to determine the most desirable route between the origin and destination nodes of the session. The COS definitions specify the characteristics that nodes and links must possess to be included in the route selected for the session. This specification enables the route-selection algorithm to determine if a node or link is acceptable. From the set that is acceptable, the algorithm calculates the best route for the session.

Because COS definitions can vary, different sessions can use different routes between the same origin and destination nodes, depending on the specified mode name. Each network node is capable of computing the least-weight (the most desirable) route to any destination.

When a session goes through both an APPN network and a subarea network, it uses two classes of service:

- In the APPN network, the session uses the class of service (COS) defined for the mode used by the transaction program. This definition is in the active configuration of the workstation. This COS is used to route the session in the APPN network.

  If you use the default blank mode, the associated COS is #CONNECT.
- In the subarea network, the name of the COS can be written in the logon mode table, in the entry for the mode name used by the session. This COS name is used by the VTAM program to route the session through the subarea network.

  If you use the default blank mode or if you do not define any COS in an entry of your logon mode table, the COS defaults to the blank entry in the ISTSDCOS table.

In both cases, each network uses the mode name to find the COS name, but the two COS names are not necessarily the same.

SNA transmission priority

The transmission priority is a value specified in the class of service. The transmission priority is sent in the BIND in the Class of Service/Transmission Priority (COS/TPF) control vector. Once the session is established, subsequent session data flows at the transmission priority specified in the COS/TPF control vector.

Data flowing on sessions using a class of service with high priority can pass data on sessions with lower priority. You should give high priority to sessions carrying interactive traffic where response time is important for example, emulator sessions. Sessions carrying high volumes of data, for example file transfers for NetView® Distribution Manager, should be given lower priority. Transmission priority support helps to prevent high volume sessions from blocking traffic on the interactive sessions.

The four transmission priorities are network, high, medium, and low. Network priority is used for network control data such as topology and directory services. The other priorities are used for user data.

Communications Server supports transmission priority for LAN, SDLC, and X.25 links. The benefit is most apparent when the network contains congested low speed links.

Route selection

After the network node server receives a response from its locate search, the topology and routing services component calculates the best route from the origin node to the destination node for the COS requested. Because the topology and routing services component sends and receives topology database updates as characteristics of any resource change, every route is calculated with the most current information.

Route selection for VTAM users

To route APPC traffic through a subarea, the workstations connected to the subarea must be defined as network nodes in Communications Server. In each network node, a link is defined that connects the node to the subarea. From the viewpoint of the network node, partner LUs on the other side of the subarea are defined as being located at the host (a LEN node). From the viewpoint of the host, each network node connected to the subarea must be defined to the VTAM program with a PU macro. All destination LU 6.2 logical units within the APPN network for a particular connection are defined under the PU (network node) as if they are actually located at the PU. However, the LUs can actually be located at other nodes within the APPN network connected to the network node. The host sees only the network node PU. The network node PU can also be a gateway PU. SETN traffic (**CP_CP_SESS_SUPPORT=NO**) is not allowed when the parameter is set to YES on the NCP and there are no PU or control point sessions.

If the PU name in the VTAM definition is the same as the control point name defined in Communications Server, be aware that you will not be allowed to define the control point as an LU in the VTAM definitions. Names must be unique in the VTAM program, whether they are PU or LU names.

The PU macro must contain **XID=YES** to use an XID exchange during activation of the PU. This parameter is coded in the NCP major node. It must not be in the PU statement of a switched major node.

For switched SNA devices, you can use a new parameter in the PU macro: **CPNAME**=cccccccc. It specifies the control point name of the network node connected to the subarea. Either **CPNAME** or **IDBLK** and **IDNUM** must be specified on a switched PU definition statement. Both can be specified. The network node provides its control point name to the VTAM program in the XID exchange during the connection sequence. The VTAM program uses the control point name to locate the corresponding PU macro. If there is no PU macro with the corresponding control point name, the VTAM program uses **IDNUM** and **IDBLK** to locate the PU macro.

To route APPC traffic from an APPN network through the subarea and out to another portion of the APPN network, the network name (**NETID**) of the owning VTAM must agree with the network ID of the APPN network. In Communications Server, the network ID (of the network node connected to the subarea) can be found using the SNA local node characteristics profile.

Intermediate session routing

Intermediate session routing is a function performed by a network node. This capability enables a network node to receive and route data destined for another node. The origin and destination of the data can either be an end node, a network node, or a LEN node. The piece of the session between two adjacent nodes is called a session stage.

High Performance Routing (HPR) support

Communications Server supports high performance routing (HPR) over Enterprise Extender (IP), synchronous data link control (SDLC), LAN, WAN, channel, Multi-Path Channel (MPC), and X.25 connections. HPR automatic network routing (ANR) minimizes the storage and processing requirements in intermediate nodes, which is a better solution than APPN intermediate session routing (ISR) for high-speed networks with low error rates. HPR improves SNA routing with these major features:

- Stateless intermediate nodes

  Stateless intermediate nodes have no knowledge of the connections that pass through them.

  An HPR intermediate node does not need any control blocks or buffer pools to support the connections that pass through it. It also does not need routing tables because the routing information is carried in each packet. Using ANR, an HPR intermediate node simply receives a packet, examines the packet header to find an identifier for the logical link to the next node and sends the packet on that link.
- An HPR intermediate node:
  - Does not recover packets because the HPR connection end points will detect and recover lost packets using the rapid transport protocol (RTP).

    If your LAN has high error rates you can also choose link-level error recovery protocol (ERP) support. However, this should not be necessary on links with low error rates.
  - Does not disassemble segmented packets and re-segment them again because the HPR connection end points will automatically choose a packet size supported by all intermediate nodes in the route.
  - Does not actively manage buffers and prevent congestion because the HPR connection end points employ flow control algorithms that prevent congestion at intermediate nodes.
  - Does not determine a packet's route by performing a table search because the HPR connection end points perform source routing and build the packet routing information using labels chosen for optimal performance by each intermediate node.

  To achieve stateless intermediate nodes:
  - The end points of the route must have buffer capacities that reflect the combination of route delay and application throughput, a value that depends on your network configuration.
  - The data links must support I-field sizes large enough to contain the HPR headers and still carry application data efficiently.
  - The data links must be very reliable because a high error rate will adversely affect the HPR throughput.

For configurations where these conditions cannot be met, APPN without HPR is still a valid implementation choice. HPR technology is an APPN option and APPN without HPR is still available and supported.

- Congestion avoidance

  An HPR network can achieve very high link utilizations. Using a technique called automatic rate based (ARB) flow control, HPR connection end points automatically sense network congestion and reduce transmission load. This avoids lost packets that, in other routing algorithms, may be caused by the lack of available buffers in intermediate nodes.

- Automatic rerouting without disruption

  When an HPR connection end point senses that a route has failed, it automatically computes a new route, reestablishes the connection, and recovers any packets lost due to the failure. Because HPR intermediate nodes are stateless (that is, they have no knowledge of the connections that pass through them), they do not participate in this recovery protocol. Because no packets are lost if the route is switched successfully, applications using the route are not disrupted by the failure.

- Software compatible with existing hardware

  HPR is a software extension of APPN; you can use HPR on your existing LAN or WAN adapters and frame relay links. Because HPR does not attempt to provide dedicated bandwidth and guaranteed real time transport (leaving this level of function to Broad Band Networking Services), the performance constraints on packet forwarding delays are the same as for APPN without HPR. The control point protocols used by HPR are those of base APPN with minor additions. HPR nodes can be installed incrementally in an existing APPN network without preplanning or coordination.

Rapid transport protocol

RTP is a set of message formats and protocols designed to utilize modern data communication media, minimize overhead in intermediate nodes, and automatically switch paths when a link in the path fails.

RTP connections are established within an *HPR subnet* and are used to transport session traffic. An HPR subnet is the portion of an APPN network that is capable of establishing RTP connections and transporting HPR session traffic. RTP connections can be thought of as *transport pipes* over which sessions are carried. These connections can carry data at very high speeds using low-level intermediate routing and minimize traffic over the links for error recovery and flow control. These flows are managed by the RTP connection endpoints.

An RTP connection's physical path can be switched automatically to reroute data around a failed node or link without disrupting the sessions. Data in the network at the time of the failure is recovered automatically.

RTP does error recovery on an end-to-end basis, rather than on a link-level basis. Performance is improved by reducing the number of flows required to do error recovery. Link-level error recovery protocols (ERPs) are also supported for all connections. ERP is a method of detecting a lost packet at one end of a link and recovering by asking the other end of the link to retransmit the packet. If ERP is used, HPR packets are sent as numbered information frames (I-FRAMES). When a frame is lost, the DLC detects the loss and the sender retransmits the frame. If ERP is not used, HPR packets are sent as unnumbered information frames

(UI-FRAMES). When a frame is lost, the DLC cannot detect the loss and HPR rapid transport protocol (RTP) must detect and recover lost packets at connection end points.

In either case, RTP always detects and recovers lost packets at connection end points. For any given connection, there are no restrictions on the number of links that use ERP, or do not use ERP.

ERP can be enabled or disabled on a link-by-link basis. Because RTP detects and recovers lost packets at connection end points, you can use either ERP links or non-ERP links when you build the network. This lets you specify link-level ERP on links that have a high rate of packet loss and maximize throughput on other links by specifying that they do not use link-level ERP. In general, the use of ERP is not recommended in LANS.

Note: ERP is always enabled in a wide area network (WAN) environment.

Flow control and congestion control are also done by RTP on an end-to-end basis. RTP uses a technique called automatic rate based (ARB) flow control to fully utilize network bandwidth when possible. RTP increases the rate at which packets are sent when the network supports this increased send rate. Congestion is automatically recognized and the send rate will be decreased accordingly when congestion occurs. The configured effective capacities of links in the connection path are used to determine both the initial send rate and the send rate increment.

Support for control flows (CF) over RTP connections is now available with HPR in Communications Server. Previously, control flows, including CP-CP sessions and route setup messages, used APPN connections while the data flows used HPR connections. Now, both control flows and data flows can use RTP connections. The benefits of this support include automatic path switching for CP-CP sessions.

Control flows automatically flow over RTP if both endpoints of the connection support this function.

Automatic Network Routing (ANR)

Automatic network routing (ANR) is a stateless routing technique enabled by RTP where a message arrives with a label that uniquely identifies the next hop in the path. Because of its simplicity, ANR can be performed at a low level with no knowledge of the connections using the path. ANR minimizes cycles and storage requirements for routing packets through intermediate nodes.

The ANR fast packet switching function improves performance in intermediate nodes by routing at a lower level than APPN and performing error recovery, segmentation, flow control, and congestion control at the end node, rather than on the intermediate node.

Intermediate ANR nodes are not aware of the SNA sessions or the RTP connections. Routing information for each packet is carried in a network header with the packet. Each node strips off the information it has used in the header before forwarding the packet, so the next node can find its routing information at a fixed place in the header. There is no need to keep the routing tables for session connectors as in base APPN, so switching packets through nodes can be done more quickly.

LU support

SNA defines LU types 0, 1, 2, 3, 4, 6.0, 6.1, 6.2, and 7. LU types 0, 1, 2, 3, 4, and 7 support communications between application programs and different kinds of workstations. LU types 6.0 and 6.1 provide communications between programs located at type 5 subarea nodes. LU type 6.2 supports communications between two programs located at type 5 subarea nodes or type 2.1 peripheral nodes, or both, and between programs and devices.

Communications Server supports LU types 0, 1, 2, and 3, which support communications with host applications that support devices such as:

LU type 0
: 4690 and retail 4700 financial terminals

LU type 1
: 3270 printers

LU type 2
: 3270 interactive displays

LU type 3
: 3270 printers

Communication occurs only between LUs of the same LU type. For example, an LU 2 communicates with another LU 2; it does not communicate with an LU 3. Communications Server also supports LU type 6.2 or APPC.

The Communications Server SNA functions enable applications to use the APPC application programming interface (API) to provide a distributed transaction processing capability in which two or more programs cooperate to carry out a processing function. This capability involves communication between the two programs so they can share local resources such as processor cycles, databases, work queues, and physical interfaces such as keyboards and displays.

Communications Server supports APPC through the APPC APIs. Refer to the following publications for more information:

- *Client/Server Communications Programming*
- *System Management Programming*

The following Communications Server functions support a range of LU types:

- SNA Gateway

  The SNA gateway enables IBM System/370™ (S/370) hosts, or zSeries architecture host computers, or both, to support LAN-attached workstations that use LU types 0, 1, 2, 3, or 6.2 for communicating through an SNA gateway workstation.

  The SNA gateway also supports LU types 1, 2, or 3 to iSeries host computers, which can process the data or pass it through to S/370 or zSeries architecture host computers.
- Conventional LU Application (LUA)

  The LUA consists of system software that supplies service routines to support LUs 0, 1, 2, and 3.

SDDLU support

The self-defining dependent LU (SDDLU) support enables you to dynamically define and activate a dependent LU at the host (VTAM). In VTAM, this is known

as dynamic definition of dependent LUs (DDDLU). SDDLU is enabled in Communications Server by coding an LU_MODEL statement on an LU definition.

To enable the DDDLU facility in VTAM, code the LUGROUP operand on the PU definition statement for the PU, and code an LU group major node. To use the IBM-supplied SDDLU exit routine that generates the LU names for you, you should also code the LUSEED operand on the PU statement.

The LUGROUP operand specifies the name of the model LU definition group that VTAM will use when dynamically defining LUs for this PU. The LU group major node contains the model definition statements. Dynamic definitions for LUs are built using the model LU definitions contained in this major node.

The LUSEED operand provides a pattern name that is used with the SDDLU exit routine to create a name for the dynamically created LU. Once the correct statements have been added to the PU statement and the LU group major node coded, these major nodes need to be active for the SDDLU function to be enabled.

Dependent Logical Unit Requester support

Dependent Logical Unit Requester (DLUR) is an architecture intended to provide dependent LU support in an APPN network. Communications Server supports all base DLUR functions and the following optional functions:

- Self-defining dependent LU (SDDLU) Support

  By using SDDLU support the DLUR-supported LUs can be dynamically defined to VTAM, thus eliminating the need to have these LUs predefined.
- TakeOver/GiveBack Support

  This support enables LU-LU sessions to remain active even when the connection between the DLUR and DLUS nodes has failed. The connection can be reestablished with the same or another DLUS, without the LU-LU sessions ever being disrupted.
- MultiSubnet Support

  This support enables the DLUR node, DLUS node, and the node containing the application to be in different subnetworks.
- DLUS-served LU registration

  An end node DLUR registers its LUs so that the network node can locate these LUs without having to pass the locate requests to the DLUR.
- Backup DLUS Support

  When the connection to the primary DLUS fails or does not become active, Communications Server will automatically try to establish a connection with the backup DLUS.

Using DLUR

To use the DLUR function, you configure a **DLUR_DEFAULTS** definition and use the link name from that definition as the host link for your LUA, dependent LU 6.2, or gateway definitions. Communications Server sends the PUNAME, CPNAME, and NODEID to the DLUS. The PUNAME is sent as part of the signalling information (CV X'0E').

- If the DLUS is uplevel (supports checking of the CV X'0E' on REQACTPU) it uses the PUNAME in its search algorithm. This is available in VTAM 4.3 with PTF or higher.
- If the DLUS is downlevel, it ignores the CV X'0E'.

- If there is no match on PUNAME or the DLUS is downlevel, the DLUS then attempts to locate a PU with matching CP name or NODEID (IDBLK/IDNUM) either predefined in a VTAM switch major node or dynamically created using the ISTEXCCS exit.

Connections to the network using the connectivity of your choice (Token Ring, SDLC, AnyNet, and so on) must be configured and active before the DLUR-to-DLUS connection can be established. Once an APPN connection exists between the DLUR and DLUS, a pair of control sessions are established between the DLUR and DLUS using a special mode, CPSVRMGR. This pair of control sessions is also referred to as the CP-SVR pipe and appears as a link to Communications Server. It can therefore be activated, deactivated, and displayed using **SNA Node Operations**.

Once the pipe is activated, SSCP-to-PU and SSCP-to-LU support can be provided to PUs and LUs that have defined the pipe as their host link. LU-to-LU sessions do not use the pipe, but will use the best path available through the network.

In the DLUR environment, any number of dedicated PUs can be defined on the LU 6.2 sessions. This enables the gateway to provide network management access through the dedicated PU to downstream workstations without requiring numerous physical links to the hosts. Figure 16 illustrates a Communications Server workstation acting as a DLUR gateway for both a workstation and a 4702 controller.

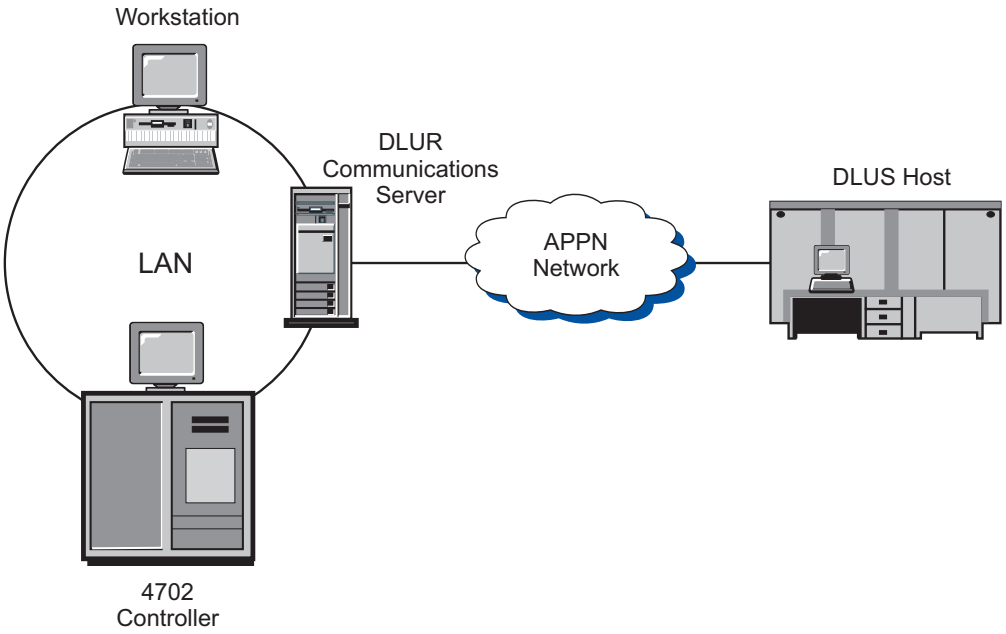

Figure 16. DLUR connection to a host through a Communications Server gateway

LU-LU sessions

Communications Server LUs can both initiate sessions and respond to session initiation requests. An LU initiates and responds to requests according to the type of LU: independent or dependent.

Independent LU

An independent LU is able to activate an LU-LU session (that is, send a BIND request) without assistance from the SSCP; therefore, it does not have an SSCP-LU

session. An independent LU is capable of sending and receiving BINDs. The BIND sender is referred to as the primary LU (PLU); the BIND receiver is referred to as the secondary LU (SLU).

Only an LU 6.2 can be an independent LU. Communications Server supports independent LU protocols to other type 2.1 nodes as well as to low entry networking-level type 5 subarea nodes.

Independent LUs can have parallel sessions between the same pair of LUs and can have multiple sessions between one LU and several other LUs. Their session limits are established on a mode name basis, which can be from 1 to 32 767.

Figure 17 illustrates how multiple and parallel sessions can be established by an independent LU. LUx supports parallel sessions with LUy and a single session with LUz. The direction of the session arrows shows the PLU-SLU relationship. LUx acts as the PLU for the session with LUz and for one of the sessions with LUy. LUx also acts as the SLU for one of the parallel sessions with LUy.

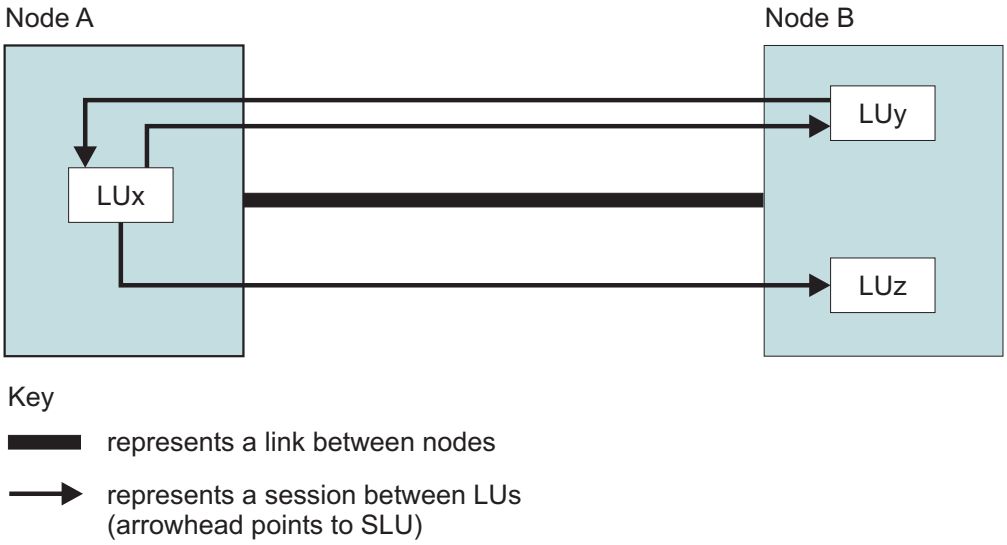

Figure 17. Multiple and parallel sessions

Dependent LU

A dependent LU is an LU that is controlled by an SNA host system. To activate an LU-LU session, a dependent LU requires assistance from an SSCP. It requires an SSCP-LU session to send a BIND. Dependent LU protocols are supported by Communications Server, but only to type 5 subarea nodes using type 2.0 protocols, not to other type 2.1 peripheral nodes. Dependent LUs act as SLUs only and have an LU-LU session limit of 1. However, multiple PU support in Communications Server enables you to establish multiple simultaneous SSCP-PU sessions with dependent LU sessions.

The dependent LU requester (DLUR) function enables Communications Server to take advantage of the enhanced SSCP support provided by a dependent LU server (DLUS). Some of the benefits of this function are:

- The dependent LUs can reside on nodes that are not adjacent to the host.
- APPN search logic finds the best path for LU-LU sessions.
- SNA dependent LUs can gain the advantages of an APPN network.

To use DLUR, configure a **DEFINE_DEPENDENT_LU_SERVER** parameter and use the link name from that definition for your LUA, dependent LU 6.2, or gateway definitions.

LU 6.2

Independent LUs are defined to the VTAM program by coding LOCADDR=0. There can be as many LUs defined with LOCADDR=0 as you want. Note, however, that not all LU 6.2s are independent LUs.

When you define the LUs of one part of the APPN network to the VTAM program, you must define them as being in the network node that connects this part of the APPN network to the subarea network. Following the PU definition of this network node, define each LU that you want to reach from the other part of the APPN network. Do not forget that control points are LUs.

An LU must be defined in VTAM to establish a session with another LU if this session goes through the subarea network. There is no effective way to get around it (such as the wildcards of APPN); VTAM must know the name of each destination LU.

Because an APPN network is intended to change easily, you should define the LUs of the APPN network in a special major node whenever it is possible. You can also define, in VTAM, LUs that do not exist yet.

Other LUs

If the network node uses its connection to the subarea network for 3270 emulation, the LU type 2 LUs of the 3270 emulation are defined in the same PU macro as the LU type 6.2 LUs of the APPN network. The link will also be used for the connection between the 3270 emulation and the host.

APPN network node T2.1 support

APPN is an enhancement to IBM's SNA and type 2.1 (T2.1) node architecture. APPN enables interconnection of systems of widely differing sizes into networks of a dynamic topology. An APPN network is easier to use, is more reliable, and provides more flexibility than traditional SNA networks. Refer to *3174 APPN Implementation Guide* for additional information on APPN network nodes.

Data compression

Data compression is the process of compressing repeated bytes or repeated data strings to shorten the length of records or blocks. This reduces the transfer time needed for communications. By reducing the amount of data transferred between host and workstation sessions, you can increase the throughput on slow speed lines and lower the cost per bit on expensive lines.

The performance gain, as measured by the number of transferred bytes, that you can expect by using data compression is often a ratio of about 2:1. This means that, with data compression active, you save about every second byte in the buffers needed for lower level protocol conversion.

Data compression is beneficial to those who need:

- Improved response times, especially on low-speed lines
- Reduced costs on lines where tariffs are based on volume of data
- Fewer leased lines

However, data compression should not be applied to every session you are running as there are disadvantages you should consider:

- Compression and decompression need additional CPU cycles
- Increased data storage (32 bytes to 9.0 KB) will be needed
- Data compression requires a secure link as defined in your APPN class-of-service (COS) table.
- Tuning of the RU size might be required to obtain the best performance. Each RU is compressed and then sent. If packets are being sent, a larger RU size on the mode will enable more full packets to be sent.

Note: The amount of increased storage depends on the compression algorithm you use. This increase is in addition to the storage required for the message.

For detailed and technical descriptions of different compression algorithms see the following publications:

- *SNA Formats*
- *Better OPM/L Text Compression IEEE Transactions on Communications, vol COM-34, no 12*, pgs 1176-1182,1986
- *IBM ITSC VTAM V3R4 and V3R4.1 Planning Guide*
- *A Technical Guide to ESA/390 Compression*

The following sections describe SNA session-level compression and Communications Server implementation.

SNA session-level compression architecture

SNA session level compression implements data compression in the LU-LU half session. With Communications Server it is available to all supported LU types, that is: LU types 0, 1, 2, 3, and 6.2. Data compression at the session level provides these advantages:

- It is more efficient to compress data before you encrypt it
- Different algorithms are more effective for different types of data
- Application programs do not have to provide their own compression logic

Two algorithms are generally defined for SNA session level compression, Run length encoding (RLE) and a form of Lempel-Ziv (LZ). Communications Server supports SNA session level compression using the following algorithms:

- NO compression.
- RLE compression; RLE is the simplest and most widely known algorithm, replacing strings or identical bytes with shorter encoded strings.
- LZ9 compression; LZ9 is a dynamic compression algorithm that compresses previously seen strings (in the current or preceding RUs) to a 9 bit code which represents the zero origin index of an entry in the compression/decompression table. The table entries store previously seen strings.
- LZ10 compression; LZ10 is a dynamic compression algorithm that compresses previously seen strings (in the current or preceding RUs) to a 10 bit code which represents the zero origin index of an entry in the compression/decompression table. The table entries store previously seen strings.

Typically, LZ compresses data better than RLE, but at a greater cost of memory and CPU function.

SNA session level compression views the session in two directions, PLU-SLU and SLU-PLU. The primary logical unit (PLU) is the LU responsible for activating the session. The secondary logical unit (SLU) is the responding LU. The PLU activates

a session by sending a Bind Session request (BIND) to the SLU, which responds with a BIND. This means that different compression algorithms can be used in the PLU-SLU and SLU-PLU directions. This is performed through BIND negotiation of the compression levels. LU 6.2 can use any combination of compression levels for a session (for example, PLU could use RLE and SLU-PLU could use LZ9). All other LU types have compression enabled or disabled. When enabled, the PLU-SLU compression level is LZ9 and SLU-PLU is RLE.

Communications Server data compression

Communications Server supports SNA session level data compression with RLE, LZ9, and LZ10 compression algorithms. With Communications Server, you can specify the use of data compression for communications over CPI-C sessions (through APPC session), APPC (LU 6.2) sessions, and LUA (LU 0, LU 1, LU 2, and LU 3) sessions.

A two-part configuration is used to enable data compression. The Communications Server node must be enabled for data compression and the LU (APPC and LUA) must first be enabled. The two node compression fields (level and tokens) are on the local node characteristics window (**NODE** keyword in the .ACG file).

The compression level field sets the maximum level that any session can be started with: NONE, RLE, LZ9, or LZ10. This field takes precedence over all compression levels configured or attempted (the only exception is stand-alone DFT, which does not require the node definition). If you are configuring a session using LUA (LU 0, LU 1, LU 2, and LU 3) to support 3270 emulation or printers, LZ9 is required for data compression. The other compression levels do not allow data compression for these LU types.

The default request unit size for compressed modes is two times the connection's basic transmission unit (BTU) size. If you are using packet switching, you might not want to use this default size; instead, use a larger size and segment the packets.

LU 6.2 compression can be enabled by:

- Using the modes supplied with Communications Server
  - #INTERC
  - #BATCHC
  - #BATCHCS
  - #INTERCS
- Changing or adding modes with compression enabled

Three mode compression fields, compression need, PLU->SLU compression level, and SLU->PLU compression level, are displayed on the Mode Definition panel (**MODE** keyword in the .ACG file).

Compression need can have two values.

Prohibited
No compression.

Requested
Use this to request data compression with the values defined in PLU->SLU compression level and SLU->PLU compression level. The requested level might not be obtained for the following reasons:
- Node settings have limited the allowable level

- The SLU has negotiated the levels down

  The SLU honors the compression levels requested by the PLU, unless limited by its node compression settings.

For more information on Communications Server compression, refer to the online help text for the product or the *Configuration File Reference*.

SNA session-level encryption

SNA session-level encryption enables you to encrypt either all of the data or selected data that is transferred between the workstation and the host. If you want to protect any workstation data by using encryption, the host must also be configured to use encryption.

An IBM SecureWay® 4758 PCI Cryptographic Coprocessor (referred to as the IBM 4758) adapter must be installed on the server to enable data confidentiality. This adapter must be initialized by following the instructions provided with the adapter.

In Communications Server, LU 6.2 session-level encryption is configured based on the mode description used for a given transaction program. There are two levels of encryption:

- APPC session level

  Strictly between two LUs, no control points get involved in translating keys
- APPN session key translation

  The control point of the originating LU (EN or NN) and possibly its NN server translate the keys in addition to the originating LU.

To configure a mode for encryption, bring up the SNA Features window and select **MODES**. Then inside the Mode Definition window, select **Setup....** The Compression and Session-level Encryption Support window appears. The parameters for the encryption configuration are in two parts:

- For session level encryption, you can specify whether encryption is optional (the LUs negotiate) or mandatory (encryption must be used).

  Note: Encryption from Communications Server to a host is always set to be mandatory.
- For APPN session key translation, you can specify that only the LU translates the key, that only the end node translates the session key, or that the end node and its network node translate the key.

Communications Server requires other products for key storage and translation. A common cryptographic architecture (CCA) product is required for key storage, managed by the utilities supplied with the IBM 4758 adapter. Communications Server calls a CCA product, which interacts with the IBM 4758 adapter to get the keys and encrypt the data.

VTAM users:

Communications Server does not encrypt the SNASVCMG session. You must specify ENCR=OPT in the APPL statement of your VTAM application definition. In working with VTAM, you must have encryption specified on the MODEENT statement. For example:

```
ENCR=B'0011' FOR MANDATORY ENCRYPTION
```

To use the VTAM encryption facility, the IBM Programmed Cryptographic Facility (PCF) must be initiated before starting VTAM.

Beginning with VTAM V3R4.1, VTAM uses a new interface to Integrated Cryptographic Service Facility/MVS (ICSF/MVS) for cryptographic services, such as providing session-level cryptography. This interface complies with the Common Cryptographic Architecture (CCA) as implemented by ICSF/MVS. With this support, you can start and stop the cryptographic service after VTAM has been started, and you can change the master key without disrupting VTAM or active LU-LU sessions.

For information on how to define data encryption, refer to *z/OS® Communications Server: SNA Network Implementation*.

Management services

Communications Server's management services (MS) are functions distributed among network components to operate, manage, and control a network. This capability is based on the SNA management services architecture documented in *Systems Network Architecture Management Services Reference*.

Focal points, service points, and entry points

Communications Server provides programming support that enables installation of management services focal point (FP), service point (SP), and entry point (EP) applications. Management services SP applications are just a variation of management services EP applications and differ only in the kinds of function they provide; otherwise, they interact with a management services focal point as management services EP applications.

Focal point

A management services focal point is a central point of control for managing a network. From a management services SP or management services EP application, the management services focal point can request certain data relating to the operation of a network such as problem and performance data or product identification.

The management services focal point can also accept certain unsolicited management services data from the nodes it manages based on the category of the management services data. An example of a management services category is MS alerts. A management services focal point can manage one or more categories of management services data, and there can be one or more management services focal points in a network. IBM Communications Server, the IBM NetView program, and the IBM OS/400® operating system are examples of products that provide management services focal point capability.

Service point

A management services SP is the function in a node that can request and capture data from devices that, by themselves, cannot serve as management services EPs, such as devices connected by LAN protocols (but not higher-level SNA protocols) to the management services SP node. Aside from collecting nonlocal data, a management services SP functions like a management services EP in its relation to the management services focal point. The IBM NetView/PC and IBM LAN Network Manager program products are examples of management services EP applications that provide management services SP functions and management services EP functions.

Entry point
A management services EP is the function in a node that captures local management services data and sends it to a management services focal point for processing, either upon request or unsolicited. Communications Server provides the management services EP function for sending alerts to the alert management services focal point. These alerts can originate within Communications Server or the DLCs it uses. Communications Server also provides programming support for applications, such as the IBM NetView/PC and IBM LAN Network Manager program products, by supplying alerts to be sent to the alert management services focal point.

Levels of SNA management services architecture

An SNA product implements a particular level (or generation) of the SNA Management Services architecture, and some products support several levels of the architecture. Communications Server can send management services data to, and receive management services data from, SNA products that implement any of three levels of the management services architecture. These levels are:

Multiple domain support (MDS) level
An SNA product that implements the MDS level of the management services architecture, such as Communications Server and IBM NetView Version 2 Release 2 (or later). It can send and receive MDS message units (MDS-MUs). IBM NetView Version 2 Release 2 provides MDS level as a subarea LU, not a control point (CP), and uses SNASVCMG-mode sessions for transporting MDS-MUs. As a focal point, it supports explicit, implicit (primary) and implicit (backup) FP-EP relationships. NetView Version 2 Release 2 also continues to support the host FP-EP relationship to EP products that do not have MDS-level support.

Migration level
An SNA product that implements the previous level of the management services architecture, such as IBM OS/400 Version 1 Release 3 Modification Level 0 (or earlier). A migration-level product can support explicit, default, and domain FP-EP relationships. The domain FP-EP relationship is inferred when the CP-CP sessions are activated to a migration-level node. A migration-level serving network node (NN) does not send MS Capabilities for FP Notification to its served end nodes (ENs), and a migration-level served EN does not accept MS Capabilities for FP Notification from its serving NN. It can send and receive CP-MSUs but not MDS-MUs. As a focal point, it supports only the alert MS category.

Network management vector transport (NMVT) level
An SNA product that implements the NMVT level of SNA management services architecture, such as IBM NetView Version 2 Release 1 (or later). NMVT is a management services request unit (RU) that flows over an active session between PU management services and control point management services. If an NMVT is routed from a workstation through a gateway, then the gateway adds its control point name to the NMVT.

Flow control

To manage the flow of data over a network, Communications Server uses adaptive session-level pacing. The pacing occurs between each pair of adjacent nodes participating in the session route. The pacing between two adjacent nodes is independent of the pacing used between other adjacent nodes in the route.

Session-level pacing

Adaptive session-level pacing uses a window-based scheme, where a sender can send only a limited number, or window, of request units per explicit grant of permission to proceed. The window size can be changed based on conditions at the receiver. This function permits a node to control the amount of data that is sent and received during normal session operation. The window control enables the receiving node to manage its rate for receiving data into its session buffers. Adaptive session-level pacing provides a node supporting many sessions a dynamic means to allocate resources to a session that has a burst of activity and to reclaim unused resources from sessions that have no activity. Adaptive session-level pacing enables the receiving node to use its available buffer resources efficiently.

Because each session stage between the endpoints is independently paced, both endpoint nodes and intermediate nodes can adapt the pacing for the sessions they handle in accordance with their own local congestion conditions. This action is the basis for global flow control and congestion management in APPN networks.

If, however, an interactive session and a session transferring a large file share a link, the interactive session data should be transmitted as quickly as possible. There are two ways to do this:

1. Assign a lower priority to the file transfer session. #BATCH uses low priority.
2. Use fixed pacing with a small window size for the file transfer session to enable interactive session data to use the link when the file transfer session is waiting for the pacing response. If connected directly to an NCP host, two-way fixed window pacing can be used to set pacing in both directions to the receive window on the defined mode.

Adaptive BIND pacing

BIND traffic can occur in bursts, particularly at node or network startup. Therefore, adaptive BIND pacing exists to control the flow of BINDs between two adjacent nodes. The same window algorithm used for session-level pacing is employed.

Segmenting and reassembly

To transmit RUs longer than the maximum-size basic transmission unit allowed by a particular link, Communications Server supports data segmentation and reassembly. These segments are reassembled into whole RUs at the partner node. This action enables the RU size defined for a session to be independent of the link that is used for the route.

High performance routing pacing

High performance routing (HPR) provides a new method of flow control called adaptive rate-based congestion control (ARB). ARB regulates traffic flow by predicting congestion in the network and reducing a node's sending rate into the network, preventing congestion rather than reacting to it.

Fixed pacing

Fixed pacing enables you to share a physical connection between two sessions. Without fixed pacing, the data to be transmitted is placed on a common data link control (DLC) queue and interactive data follows previously queued data. Fixed pacing also reduces the amount of storage that can be used to place data on the DLC queue. However, in general, adaptive pacing is the most efficient method of data transfer between nodes.

Fixed pacing can be configured on the remote end. Specify the **MAXIMUM_RECEIVE_PACING_WINDOW** keyword on a mode to limit the adaptive pacing window and provide the advantages of fixed pacing.

Transmission priority, like fixed pacing, enables sharing of a physical link between sessions. However, it will lock storage as the data is placed on the DLC queues, but it does not require the additional pacing responses that are required for fixed pacing.

Partitioning LUs among hosts

When you define multiple subarea host connections, it is required that traffic from the domain of a given host enter only on one logical link. Note that manual dial connections appear to be a single link. You must define a different PU to support each different host. Only the host links defined on the control point can have CP-CP sessions and participate in the APPN network. Links that have **USE_PU_NAME_IN_XID=1** cannot have CP-CP sessions. Otherwise, the host links can have CP-CP sessions and can also participate in APPN communication. The **LINK_STATION** keyword specifies the PU name and logical link to be used for the PU. If parallel links are required (in situations where there are more than 254 dependent LUs), one of the links must have CP-CP session support set to No.

Each dependent LU can be defined as associated with only one PU. The **LOCAL_LU** keyword specifies the host link name used for a dependent LU type 6.2. The **LU_0_TO_3** keyword specifies the host link name for LUA, and the 3270 profile specifies the host link for each 3270 emulation session.

The control point automatically defines a PU with the same name as the control point. Dependent LUs that are defined at one subarea SSCP can be associated with the control point PU. (In fact, the automatically defined LU for the control point can also be specified as dependent.) Each additional PU 2.0 for a different subarea host requires a separate link and PU definition. A PU is defined by specifying PU on a **LINK_STATION** keyword. This is unnecessary if all LUs are independent.

If the SSCP-PU session is to send alerts to the host, define the focal point link with the **NODE** keyword in the .ACG file. If a host is to receive the alerts, specify a logical link for the control point to the host. If the link to that host is not available, Communications Server logs the alerts. The only network management requests that are accepted from a PU 2.0 host are the ones for the control point PU. Replies are sent to the same host using the control point PU.

Each dependent logical unit must have a configured local address that is the same as the one configured at the host. However, use of the SNA gateway allows for a gateway address translation. A dependent LU 6.2 can act as an independent LU to a peer node; that is, it is dependent only to the subarea. Such an LU should not be a part of an APPN network with any other connection to the same subarea, that is, the subarea containing the controlling SSCP. Only one subarea connection is permitted for an APPN network, unless the subareas are independent. Although a gateway can have parallel links to the same subarea, only one can have CP-CP sessions and APPN traffic.

An SSCP in the subarea network activates the dependent LUs it controls after the link to it is established. Until an LU is activated, the LU cannot start a session. When APPC is taken down, disconnect is requested from each active host in session with a PU in the node. Each host frees the link after deactivating first the LUs and then the PU on the SSCP-PU session.

SNA gateway support

Communications Server provides a full-function Systems Network Architecture (SNA) gateway. The gateway allows multiple LAN-attached workstations to access System/370 or zSeries hosts through one or more physical connections to one or more hosts. This helps reduce the cost per workstation of host connections.

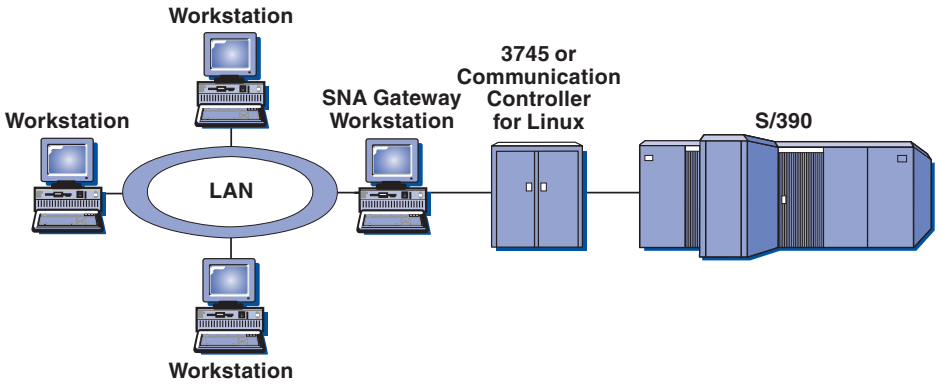

Figure 18. Example of SNA gateway configuration

The Communications Server gateway supports the SNA protocols LU 0, 1, 2, 3, and dependent LU 6.2 (APPC). With the AnyNet SNA over TCP/IP function, downstream workstations can communicate with the SNA gateway over an IP network. The gateway also supports LU 0, 1, 2, or 3 to an iSeries host using SNA pass-through. The iSeries host passes the data through to a zSeries host.

A gateway can also act as a protocol converter between workstations attached to a LAN and a WAN host line.

The LUs defined in the gateway can be dedicated to a particular workstation or pooled among multiple workstations. Pooling allows workstations to share common LUs, which increases the efficiency of the LUs and reduces the configuration and startup requirements at the host. You can also define multiple LU pools, each pool associated with a specific application. And you can define common pools that are associated with multiple hosts. When a client connects to the gateway, the gateway retrieves an LU from the pool to establish a session. The LU is returned to the pool for access by other workstations when the session is ended.

In addition, an SNA gateway can support the forwarding of network management vector transports (NMVTs) between the workstations and the host.

Each host views the SNA gateway as an SNA PU 2.0 node, supporting one or more LUs per workstation. As far as the host is concerned, all LUs belong to the SNA gateway PU. The SNA gateway can have multiple host connections simultaneously and can direct different workstation sessions to specific hosts.

To the supported workstations, the SNA gateway looks like an SNA PU 4 communications controller and forwards such host requests as BIND and UNBIND. The workstation LUs are not aware of the SNA gateway. The SNA gateway, however, is aware of all LUs at the workstations.

Downstream applications using standard SNA connectivity protocols for LU 0, 1, 2, and 3 and dependent 6.2, and communicating through an SNA gateway to a host,

are supported by Communications Server. Table 5 summarizes the SNA gateway features.

Table 5. SNAgateway summary

Feature	Description
Active workstations	254 (LAN) per adapter 128 (X.25)
DLCs	AnyNet (SNA over TCP/IP) LAN (Any NDIS** compliant network adapter X.25 SDLC (synchronous, asynchronous, and AutoSync) OEM Channel (upstream only) MPC Channel (upstream only, requires DLU Enterprise Extender
Downstream workstations	Any product that supports standard SNA connectivity protocols for LU 0, 1, 2, 3, and 6.2.
Dynamic additions and changes	Yes
Implicit workstation support	Yes
LU pooling	Yes
Maximum number of LUs	254 per PU; no limit on the number of PUs
Mode of operation	Multiple downstream PUs (not apparent to the host) PUs not visible to the host (except when through DLUR)
Multiple PU support	Yes
Segmenting support	Yes
Supported LU types	LU 0, 1, 2, 3, and dependent 6.2

Chapter 3. Planning for AnyNet support

Communications Server allows for AnyNet SNA over TCP/IP Access Node and Gateway support.

The AnyNet SNA over TCP/IP function in Communications Server enables SNA applications to communicate over interconnected IP and SNA networks.

The SNA over TCP/IP access node function enables SNA applications residing on an IP network to communicate. This function supports independent LU 6.2 and dependent LU 0, 1, 2, 3, or 6.2 either with or without dependent LU requester (DLUR). In addition, the SNA over TCP/IP access node can be used in conjunction with SNA gateway to enable SNA gateway sessions over TCP/IP.

The SNA over TCP/IP gateway function extends the reach of SNA applications by enabling SNA applications in an SNA network to communicate with SNA applications in an IP network. The SNA over TCP/IP gateway supports independent LU 6.2 sessions.

Configuring AnyNet SNA over TCP/IP

This section contains detailed information about configuring AnyNet SNA over TCP/IP.

Mapping SNA resources to IP addresses

One of the most crucial steps to enable SNA over TCP/IP communication is not done through Communication Server panels. Before SNA sessions or connections can be established, SNA over TCP/IP must determine the IP address of the partner. This is achieved by mapping the SNA identifier of the partner to an IP address using the following steps:

1. SNA over TCP/IP receives the SNA identifier from Communications Server in one of the following formats:
   - For LU names, `netid.luname`
   - For CP names, `netid.cpname`
   - For IDBLK and IDNUM values, configured as `adjacent node ID`, with two fields: `block ID` and `physical unit ID`.
2. SNA over TCP/IP takes the identifier and generates a domain name:
   - For LU names, `luname.netid.snasuffix`
   - For CP names, `cpname.netid.snasuffix`
   - For IDBLK and IDNUM values, `bbbnnnnn.snasuffix`

   Note: The default value for snasuffix is SNA.IBM.COM. SNA over TCP/IP only works in IPV4 networks. Mapping SNA LUs to IPV6 addresses is not supported. For additional information on the SNA domain name suffix, refer to the online help.

   Figure 19 on page 62 shows examples of domain names generated by SNA over TCP/IP.

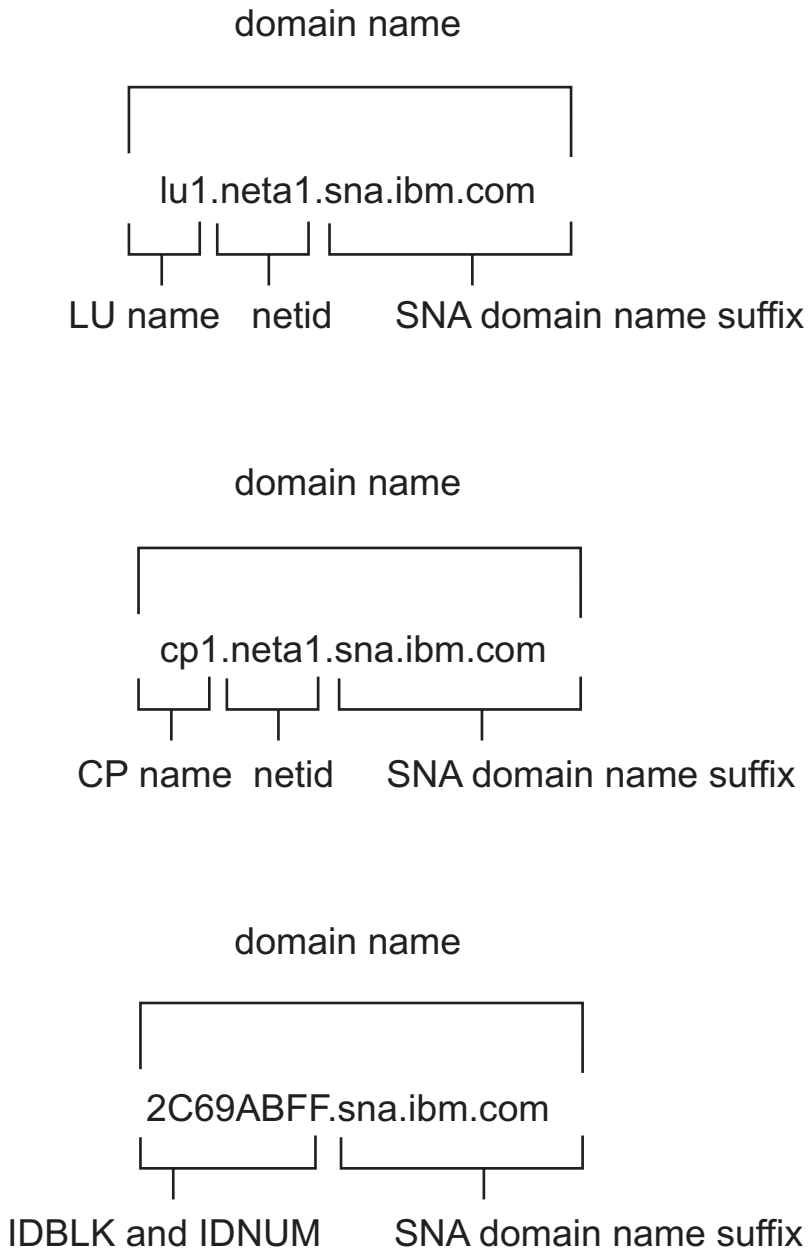

Figure 19. Formats of the domain names that SNA over TCP/IP builds

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

When the IP network includes SNA over TCP/IP gateways, consider the following additional address mapping issues:

- If the partner LU can be reached through a gateway, the domain name of the partner must be mapped to the IP address of the gateway. If there are parallel gateways, the domain name should be mapped to each gateway IP address.
- To minimize the address mapping information needed for AnyNet, consider using a naming convention in which your LU name is the same as the TCP/IP host name. For example, suppose your computer's host name is PATR.ANYNET.OURCORP.COM. If you use the LU name PATR and the netid

ANYNET, and then set the SNA domain name suffix to OURCORP.COM, AnyNet will ask TCP/IP to resolve the name PATR.ANYNET.OURCORP.COM. This name should already be in your domain name server.

Defining domain names and IP addresses

This section describes the TCP/IP name resolution function, used by AnyNet to map SNA resources to IP addresses. This function queries both the local HOSTS file and any domain name servers to convert a domain name (for example, `lu1.neta1.sna.ibm.com`) into an IP address (for example, `10.1.1.1`).

HOSTS file

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

The HOSTS file (in the drivers\etc subdirectory of your Windows system directory) lists:

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

For example, if your IP address is 10.1.1.1, network ID is NETA1, SNA resource name is LUA1, and SNA domain name suffix is the default (`sna.ibm.com`), enter the following in your HOSTS file:

```
10.1.1.1      lua1.neta1.sna.ibm.com
```

Domain Name Server

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

Each SNA identifier is mapped to a corresponding IP address by a domain name server. The location of these servers is configured in the Network section of the Control Panel.

For more information on HOSTS files and domain name servers, refer to your TCP/IP documentation. If your workstation is using the TCP/IP support in Windows, refer to the online TCP/IP documentation that is included with the Windows product.

SNA over TCP/IP gateway considerations

The following information pertains to gateways but not to access node functions.

Defining unique CP names and connection network names

For configurations that have two or more SNA over TCP/IP gateways connecting an SNA network with two or more IP networks, you must define a unique SNA control point (CP) name and a unique SNA connection network name for each IP network.

All LUs that reside on access nodes in the IP network appear to reside on a node with this CP name.

Use the reverse data file of the domain name server or the HOSTS file to define the CP name and the connection network name for a given IP network. Map the IP address 127.0.0.3 to the CP name and map the IP address 127.0.0.4 to the connection network name.

The following example shows entries in the reverse data file. For an IP network with SNA network ID NETA, CP name MYCPNAME, and connection network name MYCNET, you would define the following entries:

```
127.0.0.3 NETA.MYCPNAME.
127.0.0.4 NETA.MYCNET.
```

Notes:

1. A period is required at the end of the name only if the definition is in the DNS reverse data file. No period is used in HOSTS file definitions.
2. Do not include the SNA domain name suffix.

The AnyNet SNA over TCP/IP function of Communications Server provides a default CP name ($ANYNET.$GWCP) and a default connection network name ($ANYNET.$GWCNET). In configurations with one IP network, you can use the default by not defining a CP name or a connection network name. In configurations with multiple gateways connecting multiple IP networks, one IP network can use the default. You must, however, define a unique CP name and connection network name for all other IP networks. Figure 20 on page 65 shows how to define the CP name and the connection network name for a configuration with two IP networks.

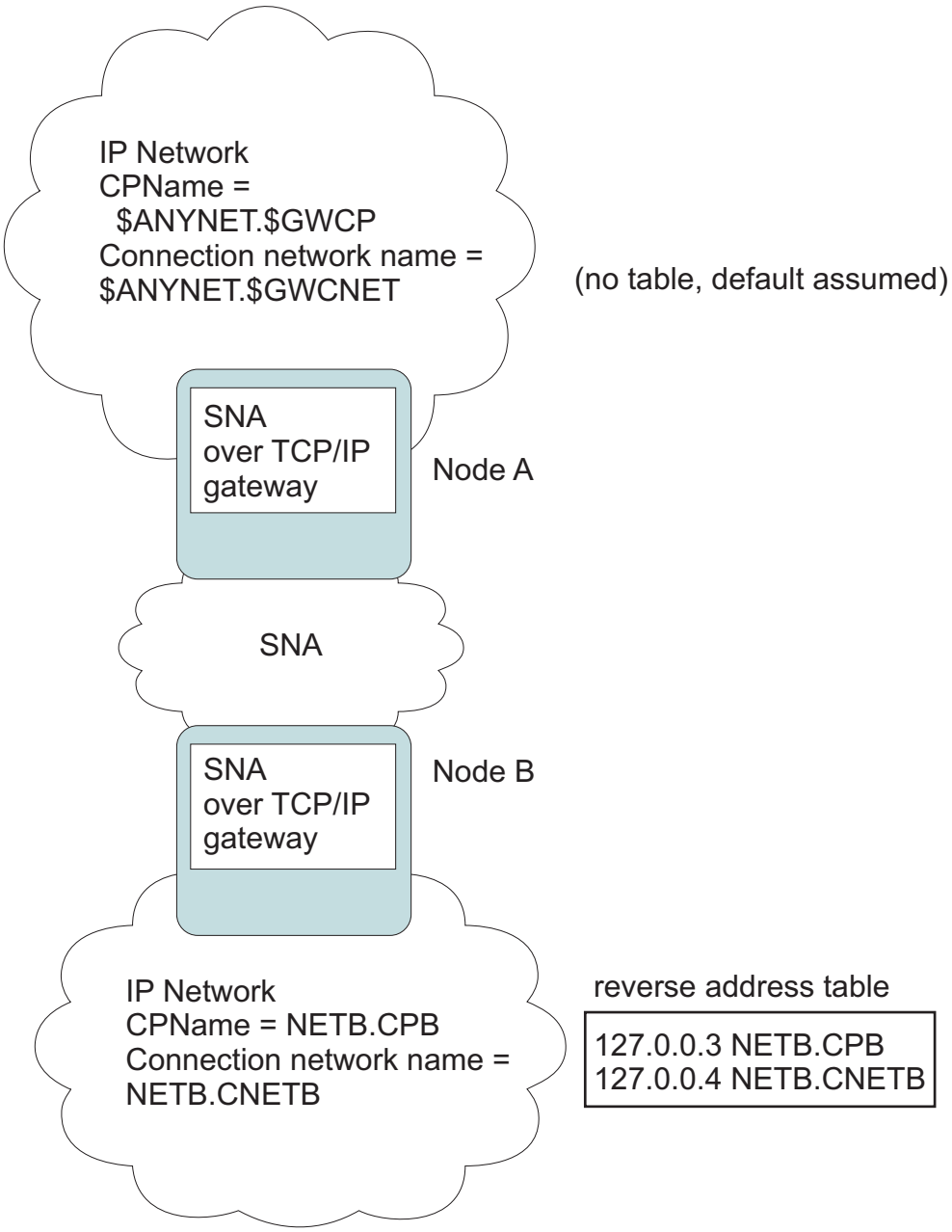

Figure 20. Defining a CP name and a connection network name

Using the wildcard entry to reduce domain name server definitions

If you are using the SNA over TCP/IP gateway and your configuration meets the following naming restrictions, you can reduce the number of domain name server entries by defining a domain name entry for each SNA network ID that can be accessed through one or more SNA over TCP/IP gateways.

- Each SNA network must have an SNA network ID that is unique among all SNA networks.
- Each IP network must have an SNA network ID that is unique among all IP networks.
- An SNA network and an IP network can have the same SNA network ID.

By coding a single domain name entry for each SNA network ID, you do not have to define a domain name entry for every LU in the SNA network that you want to communicate with over the IP network. You can use a wildcard entry (*) to specify the LU name of all LUs that have the same SNA network ID. By substituting a wildcard entry for the *luname* in the entry, you define a single domain name server entry that represents all LUs in that particular network.

Note: If you use the wildcard entry, you must use the full wildcard. Partial wildcards, such as LUA*, are not valid.

The wildcard entry is mapped to the IP address of the first SNA over TCP/IP gateway used to reach the network with that SNA network ID. As shown in Figure 21, logical units SNAAPPL1, APPC1, APPC2, and LU5 reside in network NETB and can only be reached from the IP network through an SNA over TCP/IP gateway with IP address IPgwg. If the SNA domain name suffix is SNA.IBM.COM, you define the following entry in the domain name server:

```
*.NETB.SNA.IBM.COM IPgwg
```

This entry is used for all four logical units.

Note: You have the option of defining each logical unit individually.

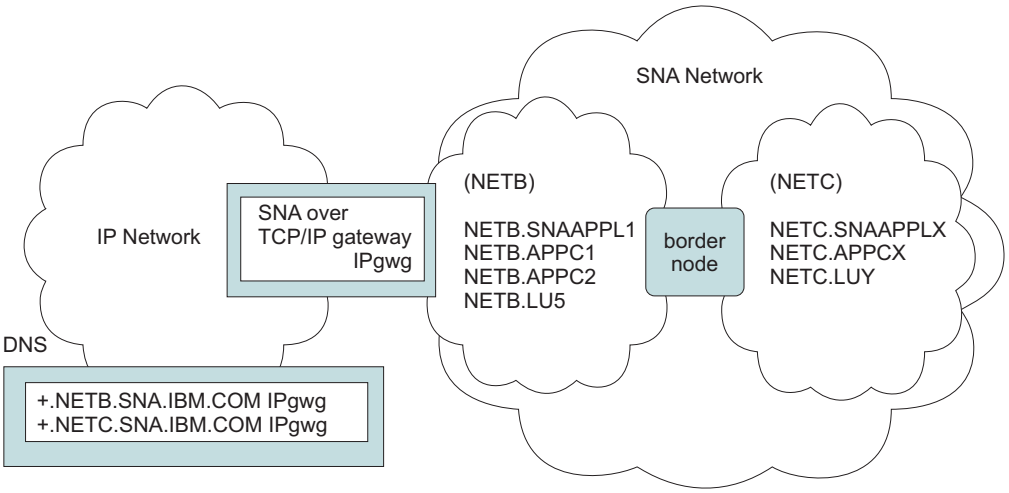

Figure 21. Domain Name Server Definitions for a Single Gateway Connected to an SNA Network with Two Network IDs

Each SNA network must have a unique entry. As shown in Figure 21, if you also have SNAAPPLX, APPCX, and LUY in network NETC, which can only be reached through the SNA over TCP/IP gateway with IP address IPgwg, the domain name server entries are as follows:

```
*.NETB.SNA.IBM.COM IPgwg
*.NETC.SNA.IBM.COM IPgwg
```

In addition, each gateway must have a unique entry. If you add a parallel SNA over TCP/IP gateway, as shown in Figure 22 on page 67, with IP address IPgwh to the preceding example, the domain name server entries are as follows:

```
*.NETB.SNA.IBM.COM IPgwg
*.NETC.SNA.IBM.COM IPgwg
*.NETB.SNA.IBM.COM IPgwh
*.NETC.SNA.IBM.COM IPgwh
```

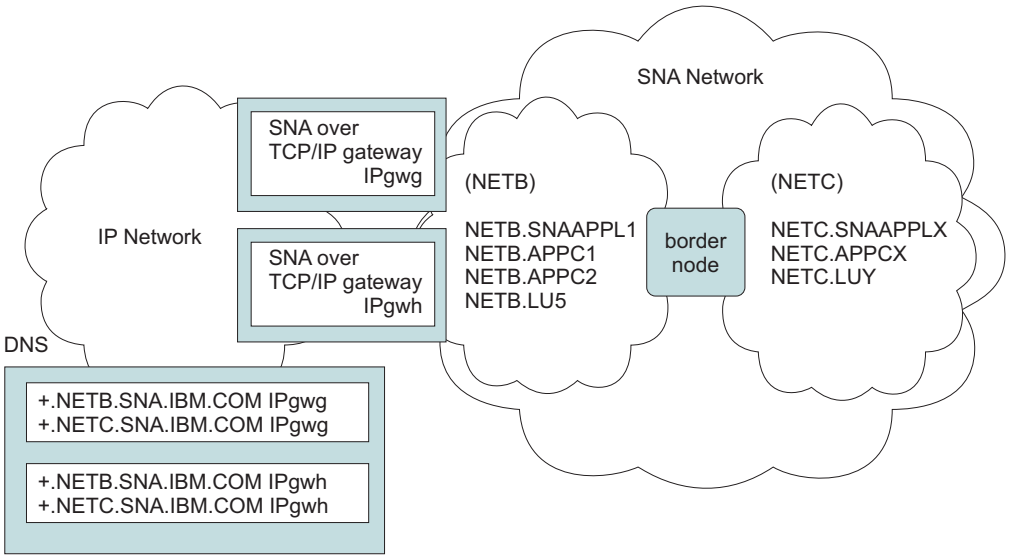

Figure 22. Domain Name Server Definitions for Parallel Gateways Connected to an SNA Network with Two Network IDs

SNA over TCP/IP access node function considerations

The following information pertains to access nodes only, and not to gateways.

How to route SNA sessions over AnyNet SNA over TCP/IP

When an SNA application initiates a session, Communications Server must first determine which transport to use - either SNA, IP, or a combination of the two.

You configure which transport you prefer by setting the routing preference. The routing preference can be set for the whole node via the default routing preference on the AnyNet over TCP/IP device or on a per LU basis when defining partner LUs.

Note: The routing preference for a node only governs sessions that are initiated from the node (access node sessions). Sessions that go through a node are not affected by the routing preference.

You can set or modify the default routing preference to one of the following:

Native first
Requests are routed over SNA. If no SNA route is available, requests are routed over TCP/IP.

Non-native
Requests are routed over TCP/IP. If no TCP/IP route is available, requests are routed over SNA.

Native only
Requests are routed only over SNA. If no SNA route is available, the connection will fail.

Non-native only
Requests are routed only over TCP/IP. If no TCP/IP route is available, the connection will fail.

AnyNet SNA over TCP/IP configuration examples

This section includes examples of AnyNet enabling SNA over IP communication. The following configuration instructions are complete only for the Windows operating system. In all examples, the SNA domain name suffix is SNA.IBM.COM.

For more information about configuring AnyNet for any other platforms mentioned in this section (such as VTAM or iSeries), refer to the appropriate product documentation.

Example 1. Running APPC or CPI-C applications over a TCP/IP network

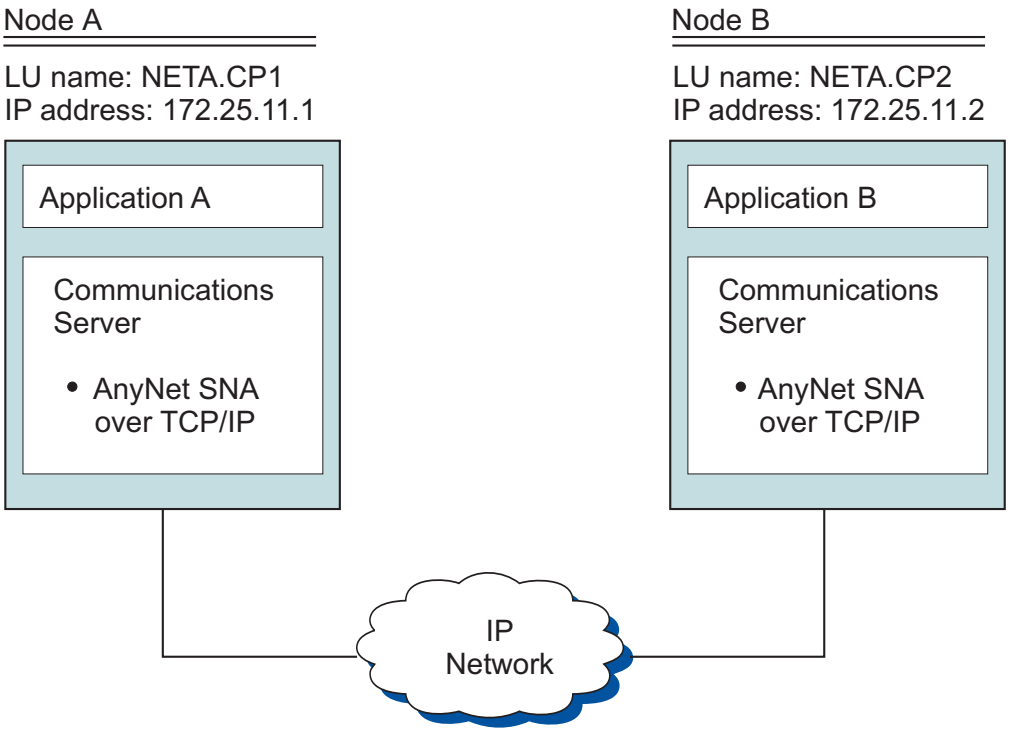

Figure 23. Running APPC or CPI-C applications over a TCP/IP network

Steps: Follow these steps to establish communications between the two Windows nodes. Note that in this example, the CP names are used as LU names.

For Node A, do as follows:

1. Add the following entry to the local HOSTS file:
   ```
   172.25.11.2 CP2.NETA.SNA.IBM.COM
   ```
2. Use **NETA.CP1** as the control point name during node setup. Ensure that the routing preference is set to route sessions over TCP/IP. Refer to the **SNA Node Configuration** help panels for more information.

For Node B, do as follows:

1. Add the following entry to the local HOSTS file:
   ```
   172.25.11.1 CP1.NETA.SNA.IBM.COM
   ```
2. Use **NETA.CP2** as the control point name during node setup. Ensure that the routing preference is set to route sessions over TCP/IP. Refer to the **SNA Node Configuration** help panels for more information.

Example 2. 3270 emulation using DLUR over a TCP/IP network

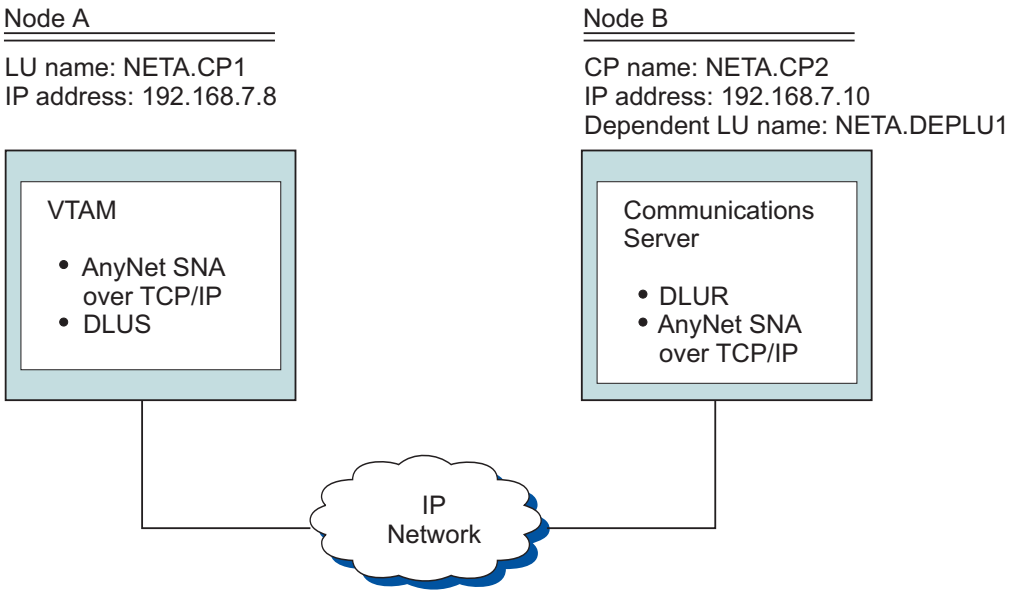

Figure 24. 3270 emulation using DLUR over a TCP/IP network

Steps: Follow these steps to establish communications between Node A and Node B.

For Node B, do as follows:

1. Add the following entry to the local HOSTS file:

   ```
   192.168.7.8 CP1.NETA.SNA.IBM.COM
   ```

2. Use **NETA.CP2** as the control point name during node setup and **NETA.CP1** as the DLUS name when configuring DLUR PUs. Ensure that the routing preference is set to route sessions over TCP/IP. Refer to the **SNA Node Configuration** help panels for more information.

For Node A, add the following entries to the HOSTS file:

```
192.168.7.10 CP2.NETA.SNA.IBM.COM
192.168.7.10 DEPLU1.NETA.SNA.IBM.COM
```

Note that MVS™ AnyNet SNA over TCP/IP currently requires DLUS/DLUR for dependent LU communication.

Example 3. Using an SNA gateway to enable 3270 emulation between SNA and TCP/IP networks

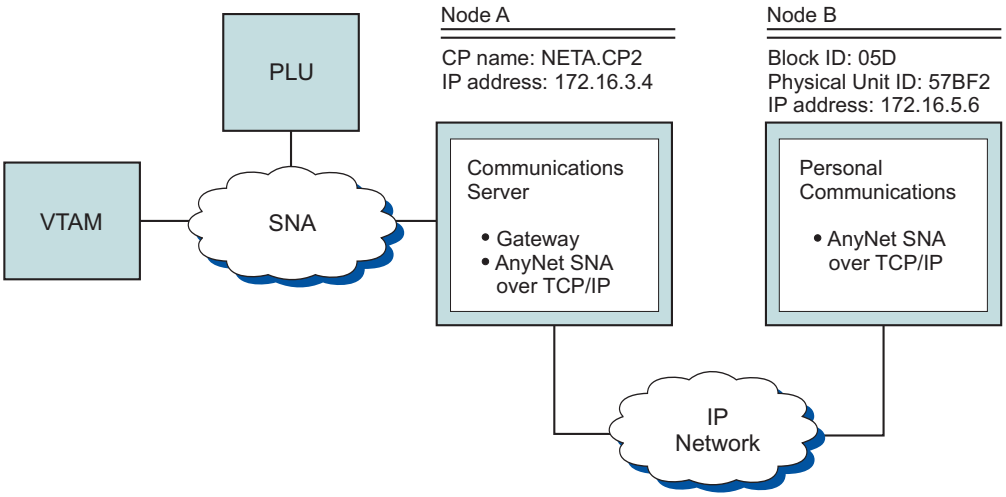

Figure 25. Using an SNA gateway to enable 3270 emulation between SNA and TCP/IP networks

Steps: Follow these steps to establish communications between Node B and the VTAM host.

For Node A, do as follows:

1. Add the following entry to the local HOSTS file:

   ```
   172.16.5.6 05D57BF2.SNA.IBM.COM
   ```

2. Use **NETA.CP2** as the control point name during node setup and use ANYNET device to assign implicit templates when defining clients. Refer to the **SNA Node Configuration** help panels for more information.

For Node B, add the following to the HOSTS file:

```
172.16.3.4 CP2.NETA.SNA.IBM.COM
```

Example 4. Using an SNA gateway for 3270 emulation over a TCP/IP network

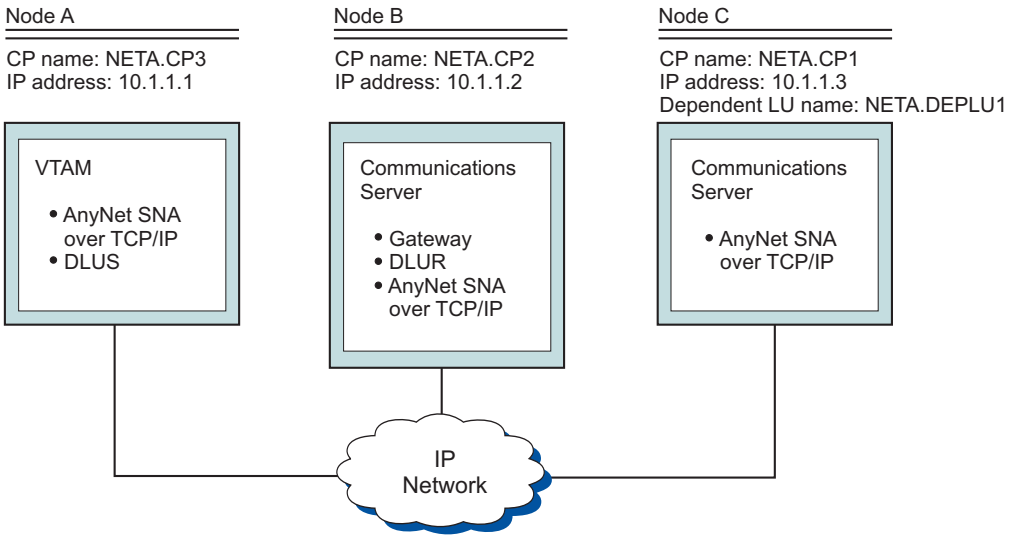

Figure 26. Using an SNA gateway for 3270 emulation over a TCP/IP network

Steps: Follow these steps to establish communications between Node C and Node A.

For Node B, do as follows:

1. Add the following entry to the local HOSTS file:

   ```
   10.1.1.3 CP1.NETA.SNA.IBM.COM
   ```

2. Use **NETA.CP2** as the control point name during node setup, **NETA.CP1** as the adjacent CP name when defining the ANYNET SNA over TCP/IP connection definition and **NETA.CP3** as the DLUS name when assigning the DLUS to a client template. Ensure that the routing preference is set to non-native for NETA.CP3. Refer to the **SNA Node Configuration** help panels for more information.

For Node C, do as follows:

1. Add the following entry to the HOSTS file:

   ```
   10.1.1.2 CP2.NETA.SNA.IBM.COM
   ```

2. Use **NETA.CP1** as the control point name during node setup, **NETA.CP2** as the adjacent CP name when defining the ANYNET SNA over TCP/IP connection definition. Refer to the **SNA Node Configuration** help panels for more information.

For Node A, add the following entry to the HOSTS file:

```
10.1.1.2 CP2.NETA.SNA.IBM.COM
10.1.1.2 DEPLU1.NETA.SNA.IBM.COM
```

Example 5. 3270 emulation from two Windows workstations on different IP networks

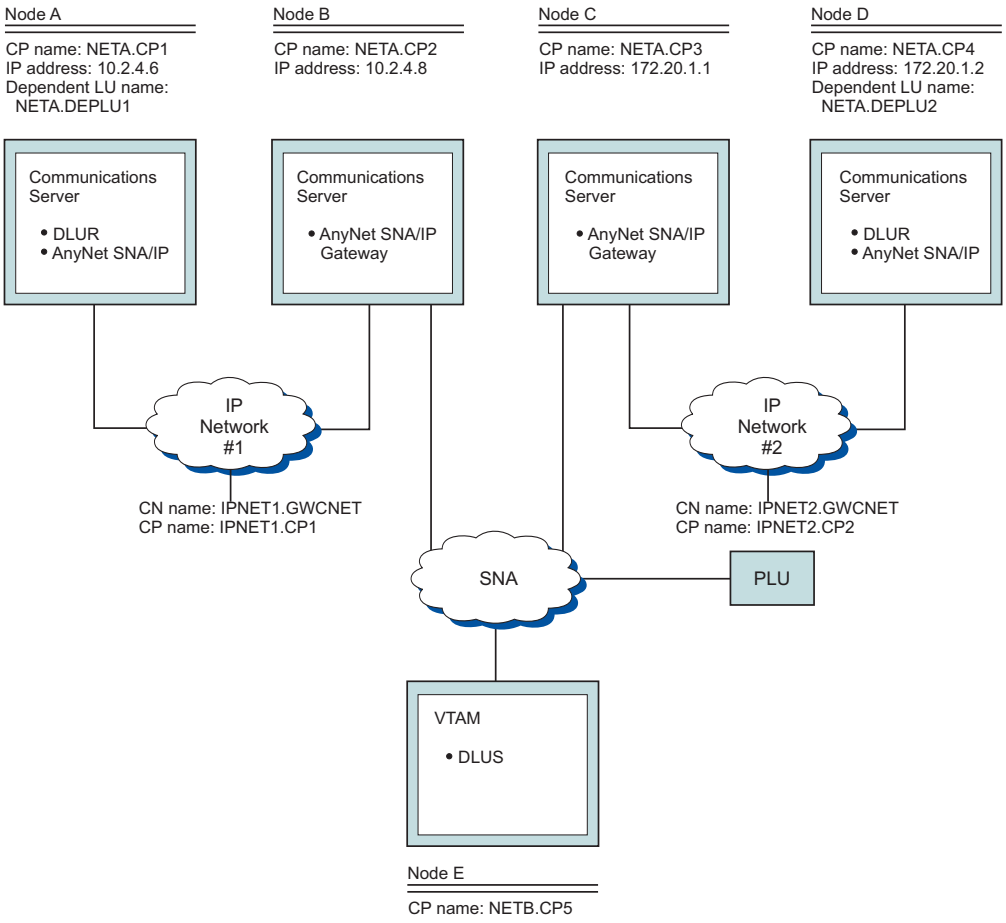

Figure 27. 3270 emulation from two Windows workstations on different IP networks

Steps: Follow these steps to establish communications from Nodes A and D to Node E.

For Node A, do as follows:

1. Add the following entries to the local HOSTS file:

   ```
   10.2.4.8 CP5.NETB.SNA.IBM.COM
   127.0.0.4 IPNET1.GWCNET
   127.0.0.3 IPNET1.CP1
   ```

2. Use **NETA.CP1** as the control point name during node setup, **NETA.CP5** as the DLUS name when configuring the DLUS PUs. Ensure that the routing preference is set to non-native for NETA.CP5. Refer to the **SNA Node Configuration** help panels for more information.

For Node B, add the following entries to the HOSTS file:

```
10.2.4.6 CP1.NETA.SNA.IBM.COM
127.0.0.2 DEPLU1.NETA.SNA.IBM.COM
10.2.4.6 DEPLU1.NETA.SNA.IBM.COM
127.0.0.4 IPNET1.GWCNET
127.0.0.3 IPNET1.CP1
```

For Node C, add the following entries to the HOSTS file:

```
172.20.1.2 CP4.NETC.SNA.IBM.COM
127.0.0.2 DEPLU2.NETC.SNA.IBM.COM
172.20.1.2 DEPLU2.NETC.SNA.IBM.COM
127.0.0.4 IPNET2.GWCNET
127.0.0.3 IPNET2.CP2
```

For Node D, do as follows:

1. Add the following entry to the HOSTS file:

   ```
   172.20.1.1 CP5.NETB.SNA.IBM.COM
   127.0.0.4 IPNET2.GWCNET
   127.0.0.3 IPNET2.CP2
   ```

2. Use **NETA.CP4** as the control point name during node setup, **NETA.CP5** as the DLUS name when configuring the DLUS PUs. Ensure that the routing preference is set to non-native for NETA.CP5. Refer to the **SNA Node Configuration** help panels for more information.

Helpful hints

This section contains helpful hints on tuning, TCP/IP connectivity via SLIP or PPP, and dynamic IP addresses.

Tuning: If you can access an LU through multiple SNA over TCP/IP gateways, and you have mapped that LU name to multiple IP addresses, increase the **CONN_RETRY_SECS and CONNWAIT_SECS** parameters of the **ANYNET_COMMOM_PARAMETERS** keyword when configuring the AnyNet base parameters. This ensures that TCP connections will be attempted to all possible adapters and gateways. It can take as long as 90 seconds for a TCP connection to fail to an inactive IP address.

Dynamic IP addresses: Generally, AnyNet SNA over TCP/IP depends on SNA resources (for example, LU names, CP names, or idblk/num) being statically mapped to IP addresses. However, depending on your configuration and how your connections are initiated, you might be able to use AnyNet SNA over TCP/IP in environments where IP addresses are dynamically assigned (for example, DHCP).

APPC or CPIC applications: An SNA over TCP/IP access node with a dynamically assigned IP address may always initiate sessions to another SNA over TCP/IP access node or gateway with a static IP address.

The only way an SNA over TCP/IP access node or gateway with a static IP address (node A) can initiate a session to a partner with a dynamic IP address (node B) is:

1. Node B initiated a session to or through node A first.
2. The session initiated in Step 1 is still active.

Dependant LU applications:

Note: The information in this section applies to LU 0, 1, 2, 3, or dependent 6.2 applications.

SNA over TCP/IP access nodes with dynamically assigned IP addresses can support dependent LU communications if the following criteria are met:

- Dependent LU communication is occurring through an SNA gateway either through:
  - Communications Server/2 4.1 with APAR JR10461 applied
  - Communications Server/2 V5
  - Communications Server for Windows 5.01 or higher

- The SNA over TCP/IP access node with a dynamically assigned IP address initiates connections to an SNA gateway with a static IP address.

Dependent LU communication through DLUS/DLUR over AnyNet SNA over TCP/IP is not supported if the DLUR node has a dynamically assigned IP address.

Chapter 4. Planning for Client/Server communication

This chapter contains information about planning for SNA API clients for Communications Server.

SNA API clients

This section contains information specific to SNA API clients.

Installation and configuration

For information about installing and configuring Communications Server SNA API clients, refer to *Quick Beginnings*.

Lightweight directory access protocol

Lightweight Directory Access Protocol (LDAP) provides a standard way to access directory services. LDAP enables you to read, search, add, and remove information from centralized or distributed databases. LDAP is similar to the older X.500 Directory Access Protocol (DAP), but without some of the overhead (such as session and presentation OSI layers). You can use **LDAP Client Configuration** to configure SNA API client information and store it on an LDAP server.

LDAP Client Configuration shows the logical organization and relationships between the clients and their configurations.

For more information on using these major areas, refer to the help panels for **LDAP Client Configuration**.

Directory information tree

The Directory Information Tree (DIT) connects all Directory Service Agents (DSAs) in a predefined model. The model is an extensible hierarchical structure consisting of a number of base objects. A typical LDAP DIT would have a root, under which *countries* (c) are defined. Below the countries are usually *organizations* (o), and below an organization are *individuals*, ((cn) or (uid)), or *organizational units* (ou). For example, an fully-distinguished LDAP entry would be specified as:

```
c=US, o=company.com, ou=Sales, cn=temp
```

Client hierarchy

The client hierarchy is relative to the LDAP DIT you logged in to. You can add users or organizational units to the client hierarchy or delete users or organization units from the client hierarchy.

Configuration hierarchy

The configuration hierarchy logically shows, through a graphical tree view, the client configuration and all the relationships between the configuration definitions. The configuration hierarchy enables you to create, modify, or remove definitions.

Common API client for Communications Server for Windows

The client shipped with Communications Server permits clients to locate and access IBM Communications Servers in the network.

Install options for configuration location

During installation of the client, the user can choose the client configuration location. There are three choices:

- INI file configuration, which is the method used with previous versions of Communications Server and is the default method when installing the SNA API client that is included with Communications Server
- Lightweight Directory Access Protocol (LDAP)

The configuration location contains server lists, user names, and all other client configuration information.

Configuration locations

The information can be stored in either INI or LDAP. For example, Communication Servers names and addresses can be stored in INI or LDAP for the API client. The common SNA API client enables you to easily integrate different server types on the same network.

Configuration utility

Use the Set Configuration Options utility to change the client configuration location (INI or LDAP) after installation.

Locating servers

The client enables the user to locate and access IBM Communications Servers using TCP/IP transport protocols.

TCP/IP: The method used to locate servers using the TCP/IP transport protocol is identical. The network administrator can control access to the servers through server configuration. See "Planning for TCP/IP scopes" on page 116 for more information.

Assigning a default local LU

You can assign a default local LU alias for each user through either the INI file configuration or LDAP configuration for 32-bit and 64-bit Windows SNA API clients.

APPC programs can use a default local LU alias rather than specify one directly. When an APPC program issues a TP_START verb with the local LU alias field set to binary zeroes or all ASCII blanks, the APPC API uses the configured default local LU alias.

EHNAPPC programs can use a default local LU alias rather than specify one directly. When the local LU field on the iSeries configuration panel is left blank, the configured default local LU alias is used.

CPI-C programs can use a default local LU alias rather than specify one directly. When the local LU alias field in the CPI-C side record is left blank, and no value has been defined using the APPCLLU environment variable, the CPI-C API uses the configured default local LU alias when initiating a conversation.

Attach Manager started programs can use a default local LU alias rather than specify one directly. When the local LU alias field in the attach manager record is left blank, the Attach Manager uses the configured default local LU alias when processing incoming conversation requests.

Assigning a default partner LU

You can assign a default partner LU alias for each user through either the INI file configuration or LDAP configuration for 32–bit Windows SNA API clients.

APPC programs can use a default partner LU alias rather than specify one directly. When an APPC program issues an ALLOCATE verb with the partner LU alias field and the fully qualified partner LU field set to binary zeroes or all ASCII blanks, the APPC API uses the configured default partner LU alias.

CPI-C programs can use a default partner LU alias rather than specify one directly. When the partner LU alias field in the CPI-C side record is left blank, the CPI-C API uses the configured default partner LU alias when initiating a conversation.

Assigning a default LUA session name

You can assign a default partner LU alias for each user through either the INI file configuration or LDAP configuration for 32–bit Windows SNA API clients.

LUA programs, such as 3270 emulators, can use a default LUA session name rather than specify one directly. When a LUA program issues an RUI_INIT or SLI_OPEN verb with the LUA LU name field set to binary zeroes or all ASCII blanks, the RUI/SLI API uses the configured default LUA session name.

Client/Server data encryption

Client/server data encryption enables the client to request encryption of the application data in communications between the SNA API client and Communications Server.

Data encryption can be on, off, or optional. Optional data encryption means that encryption is used if it is supported by the server.

Client/Server security

Communications Server enables Windows SNA API clients to use Windows domain security to authenticate the client connection to the server without reentering the userid and password. The client must be part of a Windows domain, either by participation in a Communications Server domain or logging in locally with a synchronized userid and password.

SNA API client users outside of the Windows domain are required to provide the userid and password, either through a prompt or by storing these values in the client configuration file.

The authorized users for client server are maintained in the IBMCSAPI local group, which is located either directly on the Communications Server or on the domain controller where Communications Server participates. This user group is created during installation and can be administered using the Windows User Manager application.

You can determine whether the userid and password are required for connection to the server by changing the **LogonControl** value entry at the server to zero (0), which removes the requirement of the userid and password specification by the client.

Audit trail records of client connections to the server are tracked in the Windows Event Log. You can view the records using the Windows Event Viewer under the

Application log. If you do not want to track client connections to the server, you can change the **AuditTrail** value entry at the server to zero (0).

You can access the **LogonControl** and **AuditTrail** value entries through the Windows Registry Editor under the key:

```
HKEY_LOCAL_MACHINE/SYSTEM/CurrentControlSet/Services/IBM SNA Client Services.
```

WTS

SNA API Client includes the **LUALIAS** environment variable. **LUALIAS** can be used to migrate an application to a WTS environment.

Many client applications use a hardcoded Local LU alias. When migrating to a WTS environment, this hardcoded Local LU alias must be mapped to different LU alias for each user. For example, Local LU alias *x* for user1 and *y* for user2. Instead, use the This can be done by configuring the environment variable **LUALIAS<TPsLUALIAS>**.

Configure multiple LU aliases in a WTS environment in the following way:

```
set LUALIAS<LocalLUAAlias>  = UserLUAlias
```

where: *LocalLUAlias* is the LU alias configured in the configuration file (in CPIC side info) *UserLUAlias* is the LU alias required to be used for a particular user.

Note: Note: This environment variable must be configured under user environment variables.

Tracing and logging functions

The IBM SNA API Client provides the following tracing and logging functions for problem determination:

- two message log files (MsgLog.dat) for logging information
- multiple trace files (multiple pcatrace.dat) for collecting debug information.

Infobundler can collect both the message log files and all the debug trace files.

Chapter 5. Planning for HPR

Communications Server supports high performance routing (HPR) over Enterprise Extender (IPv4 and IPv6), synchronous data link control (SDLC), LAN, channel, Multi-Path Channel (MPC), and X.25 connections.

HPR automatic network routing (ANR) minimizes the storage and processing requirements in intermediate nodes, which is an improvement over APPN intermediate session routing (ISR) for high-speed networks with low error rates. HPR provides non-disruptive switching to an alternative path.

Communications Server HPR support includes support for rapid transport protocol (RTP) and automatic network routing (ANR).

Communications Server provides an RTP_Tuning configuration to set parameters. See *Configuration Guide* for details.

HPR link characteristics

To provide the benefits listed above, HPR assumes that data links provide low rates of lost frames. HPR recovers from conditions that cause lost frames, but if the conditions persist for long periods, HPR assumes that network congestion is too high and HPR throughput is lowered. Achieving good HPR throughput on a given link involves proper configuration of the sending node, the facilities that constitute the link between the two nodes, and the receiving node.

Note: You might want to set ERP values in cases where line facilities have high frame loss.

Sending node
Typical adapters used with HPR will send frames onto the line with low rates of failure. For Ethernets, HPR behavior will be degraded if the LAN is heavily loaded with many collisions (for example, if the load is more than about 30% to 40% of the nominal Ethernet speed). For frame relay, facilities should be configured so that individual connections can burst to the access line rate without causing congestion conditions in the frame-relay network.

Line facilities
LAN and IP facilities can have substantial hidden complexity, ranging from large numbers of stations sharing a single LAN segment to bridges between LAN segments, perhaps with hidden lower speed links, to simulated LAN segments that operate over complex subnets. The low frame loss assumption must be met. For example, WAN links between split bridges must use facilities with low error rates or must compensate by using link level error recovery between the bridges. As another example, conditions that cause bridge buffers to overrun (broadcast storms or large bursts of broadcast frames) must be prevented.

Receiving node
Typical adapters used with HPR will receive frames from the line with low rates of failure. Some adapters, however, will fail to receive frames when

consecutive frames arrive too quickly. The best HPR performance is realized when the receiving adapter can handle its traffic without missing inbound frames.

If your network has a high error rate, systems administrators may notice lower than expected throughput on HPR connections.

HPR considerations

The HPR features are achieved by including additional information in frame headers. If you do not need the HPR features, the additional overhead may reduce your network performance. The HPR features are an advantage whenever there are intermediate nodes or alternate routes. However, if you require high performance between two nodes connected by a single link without non-disruptive recovery during a link failure, you should disable HPR when you configure that link.

However, EEDLC links always use HPR over TCP/IP. If there is an alternate path to the system, configure another link with a different destination address so that HPR can non-disruptively switch to the alternate path. If there is a single link, a link failure must be detected prior to link reactivation attempts. The path switch timer must be longer than the link timeout to non-disruptively recover the RTP connection.

HPR implements priority queuing. In general, interactive traffic passes through the network faster than lower priority batch traffic. Adapters that queue frames do so without regard to priority and may interfere with HPR's priority queueing. You may need to reduce the adapter's queue depth to prevent low priority traffic from delaying high priority traffic (see the adapter documentation for instructions on setting the adapter queue depth). However, EEDLC does not implement priority queuing.

Many PCs now have advanced power management features that will suspend power to devices (like disk drives) and reduce the clock rate on a CPU significantly to save power if the keyboard or mouse is not being used. On a workstation, this is a good feature, but on a server platform, this can occur while it is routing data from one adapter to another (and no disk activity is occurring). For Communications Server, disable advanced power management and hardware power management by entering the BIOS configuration during boot up (usually pressing F1 while initializing hardware) and disable these options. HPR will use more of the LAN adapter

HPR will use more of the LAN adapter memory due to increased performance and queueing. Many shared RAM adapters default to using 8 kilobytes of shared RAM. It is required that the shared RAM size is 16 kilobytes. To adjust this configuration, refer to the adapter configuration manual. This setting is located in either your machine BIOS, or on the adapter switch settings.:

Note: If you do not increase the shared RAM to 16 kilobytes, network performance is degraded.

Do not adjust the shared RAM value above 16 kilobytes, the optimal setting for the adapter memory.

If HPR links are failing under heavy traffic, you can set your receive timers to a higher value. This allows more time for the links to receive an acknowledgement for sent frames.

For SDLC links, this is the Primary Receive Timer. The receive timer can be set on either the DLC or a logical link statement.

To calculate the minimum receive timer value in tenths of seconds, use the following formula:

```
[(send_window_count * max_i_field_size * 8) /actual line speed]* 10
```

Considerations for tuning parameters on LAN DLCs

You can specify eight parameters to affect how quickly Communications Server handles recovery and failure of LAN links. You might need to adjust these parameters to achieve the best results if you are using slow links, HPR, or backup host connections.

You should consider the following parameters when defining a LAN device in **SNA Node Configuration**:

Under the **Advanced** tab:

Test retry interval
The test retry interval specifies the time between attempts to find the adjacent link station on the local area network (LAN). The number of attempts made are based on the value specified for the test retry limit.

The default value is 8.

Test retry limit
The test retry limit specifies the maximum number of attempts to find the adjacent link station on the local area network (LAN) without receiving an acknowledgment in the time set by the value for the test retry interval.

The default value is 5.

Maximum PIU size
The maximum path information unit (PIU) size is the maximum number of bytes that are contained in the data buffer used by SNA sessions for this link. If the value of the maximum PIU size exceeds the frame size supported by your local device driver, the value will be reduced to match the frame size.

The default value is 65 535.

Under the performance tab:

Idle Timeout
The idle timeout specifies the time that the LAN device driver waits for a frame to be received before declaring the link to be inoperative.

The default value is 30 seconds.

Acknowledgement delay
The acknowledgment delay specifies the time that the LAN device withholds a response to a received frame in order to allow more frames to be received and acknowledged with the same Request Ready (RR).

The default value is 100 milliseconds.

Poll response timeout
The poll response timeout specifies the time that the LAN device waits for a response to a frame sent with the POLL bit set.

The default value is 8 000 milliseconds.

Anticipated outstanding transmits
The anticipated outstanding transmits specifies the maximum number of frames the LAN device queues to a link station before sending a Receive Not Ready (RNR) to the adjacent link station.

The default value is 16 frames.

Receive buffer count
The receive buffer count specifies the number of buffers that are reserved in memory to hold data received from the host until it can be processed. Each buffer is the size of the PIU.

The default value is 32 buffers.

HPR path switch processing

HPR provides a non-disruptive path switch to a different connection if one connection fails. A retry of the failed connection is attempted before a path switch occurs. An explanation of the parameters involved in determining whether a connection has failed and the time to initiate a path switch follows.

When the value specified for **Idle timeout** is reached, a keepalive frame is sent. The default for **Idle timeout** is 30 seconds.

The value specified for **Poll response timeout** determines the time that the device waits for a response to the keepalive frame. The default for **Poll response timeout** is eight seconds (8 000 milliseconds). A keepalive frame is retried 10 times. This retry count can not be changed.

If there is no response to the keepalive frame, TEST commands are sent to the partner system. The value specified for **Test retry interval** determines the time between TEST command attempts, and the value specified for **Test retry limit** determines the maximum number of TEST command attempts. The default for **Test retry interval** is 8 seconds, and the default for **Test retry limit** is five attempts.

The following calculation indicates the time required to determine whether a connection has failed and initiate a path switch:

```
(Idle timeout) + (Poll response timeout * 10 retries) +
           (Test retry interval * Test retry limit) + 1
```

Using the defaults for the parameters, the time required to determine a connection failure and initiate a path switch can be up to 151 seconds.

```
(30 seconds) + (8 seconds * 10 retries) +
           (8 seconds * 5 attempts) + 1 = 151 seconds
```

If you are experiencing long path switch times (or failures) with HPR when a connection fails, you can reduce the time by decreasing the values for the parameters in the calculation.

Considerations for tuning parameters on EE (IPv4 or IPv6) DLCs

EEDLC timers are independent of HPR timers. Link failure detection timers should be smaller than end-to-end RTP connection timeouts for path switch to work properly. Also, all ″Reactivation″ flags must be configured (default) to retry on different link failure detections.

In IBM-EEDLC device configuration, all reactivation flags must be checked as follows:

- Reactivate after a failed start attempt
- Reactivate after a link failure
- Reactivate after the remote station issues a disconnect
- Delay applications' attempt to reactivate the link
- Maximum reactivation attempts (0-127): 0 (default)
- Reactivation Delay (0-3600 seconds): 30 (default)

Other EEDLC device parameters are:

Connect timer
15 seconds (default)

Connect retry count
3 (default)

Inactivity timer
10 seconds (default)

Maximum reactivation attempts set to zero (0), result in infinite retry. It is recommended that you keep the default at zero (0) unless there is a delta cost associated with the connection. In such case, you must manually initiate the connection after the reactivation attempts have been exhausted.

Reactivation Delay defaults to 30 seconds but can be tuned to retry the link activation more quickly after a link failure.

Connect timer is used for control commands such as TEST, XID, SABME, DISC and so on. **Connect timer** is the wait time for a response. If the timer expires without receiving a response, the command will be retransmitted based on the value defined in the **Connect retry count** parameter.

Example of HPR path switch processing in EE

For a reliable network, the default timers can be changed based on specific requirements. The following are suggested values in a stable network with a single link:

Maximum reactivation attempts
(0-127): 0 (default)

Reactivation Delay
(0-3600 seconds) 5 (change)

Connect timer
10 seconds (change)

Connect retry count
2 (change)

Inactivity timer
10 seconds (default)

Path_Switch_Attempts
12 (change in RTP_TUNING verb in ACG file)

If it is a link failure, it will be detected in:

```
(connect_retry_count + !) * Inactivity_timer = 30 seconds here.
```

After the reactivation delay, a reconnect is attempted. When it is successful the RTP connections path switch attempts will be successfully routed over the new link. By decreasing the **retry_count** and reactivation delay, the link failure will be detected

in a shorter time. By increasing the **Path_Switch_Attempts**, there will be less delay after the link is reactivated until the RTP connection is path switched.

Chapter 6. Planning for SNA Gateway

The SNA gateway supports:

- Various SNA protocols
- Various downstream workstations
- Various link types between the gateway and the host including using DLUR to DLUs at the host
- Pooling of LUs (sessions) at the gateway
- Definition of implicit workstations
- Dynamically changing or adding definitions for nonactive workstations
- Multiple hosts simultaneously
- Various link types (which can be combined) between the workstations and the gateway

Refer to *Quick Beginnings* for instructions on how to configure the SNA gateway using SNA Node Configuration.

Supported connections to workstations

The SNA gateway supports workstations connected by:

- LAN connectivities
  - Token ring
  - Ethernet
  - ATM (LAN Emulation)
- Frame Relay
- SDLC
  - Synchronous
- X.25 network
- AnyNet TCP/IP (for Microsoft TCP/IP supported connectivity)

The SNA gateway can support as many as 254 LUs for each defined PU. You can define a different PU for each of your host connections. The LUs are used by the downstream workstations to connect to the host. The number of downstream workstations supported is dependent on several factors, including the type of connectivity and the number of adapters on the gateway. For instance, if you have one LAN adapter on the gateway, one host link with 254 LUs can be used by 253 downstream workstations concurrently. With two adapters, you can double the number of downstream workstations.

SDLC workstation connections

You can choose to connect using SDLC in point-to-point and Multipoint configurations. After configuration has been completed, no special consideration is necessary to run SDLC between the workstation and the gateway.

When setting up multidrop secondary workstations, consider the various factors that control polling turnaround time. There are physical limitations that affect how

quickly the primary can poll the secondary that is farthest away. Defining all of the secondary workstations using the same parameters will simplify the following calculation.

To calculate the minimum inactivity time in seconds, use this formula:

```
NS * (SW + RW) * (IS + 2) / (LS / 8)
```

where:

```
NS = Number of stations
SW = Send window size
RW = Receive window size
IS = I-field size
LS = Line speed (bps)
```

If these values are not the same for all secondary workstations, a separate computation must be made for each group or workstation and added together to find the correct value for the inactivity timer. For example, for a multidrop link with 16 secondary workstations at 14.4 Kbps, with each workstation set for a send window size of 7, a receive window size of 7, and I-field size of 521, the calculation for the inactivity timer would be:

```
16 * (7 + 7) * (521 + 2) / (14400 / 8) = 65.1 seconds
```

Note: Always add a few extra seconds for a buffer.

X.25 workstation connections

If you configure the SNA gateway to use only X.25 permanent virtual circuit (PVC) connections between the gateway and the workstations, it is recommended that you configure all the workstation and gateway PVCs with negotiable link station roles. If you do not configure the gateway and workstation PVCs with negotiable link station roles, the workstations might not be able to establish the X.25 PVC links to the gateway.

See Chapter 12, “Planning X.25,” on page 127 for more information about PVC connections and the configuration.

Supported connections to hosts

he SNA gateway supports the following connections to a host:

- LAN connectivities
  - Token ring
  - Ethernet
- SDLC
  - Synchronous
- X.25 network
- AnyNet TCP/IP (for Microsoft TCP/IP supported connectivities)
- Enterprise Extender (for Microsoft IPv4 or IPv6 supported connectivity) using DLUR connection
- Channel (CDLC)
- Channel (MPC) with a DLUR connection

If there are connections to multiple hosts from the gateway being used by dependent LU traffic, only the host link with the control point PU name can do the following:

- Act as a host focal point
- Have CP-CP sessions with the host
- Be used for APPN functions

The links that are defined with a PU name that is not the same as the control point name can be used only for additional dependent LU connections to a host.

If a gateway host link is defined as a limited resource link, it will send a request disconnect to the host after the last LU-LU session is unbound. The gateway then passes the DACTLUs to the workstation and a DACTPU to workstations that have sessions only with that host link. When the workstation connected to the gateway is defined as a limited resource, the workstation link is disconnected if there are no other sessions. In Communications Server you can designate a workstation as a limited resource when defining the explicit client connection.

However, if the workstation has an application (or emulator) that automatically reactivates its host link, the gateway will reactivate the gateway link to its host when the workstation activates its link to the gateway. This means that incompatible workstation definitions would effectively inhibit the limited resource function at the gateway.

Host considerations

Consider the following things when planning for your host connection:

- If your environment uses a network control point (NCP), the 37xx or Communication Controller for Linux® (CCL) NCPGEN, **LINE** macro, **MAXLU** value must be at least as large as the number of LUs on the Communications Server.
- VTAM users can control the time interval during which VTAM would buffer data before sending it out to the controller (coattailing). When the VTAM coattailing parameter **DELAY=0** is specified on any PU definition, this might provide faster host send response time at the expense of more work for the host processor.

  When a nonzero value for DELAY is specified, this saves the host processor instructions at the expense of increased receive response time on the Communications Server workstation.

  For coattailing, begin with default value 0.2. A 0.2 second delay has a moderate effect on response time, but if the rate is approximately one transaction per second or greater, coattailing occurs.
- Although each downstream SNA workstation on the LAN implements a PU with its own set of LUs, the SNA gateway is implemented so that the host sees only the gateway PU. Therefore, only one PU definition is coded at the host. This PU definition contains the LUs for *all* the stations on the LAN that will connect to the host through this gateway.

  If your host VTAM supports self-defining dependent LUs (SDDLU), you might want to take advantage of the function. When configuring host LUs, specify an LU model type or provide an LU model name that matches the LUSEED operand value defined in the VTAM switched major node and used by the VTAM SDDLU exit routine.

LAN destination address configuration

You need to provide a destination address if you use one of the supported LAN connections between the gateway and the host or between the gateway and an explicitly defined workstation. When determining the correct destination address to enter in each profile, remember that the proper perspective for both addresses is to view the destination from the SNA gateway. Figure 28 shows this view.

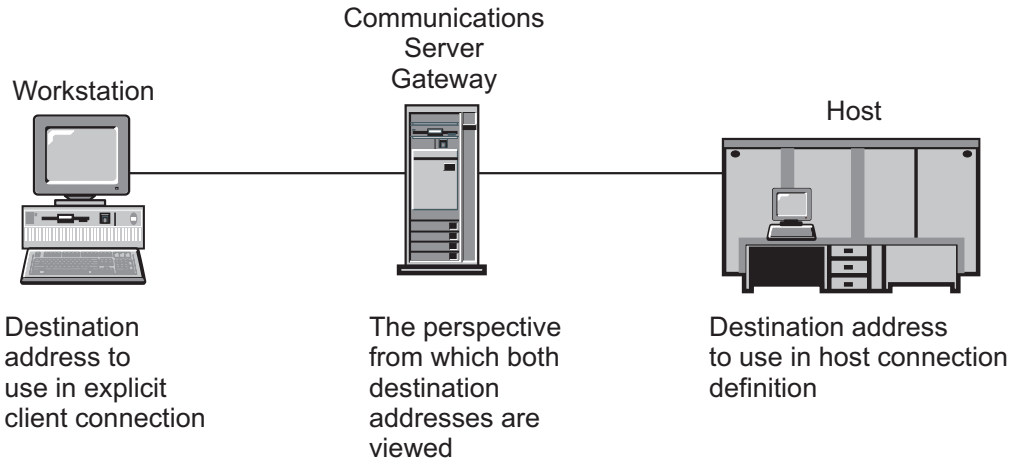

Figure 28. Perspective to use for destination address entries

Planning gateway LU definitions

The SNA gateway supports both *pooled* host LUs and *dedicated* host LUs. When LUs are configured for each host connection at the SNA gateway, they can be grouped into pools. Creating pools is often beneficial for the following reasons:

- If all sessions on all workstations do not need to communicate with the host at the same time, pooling can conserve system resources. The workstation sessions on the LAN can share the same LUs, but not at the same time. This function allows the number of LUs configured at the SNA gateway for workstation sessions to be greater than the number of LUs configured between the host and the gateway, reducing host resource requirements, reducing host workstation definition, and speeding restarts.
- Pools are easier to administer and configure. You can adjust the size of the pool by adding more LUs to it or deleting LUs from it without affecting other definitions.

  Pooled LUs are not dedicated to any particular workstation, nor do they have to be dedicated for use by only downstream gateway users. A single pool can be shared by downstream TN3270 users and SNA gateway workstations, as well as SNA API clients and local emulator sessions. If you want to configure one pool (for example, PUBLIC) for all dependent LU uses, you do not need to know how the users will be distributed across those types.
- Pools can provide better availability for the downstream workstations if you define LUs from multiple connections in the same pool. If one of the connections is active but the other is not, users can access the LUs on the active link.

Consider the following scenarios:

- The Communications Server has two host connections with 100 LUs defined on each connection, all belonging to pool A. The first link is configured as *activate at startup* and is operational. All workstations are using that link with its LUs. The

second link is configured as *auto-activate*. This combination of pools with link configuration can provide better availability in two ways. If the first link fails, the recovering dependent sessions automatically activate the second link. If the first link depletes its defined LUs, the backup link will be automatically activated when the next LU (the 101st) is required.

- You can also use the combination of pools and link configuration to provide load balancing for downstream workstations across available host connections. An example of this is for the Communications Server to have two host connections with 100 LUs defined on each connection, all belonging to pool A. Both links are configured as *activate at startup* and are operational. The workstations that are mapped to the pool of LUs will be distributed across both connections to share the load across them.

The SNA gateway assigns pooled LUs to workstation sessions when the downstream workstation connects to the gateway. The workstation sessions can be defined to use LUs on different hosts with either dedicated or pooled LUs.

Dedicated LUs do not belong to a pool. A dedicated LU can be configured for use by an explicitly defined client.

Figure 29 shows a simple configuration with workstations using dedicated LUs, pooled LUs, or both. (The connecting lines represent the LUs.)

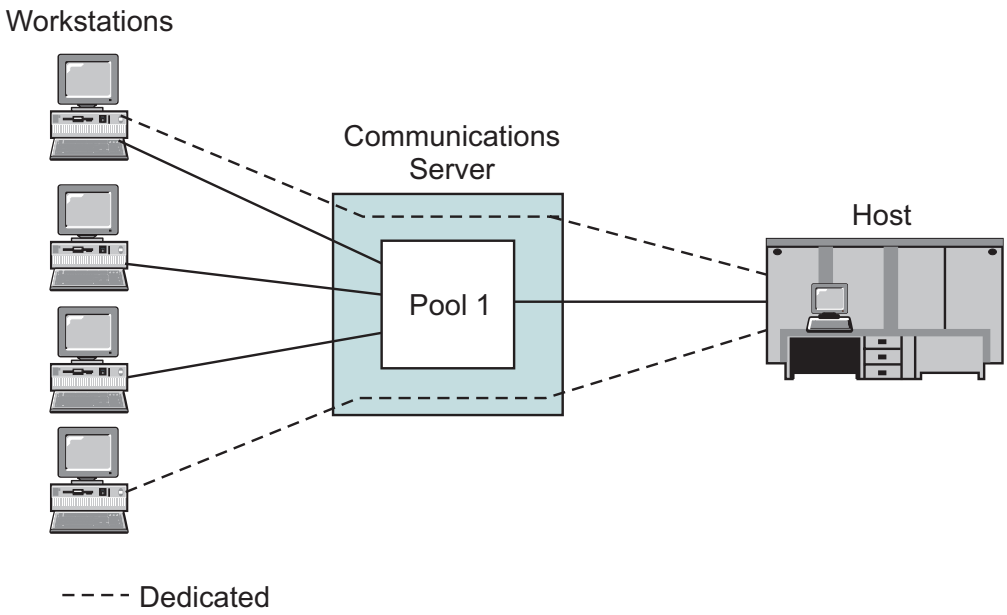

Figure 29. A simple scenario using pooled and dedicated LUs

Using explicit and implicit connections

There are two types of gateway-supported downstream workstations: explicit and implicit. Explicit workstations are workstations that have defined destination addresses over a particular DLC type (for example, Ethernet network and SDLC). To configure explicit workstations, you must know the destination address or fully qualified adjacent control point name or adjacent node ID of each workstation and also define a logical link to the gateway for each workstation. LUs that are defined for explicit workstations can be pooled or dedicated.

Implicit workstations are easier to configure, but they can use only pooled LUs. Instead of defining a link to each workstation using the gateway, define a host LU

pool (or pools) and configure the devices (DLCs) used by the workstation connections. Configure an implicit client template to be used as a model for LU definition for each workstation that connects to the gateway and does not match an explicit definition. For example, if each workstation in an Ethernet LAN has two 3270 sessions configured with NAU addresses 2 and 3, you would configure a client template with two LUs (one for address 2 and one for address 3). If both addresses are used for sessions to a single host, map both addresses to the same host pool. If however, address 2 is used to go to HOST_A and address 3 is used to go to HOST_B, map each to the appropriate host pool. In this example, each time a workstation connects to the gateway over Ethernet, which does not match an explicit definition, a link is dynamically created and the two LUs for NAU 2 and 3 are allocated from the host LU pool(s).

For implicit workstations, users connecting to the gateway need to know only the adapter address of the gateway DLC that is configured for the implicit workstations and what NAU values have been defined on the gateway. They must use these NAU values when defining their 3270 sessions and logical printers.

A downstream Personal Communications workstation connecting to Communications Server can use LAN discovery, searching for the group name `IGO2HOST`, to find the adapter address.

SNA gateway performance

Performance through any gateway is dependent upon many factors, including:

- Configuration of the gateway
- Connection to the host or hosts
- Overall processing demand placed on each host
- Connections to the supported workstations
- Configurations of the supported workstations
- Number of supported workstations
- Number of sessions with each workstation
- SNA transmission priority
- Flow control
- Pacing window for all sessions
- Traffic on non-gateway sessions

Using a workstation that implements a software gateway for other functions can also impair the performance of the gateway.

If none of the previously mentioned factors is causing a negative impact on performance, an individual workstation using supported SNA protocols should not experience any noticeable performance difference between a direct connection to the host and an indirect connection by way of a LAN through an SNA gateway connected to the host. In fact, if the host links are active at the gateway, the workstation activation might improve by removing host overhead delay. However, because of the many variables involved, you might want to conduct performance tests in your operational environment so that you can attain the desired balance between function and performance.

DLUR considerations

By using DLUR for the connection to the host, you gain flexbility in where the SNA gateway can be placed. The connection to the host can traverse any APPN network, and is not restricted to being adjacent to a HOST/NCP. A DLUR to DLUS pipe is created to the DLUS VTAM that is used for dependent session control flows.

The recommended configuration for having a DLUR connection to the host is to define the gateway to be a network node and configure the DLUS information. During configuration of the explicit and implicit clients, map them to the DLUS. This configuration is easiest because it requires no knowledge of the downstream LUs at the gateway and it provides the highest visibility of the downstream devices to VTAM, since VTAM is aware of the PU.

If you do not want VTAM to be aware of the downstream PUs, you can configure an internal PU at the gateway (rather than a host connection), and map the downstream workstations to that internal PU.

If a downstream workstation, such as Personal Communications, is DLUR-capable, you might want that workstation to route through the Communications Server using the network node capability rather than the SNA gateway function.

Chapter 7. Planning for TN3270E Server

The TN3270E server function enables you to configure your network as shown in Figure 30 on page 94 or Figure 31 on page 95.

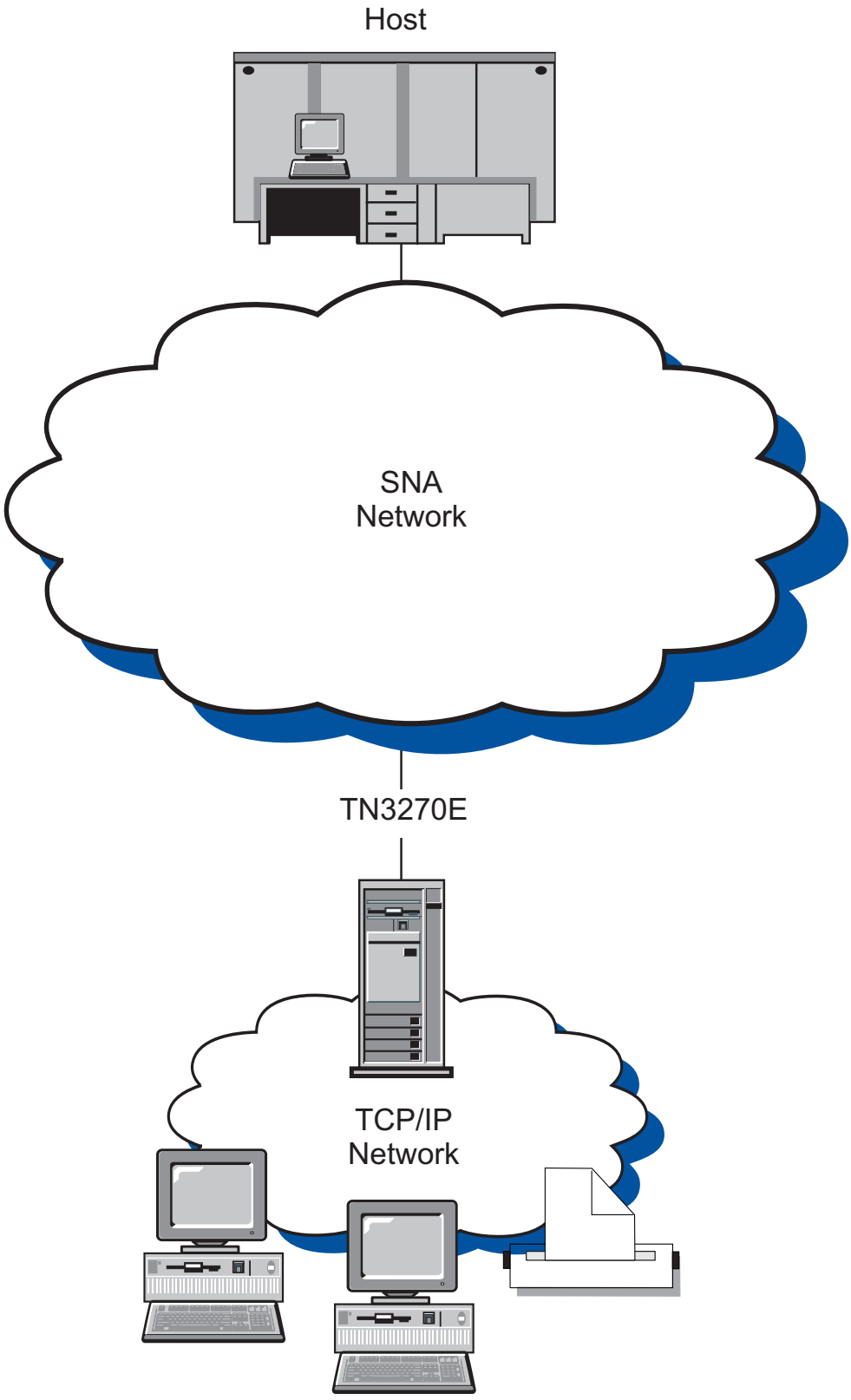

Figure 30. Communications Server configured as a TN3270E Server with a Wide Area SNA network

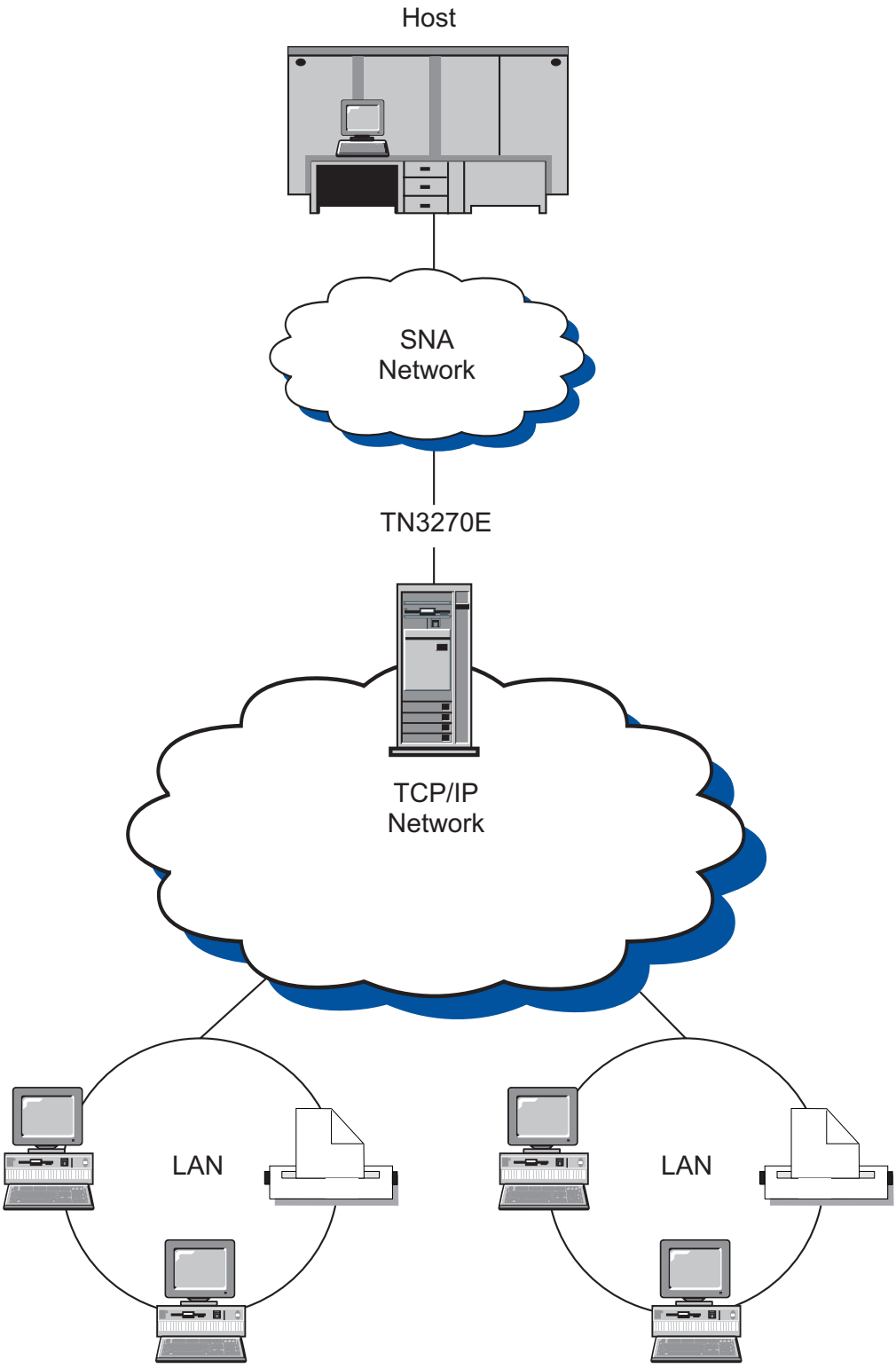

Figure 31. Communications Server configured as a TN3270E Server with a Wide Area TCP/IP network

The TN3270E server function supports:

- Multiple ports

TN3270E server allows the configuration of multiple ports with specifications for pool name, security, client authentication, the level of security, and Certificate Revocation List (CRL) support for each port configured.

- LU emulation

  TN3270E server supports LU 2, which enables users to run interactive 3270 display application programs.
- Host print

  The Telnet 3270 standard extensions (TN3270E) enable users to print from host applications to printers attached to their workstation. These printers may be locally attached or network-attached. LU 1 and LU 3 sessions are supported. TN3270E server implements the protocols outlined in RFC 1646 and RFC 1647, enabling the server to pass LU 1 and LU 3 session data to TN3270E-enabled clients, wait for client confirmation of the print request, and respond to the host.
- IP filtering

  Communications Server supports the specification of TCP/IP client filters, using IP addresses, subnetworks, hostnames, or domain names. This function enables central administration of the clients allowed to connect into the TN3270E server and specification of the LU names and pools clients can access.
- Response handling

  TN3270E-enabled clients can send both positive and negative responses, which TN3270E server sends on to the host. TN3270E server generates request responses for standard TN3270 clients.
- ATTN and SYSREQ key handling

  TN3270E server can convert and forward information to the host when the client sends an ATTN or SYSREQ key. Although clients that are not TN3270E-enabled have no explicit definition for ATTN and SYSREQ, the TN3270E server uses the following Telnet commands to implement these functions:

Table 6. Command equivalents

Telnet	TN3270E	Standard TN3270
IP	ATTN	SYSREQ
AO	SYSREQ	SYSREQ
BREAK	N/A	ATTN

- LU classes

  Communications Server categorizes user connection with LU classes. Classes consist of LUs configured with common characteristics; those that require a specific host connection, for example. This simplifies user access, groups users by application needs, and maximizes host resources.
- Secure Sockets Layer-based (SSL-based) security

  You can specify that a TN3270E server port is secure, indicating that SSL Version 3 be used for connections on the port to provide data encryption and server authentication using digital certificates. Additional configuration parameters are supported to specify the level of security for connections on the port, whether client authentication should be processed, and whether a certificate revocation list (CRL) should be checked before accepting the client connection.

Refer to *Quick Beginnings* for instructions on how to configure a TN3270E server.

Supported client workstations under TN3270E Server

TN3270E server supports any TN3270E or TN3270 client that is fully compliant with RFC 1576, 1646, 1647 or 2355.

Highlights

This section provides more information about some of the features you can configure for TN3270E server.

Changing the default port number

You can configure the ports your server uses for new connections from the **TN3270E Ports** configuration panel. The default port number is 23, but other Telnet applications, such as the TN5250 server support, may also use this port. If other Telnet applications are using port 23, you must use another port.

If you change the port number, avoid numbers that you know are used by other applications. If two applications use the same port number, one of the applications will fail.

If you change the port number, use a number greater than 1 024. Numbers less than 1 024 are reserved. For more information about reserved port numbers, see the following location on the Internet:

http://www.ietf.org/

Notify TN3270E client users when you change the port number, because they will have to configure their emulator applications to match.

Managing system traffic

There are three ways to control how often unused connections are disconnected: keepalive processing, automatic logoff, and LU takeover processing.

By default, TN3270E server does not use keepalive processing. If you use keepalive processing, you can choose either NOP or timing mark.

- NOP processing sends a Telnet NOP command after a specified keepalive frequency. This causes data to be transmitted on the connection, which causes TCP/IP to detect that the connection has broken. The server does not expect a response from the client. It can take an unpredictable amount of time for TCP/IP to detect connection outage.
- Timing mark processing sends a Telnet timing mark command to the client. If the client does not respond within the specified period, the connection is closed.

Timing mark processing causes more traffic on the system than NOP processing, but frees unused connections more quickly.

If you choose automatic logoff, the server disconnects any session that has no traffic for the specified period. Traffic from keepalive processing does not keep the connection open; data must be sent to or from the host. Printer sessions are not automatically logged off.

If your client emulators are configured to do keepalive processing, you might want to turn it off at the server, and if keepalive processing is done at the server, you might want to turn it off at the client to reduce network traffic.

If you choose LU takeover processing, the server will disconnect the session if certain conditions are satisfied. For example, when a new TN3270E session request specifies an LU that is in use at the server, the TN3270E server sends a timing mark to the client of that connection. If the client does not respond to the timing mark within the number of seconds that you specify, the server disconnects the

session and assigns the LU to the new connection request. LU takeover processing enables you to disconnect unused sessions when they are needed, without the network traffice generated by keepalive detection. Keepalive detection enables you to free unused connections more frequently, but generates more network traffic.

IP filtering

Communications Server enables you to configure TN3270E filters to specify which TCP/IP clients can connect into the server. Clients can be specified using individual IP addresses, IP subnetworks, or TCP/IP host names or domain names.

Secure Sockets Layer (SSL) support

When you configure a new TN3270E server port, you can specify that the port be used for secure connections. More than one port can be specified as secure.

To enable security, Communications Server provides a **Key Management** utility to generate the certificate and keys required by SSL. Refer to Chapter 9, "Planning for Secure Sockets Layer-based security," on page 107 for more information.

When you have configured and enabled security, TN3270E clients supporting SSL Version 3 can connect into a secure port number and establish secure connections.

Configuring SNA connections

You must configure your host connection and host LUs before you use the TN3270E server function. For a table that describes and explains host parameters, refer to *Quick Beginnings*.

Pooling

There are four classes of LU definitions specific to TN3270E server: implicit workstation, explicit workstation, implicit printer, and explicit printer. These classes correspond to the terminal-generic, terminal-specific, printer-generic, and printer-specific classes specified in RFC 1647.

Implicit workstation definitions are defined in a pool that the TN3270E server uses to satisfy requests for connections. The default workstation pool can be configured to satisfy requests where the client does not specify a workstation LU or workstation pool name. All other workstation pools must be requested by name.

You can also define a set of LU definitions used to satisfy requests for a specific LU name. These *explicit workstation* definitions ensure that a terminal device needed by a host application is not assigned to a client that does not specifically request it.

Similarly, you can define a pool of printer definitions that will be used to satisfy request for connections that do not require specific LU names (*implicit printer*) and a set used to satisfy requests for a specific LU name (*explicit printer*). The default printer pool can be configured to satisfy requests where the client does not specify a printer LU or printer pool name. All other printer pools must be requested by name.

Implicit and explicit workstation definitions can have printers that are associated with them. Each terminal definition can have a printer that is assigned to it and each printer can have an associated terminal definition. These printers are not included in the explicit or implicit printer definitions.

Associated printer definitions can only be accessed by referencing the terminal LU name. They reduce the amount of information the client user needs, because he or she only needs to know the LU name of the terminal to connect to both the terminal and the printer sessions.

Load balancing of TN3270E Servers

Communications Server also supports load balancing of TN3270 clients across multiple TN3270E servers. Refer to Chapter 10, "Planning for load balancing," on page 115 for more information.

Chapter 8. Planning for TN5250 Server

The TN5250 server function enables you to configure your network as shown in Figure 32 on page 102 or Figure 33 on page 103.

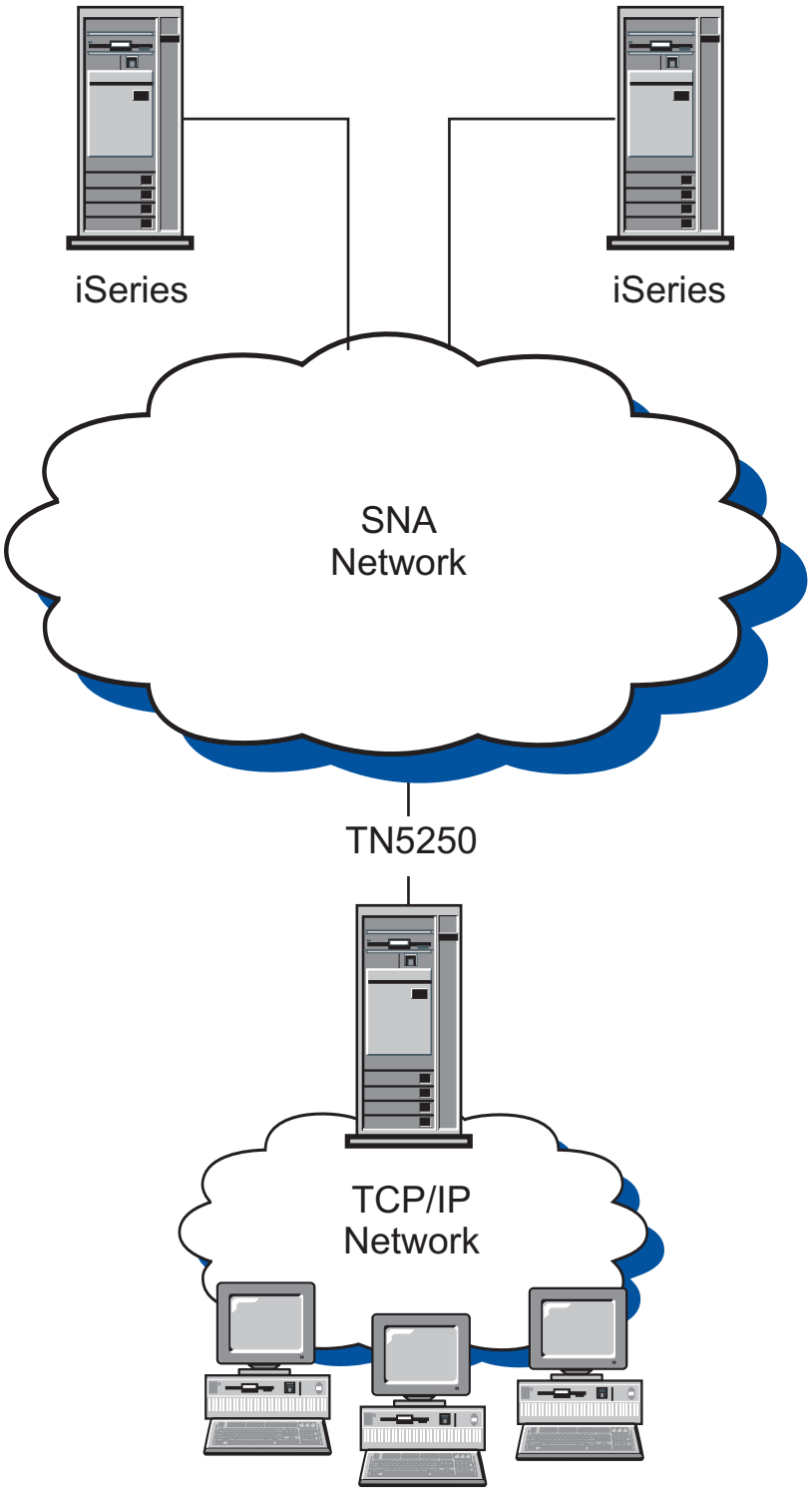

Figure 32. Communications Server configured as a TN5250 Server with a Wide Arean SNA network

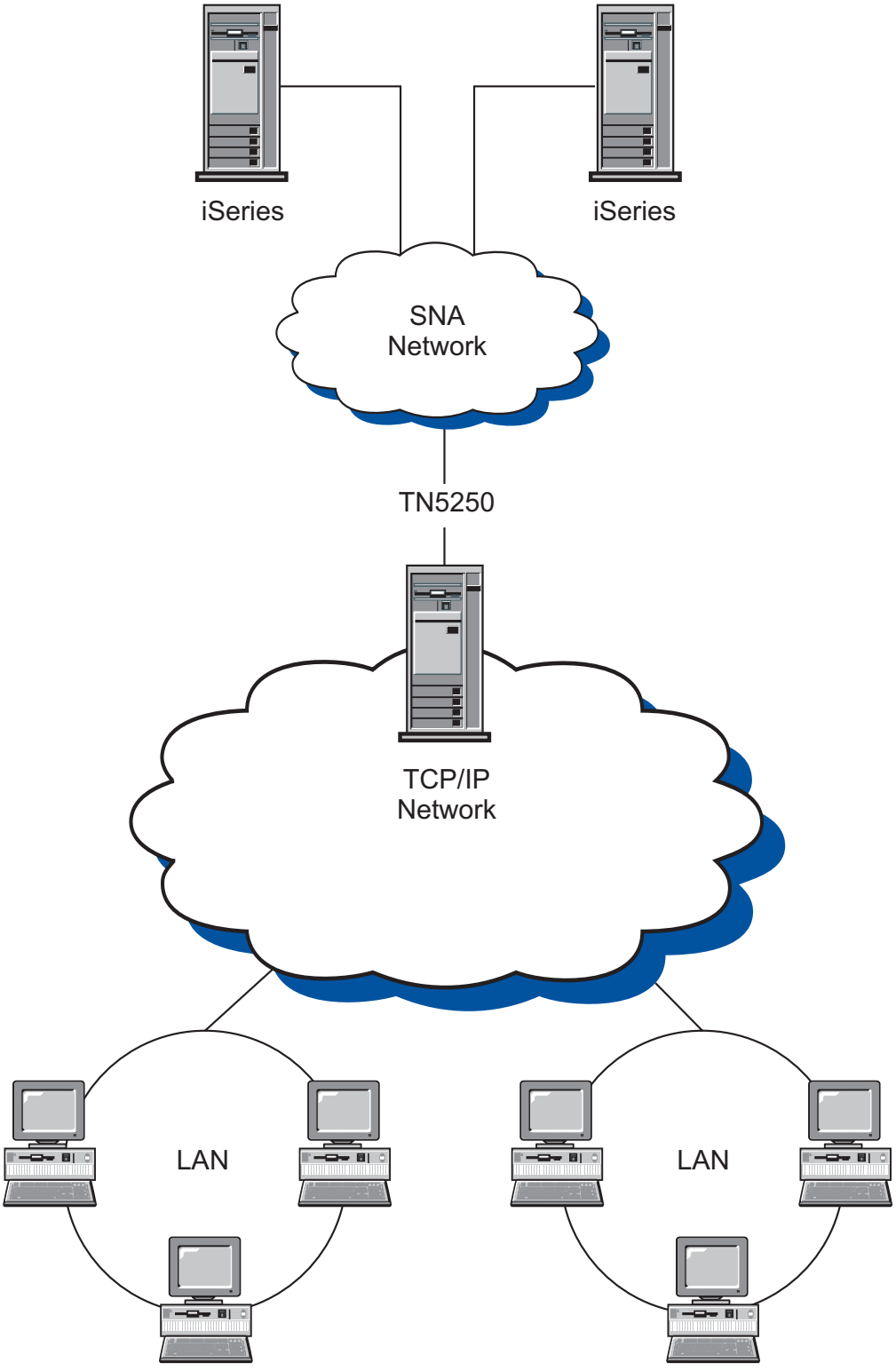

Figure 33. Communications Server configured as a TN3270E Server with a Wide Area TCP/IP network

The TN5250 server function supports:

- Multiple ports

TN5250 server allows the configuration of multiple ports with specifications for server name, security, client authentication, the level of security, and Certificate Revocation List (CRL) support for each port configured.

- Connecting to TN5250 clients with 5250 workstation data stream

  In implementing the protocols outlined in RFC 1205, the server passes 5250 workstation data to and from a TCP/IP client emulating an IBM 5250 workstation. This client is commonly known as a TN5250 client.
- Access to one or more iSeries hosts on an SNA network

  The server connects to iSeries hosts using SNA LU 6.2 protocol. Support is provided to access one or more iSeries hosts from the same or different client workstations.
- IP filtering

  Communications Server enables the specification of TCP/IP client filters, using IP addresses, subnetworks, hostnames, or domain names. This function enables central administration of the clients allowed to connect into the server, as well as support for directing clients to specific iSeries hosts.
- Secure Sockets Layer-based (SSL-based) security

  You can specify that a TN5250 server port is secure, indicating that SSL Version 3 be used for connections on the port to provide data encryption and server authentication using digital certificates. Additional configuration parameters are supported to specify the level of security for connections on the port, whether client authentication should be processed, and whether a certificate revocation list (CRL) should be checked before accepting the client connection.

Refer to *Quick Beginnings* for instructions on how to configure a TN5250 server.

Supported client workstations under TN5250 Server

TN5250 server supports any TN5250 client that is fully compliant with RFC 1205.

Highlights

This section provides more information about some of the features you can configure for TN5250 server.

Changing the default port number

When you configure the TN5250 server support, a default port is defined with port number 23, but other Telnet applications, such as the TN3270E server support, may also use this port. If other Telnet applications are running using port 23, you must use another port.

If you change the port number, avoid numbers that you know are used by other applications. If two applications use the same port number, one of the applications will fail.

If you change the port number, use a number greater than 1 024. Numbers less than 1 024 are reserved. For more information about reserved port numbers, see the following location on the Internet:

http://www.ietf.org/

Notify TN5250 client users when you change the port number, because they will have to configure their emulator applications to match.

Managing system traffic

There are two ways to control how often unused connections are disconnected: keepalive processing and automatic logoff.

By default, TN5250 server does not use keepalive processing. If you use keepalive processing, you can choose either NOP or timing mark.

- NOP processing sends a Telnet NOP command after a specified keepalive frequency. This causes data to be transmitted on the connection, which causes TCP/IP to detect that the connection has broken. The server does not expect a response from the client. It can take an unpredictable amount of time for TCP/IP to detect connection outage.
- Timing mark processing sends a Telnet timing mark command to the client. If the client does not respond within the specified period, the connection is closed.

Timing mark processing causes more traffic on the system than NOP processing, but frees unused connections more quickly.

If you choose automatic logoff, the server disconnects any session that has no traffic for the specified period. Traffic from keepalive processing does not keep the connection open; data must be sent to or from the host.

If your client emulators are configured to do keepalive processing, you might want to turn it off at the server, and if keepalive processing is done at the server, you might want to turn it off at the client to reduce network traffic.

Specifying access to iSeries hosts with multiple ports

Communications Server enables you to configure more than one iSeries, and enables you to configure more than one port for the TN5250 server to listen for incoming connections. When you configure a new TN5250 server port, you can specify the iSeries to be associated with the port. Incoming TCP/IP connections received on that port by the TN5250 server will access the iSeries associated with that port.

Notify TN5250 client users which port numbers to configure for their emulator applications to connect to specific iSeries hosts.

IP filtering

Communications Server enables you to configure TN5250 filters to specify which TCP/IP clients can connect into the server. Clients can be specified using individual IP addresses, IP subnetworks, or TCP/IP host names or domain names.

You can also configure an iSeries with a filter to specify a different iSeries than the one associated with the port. This can be used to direct the clients associated with a filter to a specific iSeries.

More than one iSeries can be specified in a filter. If a client connection attempt to the first iSeries specified is unsuccessful, connection to other iSeries hosts specified in the filter are attempted.

Secure Sockets Layer (SSL) support

When you configure a new TN5250 server port, you can specify that the port be used for secure connections. More than one port can be specified as secure.

To enable security, Communications Server provides a **Key Management** utility to generate the certificate and keys required by SSL. Refer to Chapter 9, "Planning for Secure Sockets Layer-based security," on page 107 for more information.

When you have configured and enabled security, TN5250 clients supporting SSL Version 3 can connect into a secure port number and establish secure connections.

Configuring the SNA network

You must configure your SNA network to allow the TN5250 server to connect to the iSeries hosts. Refer to *Quick Beginnings* for information on configuring your SNA network for the TN5250 server.

Load balancing of TN5250 servers

Communications Server also supports load balancing of TN5250 clients across multiple TN5250 servers. Refer to Chapter 10, "Planning for load balancing," on page 115 for more information.

Chapter 9. Planning for Secure Sockets Layer-based security

You can use Secure Sockets Layer–based (SSL-based) security on connections between TN clients and the TN3270E server or TN5250 server. This security uses SSL Version 3 to provide data encryption and authentication using signed certificates.

You can configure specific ports to support SSL-based security. If you specify security, the server must have an authenticated certificate provided by a certificate authority (CA). Communications Server provides a utility that generates and manages keys and certificates used by SSL Version 3.

You can also configure ports to request client authentication to verify that the client is authorized to establish a secure connection to the server. In addition, if client authentication is configured, you can request certificate revocation list (CRL) processing of client certificates. You specify the location of the LDAP server that maintains the CRL to determine if the client certificate has been revoked.

Note: Support for CRL processing of client certificates requires the use of the IBM Vault Registry product, which publishes Certificate Revocation Lists to an LDAP server.

You can also configure the level of security required by the port. Options include HIGHEST, MEDIUM, and AUTHENTICATION ONLY. The definition of HIGHEST is dependent on the version of Communications Server that is installed. For Communications Server supporting strong encryption, HIGHEST specifies that the port will only accept connections from clients supporting strong encryption. For Communications Server supporting export encryption only, HIGHEST is the same as MEDIUM. MEDIUM indicates that the port can establish connections with any supported level of encryption. AUTHENTICATION ONLY indicates that certificates to authenticate one or both ends of the connection are exchanged, but data is not encrypted.

This chapter gives an overview of planning for Secure Sockets Layer-based (SSL-based) security. SSL-based security is provided as an option when configuring TN3270E server and TN5250 server.

How SSL security works

Secure sockets layer (SSL) is an industry-standard protocol that uses symmetric-key and public-key cryptographic technology. Symmetric-key cryptography uses the same key to encrypt and decrypt messages. Public-key cryptography uses a pair of keys, a public key and a private key. Each server's public key is published, and the private key is kept secret. To send a secure message to the server, the client encrypts the message using the server's public key. When the server receives the message, it decrypts the message using its private key.

SSL provides three basic security services:
- Message privacy

Message privacy is achieved through a combination of public-key and symmetric key encryption. All traffic between an SSL client and an SSL server is encrypted using a key and an encryption algorithm negotiated during session setup.

- Message integrity

  The message integrity service ensures that SSL session traffic does not change in route to its final destination. SSL uses a combination of public/private keys and hash functions to ensure message integrity.

- Authentication

  Authentication is the process whereby the client and the server convince each other of their identities. The client and server identities are encoded in public-key certificates. A public-key certificate contains the following components:
  - Subject's distinguished name
  - Issuer's distinguished name
  - Subject's public key
  - Issuer's signature
  - Validity period
  - Serial number

Communications Server SSL support

Communications Server supports the following components of SSL security:

- Secure SSL sessions between TN3270E server and any TN3270 and TN3270E clients supporting SSL Version 3
- Secure SSL sessions between TN5250 server and any TN5250 clients supporting SSL Version 3
- SSL Version 3
- Message privacy and integrity
- Server-side authentication
- Optional client-side authentication
- Optional certificate revocation list (CRL) processing, when client authentication is requested.

Configuring SSL security

During TN3270E server configuration and TN5250 server configuration, you specify which ports will use SSL security for their connections. See Chapter 7, "Planning for TN3270E Server," on page 93 and Chapter 8, "Planning for TN5250 Server," on page 101 for more information on configuring secure ports.

Server authentication

To enable SSL security with server authentication, a set of public/private keys and an associated authenticated public-key certificate is required. Communications Server provides a **Key Management** utility to create, manage, and store public/private keys and certificates required by SSL.

Using the **Key Management** utility, you must do one of the following:

- Create a public/private key pair and obtain and store a certificate from one of the predefined (well--known) certificate authorities (CAs). See "Configuring SSL using a well-known trusted CA" on page 110.

This procedure requires less setup because the key database file is pre-configured with the well-known CA root certificates required to identify the CAs from whom the certificate is issued.

- Create a public/private key pair, define an unknown CA by obtaining and storing the CA's root certificate in the key database file, and obtain and store a certificate from the unknown CA. See "Configuring SSL using an unknown CA" on page 111 for more information.
- Create a self-signed certificate and store the certificate in the key database file. See 113 for more information.

Note: To ensure adequate security for your site, a self-signed certificate should be used for controlled testing purposes only.

Client authentication

If a secure port is configured for client authentication, SSL requests the client certificate. The Key Database File on the server is checked to verify the signer of the client certificate.

If the client certificate is:

- Issued by a well-known CA, no additional configuration is needed
- Issued by an unknown CA, the root certificate of the unknown CA must be obtained and stored in the key database as a signer certificate. See "Storing certificates for clients" on page 114 for more information.
- A self-signed client certificate, the self-signed certificate must be obtained and stored in the key database as a signer certificate. See "Storing certificates for clients" on page 114 for more information.

If the certificate revocation list (CRL) support is configured, the LDAP server that maintains the CRL is contacted to determine whether the client certificate has been revoked.

Note: Support for CRL processing of client certificates requires the use of the IBM Vault Registry product, which publishes Certificate Revocation Lists to an LDAP server.

Key management utility

The Communications Server **Key Management** utility allows you to create, manage and store public/private keys and certificates required for SSL communications between Communications Server and the client. Before you begin configuring SSL communications, you must open the Communications Server key database file. Use the following steps to open the file:

1. Select **Key Database File** from the main menu, and select **Open** to open an existing key database.
2. From the **Open** window, choose the **private** subdirectory of the directory where you installed the product (for example, C:\Program Files\IBM\Communications Server\PRIVATE).
3. Choose **ibmcs.kdb** as the key database.
4. At the password prompt, enter your password.

Note: The password is initially set to **ibmcs**. For adequate security, you must change the password before performing any of the **Key Management** tasks.

Changing the password

Use the following steps to change the password of the current database:

1. Select **Key Database** File from the menu, and select **Change Password** to change the key database password for the current database. The **Change Password** file dialog appears.
2. Enter the new password you want to use.
3. Enter the same password again to verify.
4. If you want the password to have an expiration date, click **Set expire time**. To ensure adequate security, the password should have an expiration date.
5. If you choose to allow the password to expire, specify the number of days until the password expires.
6. Click **Stash the password** to stash the password of the current database into the **ibmcs.sth** file. Communications Server requires that the password be stashed to access the key database file.

Configuring SSL using a well-known trusted CA

Follow the procedures in this section to set up SSL security using a certificate issued by a well-known CA. The following CA signer certificates are already stored in the key database and marked as trusted certificates:

- Thawte Personal Premium CA
- Thawte Personal Freemail CA
- Thawte Personal Basic CA
- Thawte Premium Server CA
- Thawte Server CA
- RSA secure server CA (also obtained from VeriSign)
- VeriSign class 4 public primary CA
- VeriSign class 3 public primary CA
- VeriSign class 2 public primary CA
- VeriSign class 1 public primary CA

To set up SSL security using a well-known CA, the following procedures are necessary:

- Create a key and certificate request
- Submit a certificate request to the CA
- Obtain and store a certificate in the server key database

Creating a key and certificate request

Use the following steps in the **Key Management utility** to create the public/private keys and certificate request:

1. Select **Personal Certificate Requests** from the drop-down list of the main menu and click **New** to create a new key pair and certificate request. The **Create New Key and Certificate Request** dialog appears.
2. Enter the name (label) that is used to identify the key and certificate within the database.
3. Enter the number of the key size you want to use. Choosing a larger key size results in stronger security, but requires more processing on the client and the server to establish a connection.
4. Enter the TCP/IP host name of the communications server as the common name (for example, wtr05306.raleigh.ibm.com).
5. Enter an organization name.
6. Enter an organization unit (optional).
7. Enter a city or locality (optional).
8. Enter a state or province (optional).

9. Enter a ZIP code (optional).
10. Enter a country code. You must specify at least 2 characters (for example, US).
11. Enter a certificate request file name, or use the default file name.

When you click **OK**, the information you supplied is processed. Two files are produced:

ibmcs.rdb
Private key file.

certreq.arm
This is the default name of the certificate request file. If you assigned a file name, a file with that name is created. The certificate request file is a PKCS 10-type file in armored 64 format.

Do not attempt to edit or move these files. If the **ibmcs.rdb** file cannot be found or has been corrupted when you attempt to enter the certificate into the key database, you will have to resubmit your certificate request to the CA.

Submitting a certificate request

Start a Web browser and access the Web page for the CA. Follow the instructions provided to submit the certificate request. The following are the URLs of some well-known CAs:
- VeriSign: **http://www.verisign.com/**
- Thawte: **http://www.thawte.com/**

Depending on the CA you choose, either e-mail the certificate request generated by the **Key Management** utility or incorporate the certificate request into the form or file provided by the CA.

After submitting a certificate request to a CA, you can enable SSL security by creating and storing a self-signed certificate. A self-signed certificate should be used for controlled testing purposes only. Refer to 113 for more information.

Storing a certificate in the key database

When you receive a certificate from a CA, use the **Key Management** utility to put the certificate into the key database file, **ibmcs.kdb**, located on the server.
1. Select **Personal Certificates** from the drop-down list of the **Key Management** utility and click **Receive** to receive the key pair and certificate request. The **Receive Certificate from a File** dialog appears.
2. Ensure that the data type is **BASE64-encoded ASCII data** (encoded 64 format).
3. Enter the certificate file name.
4. Enter the location (path name) of the certificate. Click **OK**. The stored certificate displays as the first item.
5. Highlight the stored certificate and click **View/Edit**. The **Key information** dialog appears.
6. Click **Set the certificate as the default**. The selected key becomes the default.

Configuring SSL using an unknown CA

To set up SSL security using an unknown CA (not already defined in the database), the following procedures are necessary:
- Create a key and certificate request
- Submit a certificate request to the CA
- Obtain the CA's root certificate and your certificate and store them in the server key database

Creating a key and certificate request

Use the following steps in the **Key Management** utility to create the public/private keys and certificate request:

1. Select **Personal Certificate Requests** from the drop-down list of the main menu and click **New** to create a new key pair and certificate request. The **Create New Key and Certificate Request** dialog appears.
2. Enter the name (label) that is used to identify the key and certificate within the database.
3. Enter the number of the key size you want to use. Choosing a larger key size results in stronger security, but requires more processing on the client and the server to establish a connection.
4. Enter the TCP/IP host name of the communications server as the common name (for example, wtr05306.raleigh.ibm.com).
5. Enter an organization name.
6. Enter an organization unit (optional).
7. Enter a city or locality (optional).
8. Enter a state or province (optional).
9. Enter a ZIP code (optional).
10. Enter a country code. You must specify at least 2 characters (for example, US).
11. Enter a certificate request file name, or use the default file name.

When you click **OK**, the information you supplied is processed. Two files are produced:

ibmcs.rdb
: Private key file.

certreq.arm
: This is the default name of the certificate request file. If you assigned a file name, a file with that name is created. The certificate request file is a PKCS 10-type file in armored 64 format.

Do not attempt to edit or move these files. If the **ibmcs.rdb** file cannot be found or has been corrupted when you attempt to enter the certificate into the key database, you will have to resubmit your certificate request to the CA.

Submitting a certificate request

Follow the procedures of the unknown CA to submit the certificate request.

Depending on the CA you choose, either e-mail the certificate request generated by the **Key Management** utility or incorporate the certificate request into the form or file provided by the CA.

After submitting a certificate request to a CA, you can enable SSL security by creating and storing a self-signed certificate. A self-signed certificate should be used for controlled testing purposes only. Refer to 113 for more information.

Storing a certificate in the key database

When you receive a certificate from a CA, contact the CA to obtain the root certificate of the CA. You must store the CA root certificate in the key database before you store the certificate that you applied for. The CA root certificate validates the certificate you applied for. Use the **Key Management** utility to store the CA root certificate.

1. Select **Signer Certificates** from the drop-down list and click **Add** to receive the CA root certificate. The **Add CA Certificate from a File** dialog appears.

2. Ensure that the data type is **BASE64-encoded ASCII data** (encoded 64 format).
3. Enter the certificate file name.
4. Enter the location (path name) of the certificate. Click **OK**. The file is marked as ***trusted*** and is stored.

To store the certificate you applied for, use the **Key Management** utility to put the certificate into the key database file, **ibmcs.kdb**, located on the server.

1. Select **Personal Certificates** from the drop-down list of the main menu and click **Receive** to receive the key pair and certificate request. The **Receive Certificate from a File** dialog appears.
2. Ensure that the data type is **BASE64-encoded ASCII data** (encoded 64 format).
3. Enter the certificate file name.
4. Enter the location (path name) of the certificate. Click **OK**. The stored certificate displays as the first item.
5. Highlight the stored certificate and click **View/Edit**. The **Key information** dialog appears.
6. Click **Set the certificate as the default**. The selected key becomes the default.

Creating a self-signed certificate

Receipt of a certificate from a well-known trusted CA can take up to three weeks. Until you receive the public server certificate, you can create a self-signed certificate to enable SSL sessions between clients and the server. A self-signed certificate should be used for controlled testing purposes only. To ensure adequate security for your site, a self-signed certificate should not be used in a production environment. To set up your site to use a self-signed certificate using the **Key Management** utility, follow these steps:

1. Select **New Self-signed Certificate** from the **Create** menu option to create a new self-signed certificate. The **Create New Self-Signed Certificate** dialog appears.
2. Enter the name (label) that is used to identify the key and certificate within the database. Select **X509 V3** as the certificate version.
3. Enter the number of the key size you want to use. Choosing a larger key size results in stronger security, but requires more processing on the client and the server to establish a connection.
4. Enter the TCP/IP host name of the communications server as the common name (for example, wtr05306.raleigh.ibm.com).
5. Enter an organization name.
6. Enter an organization unit (optional).
7. Enter a city or locality (optional).
8. Enter a state or province (optional).
9. Enter a ZIP code (optional).
10. Enter a country code. You must specify at least 2 characters (for example, US).
11. Enter the number of days the self-signed certificate is to be valid.
12. Click **OK**.
13. Click **Yes** to set the key as the default key in the key database.

If you are using IBM SecureWay Host On-Demand Version 4 or some other SSL client that requires the self-signed certificate to be designated, using the **Key Management** utility, follow these steps:

1. Select **Personal Certificates** from the drop-down list, highlight your self-signed certificate and click **Extract Certificate**. The **Extract Certificate to a File** dialog appears.
2. Select **Binary DER data** as the data type.
3. Enter the certificate file name. The file should have a file type of **der** (for example, ibmcs.der).

4. Enter the location (path name) of the certificate.
5. Click **OK**.

The certificate file must be supplied to each client authorized to connect to the server. Consult the client documentation for instructions on storing the self-signed certificate as a CA root certificate in the client's class file or key database.

Storing certificates for clients

If a client certificate is issued by an unknown CA or is a self-signed certificate, the root certificate of the unknown CA or the self-signed certificate must be stored in the key database as a signer certificate. Use the **Key Management** utility to store the certificate.

1. Select **Signer Certificates** from the drop-down list and click **Add** to receive the CA root certificate. The **Add CA Certificate from a File** dialog appears.
2. Ensure that the data type is **BASE64-encoded ASCII data** (encoded 64 format).
3. Enter the certificate file name.
4. Enter the location (path name) of the certificate. Click **OK**. The file is marked as ***trusted*** and is stored.

Chapter 10. Planning for load balancing

This chapter gives an overview of planning for load balancing. It describes planning for dependent LU and LU6.2 load balancing.

Dependant LU load balancing

Load balancing allows you to balance dependent host sessions across servers. Load balancing is supported for LUx API applications through SNA API clients, 3270 emulators, and TN3270 emulators.

When Communications Server responds to client requests, it sorts servers supporting the requested pool by load. The load for dependent LUs represents a percentage of available resources from a particular server.

The load percentage is calculated by dividing the number of active application connections by the total number of LUs (0 to 3) available. You can influence the calculated load by specifying an LU0- to-3 load factor (host session load factor) to compensate for differences between two servers, such as available memory, processor speed, and CPU utilization.

You can also use the LU0-to-3 load factor to balance the loads between Communications Servers and Novell NetWare for SAA servers.

Notes:

1. The servers participating in load balancing are determined by LU pool membership and by scope membership. All Communications Servers with LUs in pools of the same name are subject to load balancing.
2. Clients using TCP/IP protocol can participate in load balancing. However, TCP/IP clients can load balance only between IBM Communications Servers. Load balancing is determined by configured scopes.

In Communications Server, if LU pools on different servers have the same name, they function as one pool for load balancing. When assigning LUs to an LU pool, select from the host links on a server, and assign LUs from the selected host link to the LU pool. These LUs do not need to be of the same LU type. Also, the resulting pool can contain LUs from multiple links.

LU 6.2 load balancing

Load balancing allows you to balance independent LU 6.2 sessions by distributing them across multiple servers. Load balancing is supported for APPC API applications through SNA API clients and TN5250 emulators.

The load for LU6.2 represents a percentage of available resources from a particular server. The load percentage is calculated by dividing the total number of conversations over all local LUs on a particular server by the cumulative maximum session limit for all local LUs. The maximum session limit is the LU 6.2 session limit specified during configuration. If the maximum session limit is specified as zero (0), indicating there is no session limit, the default maximum local LU session limit of 512 per local LU is used when the load is calculated. The default maximum local LU session limit can also be specified during configuration.

You can influence the calculated load by specifying an LU6.2 load factor (APPC session load factor) to compensate for differences between two servers, such as available memory, processor speed, and CPU utilization. You can also use the LU6.2 load factor to balance the loads between Communications Servers and Novell NetWare for SAA servers.

Note: Clients using TCP/IP protocol can participate in LU 6.2 load balancing.

Planning for TCP/IP scopes

Scope is a parameter used to control and manage access by TCP/IP clients to servers in a network. It is the same as the service location protocol (SLP) scope as referenced in RFC 2165. The SLP is used by Communications Server to implement TCP/IP-based load balancing.

The control scope provided is necessary for two reasons:

- As your network, the number of clients, and the number of servers grow, it becomes necessary to partition access to those servers by the growing number of clients in order to reduce overall traffic on the network.
- The control scope allows administrators to organize users and servers in to administrative groups.

The meaning of the scope values are defined by the administrator of the network. These values can represent any entity. Commonly, they fall along either departmental, geographical, or organizational lines.

Once configured, clients are able to reach the SNA network through servers that are configured with the same scope or that have no associated scope (unscoped services or servers).

Note: If a server is configured as unscoped, it replies to SLP scoped and unscoped requests. If the SNA API client is configured to connect to unscoped servers, only unscoped servers will reply.

Scopes and client server security

Communications Server enables Novell NetWare for SAA clients to use Windows NT domain security to authenticate the client connection to the server without reentering the userid and password. The client must be part of a Windows NT domain, either by participation in a Communications Server domain or logging in locally with a synchronized userid and password.

The authorized users for client server are maintained in the IBMCSAPI local group, which is located either directly on the Communications Server or on the domain controller where Communications Server participates. This user group is created during installation and can be administered using the Windows NT User Manager application.

Novell NetWare for SAA client users outside of the Windows NT domain must replicate the userid and password across all servers configured with the same scope.

Where scope is configured

Each server is assigned to a scope or scopes through **SNA Node Configuration**. Clients using these servers must be configured to connect to servers within a single

specific scope or to unscoped servers. For more information on configuring clients, refer to *Client/Server Communications Programming*.

How scope relates to SLP

Communications Server scope relates directly to service location protocol (SLP) scope. Therefore, SLP directory agents may reside in the network that support the Communications Server configured scopes. If you plan to allow clients to locate Communications Server services based on scopes, consider how scope relates to the network as a whole. If there are unscoped services in a network where scopes are also used, the unscoped services are eligible to satisfy any scoped requests, which can potentially put a burden on those service agents and directory agents that support the unscoped services.

Note: If the SNA API client is configured to connect to unscoped servers, only unscoped servers will reply.

If directory agents are to be used in the site network (for upward scaling), they should be configured to handle the same scopes as are configured for the Communications Server. In addition, if unscoped services are to be used in networks with directory agents, at least one unscoped directory agent should be configured.

Communications Server includes an SLP service agent that is installed on every server where Communications Server is installed. No additional SLP support needs to be installed in the network in order for TCP/IP load balancing and Communications Server discovery to work properly.

Tracking the current load

You can track the current load for both LU 6.2 and dependent LUs using the Windows NT Performance Monitor. You can start the performance monitor from the **Administrative Tools (Common)** selection of the Programs menu.

From the performance monitor screen, select **Add to Chart** from the **Edit** pulldown. In the **Object** field, select **IBM SNA Load Balancing**. The two counters listed are:

LU0/LU3 load
Use this counter to track the load for dependent LUs.

LU6.2 load
Use this counter to track the load for LU6.2 LUs.

The results of the load tracking can be saved for future reference.

Chapter 11. Planning for backup host connections

For environments where it may be important to provide backup host connections to the primary connections in use by workstation emulator sessions, Communications Server provides you with options that enable configuration of host connections that are activated in response to failure conditions. The options you can use for this depend on the level of backup you want to provide:

- To enable full system backup between server nodes supporting LAN based IP or IPX workstation emulator sessions, include hot standby in your environment.
- To enable full system backup between server nodes supporting other than LAN based IP or IPX workstation emulator sessions, do not include hot standby in your environment. Configure a duplicate Communications Server installation on the backup system.
- To enable backup within a server node, do not use hot standby. Use normal options on connection definitions in your backup environment.

These options are explained in the following sections.

Planning for backup connections between servers using hot standby

In Communications Server, you can configure host links to activate automatically if a critical server fails. Configured connections to a host can continue to function by activating alternate connections on a backup server. This function is known as hot standby.

For the user of a workstation emulator session with a host machine whose initial connection fails due to a Communications Server failure, hot standby enables a subsequent attempt to reestablish the connection with the host machine, automatically routing through an alternate Communications Server. The initial server providing the connection is known as the ***critical server***. The alternate server providing a connection to the host machine after failure of the critical server is known as the ***backup server***. The backup server provides connections that are automatically started in response to detection of a critical server failure and licensing charges related to critical server host connectivity are automatically managed on the backup server.

A server can be both a critical server and a backup server at the same time. Servers can back each other up.

Hot standby enables backup server support for LAN based IP or IPX emulator sessions. Hot standby does not provide backup server support for the SNA environment.

In Figure 34 on page 120, the backup server is configured with a complete backup connection for HOSTCON1, which provides the primary host connectivity on the critical server. The dotted lines between the backup server and the workstation emulator clients are not active while the critical server is in operation.

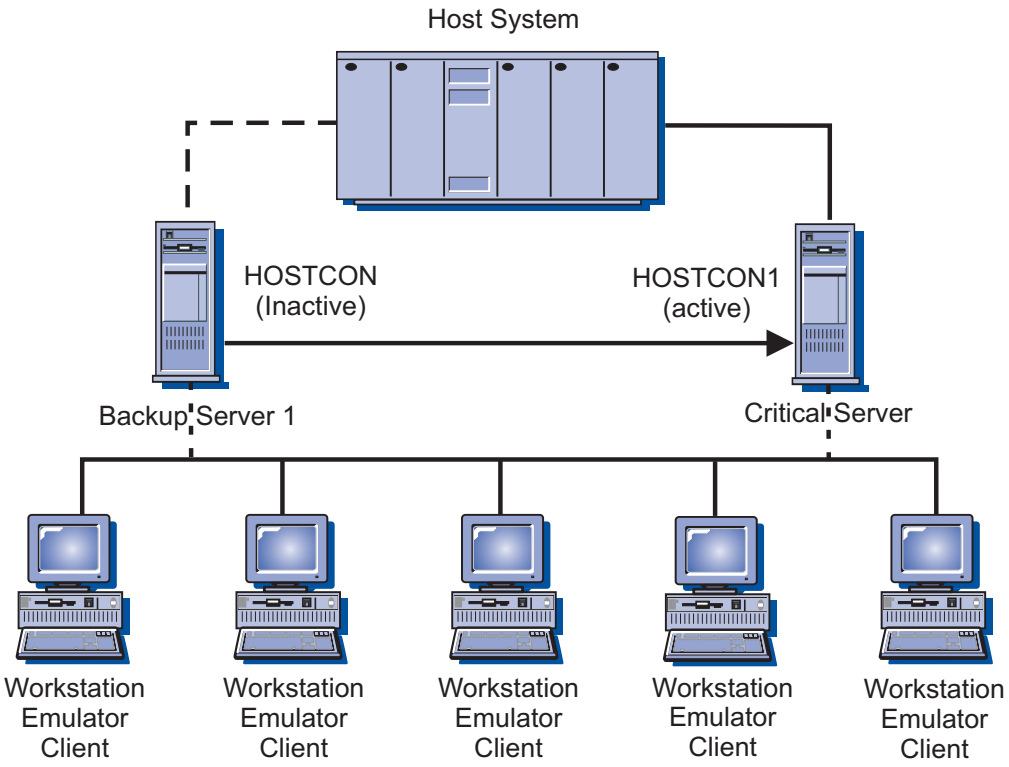

Figure 34. A critical server with one backup server

In Figure 35 on page 121, each of the backup servers are configured with a partial backup connection for HOSTCON1, which provides the primary host connectivity on the critical server. The combination of the two backup servers providing a complete backup for HOSTCON1. The dotted lines between the backup servers and the workstation emulator clients are not active while the critical server is in operation.

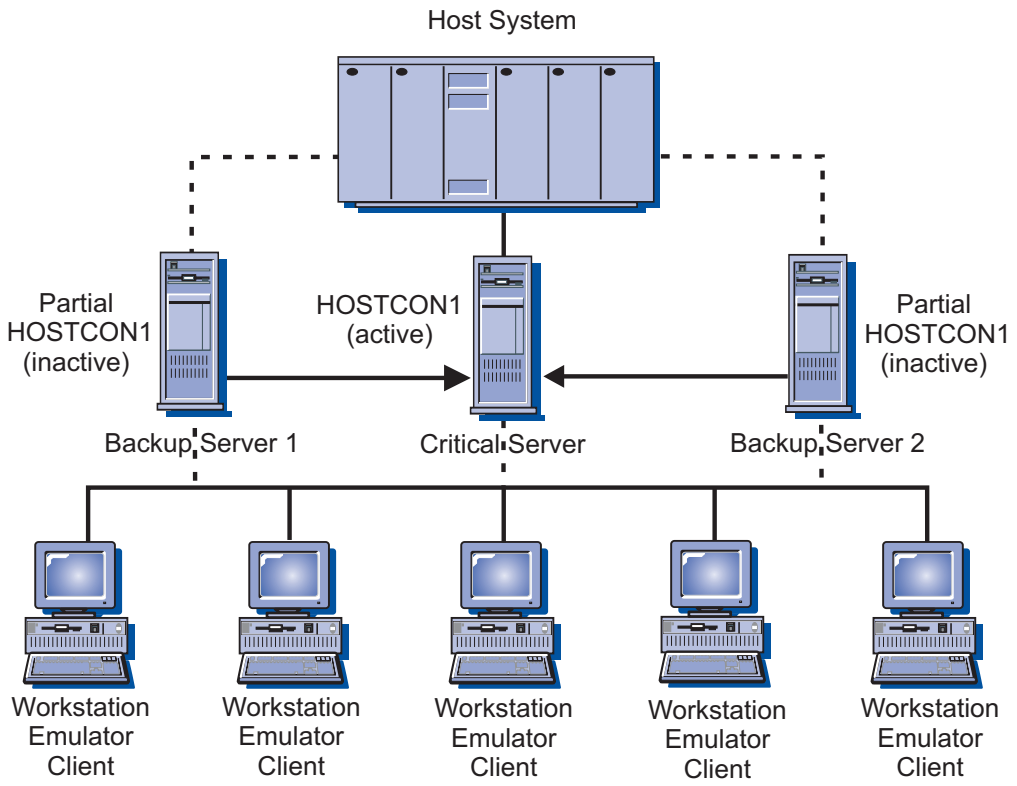

Figure 35. A critical server with two backup servers

If the critical server fails, recovery occurs as follows:

1. The backup server, the host, and the workstation emulator software on the workstation detect the failure at approximately the same time.
2. The connections named in the backup server's critical server configuration are automatically activated and their connectivity to the host machine is established.
3. The workstation establishes a new session with the backup server that has just activated its critical server connections.
4. The workstation contacts the host machine and displays a logon prompt for the user.

Note: When the critical server is active again, the backup server connections activated for it are not automatically deactivated. You must deactivate the backup server connections manually.

Using a production or dedicated backup server for hot standby

You can not install the same Communications Server licenses on more than one server running on the same network. Install Communications Server on multiple servers by purchasing an additional Communications Server package for each additional server.

Before configuring your Communications Server for hot standby, decide whether to run Communications Server as both a production server and a backup server, or as a dedicated backup server.

Using a production server

A ***production server*** is a Communications Server that takes care of your day-to-day communication requirements. This environment configures Communications Server with host connections that are activated outside of a critical server failure scenario. Activation and use of such connections requires purchasing of adequate licenses for them. You can not install the same Communications Server licenses on more than one server. To install Communications Server for use as a production server on multiple systems, you must purchase an additional Communications Server package for each additional server installation.

When you use a Communications Server as both a production server and a backup server, there are no special license considerations. You install Communications Server and the licenses you purchased for use of Communications Server as a production server.

The role of Communications Server as a backup server requires no additional licensing beyond the base licensing purchased for the critical servers it is configured to monitor. You do not need any special license considerations on the backup server. When the backup server detects a failure of a critical server, the backup server automatically activates the connections configured for the critical server and adjusts the backup server's licenses based on those in effect for the critical server. Communications Server manages hot standby licensing without your intervention.

However, if the connections configured for the critical server have already been activated outside of a critical server failure scenario, the backup server's adjustment for the critical server's licenses is not performed and separate licensing on the backup server is required.

Using a dedicated server

A ***dedicated backup*** server does not function as a production server and does not activate connections outside of a critical server failure scenario. No connections can be activated on it at startup or activated manually. Its function is to monitor one or more critical servers.

A dedicated backup server requires no additional licensing beyond the base licensing purchased for the critical servers it is configured to monitor. Communications Server manages hot standby licensing without your intervention.

Planning for the complete hot standby environment

Scenarios for using hot standby involve normal Communications Server functional scenarios for support of LAN based emulator clients over IP or IPX, with some specific hot standby operations.

To configure a hot standby environment, there are two basic dependencies:

- Two or more communications servers must be configured to provide compatible connections to a host machine. The server used as the backup must be configured with a critical server configuration for the server that supplies the primary connectivity to the host machine.
- The workstation emulator client software used to connect through the server to provide users with emulator sessions on the host machine must support some means of effecting alternate routing to the backup server in the event of a failure of the critical server.

Some emulator clients may not provide for alternate routing. Planning for a hot standby environment requires careful evaluation of emulator software and the type of connectivity used between the client and the server. The following are some choices for alternate routing:

- **Hot rollover** - This refers to support in an emulator client for configuration of multiple named servers to be contacted, in a specific order, to establish a connection through the server to the host machine for an emulator session. Hot rollover enables you to configure the backup server to be contacted when attempts to contact the critical server are unsuccessful. When the critical server fails and resources on the backup server are activated, the emulator client contacts the backup server and establishes an alternate route through it for connection to the host machine.
- **LU pools** - This refers to support in an emulator client to dynamically establish the connection to the host machine for an emulator session through the server currently responding to queries for available LUs from a named pool. LU pools support enables the backup server and critical servers to provide LUs to the same pool. The backup server is initially defined but not active. When the critical server fails and the resources on the backup server are activated, the backup server responds to requests from the emulator client for an LU from the pool and establish an alternate route through it for connection to the host machine. LU pools includes the following types of client support for LU pools:
  - Pre-SLP support: Before Communications Server support of SLP, client access of LU pools was random or in a specified server order.
  - SLP support: Communications Server SLP support provides client access of LU pools, balancing the session load across the servers contributing to the pool.
- **Multiple emulator sessions** - This refers to support in an emulator client for configuration of multiple emulator sessions, enabling the user to configure separate sessions using the backup server and critical servers to connect to the host machine. Using multiple emulator sessions is not as easy as hot rollover or LU pool support, because it requires manual intervention by the emulator user to assist with the alternate routing when a failure occurs. When the critical server connection fails, the user manually initiates the session configured to use the backup server to reestablish connection to the host machine.

  IBM Personal Communications support for TN3270E, TN5250, API, or QEL/MU emulator clients provide options for alternate routing through one or more of the hot rollover, LU pools, or multiple emulator sessions options. Some vendor emulator clients provide options for alternate routing.

Configuring hot standby on the backup server

From the **SNA Node Configuration** advanced panel, use the Configure Hot Standby Critical Servers configuration option to configure hot standby on the backup server.

You do not need to perform any hot standby configuration on the critical server, but the backup server requires hot standby configuration. Using the **SNA Node Configuration** advanced option, configure the backup server with critical server definitions. A critical server definition specifies the name of the critical server monitored by the backup server and the names of the connections activated when the critical server fails. Multiple critical server definitions can be configured, enabling a backup server to monitor and provide backup for multiple critical servers. After configuring critical server definitions on the backup server, restart the backup server node for the changes to be put into effect. For detailed information on configuring hot standby, see the online help for **SNA Node Configuration**.

Configuration of a hot standby environment

Follow these steps to configure a hot standby environment:

1. **Configure the Critical Server:** The critical server is configured without any hot standby related configuration. Configure the host connectivity to match the alternate routing option used with the emulator clients.
2. **Configure the Backup Server:** Backup server configuration involves two key concepts:
   a. Configuration of host connectivity on the backup server is similar to the critical server, except that host connections are configured for manual initiation.
   b. Critical server configuration on the backup server to define the host connections started when a critical server failure is detected.
3. **Configure the emulator clients:** The workstation emulator clients are configured to match an alternate routing option to the connectivity being supplied by the servers.
4. **Start the Critical Server and Backup Server nodes:** Start the nodes normally. **SNA Node Operations** can verify the hot standby status for the critical and backup servers.
5. **Critical Server failure:** When the critical server fails, recovery occurs as follows:
   a. The backup server, the host, and the workstation emulator software on the client detect the failure at approximately the same time.
   b. The connections named in the critical server configuration on the backup server are activated and connectivity to the host machine is established.
   c. The workstation emulator client establishes a new session with the backup server. The backup server routes sessions to the host machine through the connection activated on the backup server when the critical server failed.
   d. The workstation contacts the host machine and displays a logon prompt for the user.
6. **Critical Server back online:** When the critical server is active again, new workstation emulator sessions use the critical server host connectivity. Existing workstation emulator sessions routed through the backup server continue operation until the sessions are closed.

   You must deactivate of the backup server host connections manually. Manual deactivation of the backup server connections should be performed during off hours. You should notify users of workstation emulator sessions with the backup server prior to the deactivation of the connections.

Planning for backup connections between servers without using hot standby

Because hot standby is limited to environments supporting LAN based IP or IPX workstation emulator sessions, some environments can not use hot standby for backup connectivity. Many considerations for planning a backup environment without hot standby are similar to backup planning using hot standby. Follow these steps to configure a backup environment without using hot standby:

1. On the backup server, install a separately purchased Communications Server package with adequate licenses to provide for backup host connections to the primary server.
2. On the backup server, configure host connections used as backups to activate on demand.
3. Configure the workstation emulator clients to provide automatic routing to the resources on the backup server when the primary server fails. However, routing should be restricted to access the backup server only if a failure of the primary server is detected.

4. Both nodes are started, but only the primary node with its active upstream connections initially supply connectivity to the host.
5. When the primary server fails, the workstation emulator clients attempt subsequent connections through the backup server. Host connections on the backup server are activated.
6. The duplicate licensing installed on the backup server provides availability of the backup server to clients previously routed through the primary server.
7. When the primary server is active again, connections on the backup server are deactivated manually.

Planning for backup connections in a single server

You can not use hot standby for backup connectivity in a single server, because hot standby only applies to environments with backup connections between servers. However, some considerations for planning a backup environment within a single server are similar to backup planning using hot standby. The differences are summarized by the following steps for backup connections in a single server:

1. On the server acting as the gateway to the host, configure primary and backup connections that supply LUs to the same LU pool, but with the backup connection configured to activate on demand.
2. Configure the workstation emulator clients to connect through the LU pool.
3. When the primary connection fails, or the LUs in the pool are exhausted, an attempt to connect to the pool initiates activation of the backup connection.
4. When the primary server is active again, connections on the backup server are deactivated manually.

Chapter 12. Planning X.25

You need to plan carefully for X.25 configuration. You need to consider the following:

- To use a public PSDN, you need to apply for a network subscription for each line (link) required. It might take some time for a line to be supplied, so check well in advance with the X.25 network provider.
- What functions are available on the X.25 network and what functions are required. The network provider usually supplies a network technical guide, which you should obtain and consult.
- Collect information on logical channels you require for PVCs and SVCs.
  - Up to 16 groups of 256 logical channels can be defined.
  - The group number ranges from 0 through 15.
  - The logical channel number within the group ranges from 0 through 255.

    Multiplying the group number by 256 and adding the logical channel number gives a number in the range from 0 through 4095, which uniquely identifies a logical channel. This combination is sometimes referred to as the logical channel number.

    Notes:

    1. If you specify the CCITT as 1984 or 1988, the maximum number of logical channels is 4095.
    2. If you specify the CCITT as 1980, the maximum number of logical channels is 1024.
  - Groups of logical channels can be incoming-only (reserved for calls from other DTEs), outgoing-only (reserved for calls to other DTEs), or two-way.
- How many virtual circuits your use of SNA requires
- What PVCs you require
- The mix of incoming-only, outgoing-only, and two-way logical channels
- The maximum packet size you require and what is supported by the X.25 network.
- Details of which optional X.25 facilities your X.25 application programs will use.

  The CCITT X.25 recommendations define a number of optional facilities that your network might support.

  Some facilities have parameters that can be specified or negotiated by an application program during call setup and clearing using the facilities fields in the call request, call accepted, and clear request packets.

  The X.25 connection can be configured to support the following facilities during call setup:
  - Closed user group
  - Network user identification
  - Packet and window size negotiation
  - Reverse charging

  Define the facilities during X.25 DLC configuration.

- Details of any special requirements for frame and packet parameters. The default values for these configuration parameters have been chosen carefully to suit most situations. However, your network provider might require or recommend some values.

  To avoid problems, your configuration parameters must match the details of your network subscription. For instance, if you configured 20 SVC logical channels but only subscribed to 10, the DCE would detect an error condition when you attempted to use the 11th logical channel.

Possible X.25 connections

Figure 36 shows how an X.25 network can provide a connection from an SNA gateway to a host.

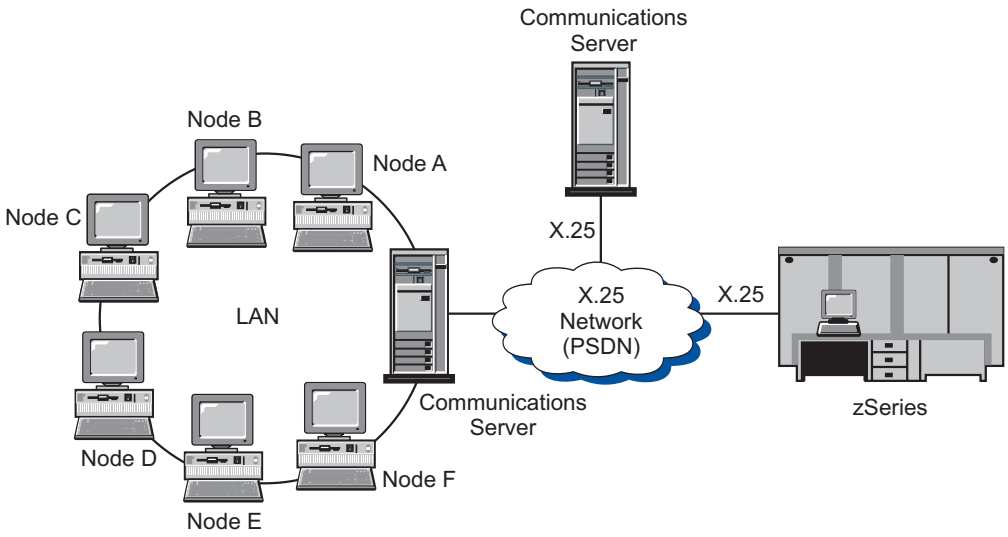

Figure 36. X.25 and SNA gateway connectivity

X.25 network changes

Actual charges depend on the billing practices of the X.25 network provider, but charges usually consist of some or all of the following elements:

- Installation charge

  A one-time installation charge when you first subscribe to the X.25 network.
- Periodic rental charge

  This charge depends on the facilities supplied by your network provider. For instance, you can expect to pay a charge for each optional network facility that you subscribe to. You can also expect to pay a charge for each available logical channel. PVCs generally cost more than SVCs because they require dedicated network resources.
- Usage charge

  This charge depends on the length of time for which you are logically connected to the network, possibly subject to a minimum charge.
- Volume charge

  This charge is related to the amount of data transmitted over the network during a billing period.

Chapter 13. Configuring host connectivity

You can connect workstations running Communications Server to zSeries (including System/370) or iSeries hosts. Many of the values you specify for Communications Server configuration are also used when you configure the host. This chapter describes the mapping between the configuration parameters you specify during Communications Server configuration and the parameters you specify on the hosts. This chapter also contains tables that list the parameters that can be used to eliminate the common parameters on each macro to be issued between Communications Server and the host.

Note: Many NCP parameters issued from the LINE, PU, and LU macros can be specified in the GROUP macro for readability and ease of coding. This matching of parameters eliminates the need to include common parameters on each subsequent macro and still allows the capability to override when appropriate.

NCP Token Ring (NTRI) with VTAM and Communications Server

Table 7 compares host parameters to Communications Server parameters for IBM Token-Ring Network connections.

Table 7. Host/PC Parameters Cross-Reference: IBM Token-Ring Network NTRI Definition

NTRI/NCP	Communications Server Profile/Parameters	Remarks
GROUP ECLTYPE = PHY	DLC Type: LAN	
LINE LOCADDR	Configure Connections LAN **Destination address**	The value set for this parameter under Communications Server must be the address of the IBM Token-Ring Network adapter in the host controller.

Token Ring with VTAM switched network and Communications Server

Table 8 compares host parameters to Communications Server parameters for IBM Token-Ring Network connections using a VTAM switched network.

Table 8. Host/PC Parameters Cross-Reference: IBM Token-Ring network VTAM switched net

VTAM	Communications Server Profile/Parameters	Remarks
IDBLK=	Configure Node Local node ID **Block ID**	The host IDBLK must match the first 3 hexadecimal characters of the **Local node ID**, which defaults to X'05D' for Communications Server. Specify this parameter when using switched lines, or omit this parameter if you use CPNAME=.

Table 8. Host/PC Parameters Cross-Reference: IBM Token-Ring network VTAM switched net (continued)

IDNUM=	Configure Node Local node ID **Physical Unit ID**	The host IDNUM must match the last 5 hexadecimal characters of the **Local node ID**. Specify this parameter when using switched lines, or omit this parameter if you use CPNAME=.
CPNAME=	Node Definition Configure Node **Fully qualified CP name (second field)**	Specify this parameter when using switched lines, or omit when using IDBLK= and IDNUM=.
LOCADDR	Configure Host LUs **NAU address**	These parameters must have the same value. For independent LUs, VTAM recommends creating a CDRSC definition or using the DYNLU operand instead of coding LOCADDR=0. For more information on coding CDRSC definitions for independent LUs, refer to the VTAM Network Implementation Guide No host definitions are required for dependent LUs if SDDLU or dynamic definition is used for independent LUs on VTAM 3.4 or later.
MAXDATA	Configure LAN Device **Maximum PIU size** (265-62235)	These values do not have to match.
MAXOUT	Configure LAN Device **Receive buffer count** (2-64)	The **Receive buffer count** in Communications Server should be equal to the MAXOUT parameter on the host.

9370 IBM Token-Ring network adapter and Communications Server

Table 9 compares host parameters to Communications Server parameters for IBM Token-Ring Network connections using a VTAM switched network.

Table 9. Host/PC Parameters Cross-Reference: IBM Token-Ring network 9370 VTAM LAN

VTAM	Communications Server Profile/Parameters	Remarks
VBUILD TYPE=LAN	DLC Type: LAN	
PORT MACADDR=	Configure Connection LAN **Destination address**	The destination address specified in the Communications Server customization must be the address of the 9370 IBM Token-Ring adapter.
PORT SAPADDR=04	Configure Connection LAN **Remote SAP**	These values must match.

Table 9. Host/PC Parameters Cross-Reference: IBM Token-Ring network 9370 VTAM LAN (continued)

ADDR=04 (outgoing only)	Configure Connection LAN **Local SAP**	These values must match.

Ethernet with VTAM switched network and Communications Server

Table 10 compares host parameters to Communications Server parameters for Ethernet network connections using a VTAM switched network.

Table 10. Host/PC Parameters Cross-Reference: Ethernet network VTAM switched network

VTAM	Communications Server Profile/Parameters	Remarks
IDBLK	Configure Node Local node ID **Block ID**	The host IDBLK must match the first 3 hexadecimal characters of the **Local node ID**, which defaults to X'05D' for Communications Server. Specify this parameter when using switched lines, or omit this parameter if you use CPNAME=.
IDNUM	Configure Node Local node ID **Physical Unit ID**	The host IDNUM must match the last 5 hexadecimal characters of the **Local node ID**. Specify this parameter when using switched lines, or omit this parameter if you use CPNAME=.
CPNAME	Node Definition Configure Node **Fully qualified CP name** (second field)	Specify this parameter when using switched lines, or omit when using IDBLK= and IDNUM=.
LOCADDR	Configure Host LUs **NAU address**	These parameters must have the same value. For independent LUs, VTAM recommends creating a CDRSC definition or using the DYNLU operand instead of coding LOCADDR=0. For more information on coding CDRSC definitions for independent LUs, refer to the *VTAM Network Implementation Guid*e. No host definitions are required for dependent LUs if SDDLU or dynamic definition is used for independent LUs on VTAM 3.4 or later.
MAXDATA	Configure LAN Device **NAU address** (265-65535)	These values do not have to match.

Table 10. Host/PC Parameters Cross-Reference: Ethernet network VTAM switched network (continued)

MAXOUT	Configure LAN Device **NAU address** (2-64)	The Receive buffer count in Communications Server should be equal to the MAXOUT parameter on the host.

VTAM/NCP and Communications Server (SDLC)

Table 11 compares VTAM/NCP host parameters to Communications Server parameters for SDLC connections.

Table 11. Host/PC Parameters Cross-Reference: SDLC

VTAM/NCP	Communications Server Profile/Parameters	Remarks
NETID=	Configure Node **Fully qualified CP name** (first field)	This parameter can be used for both leased and switched lines. This parameter identifies the SNA network you connect to using Communications Server.
NAME PU		Always use a NAME PU. This PU name should be a name that is not the same as the local CP name so you can define an LU for the control point.
IDBLK=	Configure Node Local node ID **Block ID**	The host IDBLK must match the first 3 hexadecimal characters of the **Local node ID**, which defaults to X'05D' for Communications Server. Specify this parameter when using switched lines, or omit this parameter if you use CPNAME=.
IDNUM=	Configure Node Local Node ID **Physical Unit ID**	The host IDNUM must match the last 5 hexadecimal characters of the Local node ID. Specify this parameter when using switched lines, or omit this parameter if you use CPNAME=.
CPNAME	Node Definition Configure Node **Fully qualified CP name** (second field)	Specify this parameter when using switched lines, or omit when using IDBLK= and IDNUM=.
LNCTL=SDLC	DLC Type= SDLC-WAC,SDLC-MPA, COM Port (depends on hardware used)	SDLC must be selected here. This parameter is used for both leased and switched lines.

Table 11. Host/PC Parameters Cross-Reference: SDLC (continued)

LOCADDR	Configure Host LUs **NAU address**	These parameters must have the same value. For independent LUs, VTAM recommends creating a CDRSC definition or using the DYNLU operand instead of coding LOCADDR=0. For more information on coding CDRSC definitions for independent LUs, refer to the VTAM Network Implementation Guide No host definitions are required for dependent LUs if SDDLU or dynamic definition is used for independent LUs on VTAM 3.4 or later.
NRZI=YES/NO	Configure Connections SDLC-WAC,SDLC-MPA, COM Port **NRZI encoding**	Communications Server uses USE_NRZI_ENCODING=0 and USE_NRZI_ENCODING=1. This parameter is found only on PCMs for Modem Type: Synchronous Switched, Synchronous Non-Switched, and AutoSync. The encoding scheme, NRZI or NRZ, must be configured the same way at both ends of a connection. If the encoding schemes do not match, the modems will connect but data transfer will not succeed and the connection will be disconnected by Communications Server after a link-level timeout (typically 40 seconds). We recommend the NRZ encoding scheme when you connect to an X.25 network using an IBM Wide Area Connector.
DUPLEX=FULL/ HALF	Configure Connections SDLC-WAC and COM Port only **Full-duplex communication**	The values set on each side must be consistent with the modem type and configuration. This parameter is used for both leased and switched lines.

Table 11. Host/PC Parameters Cross-Reference: SDLC (continued)

(TRANSFER * BFRS) -47	Configure Devices SDLC-WAC,SDLC MPA, COM Port only **Maximum PIU** (265-4105)	The Maximum I-field size value must be less than or equal to the (TRANSFR * BFRS) - 47 indicated on the host. This parameter is used for both leased and switched lines.
MAXDATA=	Configure Devices SDLC-WAC,SDLC MPA, COM Port only **Maximum PIU** (265-4105)	These values do not have to match. This parameter is used for both leased and switched lines. The lowest value will be used.
MAXOUT=	Configure Devices SDLC-WAC,SDLC MPA, COM Port only **Maximum PIU** (265-4105)	The **Receive window count** in Communications Server should be equal to the MAXOUT parameter on the host. This parameter is used for both leased and switched lines.
ADDR	Configure Devices SDLC-WAC,SDLC MPA, COM Port only **Link station address** (01-FE)	These parameters must have the same value. This is coded in hexadecimal for both Communications Server and VTAM/NCP. This parameter is used for both leased and switched lines.

VTAM and Communications Server (APPC)

Table 12 compares VTAM host parameters to Communications Server parameters for APPC connections.

Table 12. Host/PC Parameters Cross-Reference: APPC

VTAM	Communications Server Profile/Parameters	Remarks
LUNAME	Configure Host LU **Local LU name** or: Configure Node **Fully qualified CP name** (second field)	

Table 12. Host/PC Parameters Cross-Reference: APPC (continued)

LOCADDR	Configure Host LUs **NAU address**	These parameters must have the same value. For independent LUs, VTAM recommends creating a CDRSC definition or using the DYNLU operand instead of coding LOCADDR=0. For more information on coding CDRSC definitions for independent LUs, refer to the VTAM Network Implementation Guide No host definitions are required for dependent LUs if SDDLU or dynamic definition is used for independent LUs on VTAM 3.4 or later.
NETID APPL	Configure Partner LU 6.2 **Partner LU name**	
VTAM Interpret Table	Configure Partner LU 6.2 **Partner LU name**	This is for dependent LUs only.
LOGMODE Table LOGMODE=	Configure Mode **Mode name**	These parameters must match.
LOGMODE Table RUSIZES=	Configure Mode **Maximum RU size** (256-32767)	Final values negotiated.

VTAM and Communications Server (MPC)

Table 13 and Table 14 compare IOCP/HCD and VTAM host parameters to Communications Server parameters for Multi-Path Channel (MPC) connections.

Table 13. IOCP/HCD Parameters Cross-Reference: MPC

IOCP/HCD	Communications Server Profile/Parameters	Remarks
IOCP IODEVICE UNIT=SCTC	DLC Type: IBM-MPC	
HCD Device Type SCTC	DLC Type: IBM-MPC	

Table 14. VTAM Parameters Cross-Reference: MPC

VTAM Definition Statement	Operand	Communications Server Profile/Parameters	Remarks
VBUILD,TYPE=TRL			

Table 14. VTAM Parameters Cross-Reference: MPC (continued)

VTAM Definition Statement	Operand	Communications Server Profile/Parameters	Remarks
	LNCTL=MPC	DLC Type: IBM-MPC	
	MPCLEVEL=HPDT		Communications Server IBM-MPC connections require HPR/HPDT
	MAXBFRU	Configure MPC DLC **Maximum PIU size**	May affect performance and/or VTAM storage use
	MAXREADS	Configure MPC DLC **Receive Buffer Count**	May affect performance and/or VTAM storage use
	READ	Configure MPC DLC	The list of VTAM READ IODEVICE addresses must correspond to the Communications Server MPC DLC list of READ subchannels
	WRITE	Configure MPC DLC	The list of VTAM READ IODEVICE addresses must correspond to the Communications Server MPC DLC list of WRITE subchannels
VBUILD,TYPE=LOCAL			

Table 14. VTAM Parameters Cross-Reference: MPC (continued)

VTAM Definition Statement	Operand	Communications Server Profile/Parameters	Remarks
	CONNTYPE=APPN	Configure Connections IBM-MPC **APPN Support**	Communications Server IBM-MPC connection requires APPN.
	XID=YES	Configure Connections IBM-MPC **APPN Support**	Communications Server IBM-MPC connection requires PU2.1.
	CP-CP=YES	Configure Connections IBM-MPC **APPN**	Communications Server IBM-MPC connection requires APPN.
	HPR=YES	Configure Connections IBM-MPC **HPR Support**	Communications Server IBM-MPC connection requires HPR.
	NN	Configure Node **Node type**	VTAM PU node type must match Communications Server definition.
	DELAY		May affect performance and/or VTAM CPU use.

ES/9000® and Communications Server (SDLC)

Table 15 compares 9370 Information System parameters to Communications Server parameters for SDLC connections.

Table 15. Host/PC Parameters Cross-Reference: 9370 Information System via SDLC

9370/VTAM	Communications Server Profile/Parameters	Remarks
QFI: Non-return -to-zero inverted (microcode configuration)	Configure Connections SDLC-WAC,SDLC-MPA, COM Port **NRZI encoding**	These parameters must match. This parameter is used for both leased and switched lines. Communications Server uses USE_NRZI_ENCODING=0 and USE_NRZI_ENCODING=1. This is the synchronous data stream.
QFI: Protocol (microcode configuration) VTAM: GROUP LNCRL=SDLC	DLC Type SDLC-WAC,SDLC-MPA, COM Port	The microcode and VTAM parameters must match. This parameter is used for both leased and switched lines.

Table 15. Host/PC Parameters Cross-Reference: 9370 Information System via SDLC (continued)

MAXDATA	Configure Devices SDLC-WAC,SDLC-MPA, COM Port **Maximum PIU Size** (265-4105)	These values do not have to match. This parameter is used for both leased and switched lines.
MAXBFRU * IOBUF size	Configure Partner LU 6.2 SDLC-WAC,SDLC-MPA, COM Port **Maximum PIU Size** (265-4105)	The Maximum I-field size value must be equal to or less than the MAXBFRU times IOBUF size indicated on the host. This parameter is used for both leased and switched lines.
MAXOUT=	Configure Devices **Receive Window Count** (1-30)	The **Receive window count** in Communications Server should be equal to the MAXOUT parameter on the host. This parameter is used for both leased and switched lines.
ADDR=	Configure Devices **Link station address** (01-FE)	The values set for these parameters must match. This parameter is used for both leased and switched lines.
IDBLK=	Configure Node Local Node ID **Block ID**	The host IDBLK must match the first 3 hexadecimal characters of the **Local node ID**, which defaults to X'05D' for Communications Server. Specify this parameter when using switched lines, or omit this parameter if you use CPNAME=.
IDNUM=	Configure Node Local Node ID **Physical unit ID**	The host IDNUM must match the last 5 hexadecimal characters of the **Local node ID**. Specify this parameter when using switched lines, or omit this parameter if you use CPNAME=.
CPNAME=	Configure Node **Fully qualified CP name** (second field)	Specify this parameter when using switched lines, or omit when using IDBLK= or IDNUM=.

Table 15. Host/PC Parameters Cross-Reference: 9370 Information System via SDLC (continued)

LOCADDR	Configure Host LUs **NAU address**	LOCADDR and NAU parameters must have the same value. For independent LUs, VTAM recommends creating a CDRSC definition or using the DYNLU operand instead of coding LOCADDR=0. For more information on coding CDRSC definitions for independent LUs, refer to the *VTAM Network Implementation Guide*. This parameter is used for both leased and switched lines.

Host and Communications Server (DLUR)

Table 16 compares host and Communications Server parameters for VTAM connections over DLUR.

Table 16. VTAM Communications Server Parameters Cross-Reference: Using DLUR

VTAM	Communications Server	Remarks
IDBLK=	Configure DLUR PUs **Block ID**	The IDBLK must match the first 3 hexadecimal characters of the Node ID, which defaults to X'05D' for Communications Server. Specify this parameter when NOT using CPNAME=.
IDNUM=	Configure DLUR PUs **Physical unit ID**	The host IDNUM must match the last 5 hexadecimal characters of the Node ID. Specify this parameter when NOT using CPNAME=.
CPNAME=	Configure DLUR PUs **PU name**	The CPNAME must match the local PU name. Specify this parameter when NOT using IDNUM= and IDBLK=.

Table 17 lists the following parameters on the PATH statement that are used when establishing a connection from the host to the workstation.

Table 17. VTAM Communications Server DLUR Parameter Matching Connection from Host to Workstation

VTAM	Communications Server	Remarks

Table 17. VTAM Communications Server DLUR Parameter Matching Connection from Host to Workstation (continued)

DLURNAME=	Configure DLUR **Fully qualified CP name** (second field)	The DLURNAME provides the name of the DLUR node that owns the DLUR PU. For Communications Server, this is the fully qualified CP name.
DLCADDR= (1,C,INTPU)		The DLURNAME provides the name of the DLUR node that owns the DLUR PU. For Communications Server, this is the fully qualified CP name.
DLCADDR= (2,X,zzzzzzzz)	Configure DLUR PUs **Block ID and Physical unit ID**	The zzzzzzzz in DLCADDR=(2,X,zzzzzzzzz) must match the Node ID. Specify this parameter when NOT using DLCADDR=(3,C,puname).
DLCADDR= (3,C,puname)	Configure DLUR PUs **PU name**	The PU name in DLCADDR=(3,C,puname) must match Local PU name. Specify this parameter when NOT using DLCADDR=(2,X,zzzzzzzz).

NPSI and Communications Server (X.25)

Communications Server X.25 support enables the transmission of SNA frames across an X.25 network. X.25 treats SNA frames as raw data and transmits them as data-within-data packets. The two network nodes are not directly connected, but both are connected to an intermediary called the packet-switching data network (PSDN). The X.25 network can comprise any of the X.25 connectivities.

The software that interacts with the PSDN and packages the SNA data into packets and extracts it intact is an integral part of Communications Server. On the S/370 host side, the X.25 functions are handled by a separate software product called X.25 Network Control Program Packet Switching Interface (NPSI). The NPSI definitions are generated along with the NCP and the NPSI software runs in harmony with the NCP in the same 3705, 3725, 3720, or 3745 controller.

The majority of the X.25 parameters specified in both Communications Server and NPSI take values that are agreed upon by you and the X.25 network provider rather than between the two nodes. For example, with an SDLC connection on a non-switched line, the host controller (NCP) is configured to poll a particular secondary address. Communications Server is configured to respond to that address on a physical link between the Communications Server workstation and the host.

NPSI Connections over X.25

Table 18 compares host and Communications Server parameters for NPSI connections over X.25.

Table 18. Host/PC Parameter Cross-Reference: NPSI Connections over X.25

NPSI	Communications Server Profile/Parameters	Remarks
X25.LINE TYPE=	DLC Type X.25 COM Port, X.25-WAC	Specify **TYPE=P** for PVCs and **TYPE=S** for SVCs for NPSI X.25 line definition.
CALL= (only for type=S)	DLC Type X.25 COM Port, X.25-WAC **Virtual circuit ranges**	Virtual circuit ranges must match your X.25 network subscription.
MAXDATA=	Configure Devices X.25 COM Port, X.25-WAC **Maximum PIU size** (256-4105)	These values do not have to match.
VWINDOW	Configure Devices X.25 COM Port, X.25-WAC **Window size** (1-7)	The **SVC/PVC window size** must match the X.25 subscriptions.
MWINDOW	Configure Devices X.25 COM Port, X.25-WAC **Window size** (1-7)	
FRMLGTH	Configure Devices X.25 COM Port, X.25-WAC Frame sequence **Modulo**	The host **FRMLGTH** value must be at least MAXPKT +3 when running Modulo 8 or MAXPKT +4 when running Modulo 128. The host and Communications Server Modulo size and the packet sizes must match your X.25 subscription. MAXPKTL Configure Devices X.25 COM Port, X.25-WAC Packet size The host SVC/PVC
MAXPDTL	Configure Devices X.25 COM Port, X.25-WAC **Packet size**	The host **SVC/PVC packet sizes** must match the X.25 subscription value.

VTAM/NCP and Communications Server (X.25)

Table 19 compares host and Communications Server parameters for X.25 (VTAM/NCP).

Table 19. Host/PC Parameter Cross-Reference: X.25 (VTAM/NCP)

VTAM/NCP	Communications Server Profile/Parameters	Remarks
NETID=	Configure Node **Fully qualified CP** name (first field)	This identifies the SNA network you connect to using Communications Server.

Table 19. Host/PC Parameter Cross-Reference: X.25 (VTAM/NCP) (continued)

PU NAME		Always use a PU name. This PU name should not be the same as the local CP name so you can define an LU for the control point.
IDBLK=	Configure Node Local node ID **Block ID**	First 3 hex digits of the **Local node ID**, which defaults to X'05D' for Communications Server. Specify this parameter when using switched lines, or omit this parameter if you use CPNAME=.
IDNUM=	Configure Node Local node ID **Physical unit ID**	The host IDNUM must match the last 5 hexadecimal characters of the **Local node ID**. Specify this parameter when using switched lines, or omit this parameter if you use CPNAME=.
CPNAME=	Configure Node **Fully qualified CP name** (second field)	Specify this parameter when using switched lines, or omit when using IDBLK= and IDNUM=.
(TRANSFR * BFRS) - 47	Configure Devices X.25 COM Port, X.25-WAC **Maximum PIU size** (265-4105)	The **Maximum PIU size** must be equal to the value set on the host.
MAXDATA	Configure Devices X.25 COM Port, X.25-WAC **Maximum PIU size** (265-4105)	These values do not have to match.
LOCADDR	Configure Host LUs **NAU address**	These parameters must have the same value. For independent LUs, VTAM recommends creating a CDRSC definition or using the DYNLU operand instead of coding LOCADDR=0. For more information on coding CDRSC definitions for independent LUs, refer to the *VTAM Network Implementation Guide.* No host definitions are required for dependent LUs if SDDLU or dynamic definition is used for independent LUs on VTAM 3.4 or later.

APPC over X.25

Table 20 matches the parameters for a Communications Server workstation running advanced program-to-program communications (APPC) connecting to a host by way of X.25.

Table 20. Host/PC Parameter Cross-Reference: APPC on an IBM Token-Ring network using X.25 to a host

NCP	Communications Server Profile/Parameters	Remarks
LU NAME	Configure Local LU 6.2 **Local LU name** or: Configure Node **Fully qualified CP name** (second field)	
LOCADDR	Configure Host LUs **NAU address**	These parameters must have the same value. For independent LUs, VTAM recommends creating a CDRSC definition or using the DYNLU operand instead of coding LOCADDR=0. For more information on coding CDRSC definitions for independent LUs, refer to the *VTAM Network Implementation Guide.* No host definitions are required for dependent LUs if SDDLU or dynamic definition is used for independent LUs on VTAM 3.4 or later.
NETID APPL	Configure Partner LU 6.2 **Partner LU name**	
VTAM Interpret Table	Configure Partner LU 6.2 **Partner LU name**	This is for dependant LUs only.
LOGMODE Table LOGMODE=	Configure Mode **Mode name**	These values must match.

iSeries host line definitions

iSeries host line definitions include matching OS/400 host program parameters with Communications Server parameters. Table 21 contains the SDLC line parameters.

Table 21. iSeries Communications Server Parameters Cross-Reference: OS/400 host program line definition

OS/400	Communications Server Profile/Parameters	Remarks

Table 21. iSeries Communications Server Parameters Cross-Reference: OS/400 host program line definition (continued)

NRZI= Data Encoding	Configure Devices SDLC-WAC,SDLC-MPA, COM Port **NRZI encoding**	Communications Server uses USE_NRZI_ENCODING=0 and USE_NRZI_ENCODING=1. This parameter is found only on PCMs for Modem Type: Synchronous Switched, Synchronous Non-Switched, and AutoSync.
Modem data rate select	Configure Devices COM Port only **Configure modem**	These parameters on both systems for modem connection types, asynchronous switched, must match.
Maximum frame size (MAXFRAME)	Configure Devices SDLC-WAC,SDLC-MPA, COM Port **Receive window count** (1-30)	Asynchronous non-switched values do not have to match.
Maximum outstanding frames (MAXOUT)	Configure Devices SDLC-WAC,SDLC-MPA, COM Port **Receive window count** (1-30)	The Receive window count in Communications Server should be equal to the MAXOUT parameter on the iSeries host.

iSeries host controller definitions

Defining the iSeries host controller includes matching OS/400 host program parameters with Communications Server parameters.

Table 22 shows the OS/400 host program parameters.

Table 22. iSeries Communications Server Parameters Cross-Reference: controller parameters

OS/400	Communications Server Profile/Parameters	Remarks
Linktype: *SDLC	DLC Type SDLC-WAC,SDLC-MPA,COM Port	Link type on both systems must match. This parameter is on the line description, but include for completeness.
Attached Non-switched Line	DLC Type SDLC-WAC,SDLC-MPA,COM Port **Switched Line**	This parameter is for non-switched connections. This value must match a previously defined host line definition.
Switched Line List	(none)	This parameter is for switched connections. This value must match a previously defined host line definition.
Maximum frame size	Configure Devices SDLC-WAC, SDLC-MPA, COM Port **Maximum PIU size** (265-4105)	These values do not have to match.

Table 22. iSeries Communications Server Parameters Cross-Reference: controller parameters (continued)

Remote network identifier	Configure Node **Fully qualified CP name** (first field)	These values must match.
Remote control point name	Configure Node **Fully qualified CP name** (second field)	These values must match.
Data link role:	Configure Connections SDLC-WAC,SDLC-MPA,COM Port **Link station role**	The parameters on both systems are complementary. If the host role is primary the Communications Server workstation must be configured as secondary or negotiable. Unless multipoint (*MP) is used, it is best to set the data link role to negotiable (*NEG).
Station address	Configure Connections SDLC-WAC,SDLC-MPA,COM Port **Link station address** (01-FE)	These values must match.
APPN CP session support: (Yes/No)	Configure Connections SDLC-WAC,SDLC-MPA,COM Port **APPN Support**	These must match if CP-CP sessions are desired.
APPN node type:	Configure Connections SDLC-WAC,SDLC-MPA,COM Port **Adjacent CP type**	These values must match.

iSeries host device definitions

The iSeries host can create a device definition when Communications Server workstation connects without a previously defined device definition.

Table 23 contains the OS/400 host program device parameters.

Table 23. iSeries Communications Server Parameters Cross-Reference: device parameters

OS/400	Communications Server Profile/Parameters	Remarks
Remote location name	Configure Local LU 6.2 **Local LU name** or: Configure Node **Fully qualified CP name** (second field)	These values must match.
Attached controller	(none)	This parameter is for non-switched connections. This value must match a previously defined host line definition.
Local location name		Use the default provided.

Table 23. iSeries Communications Server Parameters Cross-Reference: device parameters (continued)

Remote network identifier	Configure Node **Fully qualified CP name** (first field)	Use the default provided.
Mode	Configure Mode **Mode name**	Use the default provided.

iSeries host line description and Communications Server

Table 24 contains the iSeries host program line description.

Table 24. Host/PC parameters Cross-Reference: line descriptions

OS/400	Communications Server Profile/Parameters	Remarks
Local adapter address:	Configure Local LU 6.2 **LAN destination address**	The two addresses must match to communicate.

iSeries host controller description and Communications Server

Table 25 contains the OS/400 host program controller description.

Table 25. Host/PC Parameters Cross-Reference: controller description

OS/400	Communications Server Profile/Parameters	Remarks
Active switched line	(none)	This parameter is for non-switched connections. This value must match a previously defined host line definition.
Remote control point	Configure Node **Fully qualified CP name** (second field)	The names must match.
Remote network identifier	Configure Node **Fully qualified CP name** (first field)	Use the default or match the parameters if different from the iSeries local network ID.
Model controller description		This parameter must be YES if the iSeries uses a connection network.
Connection network ID and Connection network CP name.	Configure Connection Networks **Connection network name**	These values must match for all nodes participating in a connection network.

iSeries host line definitions (X.25)

iSeries host line definitions include matching OS/400 host program parameters with Communications Server parameters. Table 26 contains the X.25 line parameters.

Table 26. iSeries Communications Server Parameters Cross-Reference: OS/400 host program line definition (X.25)

OS/400	Communications Server Profile/Parameters	Remarks
Local Network Address	Configure Devices **Local Address**	Both addresses are usually determined by your network provider.
Connection Type	Configure Devices X.25-COM Port X.25-WAC **Switched line**	Connection type on both systems must match.
X.25 DCE	N/A	Both addresses are usually determined by your network provider.
Maximum Frame Size	Configure Devices X.25-COM Port X.25-WAC **Maximum PIU Size**	Maximum frame size on both systems must match.
Default Packet Size	Configure Devices X.25-COM Port X.25-WAC **Packet Size**	This value is determined by your network provider, and is the same for Transmit and Receive.
Maximum Packet Size	N/A	This value is determined by your network provider.
Modulus	Configure Devices X.25-COM Port X.25-WAC **Sequencing**	This value is determined by your network provider.
Default Window Size	Configure Devices X.25-COM Port X.25-WAC **Window Size**	This value is determined by your network provider, and is the same for Transmit and Receive.
Insert Network Address	Configure Devices X.25-COM Port X.25-WAC	This value is determined by your network provider, and should be the same for both systems.
Logical Channel Entries	Configure Devices X.25-COM Port X.25-WAC **Virtual Circuit Ranges**	This value is determined by your network provider, and your X.25 virtual circuit type.

iSeries host controller definitions (X.25 PVC)

Defining the iSeries host controller includes matching OS/400 host program parameters with Communications Server parameters. Table 27 shows the OS/400 host program parameters for X.25 PVC.

Table 27. iSeries Communications Server X.25 PVC Parameters Cross-Reference: controller parameters

OS/400	Communications Server Profile/Parameters	Remarks
X.25 logical channel ID	Configure Connections X.25-COM Port X.25-WAC **Logical Channel Number**	These values are determined by your network provider.
Linktype: *X25	Configure Connections	Link types on both systems must match.
Switched Connection:*NO		All PVCs are non-switched.
Maximum Frame Size	Configure Devices X.25-COM Port X.25-WAC **Maximum PIU Size**	The Maximum Frame Size values do not need to match.
Data Link Role	Configure Devices X.25-COM Port X.25-WAC **Link Station Role**	The parameters on both systems are complementary. If the host role is primary the Communications Server workstation must be configured as secondary or negotiable.
X.25 Network Level	Configure Devices X.25-COM Port X.25-WAC **CCITT Compliance**	These values are determined by your network provider.
X.25 Link Level Protocol: *QLLC		This value must be QLLC.

iSeries host controller definitions (X.25 SVC)

Defining the iSeries host controller includes matching OS/400 host program parameters with Communications Server parameters. Table 28 shows the OS/400 host program parameters for X.25 SVC.

Table 28. iSeries Communications Server X.25 SVC Parameters Cross-Reference: controller parameters

OS/400	Communications Server Profile/Parameters	Remarks
Linktype: *X25	Configure Connections	Link types on both systems must match.
Switched Connection:*YES		All SVCs are switched.

Table 28. iSeries Communications Server X.25 SVC Parameters Cross-Reference: controller parameters (continued)

Maximum Frame Size	Configure Devices X.25-COM Port X.25-WAC **Maximum PIU Size**	The Maximum Frame Size values do not need to match.
Connection Number	Configure Devices X.25-COM Port X.25-WAC **Local DTE address**	This value is determined by your network provider.
Data Link Role	Configure Devices X.25-COM Port X.25-WAC **Link Station Role**	The parameters on both systems are complementary. If the host role is primary the Communications Server workstation must be configured as secondary or negotiable.
X.25 Network Level	Configure Devices X.25-COM Port X.25-WAC **CCITT Compliance**	These values are determined by your network provider.
X.25 Link Level Protocol: *QLLC		This value must be QLLC.

Chapter 14. Implementing Communications Server configurations

A suggested procedure for installing and configuring the software for your network follows:

1. Plan for hardware.
2. Plan for software applications.
3. Create your naming conventions.
4. Define network addresses.
5. Select configuration and installation tools.
6. Choosing Communications Server features.
7. Create configurations and install.
8. Create user materials.
9. Maintain your network.

This chapter provides the information necessary for completing the suggested procedure.

Introduction

A template configuration defines the configuration information common to a group of servers. You may want to create one or more template configurations. Each server configuration file starts with the template configuration, and can contain the few parameters required to customize the configuration for that server. This results in simplified server configuration.

The template configuration can also be used to specify configuration options that cannot be specified using response files.

For information on how to use response files and template files for configuration and installation, refer to "Configuration with template and response files" on page 159.

Remember to accumulate totals for RAM and fixed-disk space requirements for each workstation as you gather information in the following steps. That way, you can ensure that you have adequate hardware for your users when they start using the software you have chosen for them.

Step 1. Planning for hardware

If you have already determined that you can use existing servers, you need to make sure that the existing hardware has adequate memory and fixed-disk space. The existing hardware must also have the correct system units, displays, printers, keyboards, adapters, modems, and cables for the software that is going to be installed.

If you are going to acquire new servers, record the hardware on the worksheets you prepare for those users.

Storage

Refer to *Quick Beginnings* for storage requirements for Communications Server.

Memory capacity

When planning the hardware for running Communications Server, it is important to assess how the server will be used in terms of capacity. Determine what type of sessions Communications Server will support. These include, but are not limited to, TN3270E sessions, SNA API client sessions, and traditional SNA sessions to a host. Based on this number, you can calculate the average load on the server and determine the correct amount of memory for the server.

Note: These estimates are in addition to the requirements of the base operating system and any other applications running simultaneously on the same system.

Use Table 29 to determine what the average load on the server will be:

Table 29. Memory capacity

Client session type	Memory usage per session
SNA Gateway (traditional)	29.9 KB
SNA API client	25.2 KB
TN3270E	15.8 KB

The estimated memory consumption of an active Communications Server without any active links or sessions is 21 MB. To minimize performance loss, it is best to reduce the amount of paging (swapping of memory segments out to disk) that occurs. Try to have as much real memory available in the system as required by the running software, with an additional 5 - 10 MB left as a buffer. For example, a Communications Server installation supporting 1 000 TN3270E client sessions would require the following amounts of memory (based on the values in Table 29).

Microsoft Windows Server operating system	19.0 MB (estimated)
Communications Server for Windows	21.0 MB
TN3270E sessions (1 000 X 15.8 KB)	15.8 MB
Contingency	10.0 MB
Total memory recommended	65.8 MB

Note: The 65.8 MB is not a required amount of memory for 1 000 TN3270E sessions. It is a recommendation for best performance.

CPU speed

When determining the minimum processor speed required, you need to understand the average number of transactions that will have to be processed by the server each minute. A transaction is any exchange of information between the client and the host through the server, whether it is a screen refresh or a requested database entry.

When calculating the average CPU load, multiply the average number of sessions with the average number of transactions per minute per session and the scale factor from the following table (based on session type).

Client session type	CPU load factor (100 Mhz)	CPU load factor (166 Mhz)
SNA Gateway (traditional)	.004	.002
SNA API client	.021	.011
TN3270E	.011	.006

For example, to support 1 000 concurrent TN3270E sessions, with an average of 6.7 transactions per minute, the average load on the 166 Mhz CPU would be 1 000 X 6.7 X .006, or 40.2%. Performance degrades quickly when the CPU utilization exceeds 80%. A faster CPU would improve Communications Server and system performance. More users and transactions would be able to take advantage of the server simultaneously. While there is no maximum CPU speed supported by Communications Server, 100 Mhz is the recommended minimum.

Step 2. Planning for software applications

Software applications have requirements in addition to the requirements for Communications Server. For more information, refer to the software application documentation.

Step 3. Creating your naming conventions

Networks created with Communications Server require that you create and use numerous names for multiple objects on the network. Some of these names can be the same from server to server, but other names must be unique on the network to avoid conflicts among the servers trying to access network resources.

For example, two LANs might have some of the same domain names defined. As long as these LANs were not connected, no conflicts would occur. However, if they should be connected later to a backbone LAN, their names would then be in conflict on the LAN.

This means that you must create naming conventions. Naming conventions are rules and standards you use to assign names to the various network resources.

The following sections list the naming conventions for which you need to plan. If you already know what naming conventions you are going to use in your planning, record them while you plan for your network.

Criteria for naming

When you are creating naming conventions, you should determine:

- What criteria to use to create names and addresses for your network
- How you are going to keep these names and addresses consistent and, where necessary, unique within the network

If you are attaching a server to a host computer network, for example, you might find that most of the names you must use in your network are chosen for you by host personnel. In that case, for the physical units and logical units required by the host, you must record those given names for your server configuration files.

As another example, you might choose to name the servers of your network after the names of the people who use them. For a small network, this could be workable as long as the number of names is small and you are able to keep them unique. However, this would not work for a larger network because people's names are not generally unique. You have to create another convention for workstation names for a large network.

Whatever criteria you choose for the names, be sure to record them. That way, you can refer to the appropriate information any time you add new resources to your network.

Types of names and their restrictions

Most names have to be unique within the network in which you use them. The following sections list the types of names you might encounter when you are planning for, installing, and configuring a network. You should preview these names before you begin planning your network so that you are familiar with them.

The following information is provided for each name:
- A definition
- Whether the name must be unique within a network
- Naming restrictions

Names used by more than one component

Names used by more than one component are:
- Network IDs
- Passwords
- User IDs
- Network addressable unit (NAU) names

Network IDs: Network IDs are names given to networks and are used by all the servers and workstations (nodes) within the specific network to maintain a unique identity throughout all the connected networks. The network ID is also used in error logs and network management alerts associated with network system errors.

There are two ways of looking at networks. One way is as a physical network that consists of a "ring" in a token-ring environment or a "string" in an Ethernet or PC network environment. The other way is as a logical network that might not be the same as the physical network. Two or more physical networks (for example, two token rings and an Ethernet string) could be connected with the intent of keeping them in the same logical network.

The network IDs are unique among logical networks; otherwise the networks would be logically the same network. Within a logical network, LU names must be unique to avoid naming conflicts. Between logical networks, the network ID guarantees unique names. An LU name could be the same on two logical networks; however, the network ID for each logical network makes the fully qualified LU names unique. Even if the networks are not currently connected, the network IDs should be unique if you plan to bridge the networks in the future.

You should register your network IDs with IBM. This ensures that SNA networks can be interconnected at a later time without addressing conflicts. Contact an IBM Branch Office for more information about registering your network IDs.

The restrictions for network IDs are:
- Length: From 1 to 8 characters
- Allowable characters: A to Z, 0 to 9, $, @, #

Passwords: Passwords are security functions required by appropriately configured applications and services to protect data and to restrict access to resources.

Passwords do not have to be unique within the network. Passwords are user-specific.

The restrictions for LU-LU passwords are:
- Length: From 1 to 8 characters
- Allowable characters: Any hexadecimal string

The restrictions for other passwords, such as those in CPI-C and iSeries connections, are:
- Length: From 1 to 10 characters
- Allowable characters: Any ASCII string

User IDs: User IDs are unique identifying names you give to the users of your network resources so that they can access database, LAN, or host resources with emulation.

User IDs must be unique in the network.

The restrictions for user IDs are:
- Length: From 1 to 8 characters
- Allowable characters: A to Z, a to z, 0 to 9, $, @, # Network Addressable Unit (NAU) Names

Network Addressable Unit (NAU) names

The types of names that you might have to specify in Communications Server are:
- Control point (CP) names (local node names)
- Logical unit (LU) names

Control Point (CP) names (Local Node Names): The control point (CP) is responsible for managing the node and its resources. In an APPN end node, the control point must communicate with the control point in an adjacent network node to obtain APPN network services. In an APPN network node, the control point must communicate with the control points in adjacent network nodes to manage the network. The control point directs such functions as adapter activation and deactivation, and link activation and deactivation, and assists LUs in session initiation and termination.

The control point name is the second half of the fully qualified CP name in the NODE definition of Communications Server.

Control point names must be unique within a network. However, a node can have multiple PU names that are defined in the connecting (**LINK_STATION**) definition and exchanged on XID3 to different hosts. These multiple PU names must be unique within the node and in the host being connected.

The PU and control point are not the same to subarea VTAM. A PU name for each peripheral node in a VTAM domain is defined at that VTAM, and represents the view VTAM has of the peripheral nodes. The PU names at VTAM are not known to the peripheral nodes; that is, VTAM does not send the PU names to the peripheral nodes. If you want the PU name at a peripheral node to be the same as the one defined at VTAM, you must coordinate this. It is recommended that you do this, but for SNA it is not required.

To VTAM, the control point is an LU, used for activating LU 6.2 sessions between the control point and a VTAM LU (for example, CICS®). The control point name defined at the peripheral node must match an LU definition at VTAM if VTAM initiates LU 6.2 sessions to the control point. Otherwise, VTAM learns the control point name when the peripheral node initiates a session to a VTAM LU.

For Communications Server, the control point name (not including the network ID) is treated as both the local node control point name and its PU name. The only flow that contains the peripheral node PU name is an alert. However, when Communications Server (APPC/APPN) sends an alert, it includes the control point name (resource type is CP) in the alert, not a PU name. If emulators send a PU name in the alerts, the name is the same as the control point name (again, because Communications Server uses the control point name as the node PU name). Host focal point can be defined only on a link where the PU name is the same as the control point name. Further, host links with a PU name other than the control point name cannot have CP-CP sessions with the host or route APPN traffic over the link. All alerts include the control point name even if the alert is caused by a condition on a link using another PU.

The restrictions for local node names are:
- Length: From 1 to 8 characters
- Allowable characters: A to Z, 0 to 9, $, @, #
- First character: A to Z, $, @, #

Logical Unit (LU) names: Logical unit (LU) names are names given to SNA logical entities within a node that provide support functions for transaction processing. This enables them to communicate to other LUs on the network, including host applications.

The restrictions for LU names are:
- Length: From 1 to 8 characters
- Allowable characters: A to Z, 0 to 9, $, @, #

Sockets over SNA LU names: The Sockets over SNA Gateway must have an LU name configured for the Gateway to successfully initialize. Sockets over SNA Gateway will dynamically define the configured LU name to Communications Server upon initialization.

Using a predefined naming convention for Sockets over SNA LU, names can help you:
- Control which names are Sockets over SNA Gateway LUs and which names represent other LUs in your network.
- Analyze and resolve any mapping and routing errors should they occur.
- Enable algorithmic mapping to be used when mapping IP addresses to Sockets over SNA LU names.

Step 4. Defining network addresses

Ensure the consistency and uniqueness of the addresses within your network. Each address must be unique. The addresses that you define will depend on how you configure your network. The following sections describe the addresses for:
- LAN
- SDLC
- X.25

Record the addresses that you use to make sure, when necessary, that none of the addresses conflict and that they are consistent with the naming conventions you have chosen.

LAN adapter addresses

LAN adapter addresses are 12-character hexadecimal numbers encoded in the adapter card by the manufacturer (universally administered adapter address) or

assigned by the network administrator (locally administered address). Each network adapter card in the workstation you are configuring for LAN communications must have a unique address.

You can use the universally administered addresses, also referred to as "burned-in addresses", for your network adapter cards, or you can assign locally administered addresses. LAN adapter addresses must be unique within the network. If you use locally administered addresses, ensure that the addresses are unique within the network.

Locally administered addresses offer a significant advantage in the event of an adapter failure that requires replacement of the adapter. You can transfer the existing address to the replacement adapter and avoid changing any of the configurations referring to this address. If you use universally administered addresses, you have to change the network adapter card address for all of the workstations that access the defective adapter card.

The limitations for configuring LAN adapter addresses are:
- **Length**: 12 characters
- **Allowable characters**: A, B, C, D, E, F, 0-9
- **Range**: X'400000000000' through X'7FFF FFFF FFFF'

On the LAN, you might choose to use locally administered LAN adapter addresses. Thus, you are assigning hexadecimal numbers as the LAN adapter addresses to each of the LAN adapters instead of using the universal LAN adapter addresses built into the LAN adapters. You might decide to use a convention with the following criteria:
- First 4 digits are always X'4000' v
- Next 3 digits are always X'268' to represent your department
- Final digits are in the range from X'01000' to X'FFFFF' to represent the individual workstations of your network

Universally administered addresses on Ethernet are in Ethernet format. You can specify the format type (Ethernet or Token-ring) when using locally administered addresses. When configuring the destination address in SNA connections, make sure that the address format is the same as that specified at the remote. With bridging, it is possible to be on a token-ring locally and have the remote station on an Ethernet and using an Ethernet format address (byte-swapped).

SDLC secondary station addresses

Station addresses are used to identify a secondary station to the network.

Secondary station addresses must be unique within a network. The primary station will communicate with a secondary station by using the secondary station address. The secondary will communicate with the primary by using its own address.

For point-to-point connections, if the secondary station supports the broadcast address X'FF', the primary will learn the remote secondary address. The secondary can specify any value between X'01' and X'FE'.

For primaries that do not support the broadcast address, the secondary station must be set to the same value that is defined at the primary. The value must be between X'01' and X'FE'.

Note: Most stations will support the broadcast address, so the secondary station address at the primary should use X'FF'.

Negotiable stations have local secondary station addresses configured between X'01' and X'FE'. The secondary address of the station negotiated to secondary will be used.

For secondaries on a multipoint connection, the address must match the value specified at the node providing the multipoint primary server function. The address will be in the X'01' to X'FE' range.

X.25 addresses

X.25 addresses are used to identify resources communicating on X.25 networks. X.25 networks implement the CCITT recommendation defining the interface between data terminal equipment and packet-switching networks. X.25 addresses must be unique within a network. You can get these addresses from your X.25 network provider.

Internet addresses

Internet Protocol (IP) addresses are used to route data through the network. Every TCP/IP host is assigned at least one unique IP address. The IP address assigned to the host does not define a host on the network; it defines a network interface on that host to a network.

A Communications Server node must have a unique IP address for each network interface routing TCP/IP data through the node. For example, a Communications Server node that is routing TCP/IP traffic over an SNA network (using the EEDLC IPv4 and IPv6 needs unique IP addresses for both the networks. Each Windows operating system has only one host name although a Windows Server 2003, Windows XP and Windows Vista can support both IPv4 and IPv6 connections.

An IP address consists of a 2-part, 32-bit address field. The first part of the address field contains the network address; the second part contains the host address.

IPv4 addresses are usually specified in dotted decimal format. For example:

```
a.b.c.d
```

where a,b,c and d are values between 0 and 255.

IPv6 addresses are usually specified by hostname but the actual address can be also be used in the form:

```
2002:905:D0A1::905:DOAL
```

Step 5. Selecting configuration and installation tools

Communications Server uses the configuration tools explained in the following sections. Refer to *Quick Beginnings* for more information about these tools.

Node configuration application

The **SNA Node Configuration** application is a graphic window application that enables you to manage SNA configuration information. The application uses a tree view to organize the SNA configuration data to show the relationships between definitions. The user is given task assistance when building a configuration through an integrated task list and context sensitive help. The application is responsible for building the configuration files for the user and verifying the data provided.

Remote configuration

SNA Node Configuration can also be used to connect to a remote Communications Server and directly configure its resources. The user is able to remotely manage the configuration for a Communications Server anywhere in the network.

A remote administration client installs only the administrative applications on a client, including **SNA Node Configuration**. From this client, a user is able to fully administer and configure any Communications Server in the network.

Windows 2000, Windows XP, Windows Server 2003 and Windows Vista Remote Administration Clients use Windows domain security to authenticate the client connection to the server without reentering the userid and password. The client must be part of a Windows domain, either by participation in a Communications Server domain or logging in locally with a synchronized userid and password.

Remote administration client users outside of the Windows domain are required to provide the userid and password, either through a prompt or by storing these values in the client configuration file.

The authorized users for remote administration clients are maintained in the IBMCSADMIN local group, which is located either directly on the Communications Server or on the domain controller where Communications Server participates. This user group is created during installation and can be administered using the Windows User Manager application. Remote administration client users must be given user rights in the IBMCSADMIN group to log on locally to the server.

ASCII configuration file

Communications Server **SNA Node Configuration** stores its configuration data into a human-readable ASCII configuration file. This enables the user to modify configuration files without using **SNA Node Configuration**. (Refer to the *Configuration File Reference* for more information about this file and its syntax.) Using this file, a network administrator can quickly make changes to the configuration using automated tasks such as scripting or software distribution services such as Tivoli Software Distribution support or Microsoft System Management Server.

When creating configurations for a large number of servers to implement, the network administrator can create a template configuration file that represents the common configuration elements for all servers. Using a response file with only those changes necessary for each server, the administrator can distribute the template and response file and merge the two to create the target configuration. For detailed information on how to use template files and response files for configuration and installation, refer to “Configuration with Template and Response Files”.

Configuration with template and response files

The Communications Server template enables you to create or modify a configuration using an editor. You can configure all of the Communications Server configuration keywords and parameters with response files. The template has the same format as Communication Server configuration (.ACG) files.

Note: The format of the .ACG files is documented in the OCDNTS50.DAT file contained in the Communications Server installation directory (for example, C:\Program Files\IBM\Communications Server). Refer to the *Configuration File Reference* for more information on the keywords and parameters used in the .ACG files.

Template files can ease the mass distribution of configurations to remote servers. A template file can specify the keywords which are common to several servers. For example, if you have multiple servers to configure as SNA gateways with implicit client support, many of the keywords will be identical. You can create a template configuration file that reflects those common keywords.

When you create configurations using template and response files, the verification utility searches directories in the following order:
1. The local directory where the verification utility is being invoked.
2. The PRIVATE subdirectory in the Communications Server installation directory.

To ensure that the verification utility can locate the template and response files, you should store them in the PRIVATE subdirectory. The PRIVATE subdirectory is also where the configuration (.ACG) files are stored.

Key fields

The key field is the parameter in a keyword that names the keyword and uniquely identifies it from other keywords of the same type. The @KEY_NAME parameter specifies the key field for the keyword.

The key field is always the first parameter in a keyword that has a key field (for example, **MODE_NAME** in the **MODE** keyword).

Some keywords do not have key fields because they can only be specified once in a configuration file. An example of a keyword that can only be specified once is the **TN3270E_DEF** keyword.

Adding keywords to a template file

When using the response file to add a new keyword definition, the entire keyword must be provided. The key field must be provided along with a unique value. If any subfields are omitted from the keyword, the defaults for those fields are used. For example, to add a **MODE** keyword to the configuration, the response file might contain the following keyword:

```
MODE=(
MODE_NAME=MYMODE
COS_NAME=#INTER
CRYPTOGRAPHY=NONE
DEFAULT_RU_SIZE=1
MAX_NEGOTIABLE_SESSION_LIMIT=8192
MAX_RU_SIZE_UPPER_BOUND=4096
MIN_CONWINNERS_SOURCE=4096
)
```

The content of the response file assumes that a **MODE** keyword with the parameter of MODE_NAME=MYMODE does not exist in the template. If it does, the parameters would have been updated with the values provided in the response file.

If the MODE_NAME parameter was omitted from the response file, an error would occur during the configuration verification because the MODE_NAME parameter could not be uniquely identified. Not all parameters available for the

MODE keyword were specified in the response file. The remaining parameters use the defaults assigned by the OCDSNT50.DAT file. The resulting addition to the configuration would look like this:

```
MODE=(
MODE_NAME=MYMODE
AUTO_ACT=0
COMPRESSION=PROHIBITED
COS_NAME=#INTER
CRYPTOGRAPHY=NONE
DEFAULT_RU_SIZE=1
MAX_NEGOTIABLE_SESSION_LIMIT=8192
MAX_RU_SIZE_UPPER_BOUND=4096
MIN_CONWINNERS_SOURCE=4096
PLU_MODE_SESSION_LIMIT=8192
RECEIVE_PACING_WINDOW=20
)
```

Modifying a keyword in a template file

When using the reponse file to modify an existing keyword definition, the original keyword should exist in the template file. If it does not exist in the template file, the response file adds an entry to the new configuration. The key parameter must be specified in the response file to identify the target keyword. Only those parameters specified in the reponse file keyword are updated in the template file's keyword. Parameters not specified in the response file are left unchanged. For example, if the following **MODE** keyword is in the template file:

```
MODE=(
MODE_NAME=#INTER
AUTO_ACT=0
COMPRESSION=PROHIBITED
COS_NAME=#INTER
CRYPTOGRAPHY=NONE
DEFAULT_RU_SIZE=1
MAX_NEGOTIABLE_SESSION_LIMIT=8192
MAX_RU_SIZE_UPPER_BOUND=4096
MIN_CONWINNERS_SOURCE=4096
PLU_MODE_SESSION_LIMIT=8192
RECEIVE_PACING_WINDOW=20 )
```

and the following keyword is specified in the response file:

```
MODE=(
MODE_NAME=#INTER
AUTO_ACT=10
)
```

the resulting configuration would have the following MODE keyword definition:

```
MODE=(
MODE_NAME=#INTER
AUTO_ACT=10
COMPRESSION=PROHIBITED
COS_NAME=#INTER
CRYPTOGRAPHY=NONE
DEFAULT_RU_SIZE=1
MAX_NEGOTIABLE_SESSION_LIMIT=8192
MAX_RU_SIZE_UPPER_BOUND=4096
MIN_CONWINNERS_SOURCE=4096
PLU_MODE_SESSION_LIMIT=8192
RECEIVE_PACING_WINDOW=20
)
```

Deleting a keyword from a template file

When using the response file to delete a keyword from the template, the key parameter and value that identify the keyword must be specified, along with the keyword DELETE. For example, if the template file specifies the following keyword:

```
MODE=(
MODE_NAME=#INTER
AUTO_ACT=0
COMPRESSION=PROHIBITED
COS_NAME=#INTER
CRYPTOGRAPHY=NONE
DEFAULT_RU_SIZE=1
MAX_NEGOTIABLE_SESSION_LIMIT=8192
MAX_RU_SIZE_UPPER_BOUND=4096
MIN_CONWINNERS_SOURCE=4096
PLU_MODE_SESSION_LIMIT=8192
RECEIVE_PACING_WINDOW=20
)
```

and the response is

```
MODE=(
MODE_NAME=#INTER
DELETE
)
```

the resulting configuration does not contain the #INTER mode definition. The **DELETE** keyword can appear after a *parameter=value* specification or on a line by itself, either preceding or following the parameter. For example, the following uses of the **DELETE** keyword are valid:

```
MODE=(
MODE_NAME=#INTER
DELETE
)
MODE=(
DELETE
MODE_NAME=#INTER
)
MODE=(
MODE_NAME=#INTER DELETE
)
```

The **DELETE** keyword can not appear in front of a parameter=value specification on the same line. For example, the following use of the **DELETE** keyword is not valid:

```
MODE=(
DELETE MODE_NAME=#INTER
)
```

To delete all keywords of a particular type, or to delete one keyword that does not have a key field, only the keyword and the **DELETE** keyword are necessary. For example,

```
TN3270E_DEF=(
DELETE
)
```

Step 6. Choosing Communications Server features

The supported features for Communications Server are described below. Some or all of these functions might be supported for your connection type:

SNA gateway configuration

By configuring an SNA gateway, you can exchange data between downstream client workstations and a Systems Network Architecture (SNA) mainframe host computer. The gateway performs the function of a concentrator, making LU resources from one or more hosts and host PUs available to downstream SNA client workstations, over any communications media supported by Communications Server.

TN3270E server configuration

By configuring a TN3270E server, you can exchange data between downstream TCP/IP client workstations running TN3270E (or TN3270) applications and a Systems Network Architecture (SNA) mainframe host computer. The server performs the function of a concentrator, making LU resources from one or more hosts and host PUs available to downstream TN3270E (or TN3270) sessions, over any communications media supported by Communications Server.

TN5250 server configuration

By configuring a TN5250 server, you can exchange data between downstream client workstations on a TCP/IP network running TN5250 applications and an iSeries on an SNA network. The server performs the function of a gateway, making iSeries display sessions available to downstream TN5250 clients, over any communications media supported by Communications Server.

APPN network node configuration

Network nodes are responsible for maintaining and routing information through an APPN network. They maintain a directory of all resources (LUs) available on directly attached end nodes, as well as maintain a topology of all network nodes in the APPN network. When an independent LU 6.2 session is requested by a node in the network, the network node is responsible for locating the remote resource requested by the session and establishing and routing the session between the two nodes.

DLUR/DLUS configuration

Communications Server supports dependent LU sessions over APPN networks. This requires Communications Server to function as a dependent LU requester (DLUR) that requests routing services from a dependent LU server (DLUS), which resides on a host.

Communications Server supports DLUR for both local sessions and devices, as well as downstream sessions and devices. In either case, the local node must be configured to connect into an APPN network.

AnyNet SNA over TCP/IP gateway configuration

The SNA over TCP/IP gateway routes SNA traffic (independent LU 6.2) between TCP/IP and SNA networks. The SNA over TCP/IP gateway, for example, enables APPC or CPI-C applications running on another AnyNet SNA over TCP/IP access node to communicate with SNA peer nodes on an SNA network.

The AnyNet SNA over TCP/IP Gateway must be configured as an APPN network node. This will enable APPC sessions to be routed to appropriate SNA peer nodes.

SNA API client configuration

SNA API clients are separate machines from the Communications Server which have the ability to run SNA applications while using the SNA code on the Communications Server node to actually manage the session. Communications Server simply establishes and manages a session as if the origin were the local node, but forwards the application data to the SNA API client instead of to its own application APIs.

SNA API clients give you the ability to run SNA applications without having to install an SNA communications stack, such as the Communications Server for Windows, on the same machine. Smaller, less powerful machines can thus be used to run the SNA applications while a centralized, more powerful machine can be dedicated as the SNA server for these SNA API clients.

SNA API clients support two types of applications: APPC (independent LU 6.2) applications and LUA API applications, such as 3270 emulators.

CPI-C or APPC configuration

Communications Server supports CPI-C or APPC applications (for example, APPC3270 or 5250 emulation to an iSeries system). APPC networking is very flexible and versatile and, therefore, it can be complicated. The complexity of this configuration depends on how you will use Communications Server and how much work your application will do for you.

Dependent LU 6.2 sessions to a host

Communications Server supports dependent LU 6.2 sessions with a host PU.

3270 configuration

Communications Server supports 3270 sessions with a host.

Focal point configuration

Communications Server supports remote focal point locations for routing alert information for specific management services applications. The local node will locate remote focal points through an APPN network.

Step 6.1 Planning for application programs

Specifically, check requirements for:

- Modes
- Transaction programs (TPs)
- LU definitions (host, LU 6.2)
- Partner LUs

Step 7. Creating configurations and installing

You need to determine how Communications Server will be configured and installed on your user's servers. You can:

- Give your users instructions for installing and configuring Communications Server. You can either prepare your own instructions or print copies of the Quick Beginnings. You should also provide the specific information for each server, such as the network ID, control point name, and the network address.

- You should give your users instructions for installing Communications Server with an existing configuration or template file. You must also provide a response file tailored for each server, as well as Communications Server CDs or access to CD images.
- Configure and install Communications Server on user servers yourself. Unless you have a very small group of users, you might want to consider using a product that aids configuration, installation, and distribution of software for multiple workstations. Examples of this type of product include IBM NetView Distribution Manager/2 (NVDM/2), Tivoli Software Distribution Support, and Microsoft System Management Server (SMS).

Using response file configuration and installation

If you choose to use response file configuration and installation, refer to "Configuration with template and response files" on page 159.

Step 8. Creating user materials

After you have planned for your network and determined how you are going to implement your plan, you need to create user materials. This means that you should prepare documentation for installation, configuration, and daily use, and prepare backup procedures.

Preparing documentation

Prepare a set of customized documentation to assist your users in installing, configuring, and using Communications Server and local applications for their particular needs. The following sections contain suggestions for the kind of information you should include.

Configuration and installation materials and procedures

Quick Beginnings and online installation helps are available to assist your users in installing Communications Server. Your instructions should tell the user which of the following steps to perform when installing software:

- When to back up configuration files
- Whether to accept the defaults to create a Communications Server configuration file You might need to provide network information, such as LAN adapter addresses, network names, and so forth.
- When to install with a response file that you have provided
- When to install with a customized configuration file that you have provided

If necessary, provide your users with the appropriate documentation.

Communications Server materials and procedures

The following materials are recommended for using Communications Server functions and APIs:

- Quick Beginnings
- User-written programs for the chosen Communications Server APIs

Starting and stopping Communications Server

For the procedure to start or stop Communications Server, refer to *Quick Beginnings*.

If you have different instructions for starting or stopping, supply these instructions to your users.

- Host help telephone number for problem determination (contact your host personnel)

- Logon procedures for host systems or applications
- Procedure for stopping the host application Contact your host personnel.
- Logoff procedures

Your instructions should include any special requirements for logging off systems or applications. This information can be obtained from your host personnel.

Materials and procedures for application programs

You should provide your users with procedures and other information for the application programs you might use on your network. Generally, you should provide:

- Procedures for starting application programs
- Procedures for running application programs v Procedures for responding to messages generated by the application program
- Procedures for problem determination
- If necessary, procedures for stopping the application program

Contact the application programmer for the preceding information.

Problem determination materials and procedures

For the procedures on problem determination and reporting, refer to *Quick Beginnings*.

Preparing backup procedures

Occasionally, your users might erase or change the configuration files, registry, file system, application programs, and other locally created programs or files. Also, your servers might experience erasures or unacceptable changes as well, especially when many users are accessing your server workstations throughout the workday.

Because of this, you need to prepare and document the backup procedures for your network. You might also want to change the attributes of selected files on your servers to Read-Only so your users cannot change them.

Step 9. Maintaining your network

When you have completed planning for, installing, and configuring your network, and it is running on a daily basis, your remaining task is maintenance. You must plan for adding, changing, or deleting resources and users in your network, and plan for problem-solving.

You need to perform the same level of planning and implementation steps for the changes to your network that you performed for initial setup. You should follow the same steps for these changes to your network that you used for the initial planning, installation, and configuration.

Communications Server provides these tools to help you monitor the day-to-day performance of your network:

- **SNA Node Operations**
- Command line utilities
- Remote Administration Client
- System Management API
- SNMP subagent
- Performance Monitor

The following products can also assist you in daily management of your network:

- IBM NetView

- IBM LAN Network Manager
- Tivoli Software Distribution Support

Chapter 15. System management facilities

This chapter provides an overview of the Communications Server system management facilities and describes the capabilities that are common to each facility. Each facility also has some functions that are unique, and those functions are described for each facility.

The system management facilities enable you to monitor and control the communications resources of Communications Server. They also enable you to adjust these resources to improve the efficiency of SNA communication services or to monitor and test these services during problem determination.

Refer to *Quick Beginnings* for a list of resources that you can manage.

You can manage Communications Server with the following facilities:
- **SNA Node Operations**
- Remote administration
- Command-line utilities
- SNA Node Operations ActiveX control
- APPN MIB support
- zSeries Remote Operations Support (ROPS)

You can use any combination of these facilities for Communications Server management.

Common capabilities of system management facilities

Each system management facility can perform the following actions for resources:
- Display resource information
- Start resources
- Stop resources
- Delete resources
- Initiate path switches

You can perform a subset of these actions for each resource, as shown in Table 30.

Table 30. Management activities for Communications Server resources

Resource	Display	Start	Stop	Delete	Path switch
Connections	X	X	X	X	
CPI-C side information	X			X	
Devices	X	X	X	X	
DLUR PUs	X	X	X		
Local LU 0 to 3	X			X	
Local LU 6.2	X			X	
LU 6.2 sessions	X		X		
PU 2.0/DLUR gateway	X			X	
RTP connections	X				X

Table 30. Management activities for Communications Server resources (continued)

Resource	Display	Start	Stop	Delete	Path switch
TN3270e sessions	X		X		
TN5250 sessions	X		X		

SNA Node Operations

SNA Node Operations is an online facility for monitoring and controlling communications resources maintained by Communications Server. **SNA Node Operations** allows you to modify, delete, and monitor resources in your network.

You can use **SNA Node Operations** to help manage the resources in your network. The systems management programming capabilities of Communications Server enable you to configure and manage the nodes within your SNA network. **SNA Node Operations** provides a tree-view diagram of your configuration that enables you to select and modify resources in a hierarchical graphic interface.

You can manage multiple windows within the **SNA Node Operations** main window.

You can define more than one Communications Server configuration and switch between configurations as your needs require. When you start Communications Server, the default configuration is offered as the first choice for node configuration or node operation unless you specify another configuration. Once you have completed a new configuration, you can replace the default configuration with your new one. Use SNA Node Operations to apply your new configuration and ensure that the relevant part of the product runs as you have configured it.

SNA Node Operations enables you to launch other Communications Server programs, such as **Log Viewer, Trace Facility**, or **SNA Node Configuration**.

Remote administration

SNA Node Operations can also be used to connect to a remote Communications Server and directly administer its resources. The user is able to remotely manage the resources for a Communications Server anywhere in the network.

A remote administration client installs only the administrative applications on a client, including SNA Node Operations. From this client, a user is able to fully administer any Communications Server in the network.

Windows remote administration clients use Windows domain security to authenticate the client connection to the server without reentering the userid and password. The client must be part of a Windows domain, either by participation in a Communications Server domain or logging in locally with a synchronized userid and password.

Remote administration client users outside of the Windows domain are required to provide the userid and password, either through a prompt or by storing these values in the client configuration file.

The authorized users for remote administration clients are maintained in the IBMCSADMIN local group, which is located either directly on the

Communications Server or on the domain controller where Communications Server participates. This user group is created during installation and can be administered using the Windows User Manager application. Remote administration client users must be given user rights in the IBMCSADMIN group to log on locally to the server.

Command-line programs

Command line programs enable you to issue commands from a Windows prompt window. The output of commands is displayed on the screen, but the output can be redirected to a file.

Refer to *Quick Beginnings* for a list of command line programs you can use.

SNA Node Operations ActiveX control

There is an ActiveX control version of **SNA Node Operations**. This control allows you to imbed **SNA Node Operations** into any application that allows embedded controls such as the Microsoft Management Console (MMC). The name of the control is CSNTACTX and it is located in the product installation directory. This control has all of the **SNA Node Operations** functions and provides the benefit of embedding **SNA Node Operations** into other applications, enabling cross product management.

APPN MIB support

Communications Server supports simple network management protocol (SNMP) requests for APPN management information from any SNMP management system.

The SNMP Service is installed from the Services tab under the Network pulldown of the Windows NT control panel. After the SNMP Service software is installed on your computer, you must configure it with valid information for SNMP to operate.

You must be logged on as a member of the Administrator group for the local computer to configure SNMP.

The SNMP configuration information identifies communities and trap destinations.

zSeries Remote Operations Support (ROPS) for the NetView program

When RUNCMD is issued from NetView, the value for the APPL parameter must be specified as APPL=CSRCMDS.

When the RUNCMD command is issued, all information, including the ID and password, is transmitted in the clear and is not secured in any way, unless the NetView suppression character is used when the RUNCMD command is issued. This information is displayed on the screen and entered in the NetView log. If passwords are transmitted to the ROPS service, the user is responsible for securing the NetView terminal and its NetView logs so that sensitive data is not compromised.

Appendix A. Planning for Communications Server

Modems

Communications Server supports synchronous modems and CSU/DSU units that support the V.24bis command set. Refer to your network supplier for the recommended connection type and modem.

Session-level encryption adapters

To use session-level encryption, you must have the Programmed Cryptographic Facility program and one of the following adapters:

- The IBM SecureWay 4758 PCI Cryptographic Coprocessor Adapter
- Other adapters compliant with the IBM Common Cryptographic Architecture

Note: If you use an adapter other than the IBM 4758, you will need to use cryptographic API verbs instead. Refer to the *Communications Server Programming Guide and Reference* for more information.

Data compression considerations

Prior to using SNA data compression several configuration steps have to be worked out on the involved systems. SNA data compression is supported on:

- VTAM V3R4.1 or higher
- OS/400 V2R3 or higher

When compression is done, each RU is compressed independently and acompression header is added to the RU. The TH and RH are not compressed. No compression is done when the RU is smaller than 20 bytes, or greater than 64K bytes, or, in the case of RLE only, if the size of the compressed data, including the compression header, is greater than the original data.

Before you start to configure the appropriate features of Communications Server for data compression, ensure that your partner system is able and prepared to support data compression.

Planning the Host/VTAM connection

Before you can request data compression on a session to an S/370 or zSeries® host with 3270 emulation or LU 6.2 type communications, VTAM Version 3 Release 4.1 (VTAM V3R4.1) must be installed and configured to negotiate such requests.

Your host VTAM system programmer should supply you with the proper VTAM definitions required for data compression. Here is a summary list of the definitions required. Examples of these are given following the list.

- The VTAM startup options must contain the **CMPVTAM**=x parameter
- You need a LOGMODE entry with the parameter **COMPRES**=
- This LOGMODE entry must be in the Mode table that you are using
- There must be a VTAM APPL statement with the parameters **CMPAPPLI**= and **CMPAPPLO**= for the application that you will be using
- The VTAM LU definition for your session must specify the LOGMODE and Mode table that were defined for compression.

CMPVTAM

This is the VTAM startup options parameter indicating the maximum compression level allowed for any session connected to this host. The syntax of this parameter is:

```
CMPVTAM=n
```

where n is the level of compression allowed for a request, ranging from 0 to 4. The recommended value is 4, which allows all levels of data compression to be used.

COMPRES

This is the VTAM MODEENT table entry in the LOGMODE definition that will be used by your LU. The syntax is:

```
COMPRES=value
```

where value stands for SYSTEM, REQUESTED or PROHIBITED. SYSTEM should be coded for all supported LU types.

CMPAPPLO

This is the VTAM application (APPL) definition parameter that sets the requested compression level for outbound Request/Response Unit (RU) data. The syntax is:

```
CMPAPPLO=n
```

where *n* is the requested data compression level for the PLU →SLU direction of data flow, also called outbound data. For all LU type sessions supported by Communications Server, the value of *n* can be equal to 0, 1, or 2, representing the data compression level 0, level 1, and level 2. However, for LU types 0, 1, 2, and 3, Communications Server always reserves compression resources at level 2.

CMPAPPLI

This is the VTAM application (APPL) definition parameter that sets the requested compression level for inbound Request/Response Unit (RU) data. The syntax is:

```
CMPAPPLI=m
```

where *m* is the requested data compression level for the SLU → PLU data flow, also called inbound data. The value for *m* depends on the LU type used for the session:

- For 3270 LU type sessions, *m* can have the value equal to 0 or 1.
- On LU 6.2 type sessions, *m* can have the values 0, 1 or 2.

Note: VTAM always represents the primary LU (PLU) and Communications Server represents the secondary LU (SLU) for LU types 0, 1, 2 and 3.

Following are coded examples, which were extracted from the original VTAM listings, pointing out the usage and placement of the parameters discussed above. Pay special attention to the highlighted lines.

```
*********************************************************
SSCPID=20,HOSTSA=20,XNETALS=YES,                              X
GWSSCP=YES,                                                   X
NODELST=NODES1,                                               X
CMPVTAM=4,                                  X
CONFIG=K0,SUPP=NOSUP,                                         X
NETID=USIBMRA,HOSTPU=ISTPUS20,HOSTSA=20,SSCPNAME=RAK,         X
SSCPDYN=YES,SSCPORD=PRIORITY,                                 X
ASYDE=TERM,                                                   X
NOTRACE,TYPE=VTAM,IOINT=0,                                    X
NOTRACE,TYPE=SMS,ID=VTAMBUF,                                  X
PPOLOG=YES,                                                   X
NODETYPE=NN,                                                  X
CPCP=YES,                                                     X
CSALIMIT=0,                                                   X
NOTNSTAT,DYNLU=YES,                                           X
IOBUF=(3500,256,3,,1,58),                                     X
LPBUF=(1100,,2,,1,4),                                         X
LFBUF=(100,,,,1,1),                                           X
CRPLBUF=(2400,,,,1,4),                                        X
SFBUF=(60,,,,1,1)
*********************************************************
```

Figure 37. VTAM startup parameter compression level up to 4 is allowed in this VTAM

```
************************************************************************
* LOGMODE FOR COMPRESSION TEST - BASED ON D4C32XX3                     *
************************************************************************
*                                                                      *
*                   3274 MODEL 1C (REMOTE SNA) @OY02946                *
*                   PRIMARY SCREEN 24 X 80 (1920)                      *
*                   ALTERNATE SCREEN TO BE DETERMINED BY APPLICATION   *
*                                                                      *
************************************************************************
D4C3COMP MODEENT LOGMODE=D4C3COMP,                                     *
                 FMPROF=X'03',                                         *
                 TSPROF=X'03',                                    *
                 PRIPROT=X'B1',                                        *
                 SECPROT=X'90',                                        *
                 COMPROT=X'3080',                                      *
                 RUSIZES=X'87F8',                                      *
                 PSERVIC=X'028000000000000000000300',                  *
                 APPNCOS=#CONNECT,                                     *
                 COMPRES=REQD
```

Figure 38. VTAM logmode table. The MODEENT macro configured with data compression requested.

```
     VBUILD TYPE=APPL
*********************************************************************
* VTAM APPL STATEMENTS FOR CICS/MVS* 3.3                            *
*********************************************************************
RAKAC001 APPL ACBNAME=RAKAC001,                                     *
              MODETAB=MTAPPC,                                       *
              EAS=20,                                               *
              SONSCIP=YES,                                          *
              AUTH=(ACQ,VPACE,PASS),                                *
              PARSESS=YES,                                          *
              VPACING=5,                                            *
              CMPAPPLI=1                                            *
              CMPAPPLO=2
```

Figure 39. VTAM application statement. It controls the maximum supported levels for this session.

```
WTCC1102 PU ADDR=13,
            IDBLK=05D,
            IDNUM=32289,
            etc.
            DLOGMOD=D4C3COMP
            MODETAB=AMODETAB
            etc.


RACC1102 LU LOCADDR=2,DLOGMOD=D4C3COMP,MODETAB=AMODETAB
         LU
         etc.
```

Figure 40. VTAM PU and LU Definitions. The LU RACC1102 uses the logmode D4C3COMP, which is in the AMODETAB modetable.

Planning for iSeries data compression

When connected to an iSeries system, you need OS/400 V2R3 to be able to use data compression. All that is really required is to ensure that you have a MODE definition that specifies compression support. Three parameters are required on the "Create Mode Description" profile obtained through either the CRTMODD command or by going through the menus. The three parameters are:

DTACPR

This parameter controls the use of data compression. Valid parameters are shown below:

NETATR
Use the system-wide network value for this parameter.

NONE
No data compression allowed.

ALLOW
If the remote system requests data compression, the local system will allow the session to be established

REQUEST
The local system requests data compression.

REQUIRE
The use of data compression is required. If one of the systems is not able to perform data compression at the requested level, the session will not be established.

INDTACPR

This value represents the maximum level of compression used for inbound data.

OUTDTACPR

This value represents the maximum level of compression used for outbound data.

For LU 6.2 sessions, data compression levels 0, 1, and 2 are supported in both directions, that is, for both outbound and inbound data flow.

For information about enabling Communications Server for data compression, refer to *Quick Beginnings*.

Performance considerations

The performance of computer communications in general, and of Communications Server in particular, is subject to a large number of variables. The intent of the following information is to introduce some of the factors that influence the performance of communications in a Communications Server environment.

Communications performance is affected by the speed of the communications line. This speed is usually expressed as the number of bits per second (bps) that can be transmitted over the line. Typically, the higher the bps, the greater the communications performance that can be achieved on the line.

In addition to the bit rate, several other line-related factors can affect communications performance. For instance, some lines can only carry data in one direction at a time. To change the direction of communications (who can send), the station raises a *request to send* (RTS) and waits for *clear to send* (CTS) to be issued. This requires additional time for this *line turnaround* (which is modem dependent). Another factor is the transmission quality of the communications line.

When a line error occurs, most protocols will detect this and resend the data. A line that has fewer transmission errors will give greater throughput.

Greater throughput can also be achieved if the protocols operate in a *duplex* data transmission mode, where the protocol can receive and transmit simultaneously. LAN and X.25 are duplex protocols. SDLC can operate in a duplex or half-duplex mode.

To use duplex for SDLC, you should use an appropriate adapter (the MPA adapter cannot support duplex). The remote station that can also support duplex.

As line speed increases, the bit-rate capacity of the line becomes less of a constraint in performance. Instead, the constraint becomes related to delays inside the devices that are using the communication line (modems, control units, or computer internal constraints). At very high speeds, these delays can impact communication performance more than the bit rate capacity of the line.

Analyzing these situations takes some effort. For example, longer than expected communication delays can be caused by any of the following:

- An overloaded communication line
- An overloaded host system
- An overloaded 3X74 control unit
- An overloaded 37xx network controller
- Less than optimal VTAM or NCP definition
- Inadequate network design or network management
- Operating system memory overcommitment and swapping

In a high-speed communication environment, such as a LAN, the bit rate of the communication line is less significant because the communication capacity is largely determined by the speed of the communication devices. Better communication performance can often be achieved by increasing the processing power of the workstation, host, or other network components. This does not mean that the bit rate of a LAN network is never a constraint; rather, it might require several computers running at their communication capacity before the performance of the LAN significantly degrades.

To summarize, in many cases the performance of Communications Server is largely determined by factors that are related to the communication network being used. Providing communications performance at a reasonable cost involves both an understanding of the network as a whole and the analysis of its individual components.

Appendix B. Notices

This information was developed for products and services offered in the U.S.A. IBM may not offer the products, services, or features discussed in this document in other countries. 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:

IBM Corporation
TL3B/062
3039 Cornwallis Road
RTP, NC 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.

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.

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.

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

Trademarks

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

AIX
AS/400
Advanced Peer-to-Peer Networking

AnyNet
IBMNetView
OS/2
System/370
System/390
S/370
S/390
VTAM

The following product names are registered trademarks or trademarks of Tivoli Systems Inc. or IBM Corporation, in the United States, other countries, or both:

- Tivoli, Manage.
- TME
- Tivoli Ready
- Tivoli Enterprise

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Microsoft, Windows, Windows NT, Windows XP, Windows Server 2003, Windows Vista and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

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

Index

Numerics

A

B

C

D

E

F

H

I

L

M

N

O

P

R

S

T

W

X

Printed in USA

SC31-8656-03