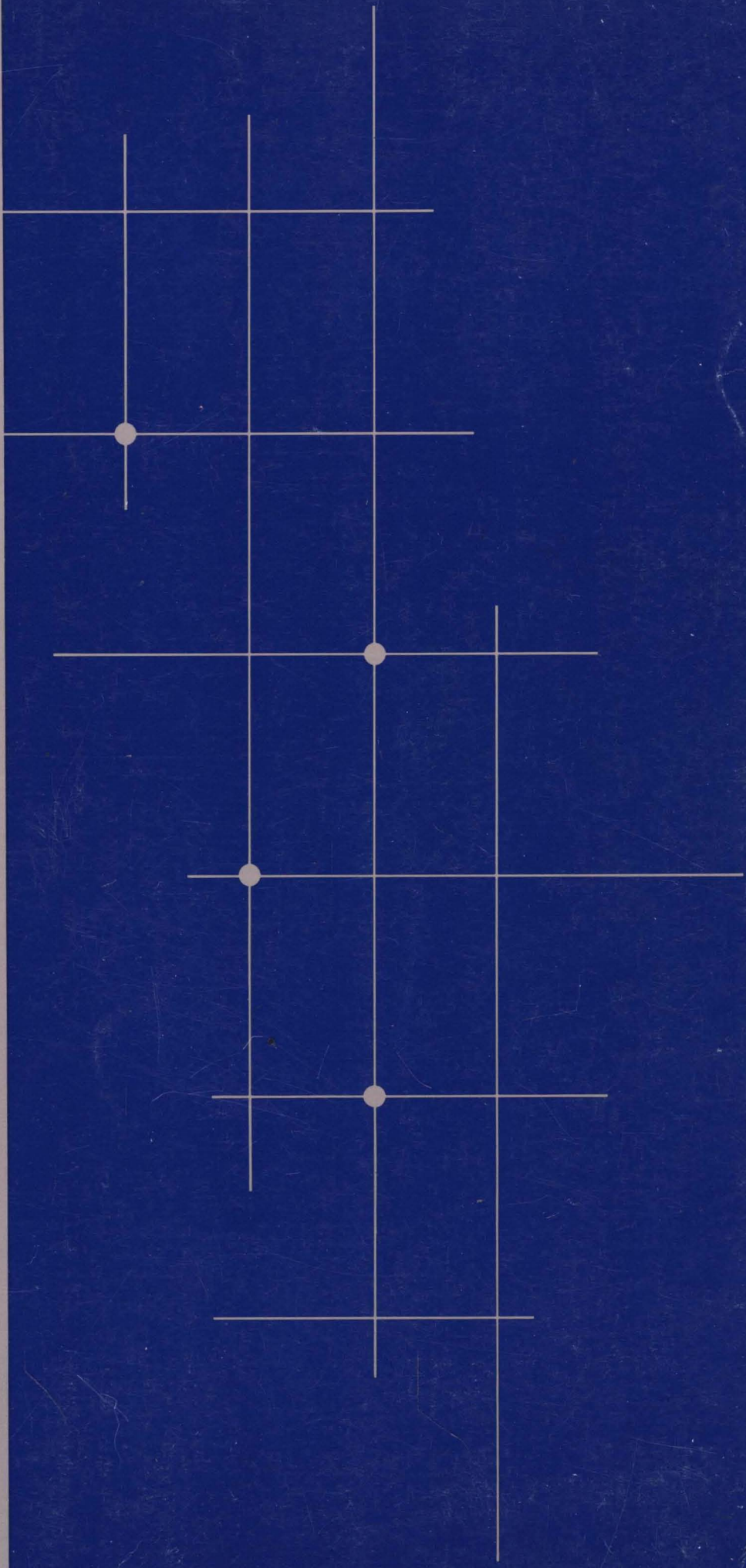


# Systems Network Architecture

Technical Overview



GC30-3073-1



# **Systems Network Architecture**

## **Technical Overview**

## **Second Edition (May 1985)**

This edition obsoletes GC30-3073-0 and replaces it in its entirety.

From time to time, changes are made to the information in IBM systems publications. Before using this publication in connection with the operation of IBM systems, consult your IBM representative to find out which editions are applicable and current.

It is possible that this material may contain reference to, or information about, IBM products (machines and programs) or services that are not announced in your country. Such references or information must not be construed to mean that IBM intends to announce such IBM products or services in your country.

Publications are not stocked at the address given below; requests for IBM publications should be made to your IBM representative or to the IBM branch office serving your locality.

A form for reader's comments is provided at the back of this publication. If the form has been removed, comments may be addressed to IBM Corporation, Information Development, Department E02, PO Box 12195, Research Triangle Park, North Carolina, U.S.A. 27709. IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation whatever. You may, of course, continue to use the information you supply.

## Preface

This book provides a technical overview of IBM's Systems Network Architecture (SNA). IBM hardware and software products implement the SNA functions that allow network users to be independent of the network's characteristics and operation. This book explains what those SNA functions are and how they provide for communication between network users.

### Who Should Read This Book

*SNA Technical Overview* is the basic SNA publication for system programmers and other data processing personnel who are responsible for defining SNA networks. System programmers should be familiar with the system generation process.

### How This Book Is Organized

Chapter 1 reviews and amplifies the SNA concepts and terms introduced in *SNA Concepts and Products*. Readers who do not need to review these SNA concepts and terms should proceed to Chapter 2.

Chapter 2 explains how system programmers define network resources to telecommunication access methods and network control programs during system definition.

Chapter 3 explains how control points activate and deactivate network resources, and how system programmers and other data processing personnel can control network activation and deactivation.

Chapter 4 explains how LU-LU sessions are initiated, activated, and deactivated, and how LU-LU sessions support communication between transaction programs.

Chapter 5 explains how route design affects the way the path control network routes data between network addressable units.

Chapter 6 explains how pacing controls the flow of data through a network.

Chapter 7 identifies the different message-unit formats that SNA defines.

Chapter 8 explains the transmission and BIND protocols that SNA defines.

Chapter 9 relates the seven architectural layers to the SNA functions they provide and compares SNA with Open Systems Interconnection (OSI).

Appendix A summarizes the different types of logical units.

Appendix B provides information on the different data streams that SNA-based products use.

Appendix C presents sequence charts that illustrate the command flows necessary to activate and deactivate network resources.

Appendix D identifies some of the ways system programmers define resources and specify routes to telecommunication access methods and network control programs.

*SNA Technical Overview* also contains a List of Abbreviations, a Glossary, and an Index.

## What Else to Read

No prerequisite reading is required for data processing personnel familiar with SNA concepts and terminology. If you are not familiar with SNA concepts and terminology, you should read *SNA Concepts and Products* before reading *SNA Technical Overview*.

Refer to the following publications for general, introductory information about SNA networks:

- *Systems Network Architecture Concepts and Products*, GC30-3072
- *Non-SNA Interconnection General Information Manual*, GC33-2023
- *IBM Synchronous Data Link Control General Information*, GA27-3093
- *X.25 SNA Guide*, GG24-1568
- *The X.25 Interface for Attaching IBM SNA Nodes to Packet-Switched Data Networks General Information Manual*, GA27-3345
- *Data Security Through Cryptography*, GC22-9062
- *Information Processing Systems - Open Systems Interconnection - Basic Reference Model*, ISO 7498
- *IBM 3270 Data Stream Programmer's Reference*, GA23-0059

Refer to the following publications for information about SNA Distribution Services (SNADS) and Document Interchange Architecture (DIA):

- *Office System Architecture Primer*, GG24-1659
- *Office Information Architectures: Concepts*, GC23-0765
- *Document Interchange Architecture: Concepts and Structures*, SC23-0759
- *Document Interchange Architecture: Technical Reference*, SC23-0781

- *Document Interchange Architecture: Transaction Programmer's Guide*, SC23-0763
- *Systems Network Architecture Format and Protocol Reference Manual: Distribution Services*, SC30-3098

Refer to the following publications for detailed information about SNA networks:

- *Systems Network Architecture Format and Protocol Reference Manual: Architectural Logic*, SC30-3112
- *Systems Network Architecture Format and Protocol Reference Manual: Architecture Logic for LU Type 6.2*, SC30-3269
- *Systems Network Architecture Reference Summary*, GA27-3136
- *Systems Network Architecture Transaction Programmer's Reference Manual for LU Type 6.2*, GC30-3084
- *Systems Network Architecture Sessions Between Logical Units*, GC20-1868
- *Installation Guide Scenarios for Release 3 Advanced Communications Function (ACF)*, G320-5839



# Contents

<b>Chapter 1. Introduction</b>	1
The Architecture	3
Network Components	4
Nodes	7
Host Subarea Nodes	7
Communication Controller Subarea Nodes	7
Peripheral Nodes	7
Subareas	9
Links	10
Data Channels	10
SDLC Links	10
Other Data Link Control Protocols	10
End Users	11
Network Addressable Units and the Path Control Network	11
Network Addressable Units	11
Logical Units	11
Physical Units	12
System Services Control Points	13
Path Control Network	16
Data Link Control	16
Path Control	18
Node Types	20
Summary	22
<b>Chapter 2. System Definition</b>	23
Defining Network Resources	25
Shared Control of Resources	25
Concurrent Sharing	26
Serial Sharing	28
Share Limit	28
Identifying Network Resources	30
Network Addresses	30
Subarea Addresses	31
Element Addresses	32
Extended Network Addressing	32
Local Addresses	33
Boundary Function	33
Network Names	34
Considerations for Interconnected Networks	36
Gateway Nodes	38
Gateway SSCPs	38
Interaction between Gateway Nodes and Gateway SSCPs	39
Network Identifiers	39
Summary	40



<b>Chapter 3. Network Activation</b>	41
Network Activation	43
Activating Sessions with Physical Units and Logical Units	43
Activating Links	44
Hierarchy of Network Activation	44
Controlling Network Activation	58
Cascaded Activation and Deactivation	58
Configuring and Reconfiguring a Network	58
Scheduled Changes	58
Unscheduled Changes	59
<b>Chapter 4. LU-LU Sessions</b>	61
Initiating LU-LU Sessions	63
Session-Initiation Requests	63
Unformatted Requests	64
Formatted Requests	64
Initiating Cross-Domain LU-LU Sessions	64
Initiating Cross-Network LU-LU Sessions	66
Activating LU-LU Sessions	68
BIND Requests	68
Nonnegotiable BIND	68
Negotiable BIND	68
Notifying the SSCP	68
Half-Sessions	69
Transaction Programs	69
Document Interchange Architecture	70
SNA Distribution Services	70
Verbs	70
Conversations	71
Invoking Transaction Programs	71
Sync Points	71
Terminating an LU-LU Session	72
Secondary LU Requests Session Termination	72
Primary LU Requests Session Termination	72
<b>Chapter 5. Route Design</b>	73
Designing Routes through the Network	75
Connecting Subarea Nodes	75
Parallel Links	75
Transmission Groups	76
Defining Paths	77
Explicit Routes	80
Routing Tables	82
Defining Virtual Routes	84
Transmission Priority	84
Assigning Session Traffic to a Virtual Route	84
Class of Service Tables	84
Selecting Virtual Routes	85
Activating and Deactivating Routes	86
Benefits of the SNA Routing Technique	87

<b>Chapter 6. Pacing</b>	89
Flow Control Algorithms	91
Global Algorithms	91
Local Algorithms	91
Pacing	91
Session-Level Pacing	92
One-Stage Pacing	93
Two-Stage Pacing	93
Virtual-Route Pacing	94
Summary	95
<b>Chapter 7. Data Formats</b>	97
Requests and Responses	99
Message Unit Formats	99
Basic Information Unit	100
Request Header	100
Request Unit	100
Response Header	100
Response Unit	101
Path Information Unit	101
Transmission Header Formats	102
Basic Link Unit	102
<b>Chapter 8. SNA Protocols</b>	107
Path Control Network Transmission Protocols	109
Sequencing	109
Blocking	110
Segmenting	111
BIND Protocols	112
Response Protocols	112
Definite response	112
Exception response	113
No response	113
Chaining Protocols	113
Definite Response Chain	114
Exception Response Chain	114
No-Response Chain	114
Bracket Protocols	114
Sequencing Protocols	115
Request and Response Mode Protocols	115
Immediate Request Mode	115
Delayed Request Mode	115
Immediate Response Mode	116
Delayed Response Mode	116
Send and Receive Mode Protocols	116
Full Duplex	116
Half-Duplex Contention	116
Half-Duplex Flip-Flop	117
Comparison of Send and Receive Mode Protocols and Transmission Medium Protocols	117
Data Security Protocols	117
Passwords and User IDs	117
Session Cryptography	118
LU-LU Session Passwords	118
Function Management Headers	119

<b>Chapter 9. Relationship of SNA Layers to Network Operation</b>	<b>121</b>
Architectural Definitions	123
Network Addressable Unit Functions	124
Path Control Network Functions	124
SNA Layers	125
Transaction Services Layer	126
Configuration Services	126
Session Services	126
Management Services	127
Presentation Services Layer	127
Data Flow Control Layer	127
Transmission Control Layer	128
Path Control Layer	128
Data Link Control Layer	128
Physical Control Layer	129
Peer-to-Peer Communication between SNA Layers	129
SNA Layer Management	131
Open Systems Interconnection	132
OSI Physical Layer	132
OSI Data Link Layer	132
OSI Network Layer	132
OSI Transport Layer	133
OSI Session Layer	133
OSI Presentation Layer	133
OSI Application Layer	133
Comparison of SNA and OSI	134
<b>Appendix A. Summary of LU Types and Representative IBM Products</b>	<b>137</b>
<b>Appendix B. Data Streams</b>	<b>139</b>
SNA Character String	139
SNA 3270 Data Stream	140
General Data Stream	140
Document Content Architecture	141
<b>Appendix C. Sequence Charts</b>	<b>143</b>
Typical Request Unit Sequences for Activating and Deactivating Network Resources	143
Typical Request Unit Sequences for Routing	151
Typical Request Unit Sequences for Activating Sessions, Deactivating Sessions, and Transferring Data	155
<b>Appendix D. Specifying Parameters for System Definition</b>	<b>175</b>
Specifying Activation, Deactivation, and Control Options	175
ACF/VTAM Options	175
Specifying Links and Associated Resources to SNA Products	176
Defining SDLC Links, Link Stations, and Transmission Groups	176
ACF/NCP Definition	176
ACF/VTAM Definition	176
Specifying Routes	176
Specifying Routes to ACF/NCP	177
Specifying Routes to ACF/VTAM	177

**List of Abbreviations** 181

**Glossary** 183

**Index** 195



## Figures

1. A Layered Architecture 3
2. Hardware and Software Components of an SNA Network 5
3. Another SNA Network Configuration 6
4. Subarea Nodes and Peripheral Nodes 8
5. Subareas 9
6. A Single-Domain Network 14
7. A Multiple-Domain Network 15
8. Link Stations 17
9. Path Control Elements 19
10. Node Types 20
11. Architectural Definition of Node Types 21
12. Concurrent Sharing of Network Resources 27
13. Serial Sharing of Network Resources 29
14. Format of a Network Address 30
15. Assigning Subarea Addresses 31
16. Constant Element Addresses 32
17. Boundary Function Components 34
18. Interconnected Network Configurations 37
19. Hierarchy of Network Activation: Part I 45
20. Hierarchy of Network Activation: Part II 46
21. Hierarchy of Network Activation: Part III 48
22. Hierarchy of Network Activation: Part IV 49
23. Hierarchy of Network Activation: Part V 51
24. Hierarchy of Network Activation: Part VI 52
25. Hierarchy of Network Activation: Part VII 53
26. Hierarchy of Network Activation: Part VIII 54
27. Activating a Host Subarea Node, a Channel-Attached Subarea Node, and a Channel between Two Subarea Nodes 56
28. Activating a Peripheral Node Attached to a Subarea Node by a Nonswitched SDLC Link 57
29. Initiating a Cross-Domain LU-LU Session 65
30. Initiating a Cross-Network LU-LU Session 67
31. A Conversation between Transaction Programs 71
32. Parallel Links between Adjacent Communication Controller Nodes 75
33. Transmission Groups 76
34. A Path between Logical Units 77
35. An Explicit Route and a Peripheral Link 79
36. Multiple Explicit Routes 81
37. Routing Table Segments for Two Explicit Routes 82
38. Session-Level Pacing 93
39. Basic Information Unit (BIU) Format 100
40. Path Information Unit (PIU) Format 101
41. Basic Link Unit (BLU) Format 102
42. Use of Data Formats 103

43.	Blocking of Path Information Units	110
44.	Segmenting of Path Information Units	111
45.	Relating SNA Layers to SNA Functions	123
46.	Definition of a Half-Session	124
47.	The Seven SNA Layers	125
48.	Communication between SNA Layers	129
49.	Communication between Two Transmission Control Layers	130
50.	Symbols and Abbreviations for Figures 51 through 56	143
51.	Activating a Host Node, a Channel-Attached Subarea Node, and the Channel between Them	144
52.	Activating Explicit and Virtual Routes between Adjacent Subarea Nodes	145
53.	Deactivating Virtual Routes, Explicit Routes, and SDLC Links	146
54.	Deactivating a Peripheral Node Attached by a Nonswitched SDLC Link	148
55.	Deactivating a Peripheral Node Attached by a Switched SDLC Link	149
56.	Deactivating a Channel-Attached Subarea Node and Associated Resources	150
57.	Symbols and Abbreviations for Figures 58 and 59	151
58.	Propagation of Explicit Route Operative (NC-ER-OP) Requests	152
59.	Propagation of Routing Information Following Activation of Multiple Transmission Groups between the Same Subareas	154
60.	Symbols and Abbreviations for Figures 61 through 72	155
61.	Activating an SSCP-SSCP Session	156
62.	Activating a Same-Domain LU-LU Session	157
63.	Activating a Cross-Domain LU-LU Session	159
64.	Activating a Cross-Network LU-LU Session	160
65.	Deactivating a Same-Domain LU-LU Session	163
66.	Deactivating a Cross-Domain LU-LU Session	164
67.	Cross-Domain Takedown Sequence	167
68.	Communication Using Brackets in a Half-Duplex Flip-Flop Mode	168
69.	Communication Using Half-Duplex Contention Protocols	170
70.	LU-LU Communication Using Half-Duplex Flip-Flop Protocols	171
71.	Protocols for Quiescing Data Flow	172
72.	Protocols for Deactivating LU-LU Sessions	173

## Summary of Amendments

### Changes for GC30-3073-1

This edition of *SNA Technical Overview* includes the following enhancements to SNA:

- Type 6.2 logical units

Type 6.2 logical units provide for program-to-program communication in an SNA network.

- Type 2.1 nodes

Type 2.1 nodes provide peripheral node support for type 6.2 logical units. SNA supports direct connection between type 2.1 nodes.

- Extended network addressing

Extended network addressing provides additional network addresses for users that have larger networks.

- SNA network interconnection

SNA network interconnection allows end users in different networks to communicate with each other while allowing you to configure, define, and manage each network independently.

This edition also includes Document Interchange Architecture (DIA) and SNA Distribution Services (SNADS), two of IBM's office information architectures. These architectures use type 6.2 logical units to communicate with one another.

Finally, this edition reflects changes and corrections to the architecture. These changes include the redefinition of the upper layer of the architecture (the transaction services layer) and the definition of the lower layer. The lower layer, the physical control layer, provides the physical interface between adjacent nodes.





## Introduction

This chapter reviews SNA concepts and terminology, thereby establishing a common level of knowledge for readers of this book.

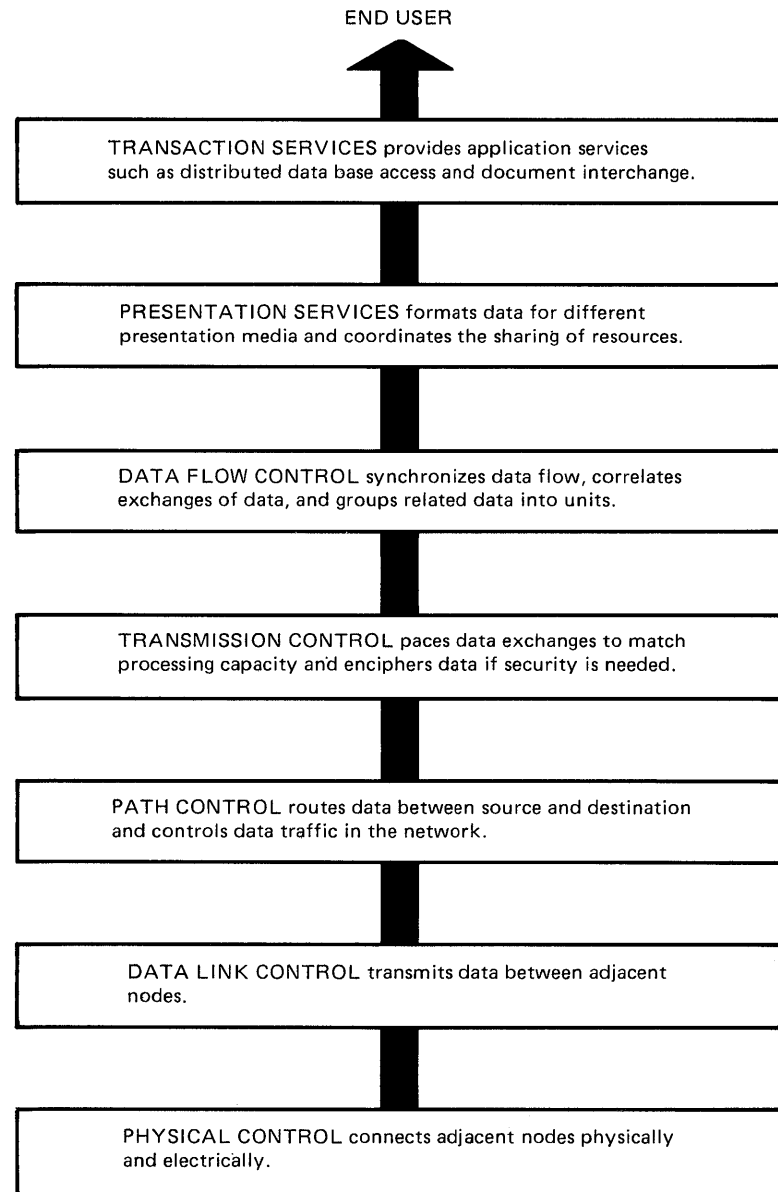
## Contents

The Architecture	3
Network Components	4
Nodes	7
Host Subarea Nodes	7
Communication Controller Subarea Nodes	7
Peripheral Nodes	7
Subareas	9
Links	10
Data Channels	10
SDLC Links	10
Other Data Link Control Protocols	10
End Users	11
Network Addressable Units and the Path Control Network	11
Network Addressable Units	11
Logical Units	11
Physical Units	12
System Services Control Points	13
Path Control Network	16
Data Link Control	16
Path Control	18
Node Types	20
Summary	22



# The Architecture

IBM's Systems Network Architecture (SNA) provides for communication between a diverse group of IBM products. SNA is a hierarchical structure that consists of seven well-defined layers. Each layer in the architecture performs a specific function. Figure 1 identifies SNA's seven layers and their functions.



**Figure 1. A Layered Architecture**

Each layer performs services for the next higher layer, requests services from the next lower layer, and communicates with corresponding layers in other SNA-based products. For example, the physical control layer:

- Manages the physical interface between its node and the transmission facilities that are attached to the node
- Performs services for the data link control layer
- Communicates with physical control layers in other SNA-based products.

As another example, the transaction services layer:

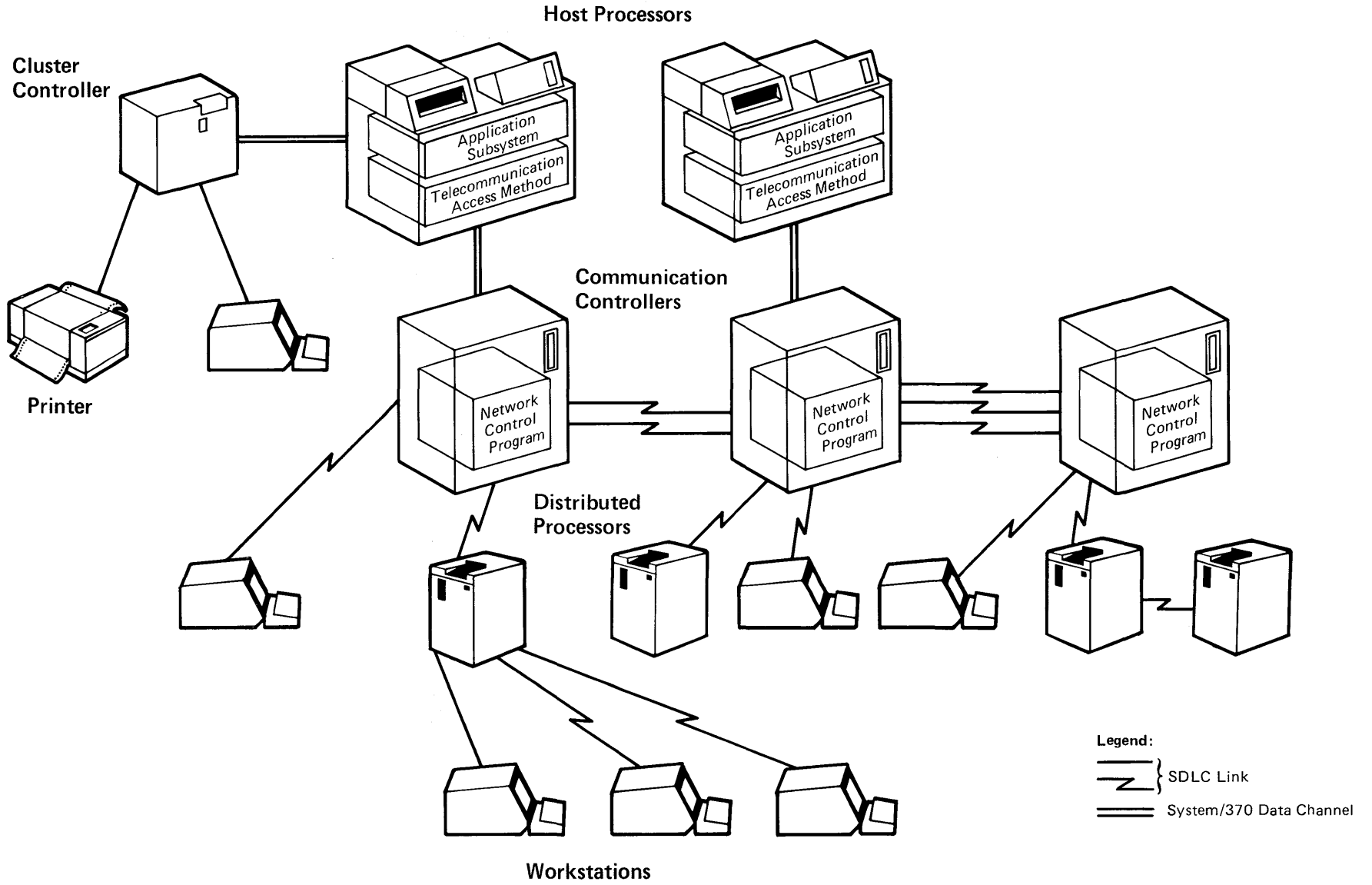
- Provides the end user access to the network
- Requests services from the presentation services layer
- Communicates with transaction services layers in other SNA-based products.

“Chapter 9. Relationship of SNA Layers to Network Operation” presents an in-depth discussion that relates network services to the specific layers that implement them.

## Network Components

Hardware and software components implement the functions of the seven architectural layers. Hardware components include processors, communication controllers, cluster controllers, workstations, and printers. The software components that implement SNA functions include telecommunication access methods, application subsystems, and network control programs. Figure 2 and Figure 3 illustrate two possible network configurations of these hardware and software components.

Figure 2. Hardware and Software Components of an SNA Network



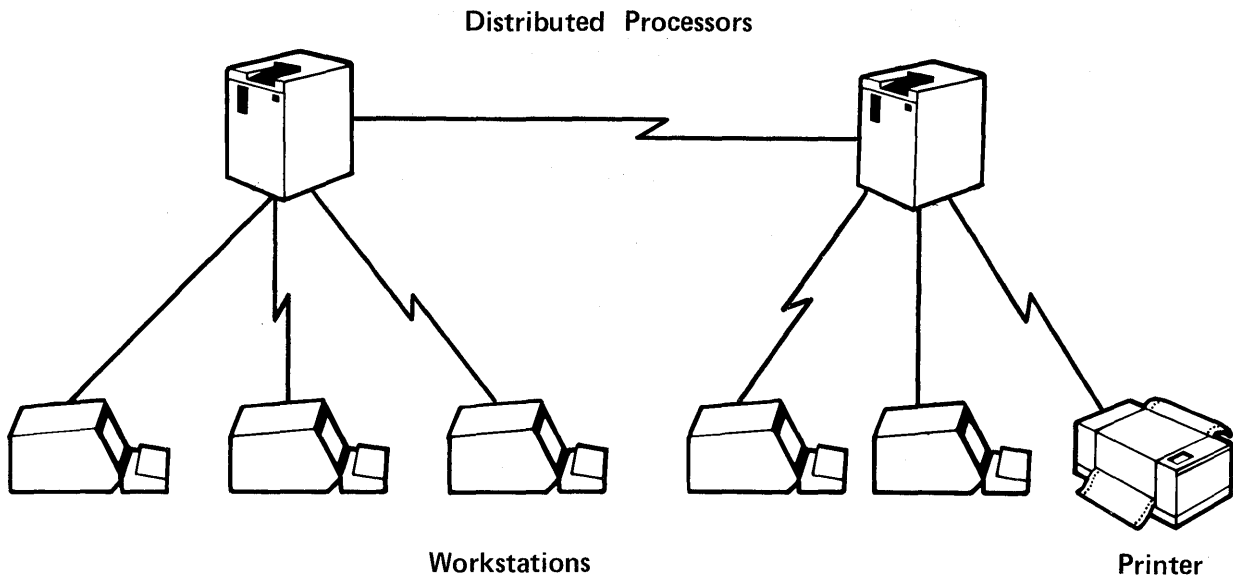


Figure 3. Another SNA Network Configuration

## Nodes

A network consists of many different hardware and software components. SNA defines a **node** as the portion of a hardware component, along with its associated software components, that implement the functions of the seven architectural layers. SNA currently defines three kinds of nodes: host subarea nodes, communication controller subarea nodes, and peripheral nodes.

### Host Subarea Nodes

A processor that contains a telecommunication access method (for example, ACF/VTAM) is a **host subarea node**. Host subarea nodes provide the SNA functions that control and manage a network.

### Communication Controller Subarea Nodes

A communication controller that contains a network control program (for example, ACF/NCP) is a **communication controller subarea node**. Communication controller subarea nodes provide the SNA functions that route and control the flow of data in a network.

### Peripheral Nodes

All other nodes are **peripheral nodes**. Peripheral nodes encompass many devices, such as cluster controllers, distributed processors, workstations, and printers.

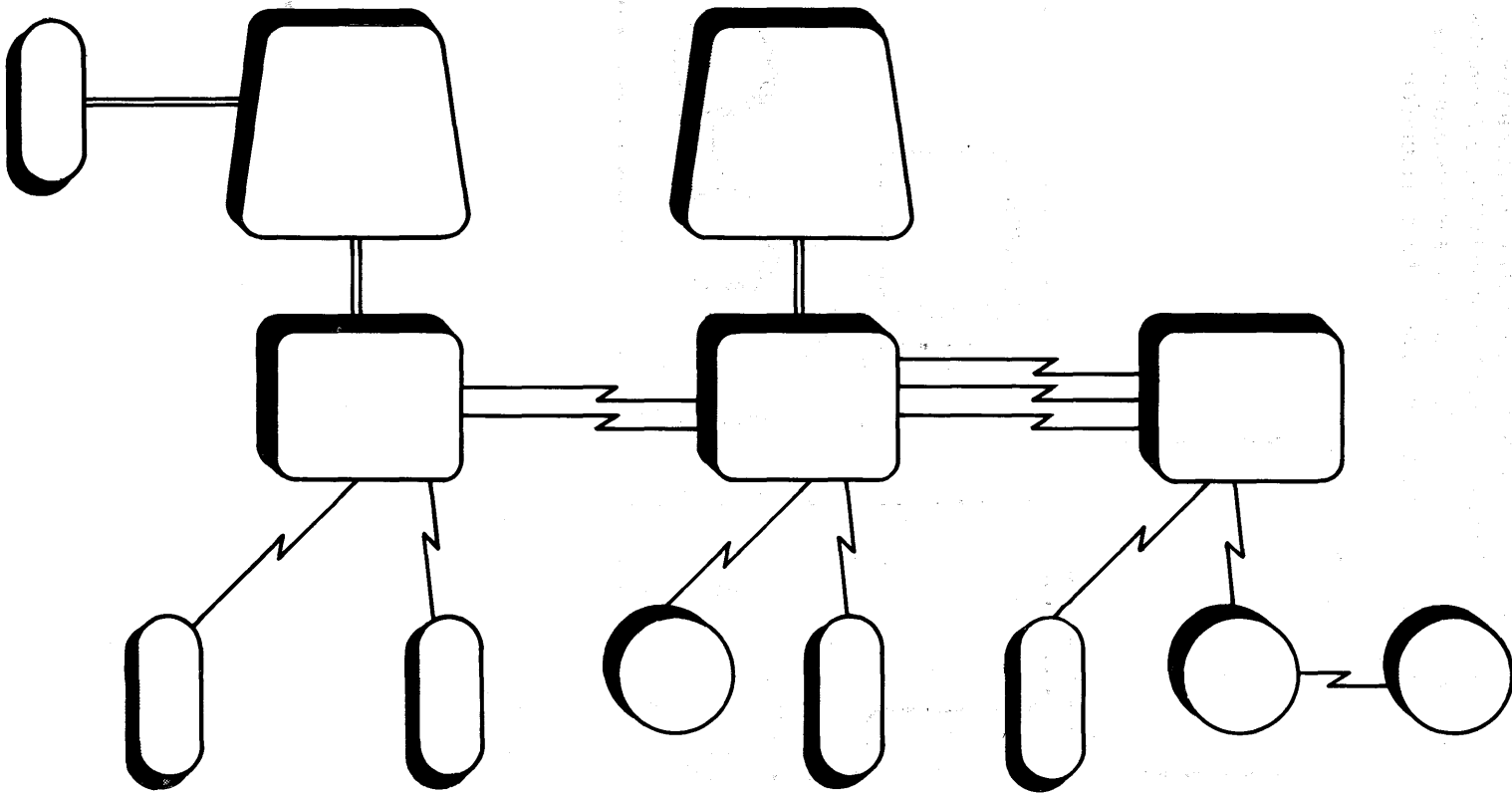
The network configuration that was shown in Figure 2 contains:

- Two host subarea nodes
- Three communication controller subarea nodes
- Eight peripheral nodes (one cluster controller, four distributed processors, and three workstations).

Figure 4 uses symbols to represent these nodes. Note that Figure 4 uses two different symbols to represent peripheral nodes. The architecture classifies peripheral nodes based on their capability for direct connection to one another (as Figure 3 illustrates) and the end-user services that they provide. The differences between the two types of peripheral nodes are discussed later in this chapter.



Figure 4. Subarea Nodes and Peripheral Nodes



Legend:



Host Subarea Node



Communication  
Controller  
Subarea Node



Peripheral Nodes



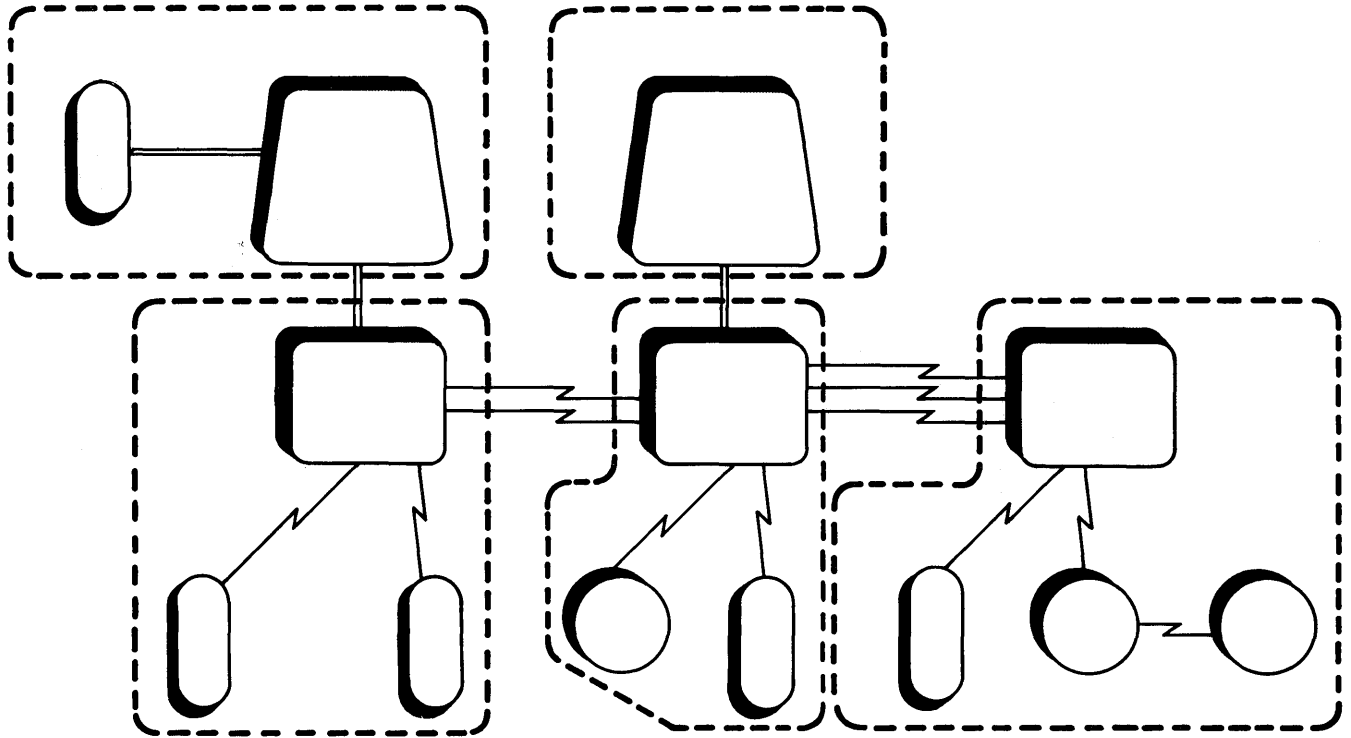
SDLC Link



S/370 Data  
Channel

## Subareas

A **subarea** consists of one subarea node and the peripheral nodes that are attached to that subarea node. The network configuration in Figure 5 contains five subarea nodes and eight peripheral nodes. Because each subarea node and its attached peripheral nodes constitute a subarea, this configuration contains five subareas.



Legend:



Host Subarea Node



Communication  
Controller  
Subarea Node



Peripheral Nodes



SDLC Link



S/370 Data  
Channel

Figure 5. Subareas

## Links

Adjacent nodes are connected to one another by one or more **links**. A link consists of a **link connection** and two or more **link stations**. A link connection physically connects two nodes; it is the physical medium of transmission, such as a telephone wire or a microwave beam. Link stations use data link control protocols to transmit data over a link connection. SNA networks support two types of data link control protocols: System/370 data channels and Synchronous Data Link Control (SDLC) links.

## Data Channels

A **data channel** transmits bits in parallel. Data channels connect host subarea nodes to other subarea nodes and to locally-attached peripheral nodes. Because of the nature of their physical connection, channel-attached nodes must be located fairly close to one another.

## SDLC Links

An **SDLC link** transmits data serially by bit. SDLC is a data link control protocol, independent of the physical medium that connects adjacent nodes. Telephone lines, microwave links, and other voice-quality communication media can use SDLC protocols. For additional SDLC information, refer to the *IBM SDLC General Information* manual.

## Other Data Link Control Protocols

SNA networks also support X.25 interface, binary synchronous communication (BSC), and start/stop data link protocols.

**X.25 interface**, an international standard recommended by the International Telegraph and Telephone Consultative Committee (CCITT), allows SNA networks to transmit data across packet-switched networks. SNA networks also support direct X.25 interface protocols between adjacent subarea nodes. For additional X.25 interface information, refer to the *X.25 SNA Guide* and *The X.25 Interface for Attaching IBM SNA Nodes to Packet-Switched Data Networks General Information Manual*.

**Binary synchronous communication** and **start/stop** data link control protocols allow SNA networks to transmit data to and receive data from non-SNA workstations. For additional information, refer to the *Non-SNA Interconnection General Information Manual*.

## End Users

For the purposes of discussing the architecture, this publication uses the term **end user** to identify both (1) individuals who interact with the network through a workstation and (2) application programs. The architecture defines end users as the ultimate sources and destinations of information that flows through a network. End users interact with the network to obtain services that the network provides—primarily, the efficient exchange of data between two points.

## Network Addressable Units and the Path Control Network

Nodes provide SNA functions that enable end users to be independent of a network's characteristics and operation. The network resources in the nodes that provide these functions fall into two categories: network addressable units and the path control network.

### Network Addressable Units

**Network addressable units (NAUs)** enable end users to send data through a network and help network operators perform network control and management functions. Network addressable units provide functions to:

- Synchronize communication between end users
- Manage the resources in each node
- Control and manage the network.

Each NAU has an address that identifies it to other NAUs and to the path control network. The path control network uses this address to route data between NAUs. SNA defines three kinds of network addressable units: logical units, physical units, and system services control points.

### Logical Units

Every end user gains access to an SNA network through a **logical unit (LU)**. Logical units manage the exchange of data between end users, acting as an intermediary between the end user and the network. There is not a one-to-one relationship between end users and LUs. The number of end users that can gain access to a network through the same LU is an implementation design option.

**LU-LU Sessions:** Before end users can communicate with one another, their respective LUs must be connected in a mutual relationship called a **session**. Because the session connects two LUs, it is called an **LU-LU session**. Multiple, concurrent sessions between the same two logical units are called parallel LU-LU sessions.

**LU Types:** The architecture defines different kinds of logical units, called **LU types**. LU types identify the particular set of SNA functions that a product provides to support end-user communication. Since SNA was announced, IBM has developed a number of LU types to handle the communication requirements of a variety of end users.

The architecture defines LU types 1, 2, 3, 4, 6.1, 6.2, and 7. (Type 5 does not exist.) LU types 2, 3, 4, and 7 support communication between application programs and different kinds of workstations. LU types 1, 4, 6.1, and 6.2 support communication between two programs.

LU type 6.2 is the most recent LU type. IBM developed LU type 6.2 for SNA program-to-program communication. You can write application programs, or use IBM-developed application subsystems, that enable an LU type 6.2 to perform the same functions that the previous LU types provided.

LU-LU sessions can exist only between logical units of the same LU type. For example, an LU type 2 can communicate only with another LU type 2; it cannot communicate with an LU type 3. A host subarea node normally supports multiple LU types in order to enable communication with any other logical unit in the network.

Appendix A details the different LU types and identifies the kind of network applications that each type supports. Appendix A also provides representative examples of IBM products that typically use each LU type.

## **Physical Units**

Every node contains one **physical unit (PU)** to manage the links that connect the node to adjacent nodes. The PU represents the processor, controller, workstation, or printer to the network. Physical units, like other NAUs, are implemented by a combination of software and hardware components within each node.

## System Services Control Points

**System services control points (SSCPs)** activate, control and deactivate network resources. Only host subarea nodes contain an SSCP.

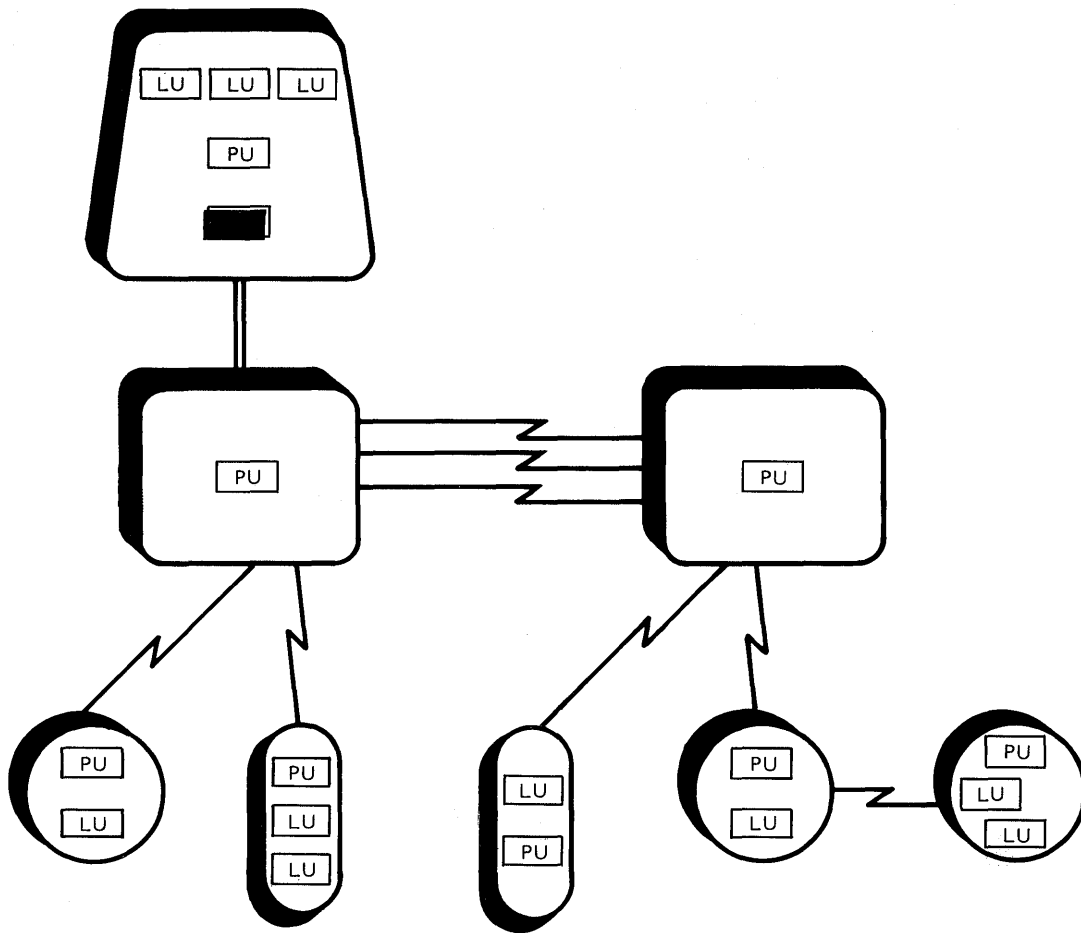
An SSCP has various functions in a network:

- It manages network resources in accordance with the commands that network operators issue.
- It coordinates the activation of sessions between logical units. For example, an SSCP provides directory services to assist a logical unit initiate an LU-LU session.
- When necessary, it acts on the physical network to activate sessions. For example, an SSCP can cause a workstation to be dialed over a switched link connection when that is necessary to activate a session.

Each SSCP manages a portion of the network; that portion is called a **domain**. When there is only one SSCP in a network, that SSCP must manage all the network resources. A network that contains only one SSCP is a **single-domain network**. Figure 6 illustrates a single-domain network.

When there are multiple SSCP's in the network, each SSCP controls a portion of the network resources. A network that contains more than one SSCP is a **multiple-domain network**. Figure 7 illustrates a multiple-domain network.

Most network configurations contain more than one host subarea node; therefore they are multiple-domain networks. To simplify the illustrations, most of the subsequent figures show single-domain networks. Multiple-domain networks are shown when they are necessary to explain the concept being introduced.



**Legend:**

PU = Physical Unit

LU = Logical Unit

SSCP = System Services Control Point

**Figure 6. A Single-Domain Network**

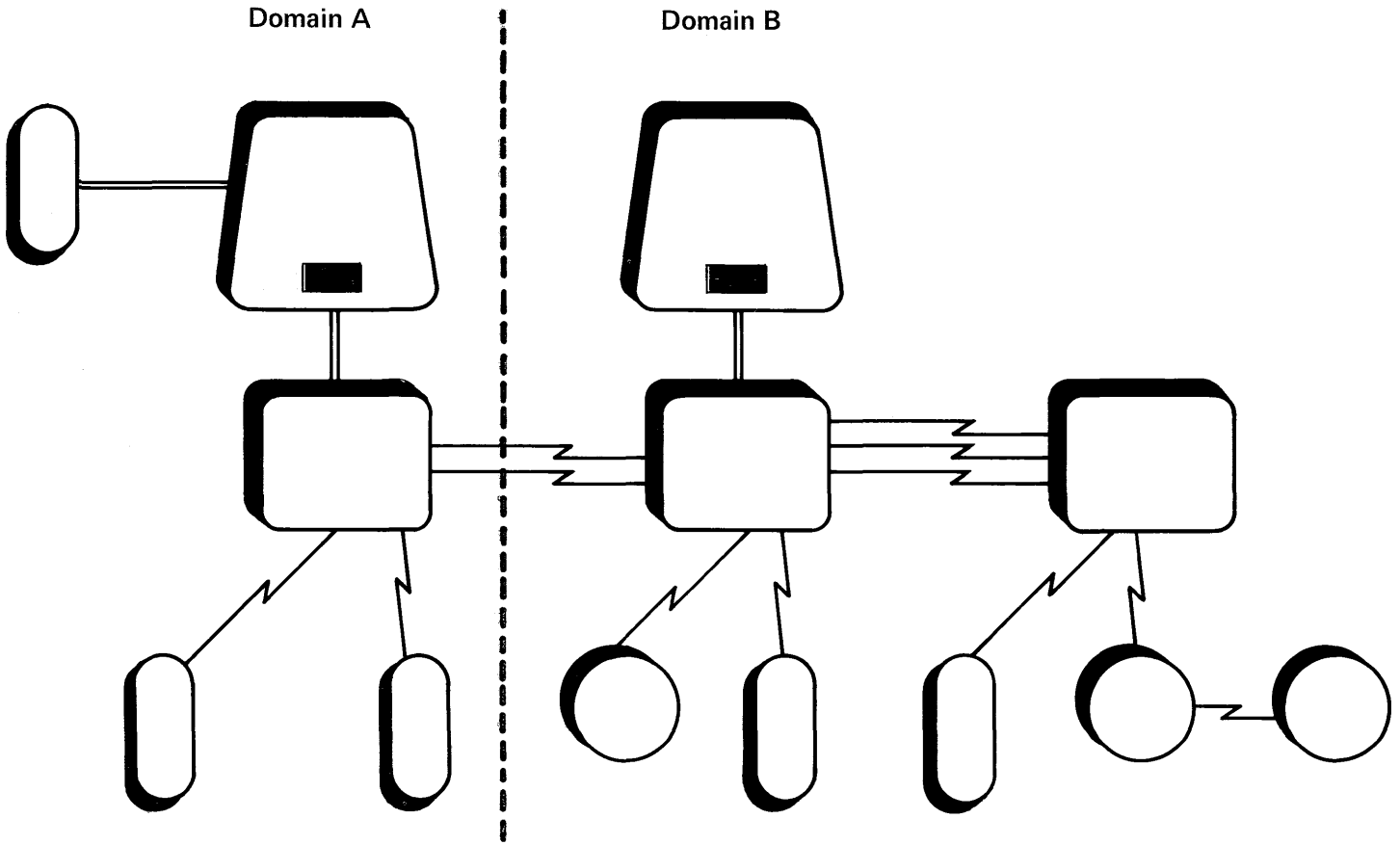


Figure 7. A Multiple-Domain Network

Legend:  
SSCP = System Services Control Point



## Path Control Network

The **path control network** routes and transmits data between network addressable units. The path control network provides functions to:

- Transmit data across links between adjacent nodes
- Route data between subarea nodes
- Route data between a subarea node and an adjacent peripheral node.

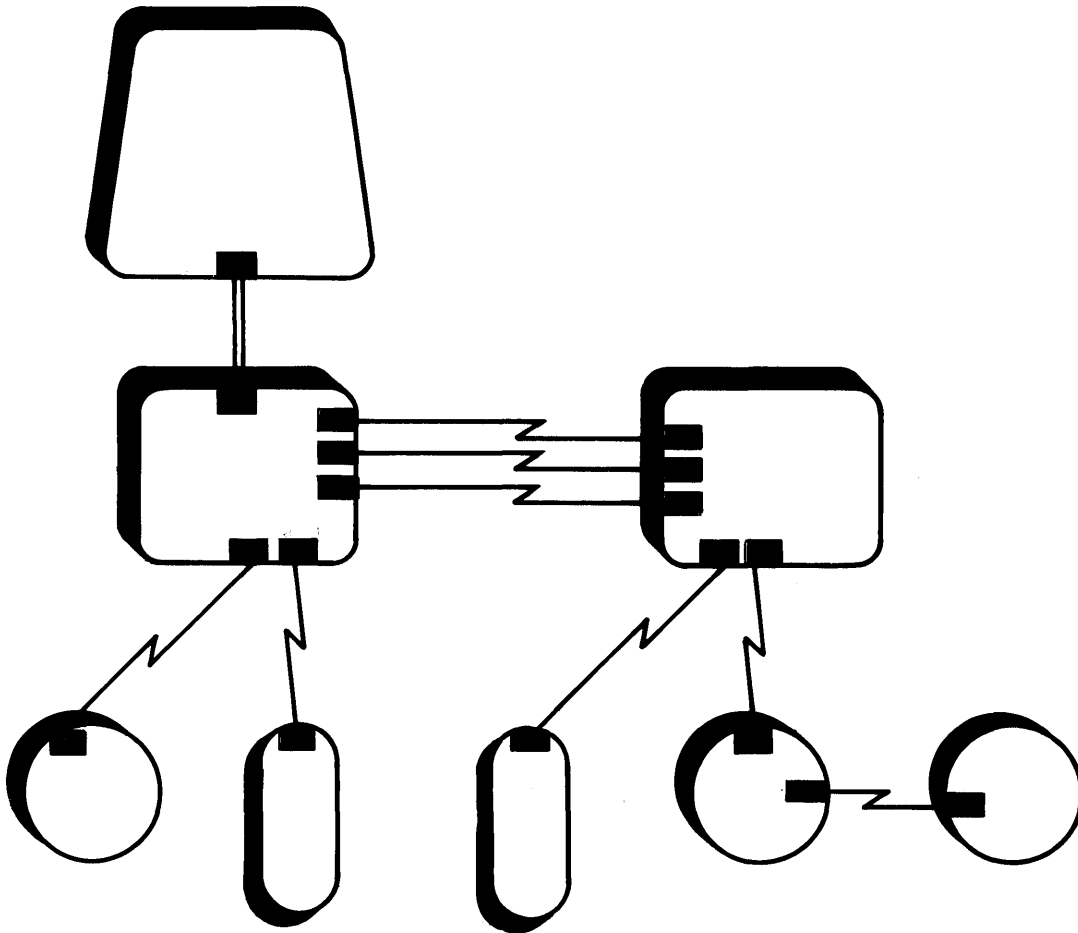
The three elements of the path control network that provide these SNA functions are data link control, subarea path control, and peripheral path control.

## Data Link Control

**Data link control** elements transmit data across links between adjacent nodes. SNA identifies data link control elements as **link stations**. SSCPs activate, control, and deactivate links through the link stations in each node.

Recall that a link consists of a link connection, which physically connects two nodes, and two or more link stations. Link stations control a link by using data link control protocols to transmit data over the link connection.

Every node contains one link station for each link that is attached to that node. Figure 8 highlights the link stations in each node.



**Legend:**  
LS = Link Station

**Figure 8. Link Stations**

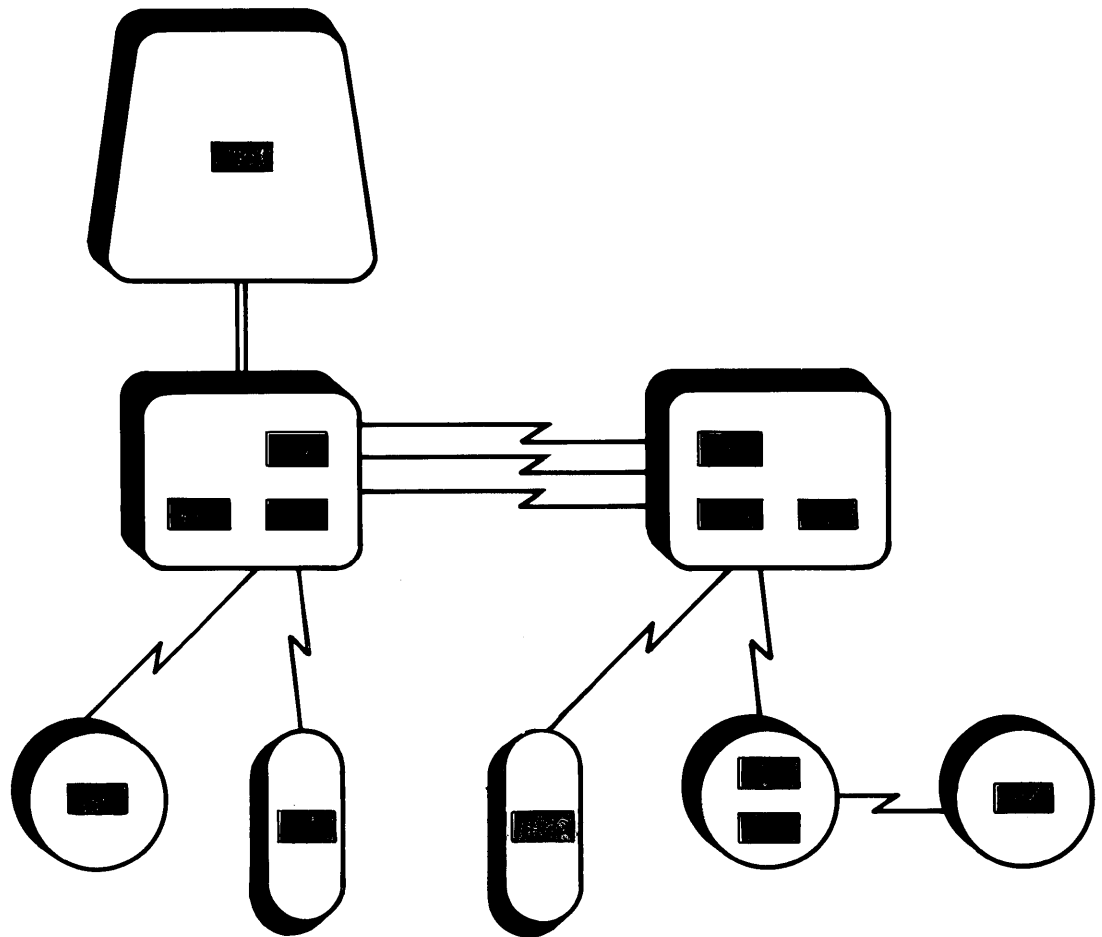
## Path Control

**Path control** elements route message units between network addressable units (NAUs). Message units, a generic term for the data that flows between network resources, contain the address of the destination NAU. Path control elements use this address to select a path for the data through the network. There are two kinds of path control elements: subarea path control and peripheral path control.

**Subarea Path Control:** **Subarea path control** elements route data between (1) subareas and (2) NAUs in subarea nodes. Subarea path control elements use network addresses to route message units through a network. Every subarea node contains one subarea path control element.

**Peripheral Path Control:** **Peripheral path control** elements route data between (1) subarea nodes and peripheral nodes and (2) NAUs in peripheral nodes. Peripheral path control elements use local addresses to route message units. Whereas a network address is unique within a network, a local address is unique only within a peripheral node.

Subarea nodes contain one peripheral path control element for each adjacent peripheral node. Peripheral nodes contain one peripheral path control element to route data to and from their subarea node, and an additional peripheral path control element for each directly-connected peripheral node. Figure 9 highlights the path control elements in a network.



Legend:

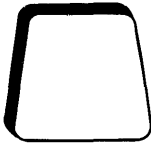



PC = Subarea Path Control Element

pc = Peripheral Path Control Element

**Figure 9. Path Control Elements**

# Node Types

Recall that a node is the portion of a hardware product, and its associated software, that supports SNA functions. Because different SNA-based products support different SNA functions, the architecture assigns **node types** to identify the functions that each product supports. The chart in Figure 10 shows the four node types that the architecture currently defines.<sup>1</sup> The architecture distinguishes these four node types by their different routing capabilities, connection capabilities, and the presence or absence of different NAU types.

NODE TYPE	DESCRIPTION	GRAPHIC REPRESENTATION USED IN THIS PUBLICATION
5	Host Subarea Node	
4	Communication Controller Subarea Node	
2.1	Peripheral Node	
2.0	Peripheral Node	

**Figure 10. Node Types**

---

<sup>1</sup> This chart does not include node type 1. Although the architecture continues to support this node type, there will be no further node type 1 implementations. Node type 1 was a peripheral node that supported only LU type 1.

Figure 11 summarizes the architectural differences between node types. As the chart shows, only type 5 nodes contain an SSCP. This is the primary architectural distinction between type 5 and type 4 nodes. Another difference between type 5 and type 4 nodes is that type 4 nodes typically do not contain logical units.

The architecture distinguishes between type 2.1 and type 2.0 nodes by (1) the different LU types that each node supports and (2) the ability of NAUs in a type 2.1 node to communicate directly with NAUs in an adjacent type 2.1 node. Type 2.0 nodes cannot be directly connected to one another; they can communicate directly only with their adjacent subarea node.

NODE TYPE	ARCHITECTURAL DESCRIPTION	PRIMARY FUNCTIONS
5	<ul style="list-style-type: none"> <li>● Subarea node</li> <li>● Contains an SSCP</li> <li>● Contains a PU type 5</li> <li>● Supports LU types 1, 2, 3, 4, 6.1, 6.2, and 7.</li> </ul>	<ul style="list-style-type: none"> <li>● Control network resources</li> <li>● Support application and transaction programs</li> <li>● Provide network operators access to the network</li> <li>● Support end-user services.</li> </ul>
4	<ul style="list-style-type: none"> <li>● Subarea node</li> <li>● Contains a PU type 4</li> </ul>	<ul style="list-style-type: none"> <li>● Route and control the flow of data through the network.</li> </ul>
2.1	<ul style="list-style-type: none"> <li>● Peripheral node</li> <li>● Contains a PU type 2.1</li> <li>● Supports LU type 6.2 in addition to LU types 1, 2, 3, and (for migration purposes only) 7</li> <li>● Supports direct link connections to other type 2.1 nodes.</li> </ul>	<ul style="list-style-type: none"> <li>● Provide end users access to the network</li> <li>● Provide end-user services.</li> </ul>
2.0	<ul style="list-style-type: none"> <li>● Peripheral node</li> <li>● Contains a PU type 2.0</li> <li>● Supports LU types 2, 3, and 7</li> <li>● Supports LU type 1 for non-SNA interconnect.</li> </ul>	<ul style="list-style-type: none"> <li>● Provide end users access to the network</li> <li>● Provide end-user services.</li> </ul>

**Figure 11. Architectural Definition of Node Types**

As Figure 11 shows, physical units carry the same numeric designation as the node in which they reside. Thus, the designation *PU type 5* identifies the physical unit in a type 5 node. The architecture allows multiple node types to exist within an SNA-based product; each node type contains a physical unit with the same numeric designation.

## Summary

Nodes are hardware and software components that implement SNA functions. The network resources that provide these functions fall into two categories: network addressable units (NAUs) and the path control network. System services control points (SSCPs), physical units (PUs), and logical units (LUs) are network addressable units. They enable end users to send data through a network and help network operators perform network control and management functions. Link stations (data link control elements), subarea path control elements, and peripheral path control elements are part of the path control network. They route and transmit data between network addressable units.

System services control points manage the configuration of a network, control other network resources, coordinate network operator and problem determination requests, and provide directory support and other session services for end users. Physical units manage and monitor each node's attached links and adjacent link stations. Logical units, acting as an intermediary between the end user and the network, manage the exchange of data between end users. Before end users can communicate with one another, their respective logical units must be connected in a mutual relationship called an LU-LU session.

Link stations are data link control elements that transmit data over the link connections between adjacent nodes. Path control elements route data between network addressable units. Subarea path control elements route data to and from NAUs in subarea nodes and between subareas. Peripheral path control elements route data to and from NAUs in peripheral nodes.

The architecture defines four node types to identify the SNA functions that each node supports: 5, 4, 2.1, and 2.0. Different node types vary in their routing and connection capabilities, and support different kinds of NAUs. Type 5 and type 4 nodes are subarea nodes; type 2.1 and type 2.0 nodes are peripheral nodes.

### System Definition

This chapter explains how parameters that you specify during the system definition process:

- Determine the domain of control for each SSCP
- Identify network resources to the path control network
- Enable network operators and end users to identify specific network resources.

### Contents

Defining Network Resources	25
Shared Control of Resources	25
Concurrent Sharing	26
Serial Sharing	28
Share Limit	28
Identifying Network Resources	30
Network Addresses	30
Subarea Addresses	31
Element Addresses	32
Extended Network Addressing	32
Local Addresses	33
Boundary Function	33
Network Names	34
Considerations for Interconnected Networks	36
Gateway Nodes	38
Gateway SSCPs	38
Interaction between Gateway Nodes and Gateway SSCPs	39
Network Identifiers	39
Summary	40





## Defining Network Resources

You must define network resources to access methods, network control programs, and other software before they can control or communicate with these resources. You define this information to the network's software through system generations.

System definition includes both (1) identifying the domains in a network and (2) defining the resources in each domain to a control point. A domain consists of one system services control point (SSCP) and the network resources that the SSCP can control. The network resources that an SSCP can control are logical units (LUs), physical units (PUs), links, and link stations.

You define the domain of control for each SSCP in access-method resource-definition statements and network-control-program macro instructions. An SSCP activates the network resources in its domain, thereby gaining control of those resources.

In single-domain networks, there is only one domain to identify because there is only one SSCP. You define all the network resources to that one SSCP. The SSCP then activates those resources in response to network-operator commands or access-method resource-definition statements.

In multiple-domain networks, the resources you define to each SSCP determine the domains in the network. As with single-domain networks, you define each network resource to an SSCP. Each SSCP in the network may control an equal number of resources, one SSCP may control the majority of network resources, or more than one SSCP may share control of some resources.

### Shared Control of Resources

To provide flexibility for both normal operations and backup recovery procedures, you can define some network resources to more than one SSCP. The architecture permits SSCPs to share some network resources concurrently, some serially, and some not at all.

Shared control of network resources allows you to:

- Back up one SSCP by another to increase network availability
- Partition control of a network by use rather than by the physical location of resources
- Shift control of network resources to different SSCPs at various times of the day in response to changing traffic loads.

## Concurrent Sharing

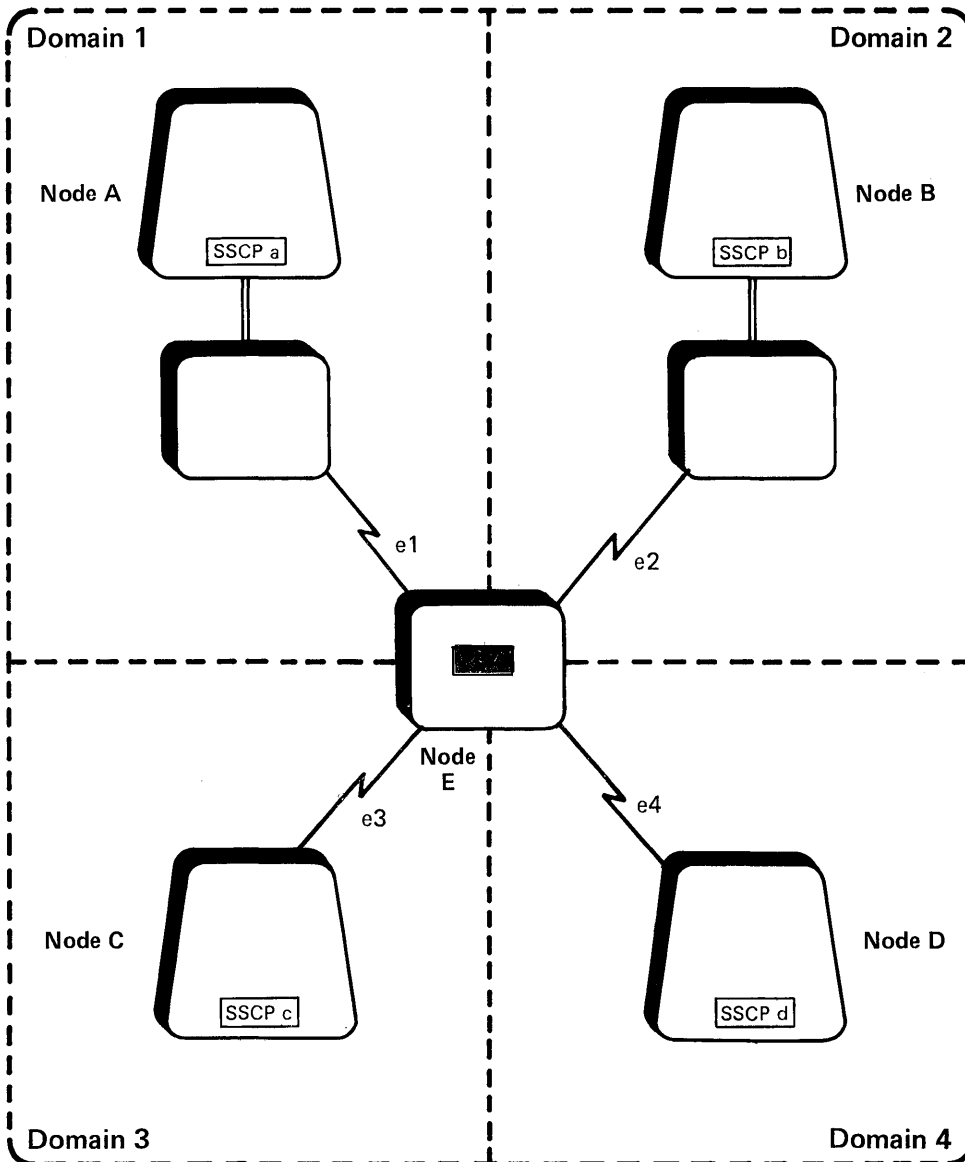
Concurrent sharing means that more than one SSCP can simultaneously control the same resource. Multiple SSCPs can concurrently share control of:

- Physical units in type 4 nodes
- Nonswitched SDLC links between subarea nodes
- Link stations for the nonswitched SDLC links between subarea nodes.

You can define these resources to more than eight SSCPs, but in actual implementation, only the first eight SSCPs that activate the resource can share its control. If one of the SSCPs fails or deactivates the resource, the other SSCPs in the network are notified. This notification serves as a signal for the other SSCPs to activate, and thus establish control of, the resource.

For example, Figure 12 shows a configuration in which four SSCPs share a physical unit in a type 4 node. The type 4 node, node *E*, belongs to all four domains.

If you define PU *e* to SSCPs *a*, *b*, *c*, and *d*, all four SSCPs can activate, and thus control, PU *e*. Any SSCPs that share control of PU *e* can also share control of the nonswitched SDLC links *e1*, *e2*, *e3*, and *e4*.





Legend:

PU = Physical Unit

SSCP = System Services Control Point

 = SDLC Link

 = S/370 Data Channel

 = Type 5 Node


 = Type 4 Node

Figure 12. Concurrent Sharing of Network Resources

## Serial Sharing

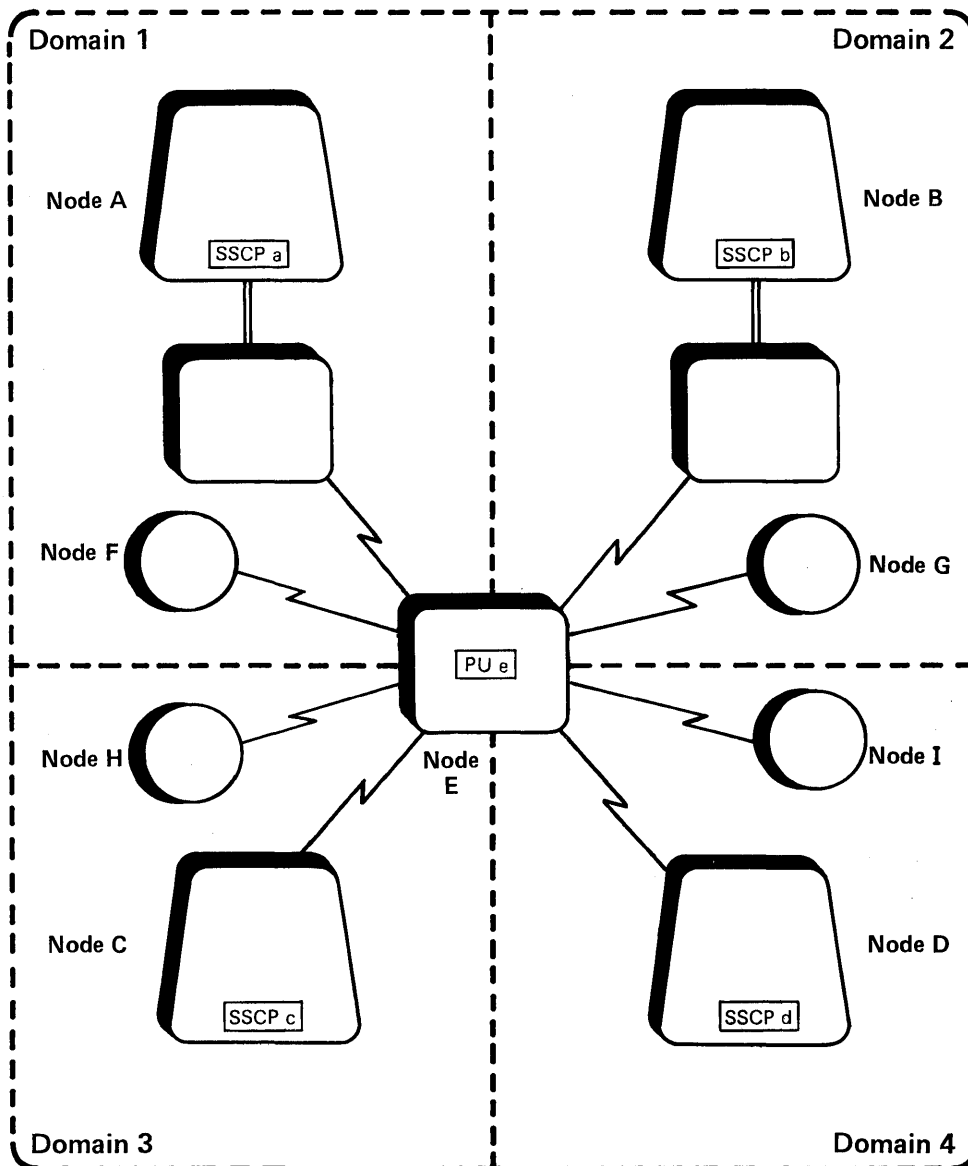
Serial sharing means that only one SSCP at a time can control a resource. When that SSCP relinquishes control, another SSCP can assume control. Four types of resources can be serially shared:

- Logical units in peripheral nodes
- Physical units in peripheral nodes
- Links and link stations that connect peripheral nodes to subarea nodes
- Switched SDLC links between subarea nodes.

Any of the SSCPs that concurrently share control of a type 4 node can serially share control of the resources in a peripheral node that is attached to that type 4 node. For example, any of the four SSCPs in Figure 13 can activate the resources in peripheral nodes *F*, *G*, *H*, and *I*; but only one SSCP at a time can control these peripheral node resources. A peripheral node's PU, LUs, links, and link stations are available to the SSCPs on a first-come, first-served basis.

## Share Limit


Each network resource has a share limit that specifies the maximum number of SSCPs that can share control of that resource. Resources that can be shared only serially have a share limit equal to 1. Resources that can be concurrently shared have a share limit greater than 1. You establish the desired share limit for each resource when coding your resource-definition statements. Appendix D provides examples of these resource-definition statements.





Legend:


PU = Physical Unit

SSCP = System Services Control Point

 = Type 2.1 node

 = Type 5 Node

 = SDLC Link

 = Type 4 Node


 = S/370 Data Channel

Figure 13. Serial Sharing of Network Resources

## Identifying Network Resources

End users and the path control network use names and addresses assigned during a system generation to identify network resources. The path control network uses (1) network addresses to route message units between subareas and to and from network addressable units (NAUs) in subarea nodes, and (2) local addresses to route message units between subarea and peripheral nodes and to and from NAUs in peripheral nodes.

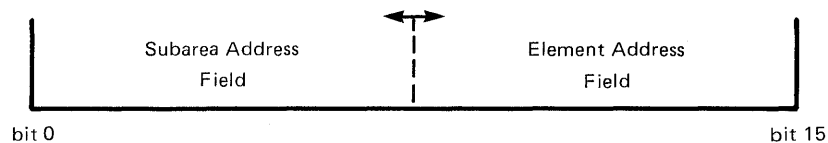
End users can communicate with one another without knowing the address of a resource. Instead of using addresses, they use names to identify network resources. An SSCP provides a directory service that translates these names into addresses for the path control network.

This section explains what a network address is and how it is generated. Also discussed are local addresses, network names, and the use of alias names and addresses for interconnected SNA networks.

## Network Addresses

**Network addresses** uniquely identify the system services control points (SSCPs), logical units (LUs), physical units (PUs), links, and link stations in an SNA network. The network address that is assigned to each network resource consists of a **subarea address** and an **element address**. Figure 14 shows the format of both the 16-bit network address and the extended, 23-bit network address.

16-bit network address:



Extended 23-bit network address:

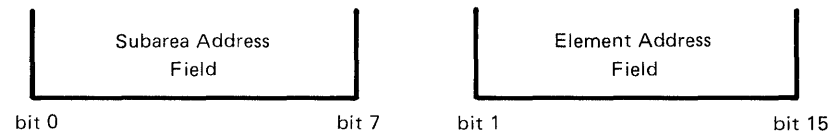


Figure 14. Format of a Network Address

## Subarea Addresses

Network addresses are based on the division of an SNA network into subareas. During system definition you assign a unique number to every subarea node (host processor and communication controller) in the network. This number becomes the subarea address field for:

- Network resources in that subarea node
- Network resources in peripheral nodes that are attached to that subarea node
- Links and link stations adjacent to that subarea node.

The unique number that you assign to each subarea node becomes the subarea address field for all the network resources in that subarea. Subarea path control elements use this subarea address to route message units between subareas. For example, consider the network configuration in Figure 15. Suppose that you assign subarea node A the number 3, subarea node B the number 5, and subarea node C the number 17. Then the subarea address for all the network resources in the subarea that contains node A will be 3. Similarly, the network resources in the subarea that contains node B will all have a subarea address of 5, and the network resources in the subarea that contains node C will all have a subarea address of 17.

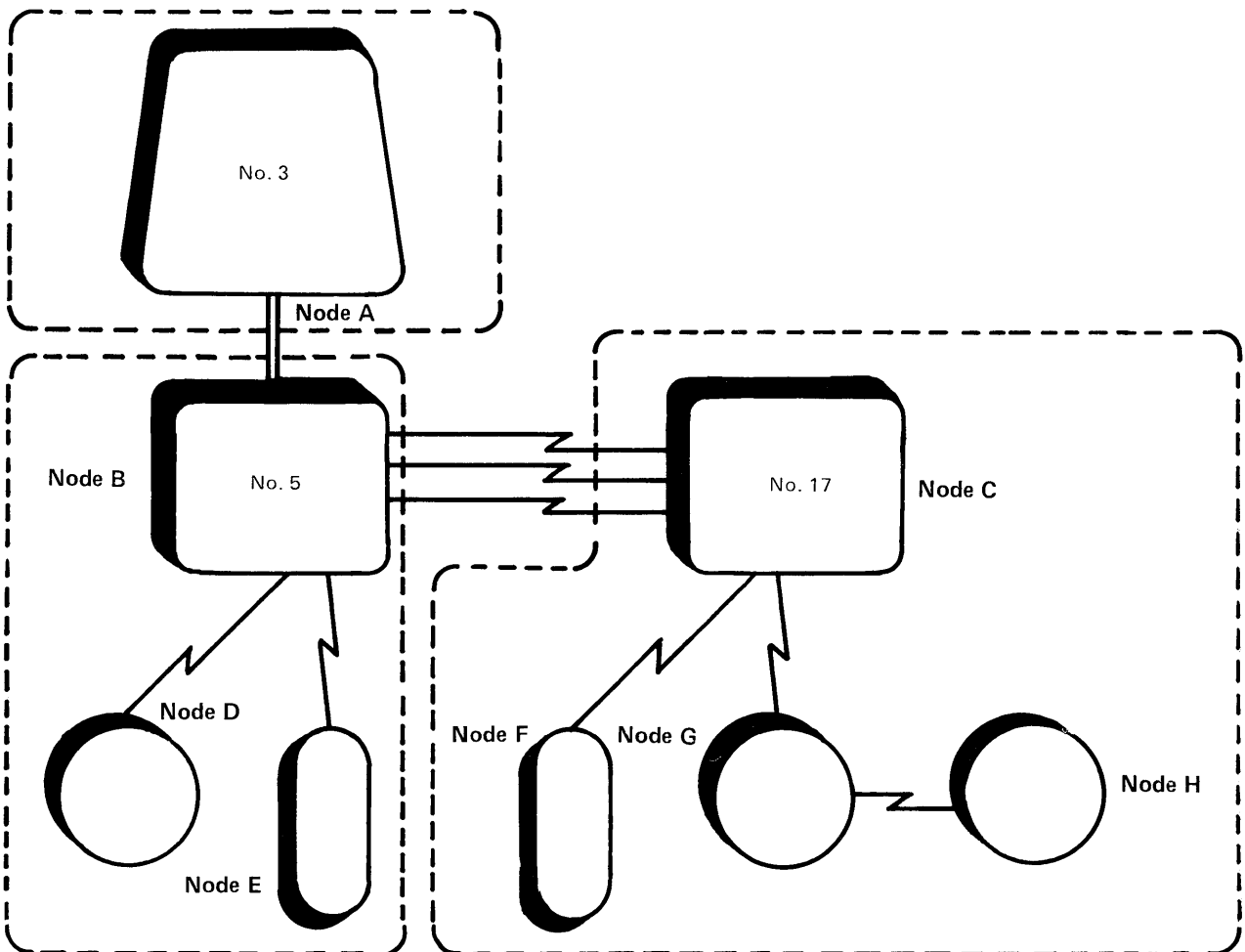


Figure 15. Assigning Subarea Addresses



## Element Addresses

Element addresses identify network resources within a subarea. Subarea path control elements in the destination subarea node use the element address field of the network address to route message units to the destination network addressable unit (NAU). Whereas each subarea address is a unique number in the network, an element address is unique only within each subarea.

You do not assign element addresses. SNA access methods and network control programs assign element addresses:

- During system generations
- During network reconfigurations
- During network activation for resources that are accessed over a switched SDLC link
- When initiating parallel LU-LU sessions.

SNA requires that certain network resources always use the same element address. Figure 16 charts these constant element addresses.

		NETWORK RESOURCE	ELEMENT ADDRESS
NODE TYPE	5	PU SSCP LUs	0 1 2 and beyond
	4	PU	0

**Figure 16. Constant Element Addresses**

SNA access methods and network control programs initially assign element addresses during a system generation. The sequence in which these programs encounter the resource-definition statements for each network resource determines the order in which they assign each element address.

## Extended Network Addressing

Prior to the announcement of extended network addressing, a network address was 16 bits long. The subarea address field ranged from 1 to 8 bits, and the remaining bits determined the maximum number of resources that could be defined within each subarea. The number of bits for the subarea address field and the number of bits for the element address field in a 16-bit network address is called the network address split. The address split you chose for the subarea and element address fields had to remain constant throughout the network.

Extending the network address from 16 bits to 23 bits provides additional addresses for users with larger networks. An extended network address uses a fixed 8-bit subarea address field to address up to 255 subarea nodes. The element address field of the extended network address uses 15 bits. This permits the assignment of up to 32,768 element addresses within each subarea. Extended network addressing enables network designers to attach more resources to each communication controller and host subarea node.

## Local Addresses

Resources in peripheral nodes have both network addresses and local addresses. Whereas network addresses uniquely identify any PU, LU, link, or link station in a network, **local addresses** uniquely identify these resources within a peripheral node. Local addresses are not the same as the element field in a network address, nor are they necessarily unique except within a peripheral node.

Peripheral nodes are solely the origin and destination of end-user data. They do not participate in overall network routing based on network addresses; they depend on boundary function support in subarea nodes to pair their local addresses with the network addresses that subarea path control elements use. The boundary function thereby insulates peripheral nodes from network address changes that result from network reconfigurations.

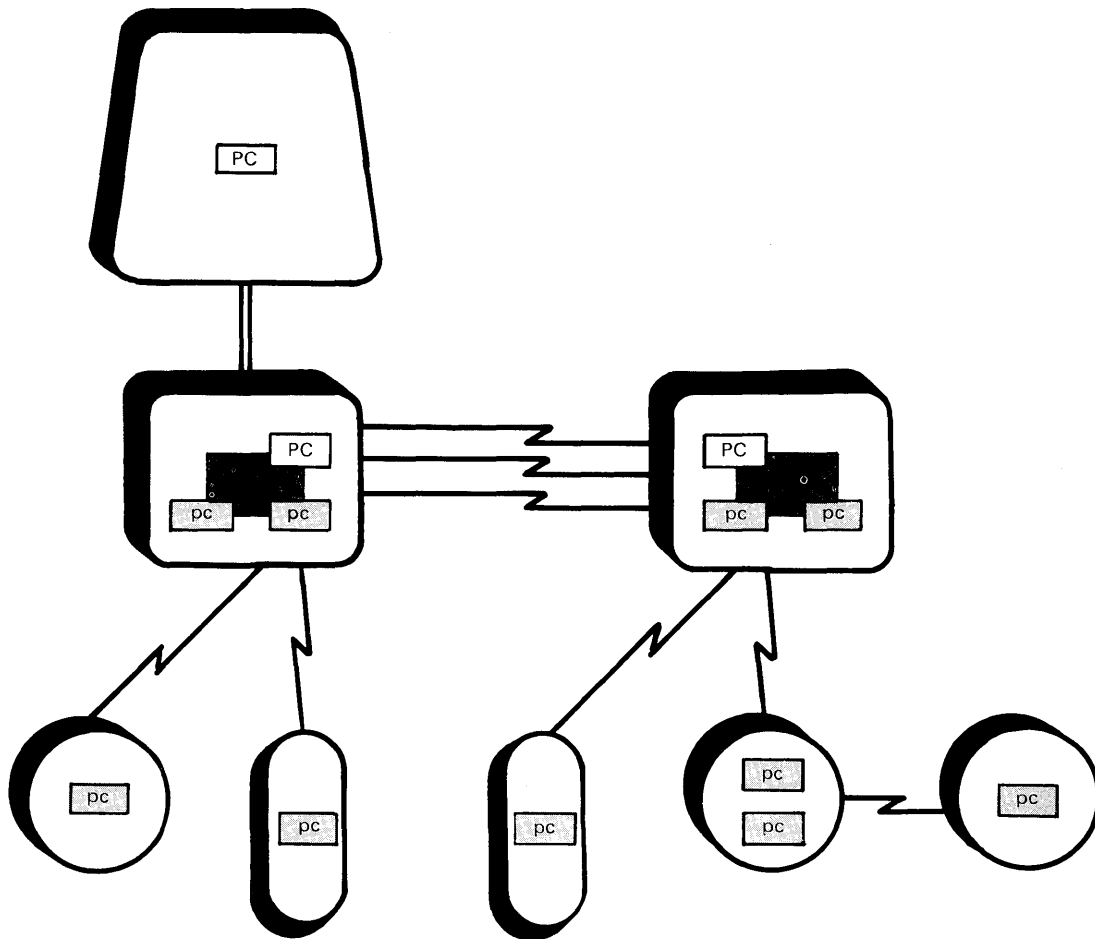
## Boundary Function

**Boundary function** components interconnect subarea and peripheral path control elements. Boundary functions are *not* part of the path control network. The architecture defines boundary functions to pair the network addresses that subarea path control elements use with the local addresses that peripheral path control elements use. Every subarea node that has peripheral nodes attached to it contains a boundary function component, as Figure 17 illustrates.

Subarea nodes perform an *intermediate routing function* when they route message units between NAUs in different subareas; they perform a *boundary function* when they route message units between a subarea node and an adjacent peripheral node. Subarea path control uses the subarea address field of the destination network address to route message units between subareas. Once a message unit reaches the destination subarea node, path control uses the element address field of the destination network address to identify the destination of the message unit.

If the destination node is a peripheral node, subarea path control uses the boundary function component to pair the network address that it uses with the equivalent local address that the destination peripheral node uses. Then a peripheral path control element in the subarea node uses this local address to route the message unit to peripheral path control in the destination node.

Peripheral path control elements use local addresses to route data from a peripheral node to an adjacent subarea node. The peripheral path control element in the subarea node that receives the message unit relies on the boundary function to pair the local address with a network address. Then subarea path control uses this network address to route the message unit through the network.



Legend:

- PC = Subarea Path Control Element
- pc = Peripheral Path Control Element
- BF = Boundary Function Component

**Figure 17. Boundary Function Components**

## Network Names

In addition to a network address, each PU, LU, link, and link station can have a network name. A **network name** is a symbolic identifier that frees application programs, network operators, and workstation operators from concern about the location of different resources within the network. You assign network names during the system definition process.

During system generations, access methods build directories that relate the network names of resources in each domain to their network addresses. The SSCP uses this directory to translate network names to network addresses for resources within its domain.

In addition, each access method builds a cooperative directory that allows SSCPs to translate names to addresses across domains. If the named resource is not in an SSCP's domain, the SSCP uses the cooperative directory to identify which SSCP can provide the name-to-address translation. Thus, SSCPs cooperatively translate a network name that an end user in one domain specifies into a network address in another domain.

The network names that you assign should be unique within your network. Use a consistent naming convention to ensure that you do not assign duplicate network names either initially (during system definition) or later (when reconfiguring the network). In situations such as back up or recovery of failing resources, names created under such a convention help network operators identify resources and their location in the network.

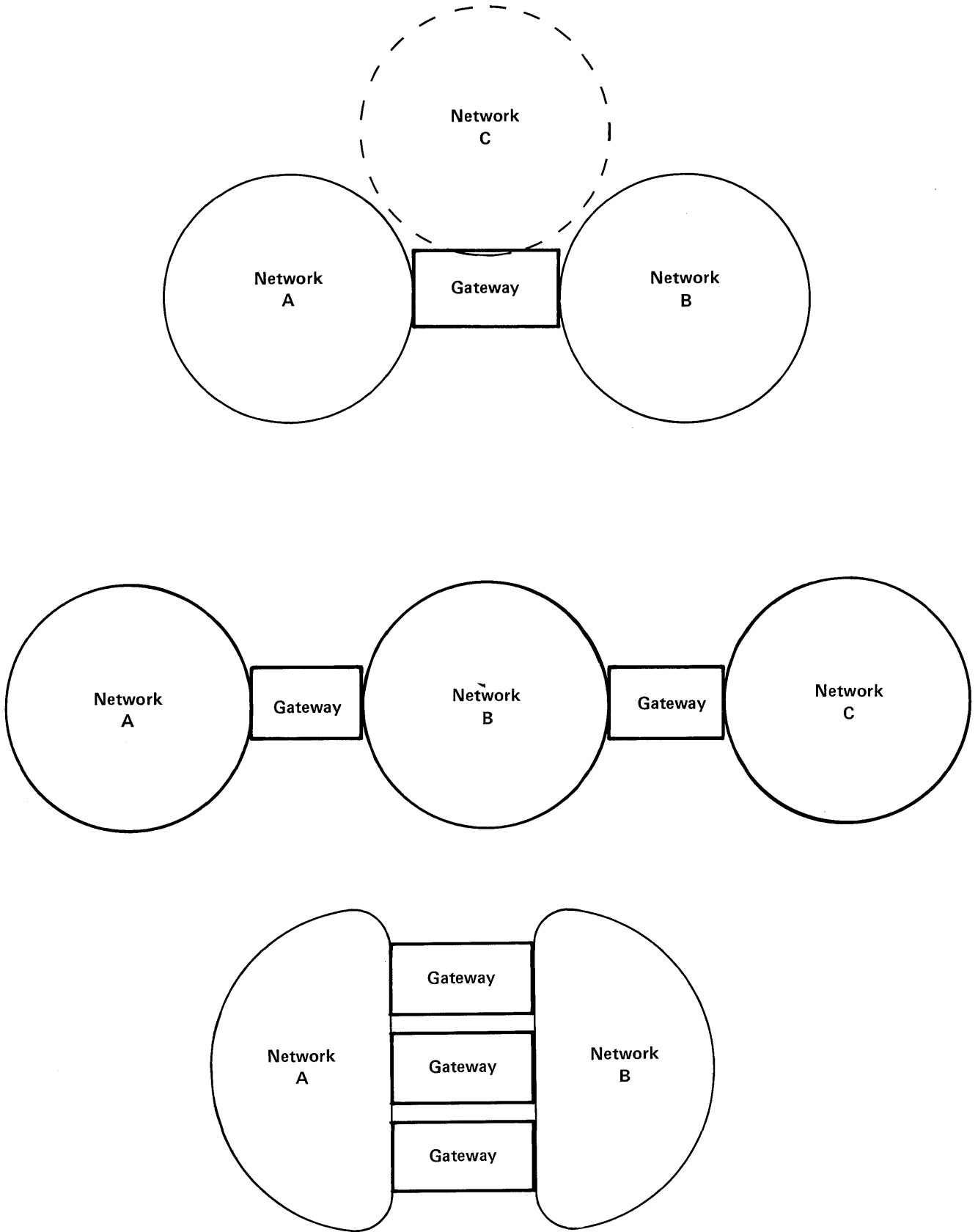
## Considerations for Interconnected Networks

The SNA network interconnection capability allows end users in different networks to communicate with each other while allowing you to configure, define, and manage each network independently. This section explains why interconnected networks do not require (1) a common network address structure using all unique network addresses and (2) a common network naming convention to maintain unique resource names.

SNA network interconnection provides a **gateway** between two or more SNA networks that permits each network to maintain the network address scheme most appropriate to its specific configuration. The subarea addresses that you assign in one network can be duplicated in interconnected networks. You can interconnect networks that do not yet support extended network addressing to networks that do support extended network addressing. In addition, you can interconnect networks that have different network address splits.

Gateways also enable interconnected SNA networks to maintain their own network naming conventions. The network names that you assign in one network can be duplicated in interconnected networks.

Figure 18 shows three configurations for interconnecting SNA networks. The first configuration shows how a single gateway can interconnect two or more networks. The second configuration shows how two gateways can interconnect two networks. In this configuration, the only resources that network B contains are the links that connect the two gateways. The third configuration shows how multiple, parallel gateways can interconnect two networks. The gateway itself consists of a **gateway node** and one or more **gateway SCCPs**.



**Figure 18. Interconnected Network Configurations**

## Gateway Nodes

**Gateway nodes** route message units between interconnected networks by translating the network addresses that resources in one network use into alias addresses that resources in an interconnected network use. An **alias address** is a network address that identifies in one network an LU or SSCP that resides in an interconnected network. A gateway node:

- Contains two or more subarea path control elements
- Contains a gateway function component
- Recognizes addresses from two or more interconnected networks
- Has a network address in each interconnected network.

Gateway nodes contain one subarea path control element for every interconnected network. Each of these path control elements routes data between one of the interconnected networks and the other gateway node path control elements.

The **gateway function** component acts as an intermediary between gateway path control elements, similar to the way a boundary function component acts as an intermediary between subarea and peripheral path control elements. Whereas boundary function components pair the network address that subarea path control uses with the local address that peripheral path control uses, gateway function components pair the network address that one network uses with the alias address that an interconnected network uses. Path control elements in gateway nodes use *real* network addresses to route data to and from the network that they represent. They use *alias* addresses to route data to and from other gateway node path control elements.

None of the interconnected networks is aware that the gateway node is performing address translations. Viewed from any one of the interconnected networks, the gateway node is a part of that network. Therefore, you must assign the gateway node a unique subarea address in each of the interconnected networks.

## Gateway SSCPs

**Gateway SSCPs** translate the network names that one network uses to the network names that an interconnected network uses. Every gateway includes at least one gateway SSCP. However, a network designer can choose to have more than one gateway SSCP per gateway. The share limit that you assign to the PU in the gateway node determines the maximum number of gateway SSCPs in that gateway.

Just as gateway nodes use real and alias network addresses to route data between interconnected networks, gateway SSCPs use real and alias names to identify resources in interconnected networks. An **alias name** is a name that identifies in one network an LU or SSCP in an interconnected network. You assign alias names during the system definition process.

## Interaction between Gateway Nodes and Gateway SSCPs

Gateway nodes and gateway SSCPs interact with one another to pair the real and alias addresses that one network uses to the real and alias addresses that an interconnected network uses.

The first message unit that a gateway node receives from an SSCP or LU in one of the interconnected networks contains (1) that SSCP's or LU's real network address and (2) an alias address that identifies one of the SSCPs or LUs in an interconnected network. However, the gateway node is initially unable to pair either (1) the origin's real address with its alias representation in the interconnected network or (2) the destination's alias address with its real address in the interconnected network. The gateway node relies on the gateway SSCP for this information.

The gateway SSCP first sends a Request Network Address Assignment (RNAA) request to the PU in the gateway node. The RNAA provides the gateway node with the alias address that identifies the origin SSCP or LU to resources in the interconnected network. Then the gateway SSCP sends another request, called a Set Control Vector (SETCV), to the gateway node. The SETCV provides the gateway node with the real address of the destination SSCP or LU in the destination network.

The gateway node can now pair (1) the alias address in the RNAA with the real address of the origin SSCP or LU and (2) the real address in the SETCV with the alias address of the destination SSCP or LU. This interaction between the gateway node and the gateway SSCP occurs only during the initiation of a session between resources in interconnected networks. Thereafter, the gateway node can pair real and alias network addresses for that session's traffic without additional information from the gateway SSCP.

## Network Identifiers

To guarantee that a network address or name is unique among interconnected networks, gateways use network identifiers. A **network identifier (ID)** uniquely identifies a network within a set of two or more interconnected networks. Gateway nodes and gateway SSCPs use network identifiers to qualify the names and addresses that they exchange with one another. You assign a unique network ID to each interconnected network during the system definition process.

A **network-ID qualified address** identifies a real network address and the network in which that address is valid. A **network-ID qualified name** identifies a real network name and the network in which that name is valid. Only gateway SSCPs and gateway nodes use network-ID qualified names and addresses. Because resources in one network are not aware of resources in an interconnected network, they do not use network-ID qualified names or addresses. Instead, they use alias network names and addresses to identify resources in an interconnected network.



## Summary

System definition includes:

- Defining network resources (PUs, LUs, links, and link stations) to one or more SSCPs
- Assigning unique subarea numbers to each subarea node
- Assigning names to network resources for use by application programs, workstation operators, and network operators.

When you define network resources to an SSCP, you are identifying the domains of the network. An SSCP activates, controls, and deactivates resources within its domain.

Network addresses uniquely identify resources within a network. Local addresses uniquely identify resources within a peripheral node. Subarea path control elements use network addresses to route message units through a network, and peripheral path control elements use local addresses to route message units to and from peripheral nodes.

Boundary function components act as an intermediary between subarea and peripheral path control elements. Boundary function components pair the network address that subarea path control elements use with the local address that peripheral path control elements use. The use of local addresses permits peripheral nodes to be independent of any network configuration changes (changes in the overall network address space).

The network names that you assign to each resource enable end users to uniquely identify network resources without regard to their actual location in the network. SSCPs provide directory services that translate network names to network addresses.

Gateways connect two or more SNA networks to one another. A gateway consists of a gateway node and one or more gateway SSCPs. Gateways act as an intermediary between interconnected networks, receiving message-unit traffic from one network, performing any necessary name or address translations, and routing the message units to one of the interconnected networks.

### Network Activation

This chapter explains how control points activate and deactivate network resources and how you can control network activation and deactivation.

### Contents

Network Activation	43
Activating Sessions with Physical Units and Logical Units	43
Activating Links	44
Hierarchy of Network Activation	44
Controlling Network Activation	58
Cascaded Activation and Deactivation	58
Configuring and Reconfiguring a Network	58
Scheduled Changes	58
Unscheduled Changes	59



## Network Activation

System services control points (SSCPs) activate network resources based on your resource-definition statements. This section explains how SSCP activate network resources and describes the hierarchy of network activation.

You begin activation of an SNA network by turning on the power to the hardware and by loading and initializing the operating systems, access methods, network control programs, and peripheral node software. Nodes become operational when their hardware and software are functioning properly. An SSCP becomes active when its type 5 node becomes operational.

### Activating Sessions with Physical Units and Logical Units

An SSCP activates sessions with the physical units (PUs) and logical units (LUs) that you defined to it during the system definition process. An SSCP must activate these sessions, called SSCP-PU and SSCP-LU sessions, before the physical units and logical units can become an active part of the network. SSCP-PU sessions and SSCP-LU sessions remain active until network deactivation.

An SSCP communicates with physical units and logical units over SSCP-PU and SSCP-LU sessions. An SSCP needs to communicate with the physical unit in each node in order to control and monitor the resources in that node. Logical units need to communicate with an SSCP before they can initiate an LU-LU session. A logical unit submits requests for LU-LU session initiation to an SSCP over the SSCP-LU session.

A **peripheral node control point (PNCP)** activates sessions with PUs and LUs that are located in type 2.1 nodes. A PNCP is a control point in a type 2.1 node that provides a subset of the SSCP functions. Peripheral node control points activate, deactivate, and manage network resources in type 2.1 nodes. Like SSCP, a PNCP becomes active when its node becomes operational.

Once the PNCP-LU sessions are active, the logical units can initiate sessions with logical units in other, directly-attached type 2.1 nodes. To initiate sessions with logical units elsewhere in the network, the LUs in a type 2.1 node must first have an active session with an SSCP, as described above.

Both an SSCP and a PNCP can activate sessions with the PU and LUs in a type 2.1 node that is attached to a subarea node. In this case, the SSCP shares control of these resources with the PNCP in that node. However, an SSCP does not activate sessions with the PU and LUs in a type 2.1 node that is not attached to a subarea node. Only a PNCP can activate sessions with these resources.

## Activating Links

An SSCP activates the links that you defined to it through access-method resource-definition statements. SSCPs activate, control, and deactivate links through the link stations (data link control elements) in each node. To activate a link, an SSCP sends commands (over the SSCP-PU session) to the physical unit in every node that the link connects to. Each physical unit then issues link-level commands to the adjacent link stations for that link. Once an SSCP activates the links between adjacent nodes, those links can carry session traffic.

A peripheral node control point (PNCP) can activate links that are attached to a type 2.1 node. **A physical unit control point (PUCP)** can activate links that are attached to a type 2.0 or type 4 node. Recall that a PNCP is a control point in a type 2.1 node that provides a subset of the SSCP functions to activate, deactivate, and manage network resources in its node. A PUCP is a control point in type 2.0 and type 4 nodes; it provides a smaller subset of the SSCP functions to activate, deactivate, and manage the links that are attached to its node.

A link must be activated from both ends. During initial network activation, the SSCP must activate a link to an adjacent node before it can activate sessions with the PU and LUs in that node. Because the SSCP has an active SSCP-PU session with the physical unit in only one of the nodes that the link connects to, the control point in the node at the other end of the link must also direct its physical unit to activate that link. The physical units in the nodes at each end of the link then issue link-level commands to the adjacent link stations, completing the link activation procedure.

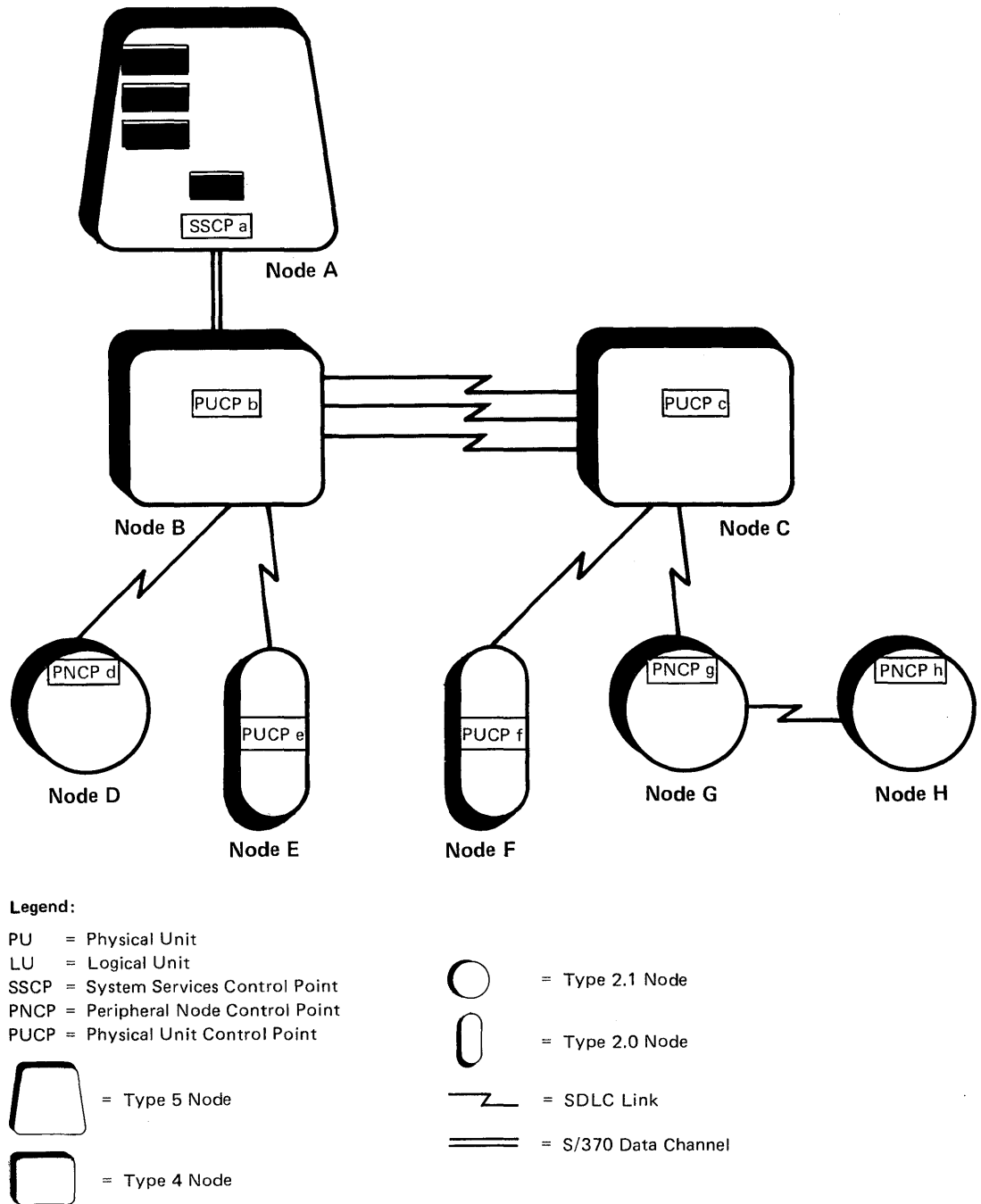
## Hierarchy of Network Activation

There is a hierarchy of network activation. Network activation begins in a host node and continues through adjacent nodes in the network. The hierarchy of network activation is:

- Physical units and logical units in host nodes
- Links to adjacent nodes
- Adjacent link stations
- Physical units in the adjacent nodes
- Logical units in the adjacent nodes.

Using a single-domain network configuration as an example, the hierarchy of network activation would be:

1. Activate the resources in the host node, as shown in Figure 19.



**Figure 19. Hierarchy of Network Activation: Part I**

a. PU a

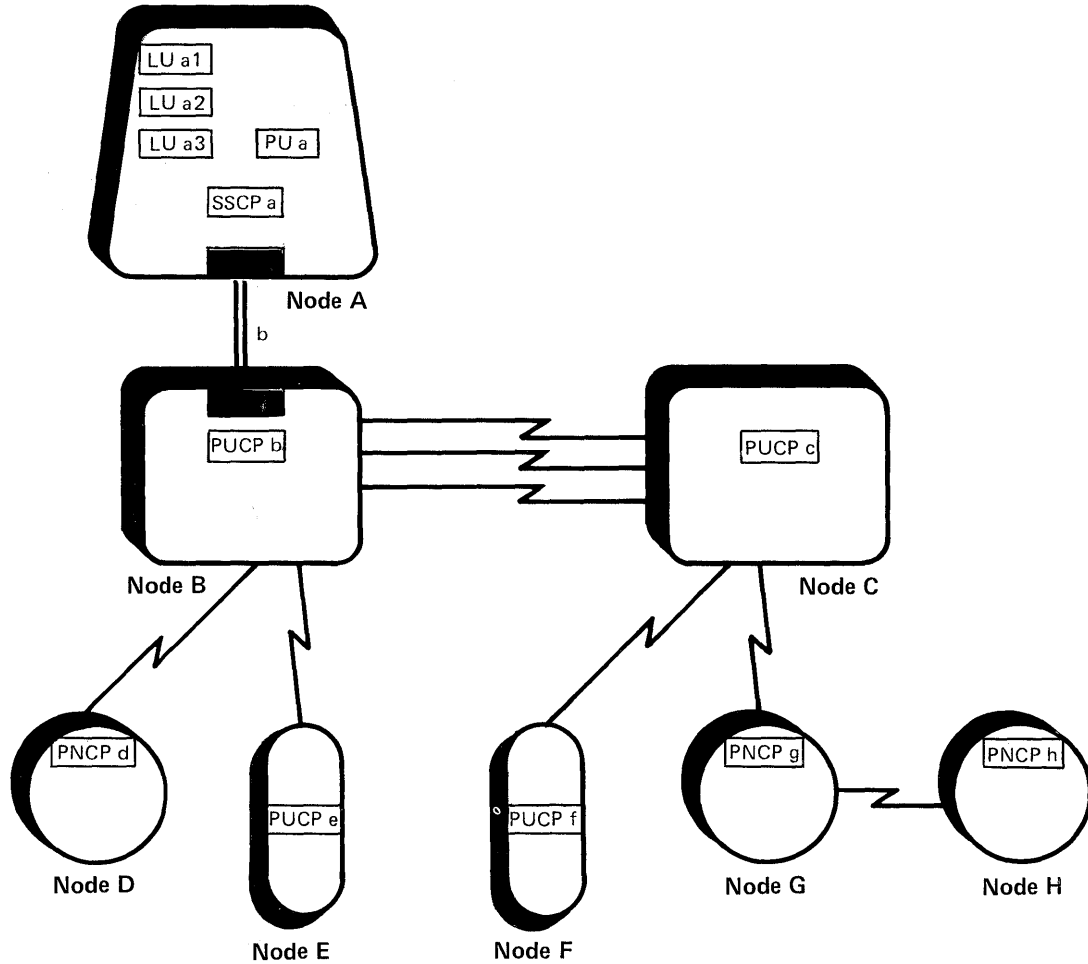
SSCP a requests the SSCP-PU session by sending an Activate Physical Unit (ACTPU) request to PU a. A positive response from PU a completes the activation of the SSCP-PU session.

b. LUs a1, a2, and a3

SSCP a requests the SSCP-LU sessions by sending Activate Logical Unit (ACTLU) requests to LUs a1, a2, and a3. Positive responses from the LUs complete the activation of these SSCP-LU sessions.

In an actual implementation, the network operator does not directly request the activation of PUs and LUs in type 5 nodes. The SSCP activates these sessions automatically.

2. Activate the link that is attached to the host node, as shown in Figure 20.



- Legend:
- PU = Physical Unit
  - LU = Logical Unit
  - SSCP = System Services Control Point
  - PNCP = Peripheral Node Control Point
  - PUCP = Physical Unit Control Point
  - LS = Link Station

Figure 20. Hierarchy of Network Activation: Part II

a. Link *b*

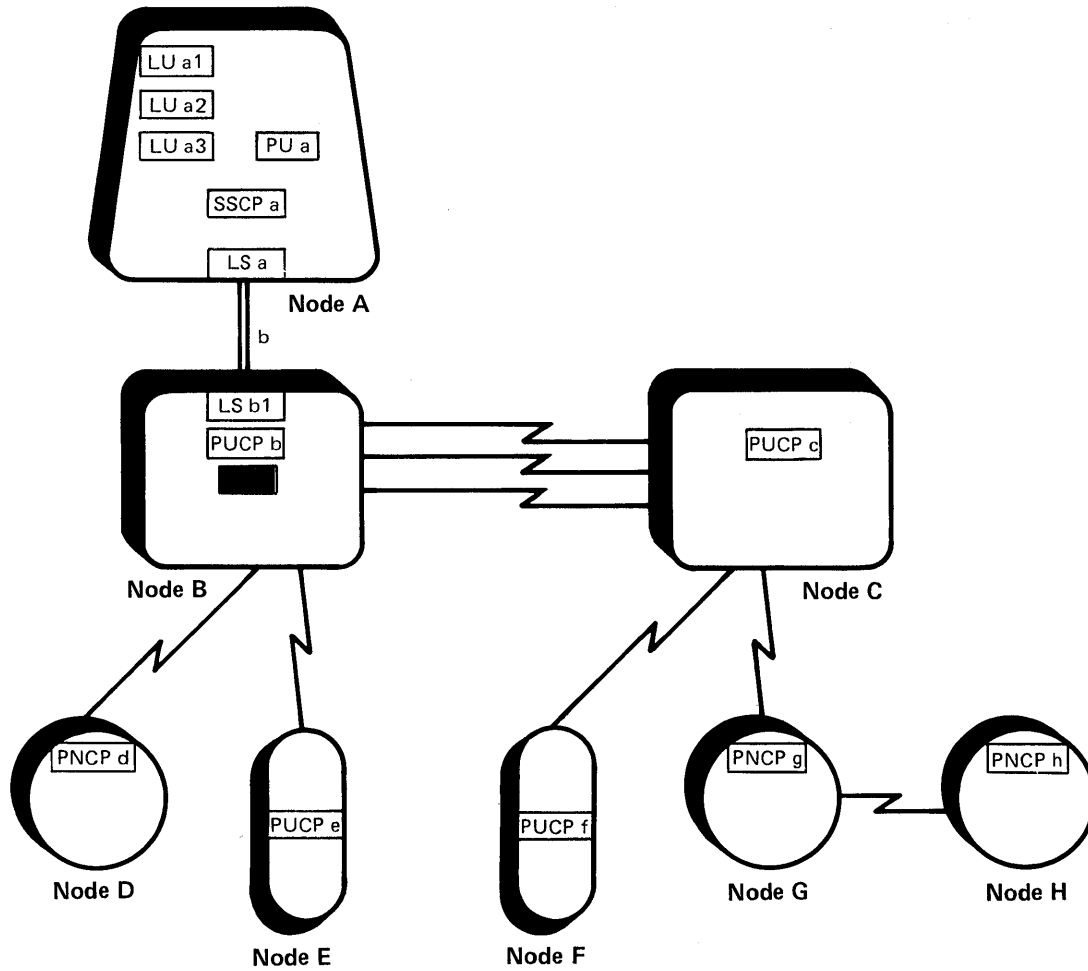
- 1) SSCP *a* sends an Activate Link (ACTLINK) request to PU *a* over the SSCP-PU session.<sup>2</sup>
- 2) PU *a* informs the SSCP over the SSCP-PU session that link *b* is operational.
- 3) SSCP *a* issues a Contact (CONTACT) command that requests PU *a* to contact LS *b1*.
- 4) LS *a* and LS *b1* exchange identifications and the data link protocol commands necessary to activate link *b*.
- 5) LS *a* informs PU *a* that it has successfully contacted its channel-attached link station.
- 6) PU *a* returns a Contacted (CONTACTED) command to SSCP *a* to inform the SSCP that LS *b1* has been contacted.

---

<sup>2</sup> A symmetric sequence is performed by the PUCP and PU in node B.



3. Activate the resources in adjacent communication controller node B, as shown in Figure 21.



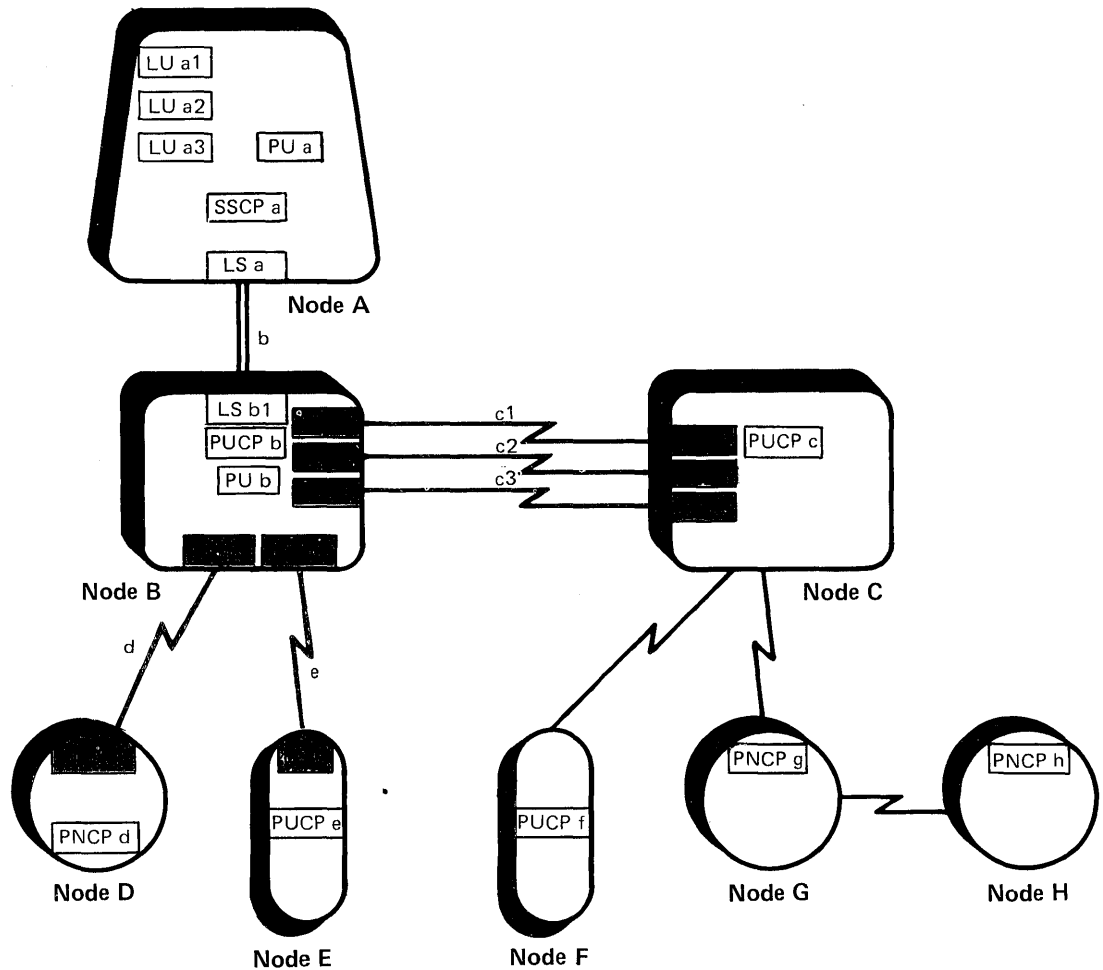
- Legend:**
- PU = Physical Unit
  - LU = Logical Unit
  - SSCP = System Services Control Point
  - PNCP = Peripheral Node Control Point
  - PUCP = Physical Unit Control Point
  - LS = Link Station

**Figure 21. Hierarchy of Network Activation: Part III**

a. PU b

The SSCP activates an SSCP-PU session with PU b.

4. Activate the links that are attached to communication controller node B, as shown in Figure 22.



**Legend:**

- PU = Physical Unit
- LU = Logical Unit
- SSCP = System Services Control Point
- PNCP = Peripheral Node Control Point
- PUCP = Physical Unit Control Point
- LS = Link Station

**Figure 22. Hierarchy of Network Activation: Part IV**

a. Links *c1*, *c2*, and *c3*

The SSCP sends PU *b* ACTLINK requests for the links that connect nodes B and C.

LS *b2* communicates with LS *c1*; LS *b3* communicates with LS *c2*; and LS *b4* communicates with LS *c3* to activate links *c1*, *c2*, and *c3*, respectively.

b. Links *d* and *e*

The SSCP sends an ACTLINK command to PU *b* to activate links *d* and *e*. LS *b5* communicates with LS *d*, and LS *b6* communicates with LS *e*. PU *b* tells the SSCP when it has successfully contacted link stations LS *d* and LS *e*.

Adjacent link stations that are connected by switched SDLC links send Exchange Identification (XID) commands and responses to one another to determine which link station is the primary link station. Then the primary link station sends a Set Normal Response Mode (SNRM) command to the secondary link station, and the secondary link station returns an Unnumbered Acknowledgment (UA) response to the primary link station.

Adjacent link stations that are connected by nonswitched SDLC links may exchange XID commands and responses before the primary link station sends an SNRM command and the secondary link station returns a UA response. For example, adjacent link stations would exchange XID commands and responses to negotiate transmission group numbers for subarea links (see "Chapter 5. Route Design") or to negotiate the primary and secondary role for directly-connected type 2.1 nodes. For additional information on the sequence of commands that adjacent link stations exchange and the definition of primary and secondary link stations, refer to the *IBM SDLC General Information* manual and the *Installation Guide Scenarios for Release 3 Advanced Communications Function (ACF)*.

5. Activate the resources in nodes that are adjacent to communication controller node B, as shown in Figure 23.

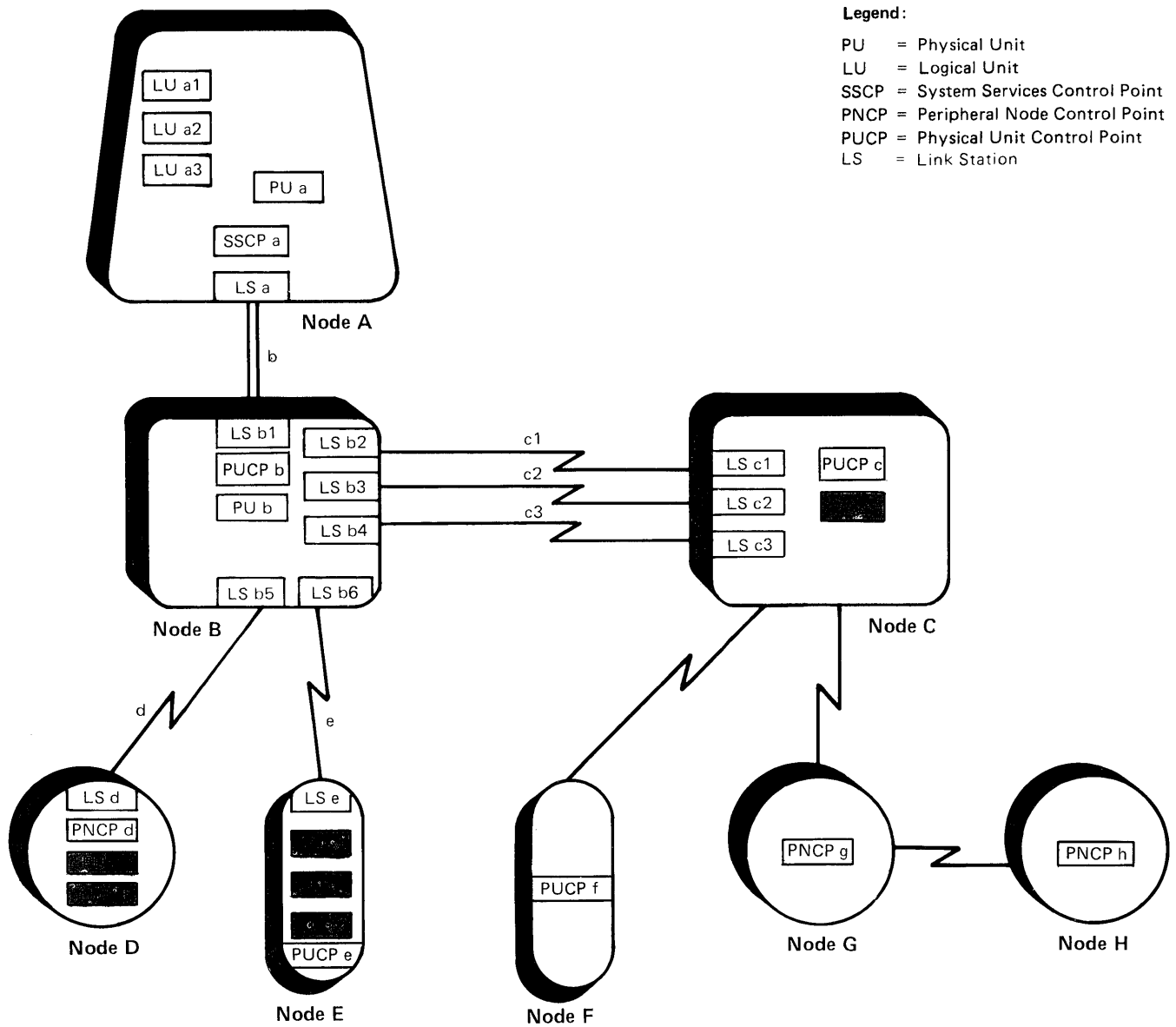


Figure 23. Hierarchy of Network Activation: Part V

- a. PUs c, d, and e
- b. LU d1 and LUs e1 and e2

The SSCP activates SSCP-PU sessions with the physical units and then activates SSCP-LU sessions with the logical units.

6. Activate the links that are attached to communication controller node C, and then activate the resources in the attached nodes, as shown in Figure 24.

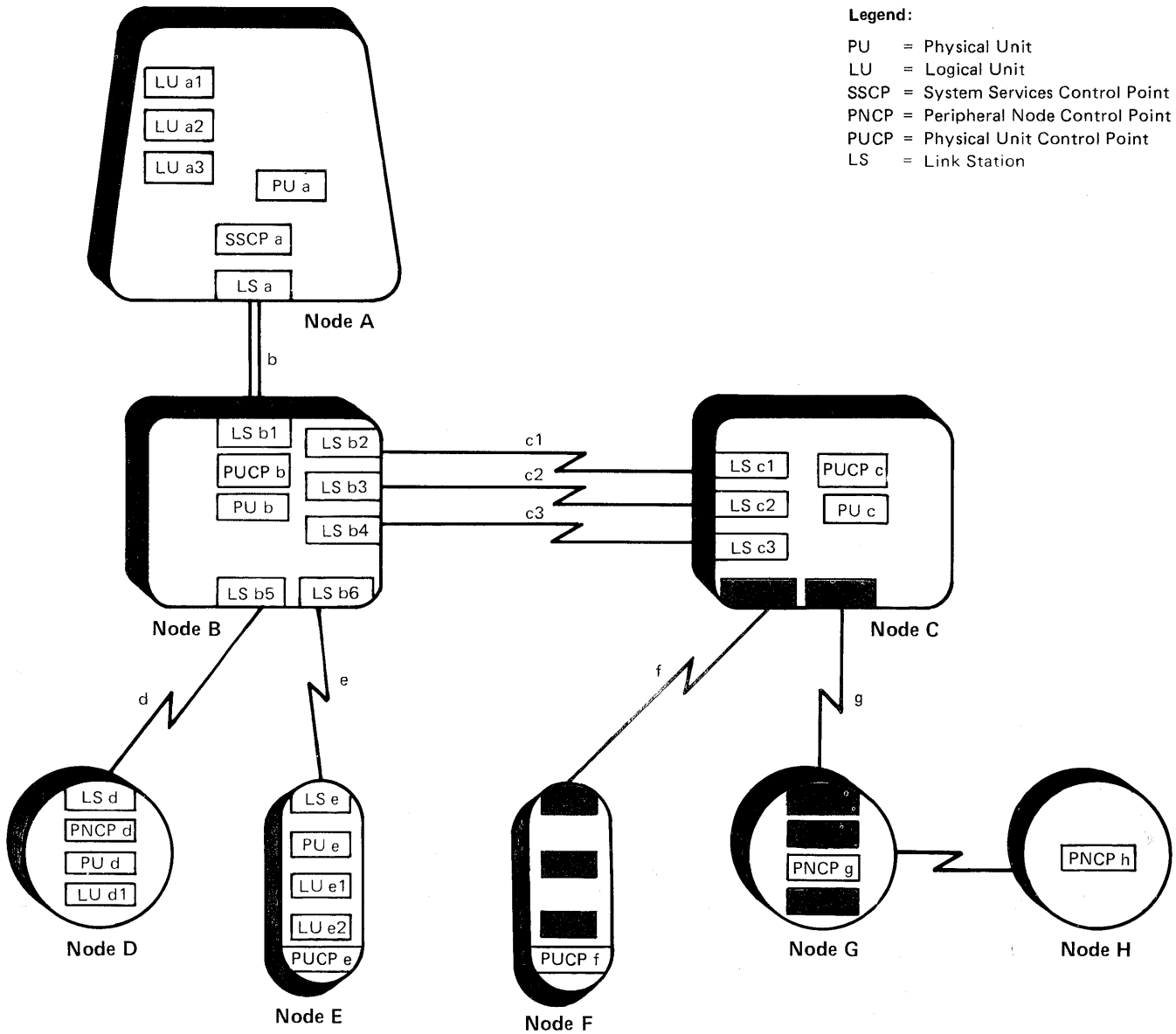


Figure 24. Hierarchy of Network Activation: Part VI

a. Links *f* and *g*

The SSCP sends PU *c* ACTLINK and CONTACT requests for these links over the SSCP-PU session.

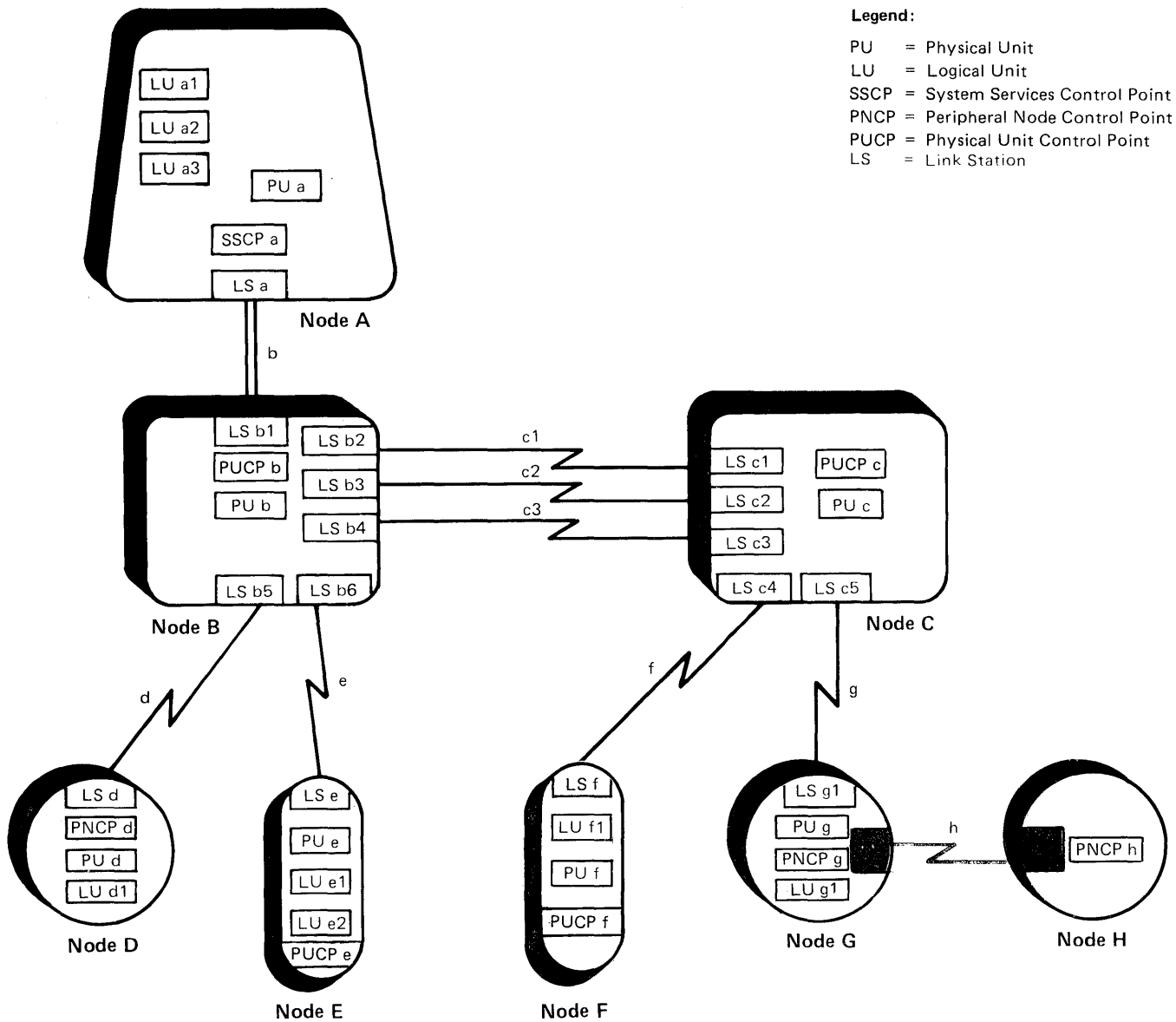
b. PUs *f* and *g*

The SSCP sends ACTPU requests to these physical units.

c. LU *f1* and LU *g1*.

The SSCP sends ACTLU requests to these logical units.

7. Local operators for type 2.1 nodes are now able to activate the link between adjacent type 2.1 peripheral nodes, as shown in Figure 25.



**Figure 25. Hierarchy of Network Activation: Part VII**

a. Link *h*

Recall that PNCPs, not SSCP, activate the links between adjacent type 2.1 nodes. The procedure for activating link *h* is the same as for links *b* through *g*, except that PNCP *g* sends PU *g* the ACTLINK command for link *h*.<sup>3</sup>

<sup>3</sup> A symmetric sequence is performed by the PNCP and PU in node H.

8. Local operators activate the resources in type 2.1 nodes that are not attached to a subarea node, shown in Figure 26.

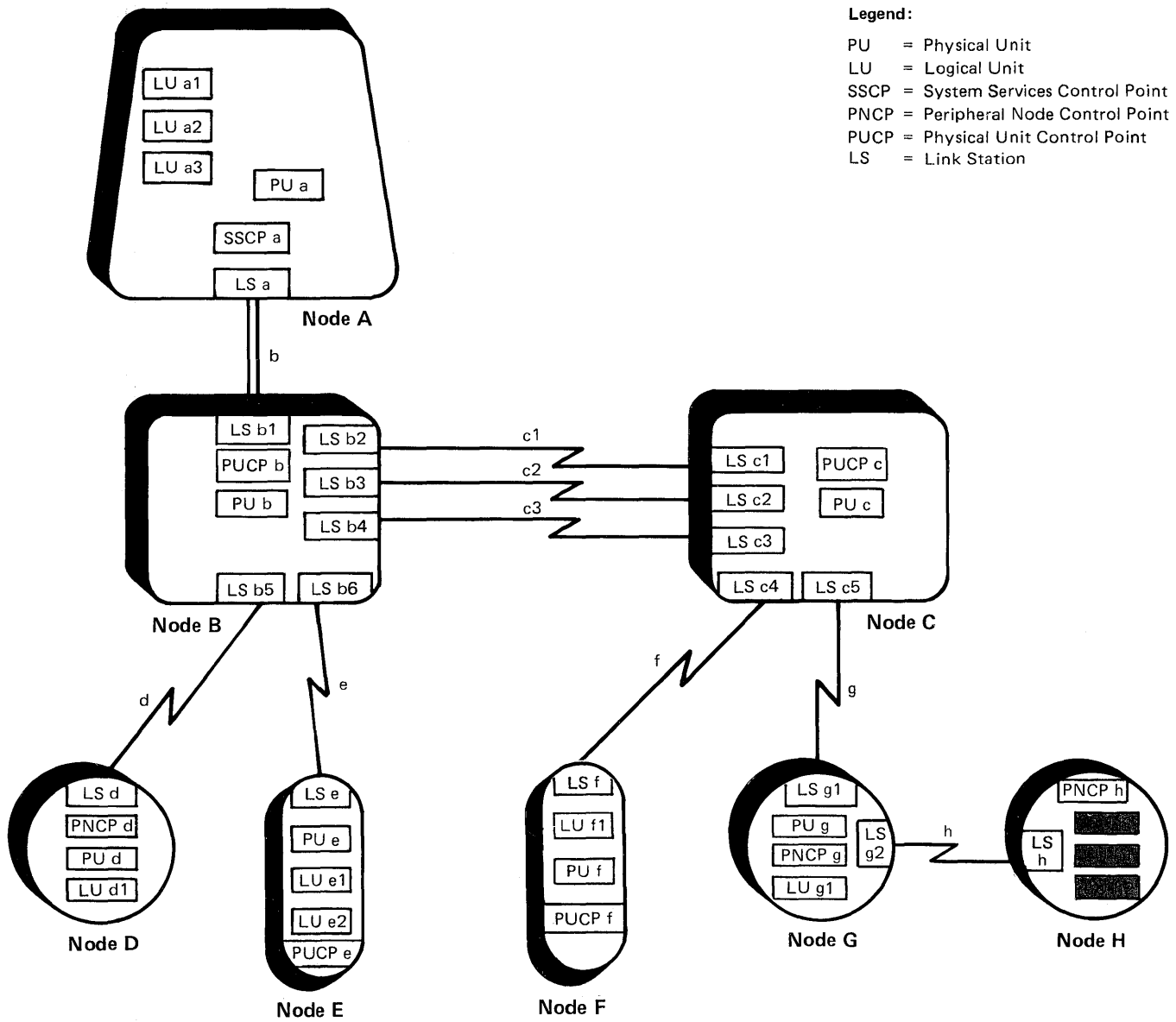


Figure 26. Hierarchy of Network Activation: Part VIII

- a. PU *h*
- b. LUs *h1* and *h2*

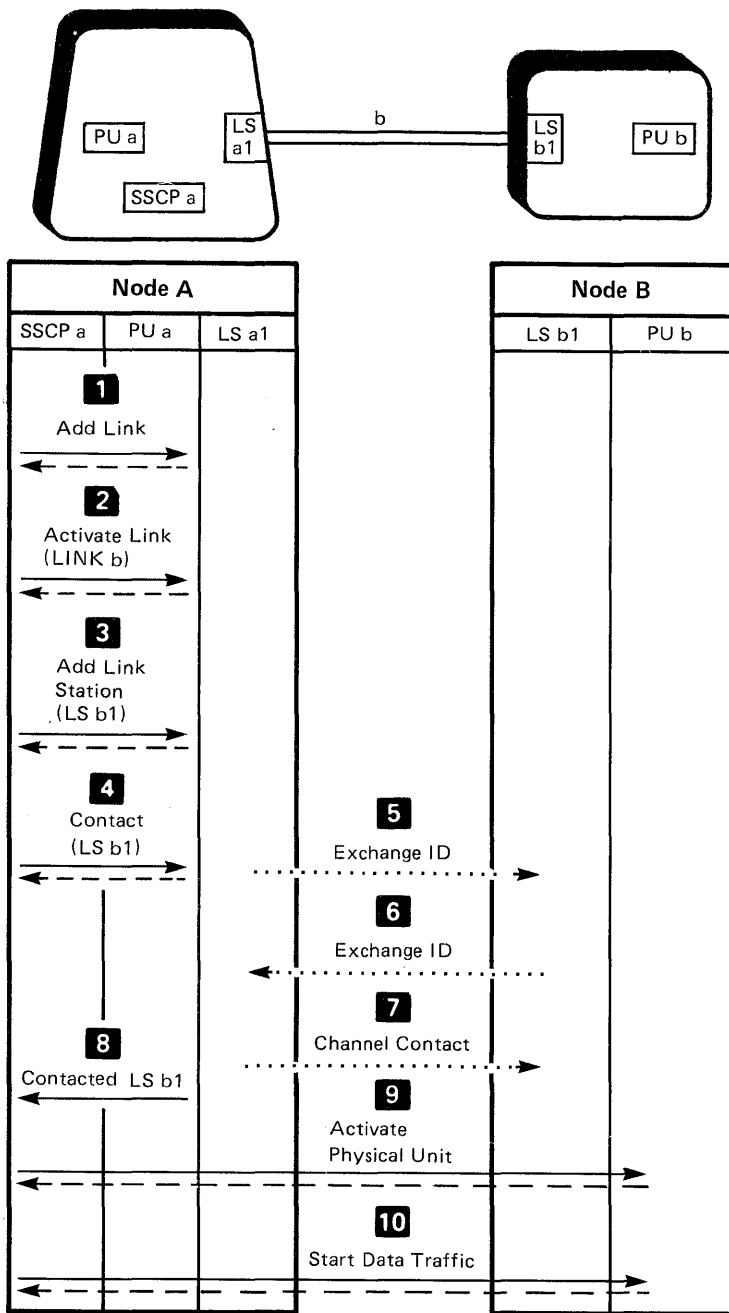
PNCP *h* activates a PNCP-PU session with PU *h* and activates PNCP-LU sessions with the logical units in its node.

The process of activating the links to adjacent nodes, then the PUs, and finally the LUs, continues until all the network resources are active. Network deactivation follows the same hierarchy as network activation, but in reverse order.

Figure 27 and Figure 28 present command sequences for activating and deactivating some of the network resources just shown. The purpose of these sequence charts is to illustrate typical activation and deactivation command flows in an SNA network.

Appendix C contains additional sequence charts that illustrate the various command flows for network activation and deactivation. Detailed discussions of the commands and their sequences are provided in the *SNA Format and Protocol Reference Manual: Architectural Logic*, the *SNA Format and Protocol Reference Manual: Architecture Logic for LU Type 6.2*, and the *SNA Reference Summary*.



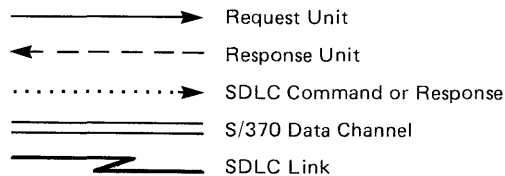


- 1** SSCP a requests PU a to furnish SSCP a with a network address for the designated link. PU a does so.
- 2** SSCP a tells PU a to activate LINK b and to prepare to issue and receive data link control commands and responses for the link.
- 3** SSCP a requests PU a to furnish SSCP a with a network address for the designated link station. PU a does so.
- 4** SSCP a tells PU a to contact the adjacent link station LS b1. The representation of the link station in Node A has network address of a1.
- 5** PU a sends PU b information about Node A, including the maximum number of bytes that Node A will accept across the channel at one time.
- 6** PU b informs PU a that the parameters sent by PU a are acceptable, and sends PU a information about Node B.
- 7** PU a completes the activation of LINK b by accepting the parameters sent by PU b.
- 8** PU a informs SSCP a that message units can now be sent to PU b through link station LS a1.
- 9** SSCP a identifies itself to PU b and assumes control of Node B by activating an SSCP-PU session with PU b.
- 10** SSCP a enables the flow of message units over the SSCP-PU session with PU b.

Legend:

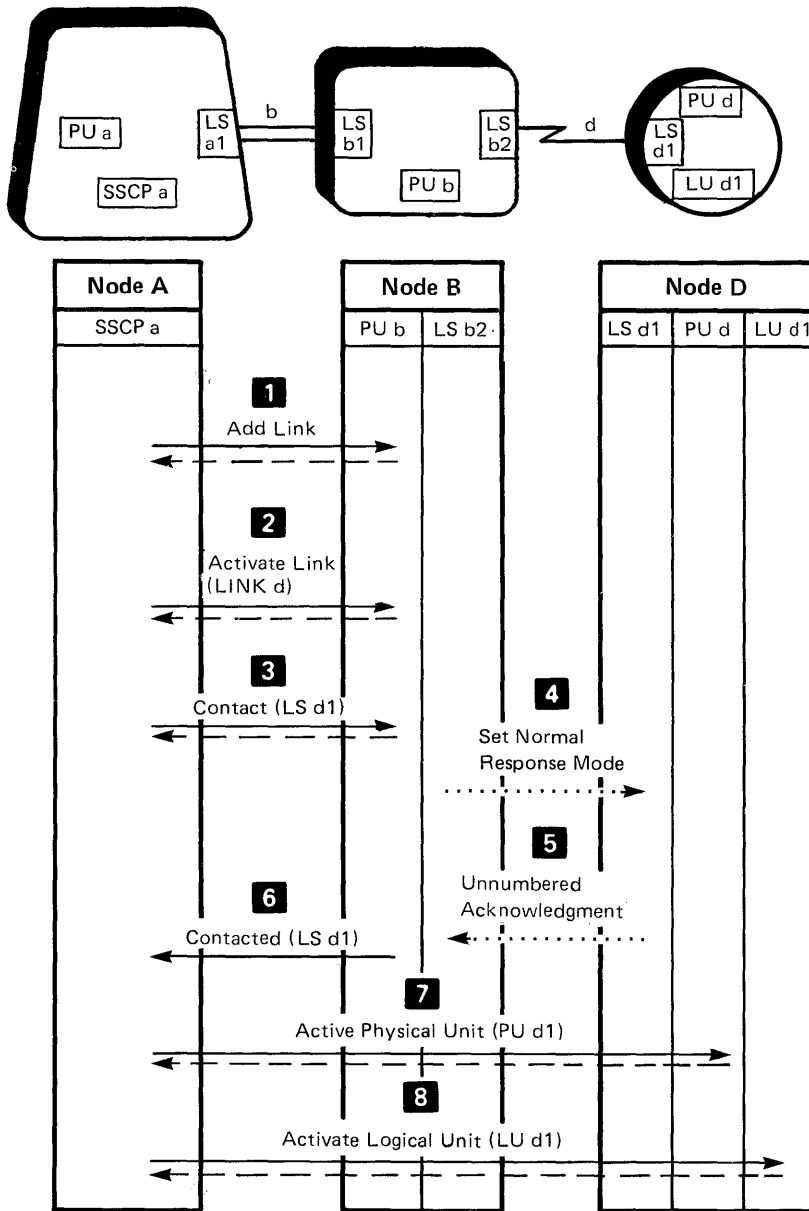


Type 5 Node  
Type 4 Node



LS = Link Station  
LU = Logical Unit  
PU = Physical Unit  
SSCP = System Services Control Point

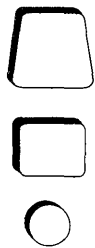
**Figure 27. Activating a Host Subarea Node, a Channel-Attached Subarea Node, and a Channel between Two Subarea Nodes**



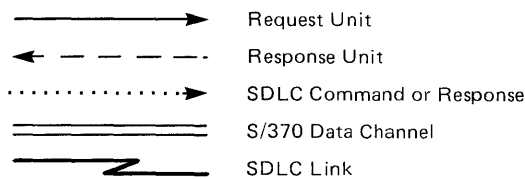
- 1** SSCP a requests PU b to furnish it with a network address for the designated link. PU b does so.
- 2** SSCP a tells PU b to activate LINK d and to prepare to issue and receive data link control commands and responses for the link.
- 3** SSCP a tells PU b to contact the adjacent link station LS d1 through link station LS b5.
 

The representation of a link station for a peripheral node has the same network address as the PU for that node, and this network address is one greater than that for the link.
- 4** PU b causes local link station LS d5 to initiate data link control procedures to contact link station LS d1. LS b5 tells LS d1 to go into normal response mode for information transfer.
- 5** LS d1 responds affirmatively to the LS b5 request.
- 6** PU b informs SSCP a that link station LS d1 is ready to receive and send message units.
- 7** SSCP a identifies itself to PU d and assumes control of Node D by activating an SSCP-PU session with PU d.
- 8** SSCP a identifies itself to LU d and assumes control of LU d by activating an SSCP-LU session with LU d.

**Legend:**



Type 5 Node  
Type 4 Node  
Type 2.1 Node



LS = Link Station  
LU = Logical Unit  
PU = Physical Unit  
SSCP = System Services Control Point

**Figure 28. Activating a Peripheral Node Attached to a Subarea Node by a Nonswitched SDLC Link**

## Controlling Network Activation

SSCPs control and manage network resources according to the parameters that you specify in your resource-definition statements. Network operators and network management programs can modify these statements, thereby changing the way an SSCP controls network resources. This section explains how an SSCP uses resource-definition statements, operator commands, and network management commands to activate, deactivate, and change the status of network resources.

### Cascaded Activation and Deactivation

The task of activating and deactivating a large network would be extremely time-consuming if a network operator had to enter separate commands for each resource, while having to remember the hierarchy of resource activation. To simplify network activation, you can specify that entire subareas, or portions of subareas, be automatically activated as the result of a single operator command. This procedure is called "cascaded" activation.

You can specify that most of the network be activated as the result of one operator command and that the remaining resources be activated by separate operator commands. What parameters you need to specify in the definition statements vary depending on the particular software programs that are in your network. You can also specify cascaded activation in the start parameters for a system generation.

Cascaded activation can begin or end at any point in the resource hierarchy. This allows you considerable flexibility when specifying parameters in your definition statements.

The cascaded deactivation capability allows a network operator to enter a single command to deactivate multiple resources. When reactivating network resources after part of the domain has failed or has been deactivated, the operator can either restore inactive resources to the status (1) they had prior to the failure or deactivation, or (2) that you defined when coding your resource-definition statements.

## Configuring and Reconfiguring a Network

You initially define a network configuration in the form of tables that are loaded into access methods and network control programs during a system generation. However, the configuration of the network is unlikely to remain exactly as you originally specified. Both scheduled and unscheduled changes are probable.

### Scheduled Changes

There are two ways to schedule network configuration changes. One way is to recode one or more of the tables that define the configuration to reflect configuration changes. This method requires a new system generation. System generations require that at least part of the network be deactivated for the amount of time necessary to load the revised information into the programs.

The other way to schedule configuration changes is through dynamic reconfiguration. Dynamic reconfiguration allows network operators to selectively add, move, or delete network resources in peripheral nodes without disrupting other network activity. The commands a network operator issues to dynamically reconfigure a network differ depending on the particular programs that are resident in the host subarea nodes. Typically, these network operator commands modify the original configuration data set (generated from your resource-definition statements during the last system generation) to reflect a new resource hierarchy. The SSCP uses the modified data set the next time it activates or deactivates network resources. A network operator can activate another data set to terminate the modifications and restore the original configuration data set.

Peripheral node resources that are connected to a subarea node by nonswitched links can be reconfigured through dynamic reconfiguration. The subarea node PU must be active, and the PUs and LUs in the peripheral nodes that are being reconfigured must be inactive.

### **Unscheduled Changes**

Unscheduled changes to a network configuration are inevitable. The hardware or software within a node may fail, or links between adjacent nodes may fail. Network operators can sometimes circumvent these failures by reconfiguring the network. But when an SSCP can no longer communicate with a type 4 node (a communication controller), the network operator cannot enter any commands to reconfigure the network.

When an SSCP can no longer communicate with a type 4 node, the network control program in the type 4 node begins a procedure called automatic network shutdown. In an orderly manner, automatic network shutdown deactivates some or all of the resources adjacent to that type 4 node. When the network control program regains its ability to communicate with an SSCP, the SSCP reactivates any resources that were affected by the automatic network shutdown procedure. How a network control program detects the loss of communication with an SSCP, and which resources it deactivates, depends upon the specific network control program that resides in the type 4 node.



### LU-LU Sessions

This chapter explains how LU-LU sessions are initiated, activated, and terminated. The chapter also discusses how LU-LU sessions support communication between two transaction programs.

### Contents

Initiating LU-LU Sessions	63
Session-Initiation Requests	63
Unformatted Requests	64
Formatted Requests	64
Initiating Cross-Domain LU-LU Sessions	64
Initiating Cross-Network LU-LU Sessions	66
Activating LU-LU Sessions	68
BIND Requests	68
Nonnegotiable BIND	68
Negotiable BIND	68
Notifying the SSCP	68
Half-Sessions	69
Transaction Programs	69
Document Interchange Architecture	70
SNA Distribution Services	70
Verbs	70
Conversations	71
Invoking Transaction Programs	71
Sync Points	71
Terminating an LU-LU Session	72
Secondary LU Requests Session Termination	72
Primary LU Requests Session Termination	72



## Initiating LU-LU Sessions

Once network resources are active, LU-LU sessions can be initiated. End users (application programs and individuals) gain access to an SNA network through logical units and exchange information over LU-LU sessions. The protocols that govern the communication between two logical units are explained in “Chapter 8. SNA Protocols.”

LU-LU sessions can be initiated in several ways:

- Either of the participating logical units can initiate an LU-LU session.
- A network operator can initiate an LU-LU session.
- In some cases, a third logical unit can initiate an LU-LU session between two other logical units.
- System definitions can specify that an LU-LU session be initiated automatically when certain resources become active.

Typically, one of the participating logical units initiates an LU-LU session. The two logical units that communicate with each other over this session are called session partners. This section discusses the session activation process when one of the participating logical units initiates the LU-LU session.

The process of activating an LU-LU session begins when a logical unit submits a session-initiation request to its control point.<sup>4</sup> Logical units must have an active session with their control point before they can request an LU-LU session.

A session-initiation request from a logical unit normally flows on the SSCP-LU session to the SSCP. But if the logical unit that is requesting the session resides in a type 2.1 node, the session-initiation request first flows on the PNCP-LU session to the PNCP. If the logical unit is requesting a session with a logical unit that is in an adjacent type 2.1 node, the PNCP performs the session-initiation process for the SSCP.

### Session-Initiation Requests

Session-initiation requests identify the two logical units that are to participate in the LU-LU session. To request a session with another logical unit, an LU sends a session-initiation request to its SSCP. The SSCP determines if (1) a path is available between the two logical units, (2) the logical units are authorized to communicate with each other, and (3) the two logical units can agree on a set of session protocols.

An SSCP accepts two kinds of session-initiation requests: unformatted requests and formatted requests. A PNCP accepts only formatted requests.

---

<sup>4</sup> Depending on the node in which it resides, a logical unit’s control point is either a system services control point (SSCP) or a peripheral node control point (PNCP).



## Unformatted Requests

An **unformatted** request contains a character string that specifies the desired session partner's network name. The SSCP calls an unformatted systems services (USS) routine to handle unformatted requests. The unformatted systems services routine uses the IBM-supplied USS definition table to convert character strings into formatted requests.

## Formatted Requests

A **formatted** request specifies the desired session partner's network name and a mode name. The SSCP calls a formatted systems services (FSS) routine to process formatted requests from a logical unit.

The formatted systems services routine uses the mode name in the formatted request as an entry into a mode table. A **mode table** (also called a BIND table) contains sets of session parameters for each LU in the network. The **mode name** in the formatted request identifies which set of session parameters in the mode table that the requesting logical unit can support.

The set of session parameters that the formatted systems services routine obtains from the mode table is called a **Bind image**. The SSCP transmits the Bind image within a Control Initiate (CINIT) request to the logical unit that is responsible for activating the LU-LU session.

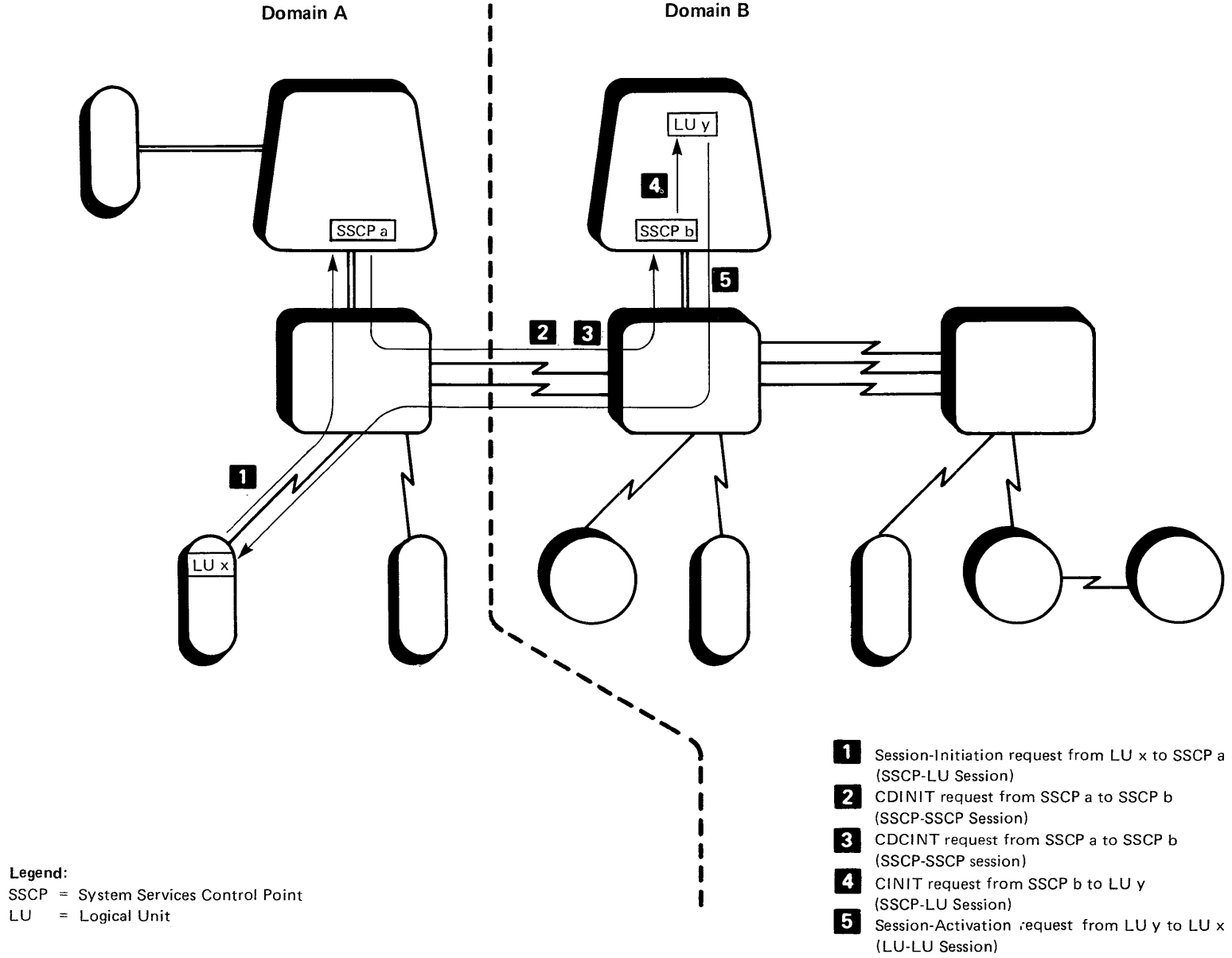
## Initiating Cross-Domain LU-LU Sessions

A **cross-domain LU-LU session** connects logical units in different domains. When the two logical units reside in different domains, each is under the control of a different SSCP. Because a different SSCP controls each of the logical units, an SSCP-SSCP session is necessary before either logical unit can activate the LU-LU session. The two SSCPs communicate over the SSCP-SSCP session to coordinate the cross-domain session initiation.

Figure 29 illustrates the sequence for initiating a cross-domain session between LU *x* and LU *y*. When SSCP *a* receives the session-initiation request from LU *x*, it checks its directory and determines that LU *y* resides in another domain, under the control of SSCP *b*. First, SSCP *a* sends a Cross-Domain Initiate (CDINIT) request to SSCP *b*. The CDINIT request identifies both the requesting LU (LU *x*) and the desired session partner (LU *y*).

Next, SSCP *a* sends a Cross-Domain Control Initiate (CDCINIT) request to SSCP *b*. The CDCINIT request passes information about LU *x* to SSCP *b* and requests that SSCP *b* send a Control Initiate (CINIT) to LU *y*. Neither LU *x* nor LU *y* is aware that the LU-LU session crosses domain boundaries.

Figure 29. Initiating a Cross-Domain LU-LU Session



## Initiating Cross-Network LU-LU Sessions

A **cross-network LU-LU session** connects logical units in different networks. Recall that a gateway consists of a gateway node and one or more gateway SSCPs. Cross-network sessions connect (1) a gateway SSCP in one network to an SSCP in an interconnected network (the second SSCP may or may not be a gateway SSCP) and (2) two logical units in interconnected networks.

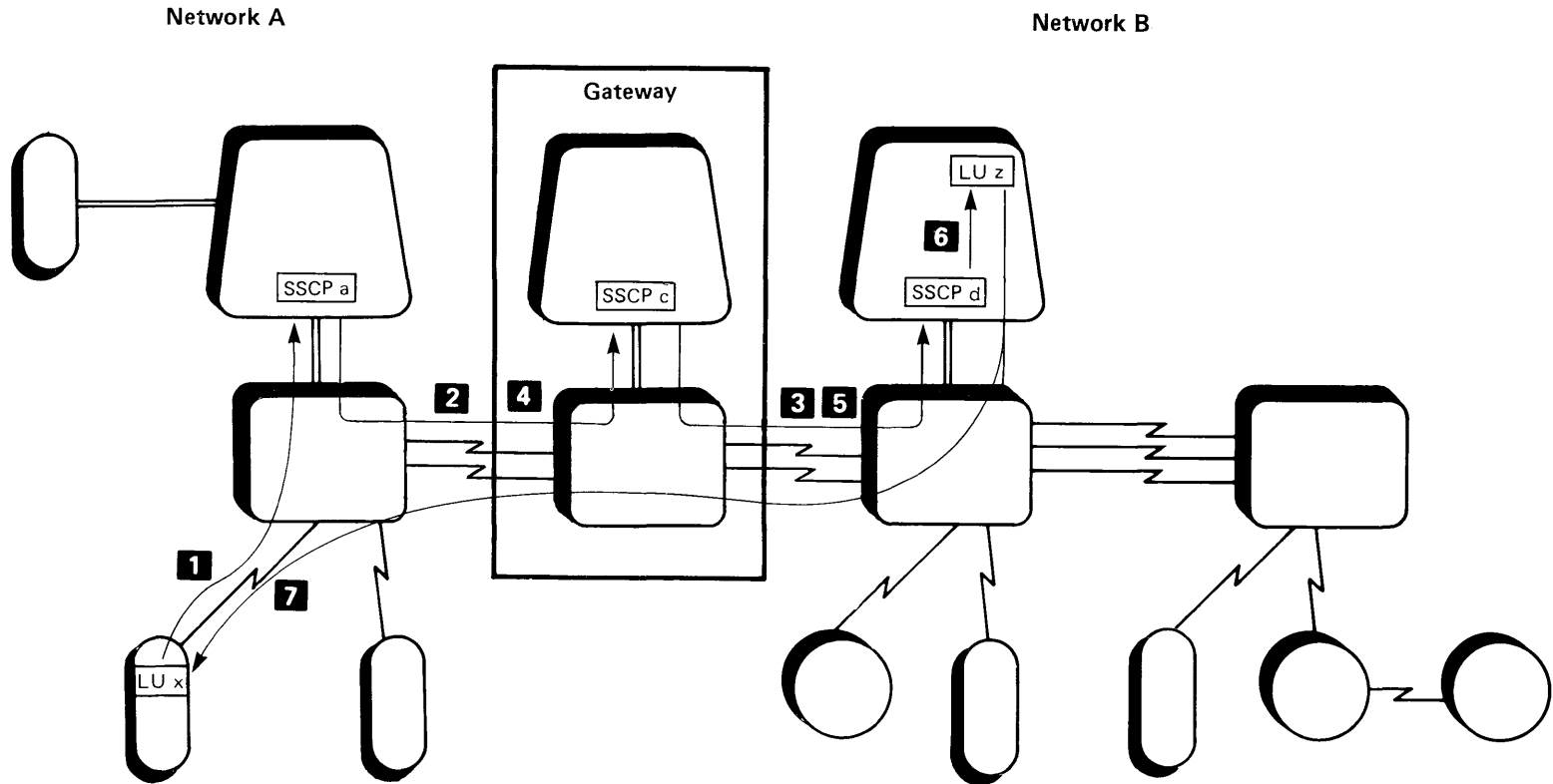
Figure 30 illustrates the sequence for initiating a cross-network session between LU *x* and LU *z*. When SSCP *a* receives the session-initiation request from LU *x*, it checks its directory and determines that LU *z* resides in another domain, under the control of SSCP *c*. First SSCP *a* sends a CDINIT request to SSCP *c*, naming LU *z* as LU *x*'s desired session partner. Then SSCP *a* sends a CDCINIT request to SSCP *c*.

Recall that an alias name identifies an LU in one network to an SSCP in an interconnected network. Both LU *x* and SSCP *a* are using an alias name to identify LU *z*. SSCP *a* is unaware that SSCP *c* is a gateway SSCP, or that LU *z* resides in a different network.

The gateway SSCP (SSCP *c*) translates the alias name of LU *z* to its real name in the interconnected network and translates the real name of LU *x* to an alias name. The gateway SSCP performs this translation before it sends the CDINIT and CDCINIT requests to SSCP *d*. These requests contain the real name of LU *z*, but an alias name for LU *x*.

The gateway SSCP has sessions with both SSCP *a* and SSCP *d*. Using these SSCP-SSCP sessions, it sends requests that it received from SSCP *a* to SSCP *d*. This transfer of requests between interconnected networks is called **SSCP rerouting**. To SSCP *a*, LU *z* appears to be in the domain of the gateway SSCP. To SSCP *d*, LU *x* appears to be in the domain of the gateway SSCP. Neither SSCP *a* nor SSCP *d* is aware that the gateway SSCP is rerouting their requests.

Figure 30. Initiating a Cross-Network LU-LU Session



**Legend:**

SSCP = System Services Control Point

LU = Logical Unit

- 1** Session-Initiation request from LU x to SSCP a (SSCP-LU Session)
- 2** CDINIT request from SSCP a to SSCP c (SSCP-SSCP Session)
- 3** CDINIT request from SSCP c to SSCP d (SSCP-SSCP Session)
- 4** CDCINIT request from SSCP a to SSCP c (SSCP-SSCP Session)
- 5** CDCINIT request from SSCP c to SSCP d (SSCP-SSCP Session)
- 6** CINIT request from SSCP d to LU z (SSCP-LU Session)
- 7** Session-Activation request from LU z to LU x (LU-LU Session)

## Activating LU-LU Sessions

Recall that the SSCP sends a Control Initiate (CINIT) request to the logical unit that is responsible for activating the LU-LU session. This CINIT request contains the set of session parameters that the requesting logical unit can support. The logical unit that is responsible for activating the LU-LU session is called the **primary LU**, and the logical unit that receives the session-activation request is called the **secondary LU**. The selected set of session parameters, the Bind image, becomes the content of the Bind Session (BIND) request that the primary LU uses to activate an LU-LU session.

### BIND Requests

A BIND request specifies, as parameters, the protocols that the primary and secondary LUs are to observe when communicating with each other over an LU-LU session. There are two kinds of BIND requests: nonnegotiable and negotiable.

#### Nonnegotiable BIND

The secondary LU can accept or reject the parameters in a nonnegotiable BIND request. If the secondary LU accepts the session parameters, it returns a positive response to the primary LU. The LU-LU session is then active, and both logical units will adhere to the parameters that the BIND request specified.

The secondary LU will reject the BIND request if the session parameters are unacceptable. The secondary LU rejects the BIND request by returning a negative response to the primary LU. The LU-LU session cannot be activated.

#### Negotiable BIND

The secondary LU can negotiate the session parameters in a negotiable BIND request. If the session parameters are unacceptable, the secondary LU returns a positive response with an alternate set of session parameters to the primary LU.

If the primary LU accepts the alternate set of session parameters, the LU-LU session becomes active. Both logical units will adhere to the alternate set of parameters. If the alternate set of session parameters is unacceptable to the primary LU, the LU-LU session cannot be activated.

Only type 4, type 6.1, and type 6.2 LUs can negotiate BIND parameters.

#### Notifying the SSCP

The primary LU sends a Session Started (SESSST) request to its SSCP when the LU-LU session becomes active. If the two logical units cannot agree on a set of session parameters, the primary LU sends a Bind Session Failure (BINDF) request instead. The BINDF indicates the reason that the LU-LU session was not activated.

## Half-Sessions

Some logical units can participate in more than one LU-LU session at the same time. A logical unit allocates a portion of its resources to support each LU-LU session. A **half-session** identifies that portion of its resources. Although not an architectural restriction, generally only logical units in type 5 nodes (host processors) support more than one LU-LU session at a time.

## Transaction Programs

A **transaction program** processes transactions in an SNA network. A transaction usually involves a specific set of initial input data that causes the execution of a specific task or job.

One example of a transaction is the entry of a customer's deposit that results in the updating of the customer's balance. A second example is the process of recording item sales, verifying checks before accepting them as tender, and receiving payment for the sold items. A third example is the transfer of a message to one or more destination points.

There are two kinds of transaction programs: application transaction programs and service transaction programs.

**Application transaction programs** are end users of an SNA network. They access the network through an LU type 6.2, and they process transactions for (1) service transaction programs and (2) other end users of the network.

**Service transaction programs** are IBM-supplied programs that the architecture defines. They are executed within an LU type 6.2, and they typically provide utility services to application transaction programs. For example, some service transaction programs provide document interchange services (using Document Interchange Architecture) that allow processors and workstations to exchange documents synchronously. Other service transaction programs provide SNA Distribution Services that allow asynchronous distribution of files and documents.

This section introduces Document Interchange Architecture and SNA Distribution Services and explains how transaction programs communicate with one another. In addition, this section explains how transaction programs invoke and coordinate error recovery with other transaction programs.

## Document Interchange Architecture

**Document Interchange Architecture (DIA)** is a program-to-program communication architecture that provides document interchange capabilities across a broad spectrum of IBM office systems. Specifically, DIA defines the protocols and data structures that enable programs to communicate processing intentions and interchange office system data in a network. DIA consists of an information interchange base that provides session services and various DIA application services. It invokes SNA Distribution Services to asynchronously interchange office system data in a network. For additional DIA information, refer to the *DIA Technical Reference*.

## SNA Distribution Services

**SNA Distribution Services (SNADS)** provides, at a user's request, the asynchronous movement and delivery of the user's material to other users. The entity that SNADS delivers (for example, a document) is called a distribution object, and the unit of work that SNADS provides is called a distribution. A distribution includes accepting a request, generating and moving copies of the distribution object across the network, and delivering the distribution object to the specified destinations.

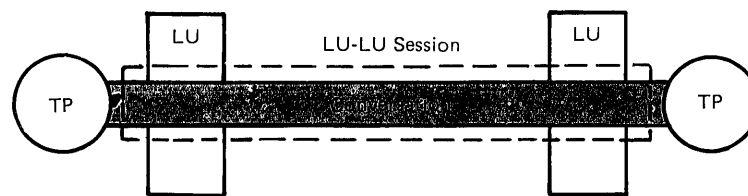
SNADS consists of a network of distribution service units that are interconnected by type 6.2 logical units. Each distribution service unit consists of a collection of service transaction programs within the LU type 6.2. For additional SNADS information, refer to the *SNA Format and Protocol Reference Manual: Distribution Services*.

## Verbs

SNA defines a protocol boundary for transaction programs that use LU type 6.2. This protocol boundary is specified by a set of **verbs** that are documented in the *SNA Transaction Programmer's Reference Manual for LU Type 6.2*. These verbs describe a generic application interface between transaction programs and the SNA network. Transaction programs issue verbs to communicate with other transaction programs.

## Conversations

Transaction programs communicate over a **conversation**. Conversations use LU-LU sessions between two type 6.2 logical units, as Figure 31 illustrates.



**Legend:**

TP = Transaction Program  
LU = Logical Unit

**Figure 31. A Conversation between Transaction Programs**

Conversations are a means of serially sharing a session from transaction to transaction. Once a conversation is allocated to a session, a send-receive relationship is established between the programs that are connected to the conversation. One program issues verbs to send data, and the other program issues verbs to receive the data. When the sending program finishes sending data, it transfers control to the other program.

## Invoking Transaction Programs

A transaction program requests a conversation with another transaction program by issuing an ALLOCATE verb that names the remote transaction program. Then the LU type 6.2 for the requesting transaction program sends the remote LU type 6.2 an Attach request to initiate the conversation. The two transaction programs are now connected by a conversation and can communicate with one another.

Refer to the *SNA Transaction Programmer's Reference Manual for LU Type 6.2* for additional information about transaction programs. Refer to the *SNA Format and Protocol Reference Manual: Architecture Logic for LU Type 6.2* for additional information about the LU type 6.2 support of transaction programs.

## Sync Points

SNA defines a **sync point** function that allows transaction programs that are processing a distributed transaction to coordinate error recovery. The sync point function protects both conversation resources and implementation-defined resources such as data bases. Any changes to these resources are logged so that they can be either backed out (reversed) if the transaction detects an error, or committed (made permanent) if the transaction is successful. A transaction program invokes the sync point function by issuing either the SYNCPT or BACKOUT verb. For additional information on the sync point function, refer to the *SNA Format and Protocol Reference Manual: Architecture Logic for LU Type 6.2* and the *SNA Transaction Programmer's Reference Manual for LU Type 6.2*.



## Terminating an LU-LU Session

An LU-LU session remains active until the end users communicating over the session are finished exchanging data. When the end users are finished communicating, one of the logical units will request that its SSCP assist in terminating the LU-LU session.

Either the primary or the secondary LU can deactivate sessions between two type 6.2 logical units. However, only the primary LU can deactivate sessions between other types of logical units. A logical unit sends an Unbind Session (UNBIND) request to its session partner to deactivate a session.

### Secondary LU Requests Session Termination

A secondary LU requests termination of an LU-LU session by sending a session-termination request to its SSCP over the SSCP-LU session. The LU sends a Terminate Self (TERM-SELF) request to ask for the SSCP's assistance in terminating the LU-LU session. The SSCP then sends a Control Terminate (CTERM) request to the primary LU over the SSCP-LU session. The CTERM notifies the primary LU that the secondary LU has requested session termination.

If the primary LU is not finished communicating with the secondary LU, it continues to communicate with the secondary LU over the LU-LU session. When the primary LU is finished communicating with the secondary LU, it sends the secondary LU an UNBIND request to deactivate the session.

### Primary LU Requests Session Termination

Typically, the secondary LU requests session termination. When the primary LU requests that an LU-LU session be terminated, it sends a Shutdown (SHUTD) request to the secondary LU. This request assures that the present work is completely processed before the session is terminated. When all outstanding work has been processed, the secondary LU returns a Shutdown Complete (SHUTC) request to the primary LU. Then the primary LU sends the secondary LU an UNBIND request to deactivate the session.

Type 6.2 logical units do not send SHUTD and SHUTC requests to request session termination. Instead, the primary LU sends a Bracket Initiation Stopped (BIS) request to the secondary LU. When all outstanding work has been processed, the secondary LU returns a BIS reply to the primary LU. Then the primary LU sends the secondary LU an UNBIND request to deactivate the session.

### Route Design

This chapter explains how the path control network uses the parameters you specify when designing routes to (1) select a path through the network and (2) assign priorities of data flow for different sessions.

### Contents

Designing Routes through the Network	75
Connecting Subarea Nodes	75
Parallel Links	75
Transmission Groups	76
Defining Paths	77
Explicit Routes	80
Routing Tables	82
Defining Virtual Routes	84
Transmission Priority	84
Assigning Session Traffic to a Virtual Route	84
Class of Service Tables	84
Selecting Virtual Routes	85
Activating and Deactivating Routes	86
Benefits of the SNA Routing Technique	87



## Designing Routes through the Network

The task of designing routes through a network ranges in difficulty from simple to complex. The degree of complexity depends on the number of subareas, connections between subareas, and types of user sessions that your network supports. When you design routes through a network, you will wish to achieve some combination of the following objectives:

- Maximize quantity of data transmitted
- Maximize data security
- Maximize route availability
- Minimize transmission time
- Minimize cost
- Minimize data loss or the necessity to retransmit data
- Minimize traffic congestion.

These objectives involve compromises. For example, you might minimize transmission times by assigning relatively little traffic to a route over a given period of time. But then you minimize transmission time over the route at the expense of the first objective listed above.

## Connecting Subarea Nodes

A single SDLC link or data channel always connects host subarea nodes to other subarea nodes. However, two or more SDLC links can connect adjacent communication controller subarea nodes.

### Parallel Links

**Parallel links** are multiple SDLC links that connect adjacent communication controller nodes while operating concurrently. Figure 32 shows three parallel links connecting adjacent communication controller nodes. The data traffic flowing between the nodes is distributed among the three parallel links. You determine the distribution of data traffic over parallel links by assigning the links to the same, or to different, transmission groups.

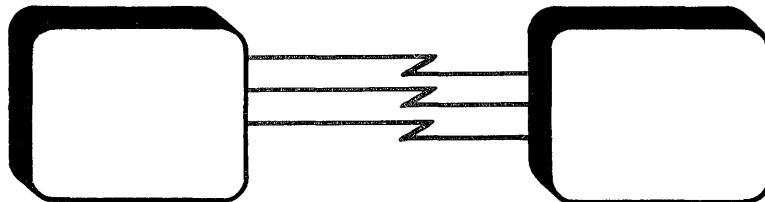


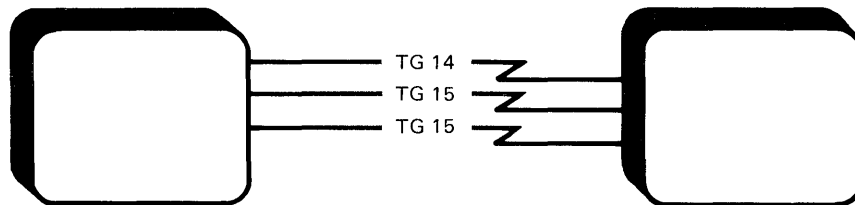
Figure 32. Parallel Links between Adjacent Communication Controller Nodes

## Transmission Groups

SNA requires that you assign each link that connects adjacent subarea nodes to a transmission group. A **transmission group** is a link or group of links that connect adjacent subarea nodes. Transmission groups appear as a single link to the path control network.

You can specify up to 16 transmission groups between adjacent communication controller nodes. You identify each transmission group by assigning the same number (called a transmission group number) to each link in the group. You can assign SDLC links to transmission group numbers 1 through 255. In actual implementations, you always assign System/370 data channels to transmission group number 1.

For example, you can group three parallel links into one, two, or three transmission groups. Figure 33 shows three parallel links grouped into two transmission groups: transmission group 14 and transmission group 15.



**Legend:**

TG = Transmission Group

**Figure 33. Transmission Groups**

A transmission group that consists of parallel links is more likely to be available than a transmission group that consists of a single link. If one of the links fails, data traffic will continue to flow on the remaining links in the group. This enables the network to maintain existing sessions without disruption.

You normally place links that have similar transmission characteristics in the same transmission group. For example, you would place the links that have a high transmission speed in one transmission group and links that have a medium transmission speed in another transmission group.

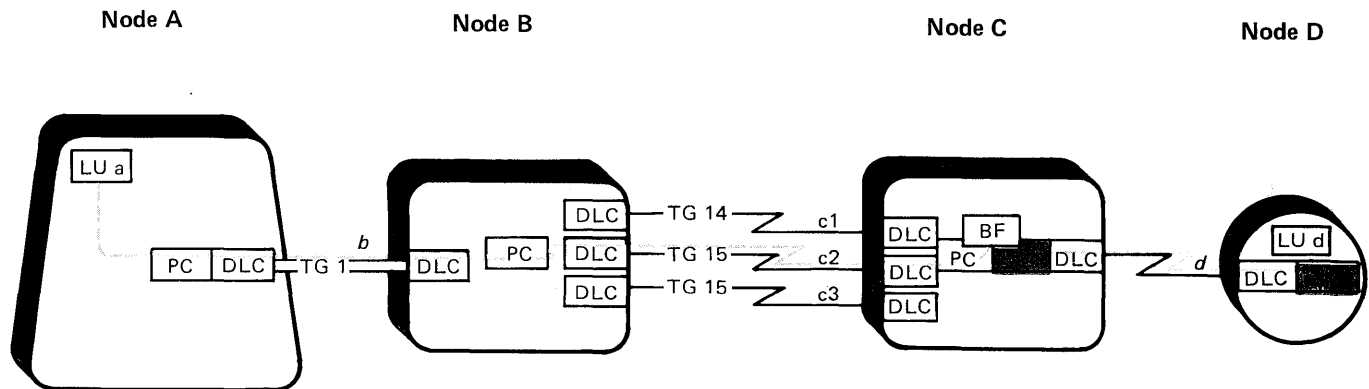
## Defining Paths

The process of designing routes involves defining one or more paths between each pair of network addressable units that need to communicate with one another. A **path** consists of a series of path control elements, data link control elements (the link stations), and link connections.

For example, consider the path of a message unit that is sent from an LU in a host subarea node to an LU in a peripheral node. Figure 34 shows a path between the two LUs.

The path shown:

- Proceeds from LU *a* in node A,
- Over link *b* (transmission group 1),
- Through node B,
- Over link *c2* or *c3* (transmission group 15),
- Through node C,
- Over link *d*,
- To LU *d* in node D.



### Legend:


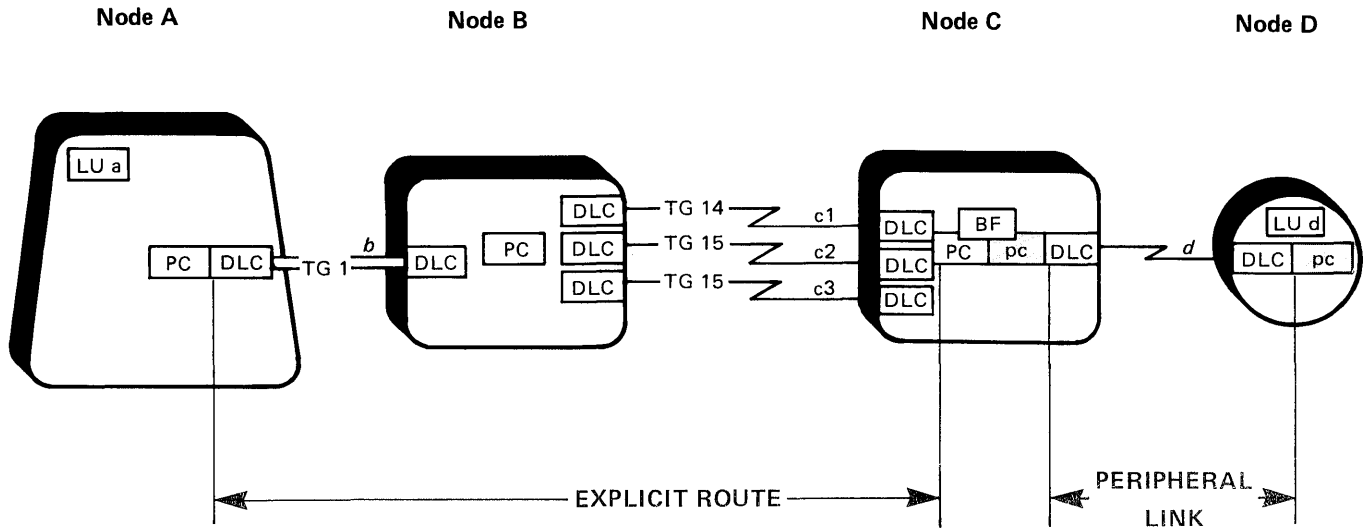
- LU = Logical Unit
- DLC = Data Link Control
- PC = Subarea Path Control Element
-  = Peripheral Path Control Element
- BF = Boundary Function
- TG = Transmission Group

Figure 34. A Path between Logical Units

This page intentionally left blank.

To define a path in an SNA network, you specify an explicit route and, if required, a peripheral link. An explicit route is the portion of the path between two subarea nodes, and a peripheral link is the portion of the path between a subarea node and a peripheral node. Figure 35 illustrates an explicit route and a peripheral link for the path between LU a and LU d.



- Legend:**
- LU = Logical Unit
  - DLC = Data Link Control
  - PC = Subarea Path Control Element
  - pc = Peripheral Path Control Element
  - BF = Boundary Function
  - TG = Transmission Group

**Figure 35. An Explicit Route and a Peripheral Link**



## Explicit Routes

SNA defines an **explicit route** as an ordered set of subarea nodes and transmission groups along a path. You can define up to sixteen explicit routes between any two subarea nodes; you assign an explicit route number to each of these routes.

Different explicit routes can include the same subarea nodes or transmission groups. For example, consider the path in Figure 36. You can define two explicit routes between subarea nodes A and C.

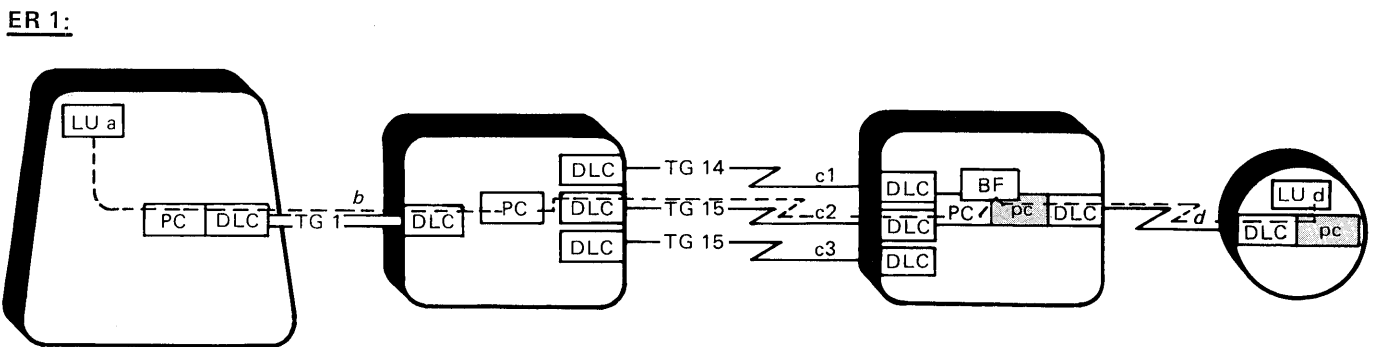
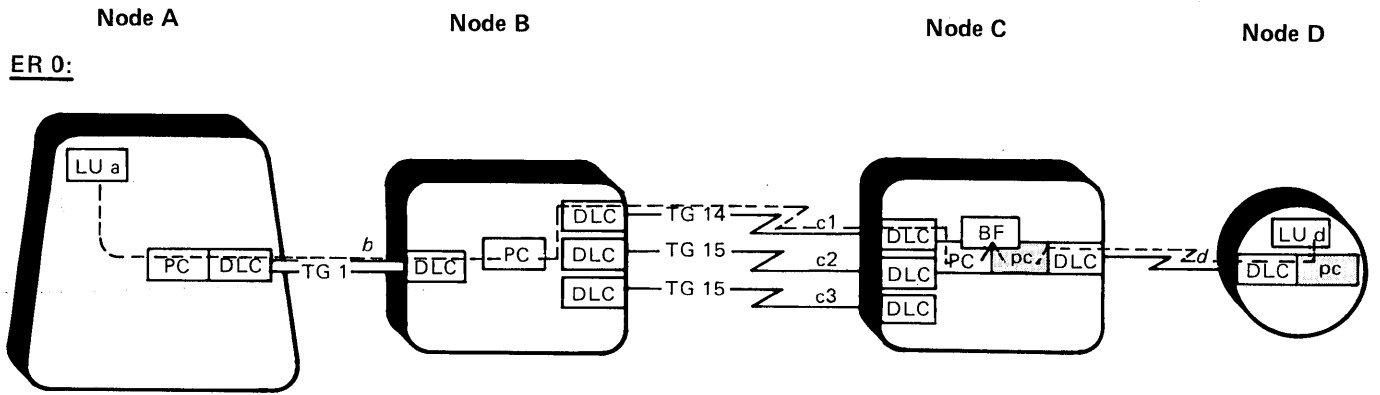
You could define explicit route 0 as:

1. Subarea node A
2. Transmission group 1
3. Subarea node B
4. Transmission group 14
5. Subarea node C.

You could define explicit route 1 as:

1. Subarea node A
2. Transmission group 1
3. Subarea node B
4. Transmission group 15
5. Subarea node C.

Explicit routes 0 and 1 include the same subarea nodes. Both explicit routes also include transmission group 1 between subarea nodes A and B. The only difference between these two explicit routes is the transmission group between subarea nodes B and C. You define more than one explicit route between subarea nodes to increase the probability that a path will be available between the two nodes, just as you define a transmission group to have multiple, parallel links.



- Legend:**
- LU = Logical Unit
  - DLC = Data Link Control
  - PC = Subarea Path Control Element
  - pc = Peripheral Path Control Element
  - BF = Boundary Function
  - TG = Transmission Group
  - ER = Explicit Route

**Figure 36. Multiple Explicit Routes**

Explicit routes are bidirectional. You can assign explicit route numbers 0 through 7 to explicit routes in the forward direction, and explicit route numbers 0 through 7 in the reverse direction. Explicit routes in the forward and reverse directions must use the same set of subarea nodes and transmission groups; however, you do not have to assign the same explicit route number to the forward and reverse directions.

## Routing Tables

Instead of defining the entire path to each node along the path, you distribute path definitions among all the nodes along the path. This simplifies system definition and saves storage space in the individual nodes. Therefore, each node contains only a part of the path specification.

The path control network routes data on a node-by-node basis. You define for each node only the information that path control needs to route data to the next subarea node along the path. You define this information in the form of routing tables. Path control in each node uses the explicit routes you define in its routing table to determine where to route the data next.

For example, consider the explicit routes (having numbers 0 and 1 in the forward direction, and numbers 2 and 1 in the reverse direction) in Figure 37. The figure shows the *simplified* routing table segments for each node that would be associated with the explicit routes.

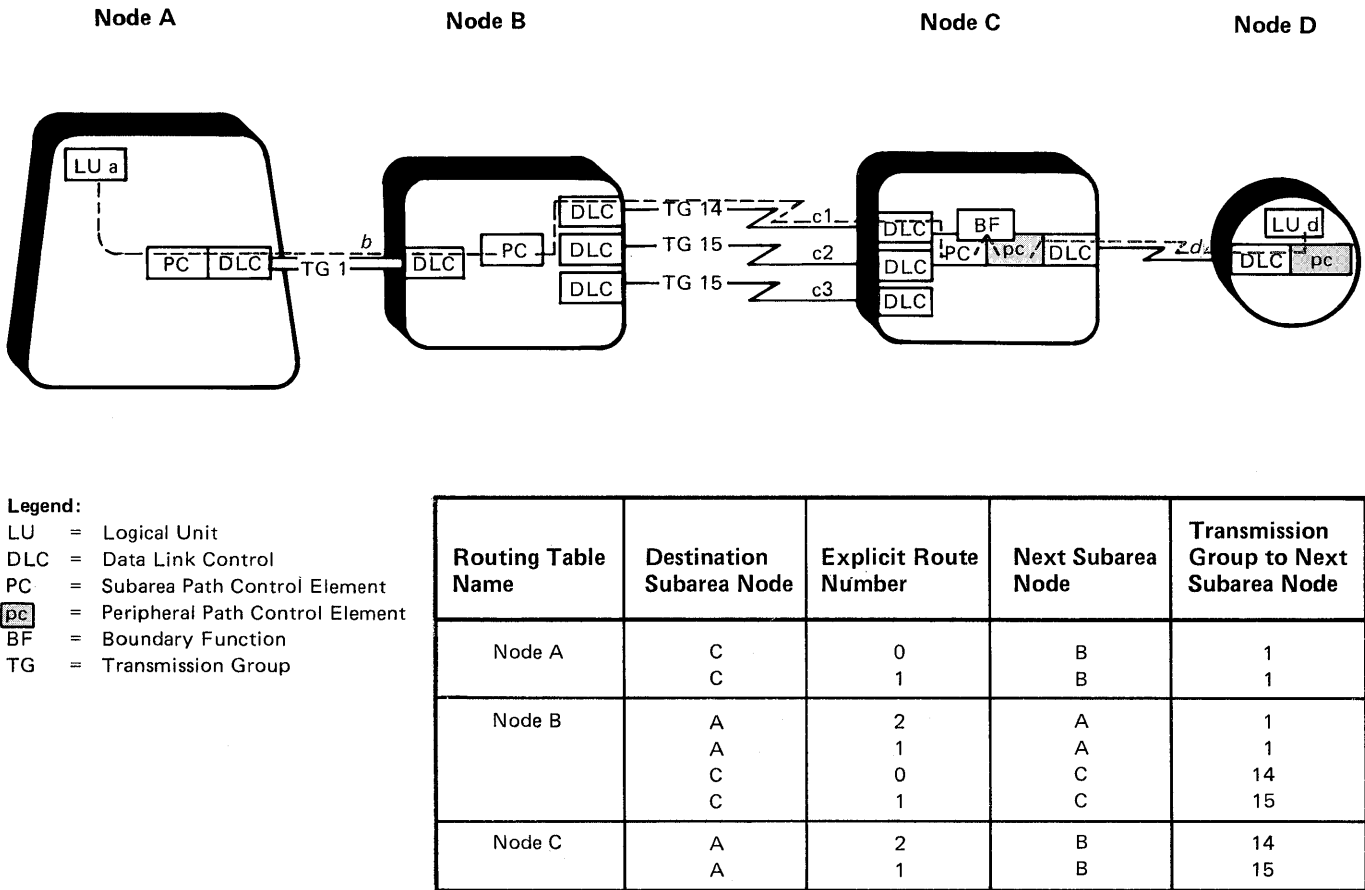


Figure 37. Routing Table Segments for Two Explicit Routes

Using the routing table segments shown in Figure 37, the routing of a message unit from LU *a* to LU *d*, over explicit route 0, would be:

#### **Node A**

1. Subarea path control uses the destination network address that the message unit contains to identify node C as the destination subarea node.
2. The node A routing table indicates that transmission group 1 should be used to route the message unit to the next subarea node along the path.

Path control gives the message unit to data link control for transmission over transmission group 1.

#### **Node B**

1. Data link control gives the message unit that it received over transmission group 1 to its subarea path control.
2. The node B routing table indicates that transmission group 14 should be used to route the message unit to the next subarea node along the path.

Subarea path control gives the message unit to data link control for transmission over transmission group 14.

#### **Node C**

1. Data link control gives the message unit that it received over transmission group 14 to its subarea path control.
2. Subarea path control uses the destination subarea address to confirm that it is the destination subarea node along the path.

Subarea path control uses the boundary function component to pair the destination network address with the local address of LU *d* in node D.

3. Subarea path control routes the message unit to the peripheral path control element in its node.
4. Using a supplemental table called an element routing table, the peripheral path control element determines that data link control should transmit the message unit over peripheral link *d* to node D.

Note that no single node knows about all of the subarea nodes and transmission groups that either explicit route 0 or 1 defines. All that a particular node knows about each explicit route in a given direction is the destination subarea node, the next subarea node along the route, and the transmission group that it uses to reach that node.

## Defining Virtual Routes

Whereas an explicit route is a physical connection between two subarea nodes, a **virtual route** is a logical connection between two subarea nodes. Each virtual route maps to an explicit route. The portion of a path between a subarea node and a peripheral node is called a **route extension**.

You define one or more virtual routes to each explicit route in the network and assign a transmission priority for data traffic that uses each virtual route. Virtual routes take on the physical characteristics (bandwidth and transmission rates) of the explicit routes that you assign them to.

You can assign up to eight virtual route numbers, each with three different transmission priorities, between two subarea nodes. The three transmission priorities, when combined with the eight possible virtual route numbers, allow you to define up to 24 virtual routes between two subareas.

### Transmission Priority

A **transmission priority** identifies the priority of message units flowing over an explicit route. You specify one of three levels of transmission priority for each virtual route: 0 (lowest), 1, or 2 (highest).

The path control network queues message units before transmitting them over a transmission group. It transmits message units that have a high transmission priority ahead of those that have a medium or low transmission priority. Subarea nodes provide an aging algorithm that periodically reorders the transmission priority of message units in a queue to assure that low priority message units do not remain in queue for an extended period of time.

## Assigning Session Traffic to a Virtual Route

When end users request an LU-LU session, they request a class of service. A **class of service (COS)** specifies a set of performance characteristics for routing data between two subareas. End users request a given class of service either directly, as a class of service name, or indirectly, by a mode name. Logical units specify the requested class of service in the session-initiation request. The SSCP uses this class of service as an entry into the class of service table.

### Class of Service Tables

You specify different classes of services in a class of service (COS) table, based upon the needs of the end users in your network. A **COS table** is a list of virtual route/transmission priority pairs. An SSCP uses the COS table to identify a particular virtual route to the path control network. All data for a given session uses the same virtual route and, therefore, is assigned the same transmission priority.

For example, the following classes of service may exist in your network:

- A class that provides response times suitable for high-priority interactive sessions
- A class that provides response times suitable for low-priority interactive sessions
- A class that provides routes that have the best availability
- A class suitable for batch processing
- A class suitable for high-security transmissions.

After you name the classes of service that your network will provide, you use these names to label entries into your COS table. Within each entry, you list all the virtual route/transmission priority pairs that you want to assign to the named class of service. The order in which you list these pairs determines the order in which the path control network selects a route for session traffic.

Not all session traffic is LU-LU session traffic. SSCPs communicate with physical units over SSCP-PU sessions, with logical units over SSCP-LU sessions, and with other SSCPs over SSCP-SSCP sessions. You must, therefore, provide at least one COS table entry for SSCP-PU, SSCP-LU, and SSCP-SSCP session traffic.

## Selecting Virtual Routes

Select the virtual routes whose characteristics best match the requirements of each session. In making the selection, you need to consider both the physical characteristics of the explicit route that is to be used and the transmission priority that you assigned to the virtual route. Some explicit routes, and therefore the virtual routes that use them, may be better than others.

For example, in your COS table you could list virtual routes assigned to explicit routes that have fewer physical elements before you list virtual routes that are assigned to longer explicit routes. Then the path control network would select the longer routes only for backup purposes when the shorter routes become inoperative. As another example, you could list virtual routes assigned to explicit routes that include multiple-link transmission groups ahead of those virtual routes assigned to explicit routes that include only single-link transmission groups.

End users require sessions with widely differing data transmission requirements. Inquiry-response sessions usually require faster data transmission and more predictable response times than data-collection sessions. Because sessions for several different kinds of applications may be in progress over a given route, you should assign priorities on a session-by-session basis. You should also assign LU-LU sessions that require rapid response times a higher transmission priority than LU-LU sessions where slower data flow is acceptable.

If none of the virtual routes in the COS table for the named class of service is active, the SSCP attempts to activate an explicit route for one of the virtual routes. If the SSCP is unable to activate an explicit route, it informs the LU that no LU-LU session can be activated. The LU requesting the session must then resubmit the session-activation request at a later time. If the LU requesting the session resides in a host node, the SSCP will notify the LU when a virtual route that can provide the requested class of service becomes available.

You can control the distribution of LU-LU session traffic, within the same class of service, across the virtual routes. You control the distribution of LU-LU session traffic across virtual routes by writing an access method exit routine that modifies the ordering of virtual route/transmission priority pairs within a COS table each time an LU-LU session is activated.

## Activating and Deactivating Routes

Access methods and network control programs in subarea nodes activate and deactivate explicit and virtual routes as needed. The network operator never enters commands to either activate or deactivate routes. Routes are automatically activated as required for session traffic flow.

To activate an explicit route, path control in the originating subarea sends an Explicit Route Activate (NC-ER-ACT) request to path control in the destination subarea node. Upon receipt of the NC-ER-ACT, path control in the destination subarea node determines the length of the explicit route by totaling the number of transmission groups in the path. Then it verifies that the route:

- Is usable
- Connects the origin and destination subarea nodes
- Is reversible
- Does not pass through any node more than once.

Explicit routes must be active before an SSCP can assign a session to a virtual route. Path control activates a virtual route automatically when a session requires a path and deactivates the route when the last session using it is deactivated. Path control deactivates explicit routes automatically when a PU or link along the path is deactivated or becomes inoperative.

## Benefits of the SNA Routing Technique

The beginning of this chapter listed some of the objectives you that consider when designing routes through a network. SNA helps you meet those objectives in the following ways:

- Rather than specifying the entire sequence of links along a path in each subarea node, SNA distributes routing information among all of the nodes along the path. Therefore, each node contains only a portion of the network's path specifications. This technique saves storage space in the individual nodes and helps users to reconfigure the network.
- SNA avoids the rigidity of always assigning sessions to the same route. An SSCP assigns sessions to virtual routes during session activation in accordance with session requirements.
- SNA helps you minimize the transmission time of data through the network by allowing parallel links in transmission groups, multiple explicit routes between subareas, and transmission priorities for virtual routes.
- SNA provides parallel links and multiple explicit routes that allow you to maximize the number of messages exchanged between two subareas in a given period of time.
- SNA allows you to increase the availability of paths through the use of parallel links in a transmission group and multiple explicit routes between subarea nodes.





### Pacing

This chapter introduces flow control algorithms and explains how these algorithms use pacing to control the flow of data through the network.

### Contents

Flow Control Algorithms	91
Global Algorithms	91
Local Algorithms	91
Pacing	91
Session-Level Pacing	92
One-Stage Pacing	93
Two-Stage Pacing	93
Virtual-Route Pacing	94
Summary	95



## Flow Control Algorithms

The number of end users sending and receiving data through a network changes over time. In a changing data processing environment, the amount of data flowing through a network often varies. You can plan for normal peak traffic periods, but unexpected traffic demands (from unanticipated end-user demand and network component failures) cause disruptions in data flow through a network.

The distribution of traffic in various parts of a network and the total traffic in a network both vary. When a link fails, the traffic that it normally carries must be diverted over other links. An already active link may not be able to carry the additional traffic load. Whenever the rate at which data enters the network exceeds the network's transmission capacity, congestion results.

When congestion results, response times lengthen and buffer resources are depleted. Severe or prolonged congestion in one part of the network affects other parts of the network, decreasing the network's overall efficiency. You can avoid network congestion by employing flow control mechanisms at both the global (network) and local (node) level.

### Global Algorithms

Global flow control algorithms provide a coordinated, network-wide mechanism to prevent congestion. They allow you to distribute traffic evenly among subarea nodes and links to avoid some nodes becoming overloaded relative to others. The purpose of global flow control is to synchronize the rate at which data enters the network with the rate at which traffic can be transmitted, thus protecting your network resources from traffic overload.

### Local Algorithms

Local flow control algorithms operate within individual nodes. They manage the network traffic through each node separately and provide feedback control to assist the global flow control algorithms. Local algorithms temporarily prevent end users from introducing more data into a congested network.

## Pacing

Both global and local algorithms implement flow control through pacing. **Pacing** controls the flow of data through a network.

You need to control the flow of data through a network because different nodes have different storage capacities for receiving and holding message units, and some nodes are capable of processing a greater amount of data than others. For example, a logical unit in a host subarea node may transmit data to a logical unit in a peripheral node faster than the peripheral logical unit can accept and process the data. Buffers in the communication controller subarea nodes that are adjacent to peripheral nodes will begin to fill up because requests cannot be

delivered to the peripheral logical unit as quickly as they are received from the host subarea node. This is one cause of congestion in a network.

SNA defines two kinds of pacing to reduce network congestion: session-level pacing and virtual-route pacing. **Session-level pacing** moderates traffic between end users to prevent an overrun of logical unit buffers. **Virtual-route pacing** synchronizes the rate of data flow over a virtual route to prevent buffer depletion in subarea nodes.

Both session-level and virtual-route pacing are based on pacing window techniques. A **pacing window** is the group of message units that a network component can send to another network component at one time. The **pacing window size** is the number of message units in the pacing window. Pacing permits only a certain number of sequential messages to be outstanding in the network between the origin and destination network components before an end-to-end acknowledgment is received. This acknowledgment, called a **pacing response**, indicates that the destination network component is ready to accept an additional pacing window.

In SNA, the first message unit in a pacing window contains a pacing request. Sometime after receiving a pacing request, the destination network component sends back a pacing response. A pacing response is a positive acknowledgment that indicates the destination's ability to receive an additional pacing window. The destination withholds the pacing response until it is ready to receive an additional pacing window.

## Session-Level Pacing

Session-level pacing allows a receiving LU to control the rate at which it receives requests. The purpose of session-level pacing is to prevent one half-session from sending data faster than the receiving half-session can process it.

Two communicating LUs frequently have an inherent saturation rate; that is, incoming traffic exceeds a node's processing and buffer storage capacity. For this reason, each sender has a maximum number of message units that can be sent prior to receiving a go-ahead pacing response from the receiver. The maximum number of message units that can be sent at one time, specified at session initiation, is called the window size.

When a receiver returns a pacing response after receiving the first message unit of a window, it indicates that it is ready to accept an additional window. If the receiver delays returning the pacing response, it indicates that it is not ready to accept the next window. Upon receiving a pacing response, a sender is allowed to send the next pacing window.

An LU-LU session can be paced in both directions. The pacing of requests flowing toward an LU is called **inbound pacing** for that LU. Similarly, the pacing of requests flowing away from an LU is called **outbound pacing**. The window sizes do not have to be the same for the two directions.

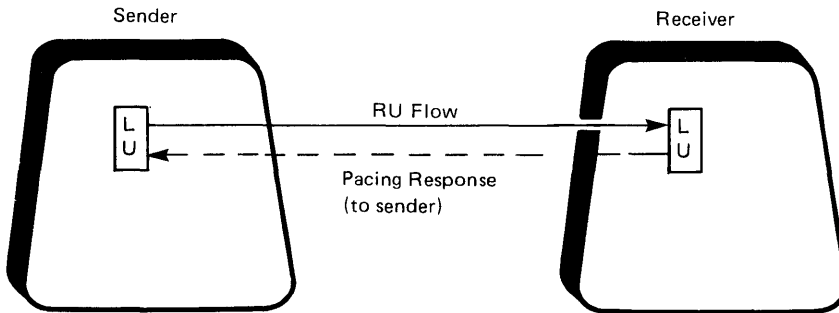
When a boundary function exists in the path, session-level pacing can have up to two stages in each direction. Figure 38 illustrates the stages of pacing.

### One-Stage Pacing

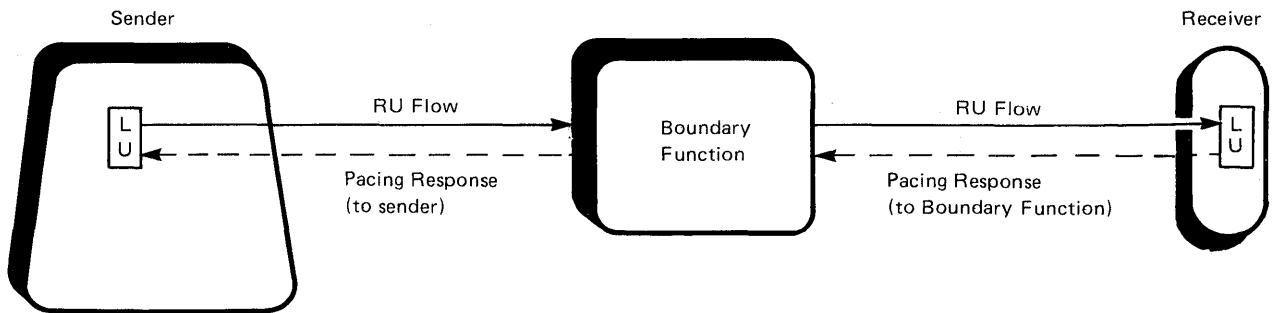
In one-stage pacing, the pacing occurs directly between the two LUs communicating over the session. For example, one-stage pacing occurs when two subarea LUs have a session with one another.

### Two-Stage Pacing

In two-stage pacing, the pacing occurs between one of the LUs and a boundary function, then between the boundary function and the other LU. Two-stage pacing occurs when a subarea LU has a session with a peripheral LU.



(a) One-stage Pacing



(b) Two-stage Pacing

**Legend:**

LU = Logical Unit

**Figure 38. Session-Level Pacing**

## Virtual-Route Pacing

Virtual-route pacing synchronizes the rate of data flow between the two end subareas of a virtual route; it monitors and controls the flow of data over a virtual route. On a network-wide basis, subarea nodes continually monitor the amount of congestion in the network. If congestion occurs, these nodes limit the amount of data they send over virtual routes. Virtual-route pacing is automatic, requiring no action by end users or network operators.

Virtual routes between one network and a gateway node are paced independently of virtual routes between the gateway node and an interconnected network. Gateway nodes, like other subarea nodes, use virtual-route pacing to limit the amount of data they send over congested virtual routes.

Activation of a virtual route includes initializing the pacing window to indicate the number of message units that a sender may transmit over a virtual route either initially, or as a result of a returned pacing response from the receiver. The pacing window size for a virtual route fluctuates between a minimum and maximum value depending on the severity of network congestion. Subarea nodes determine the severity of network congestion by totaling the number of messages that are in queue.

SNA defines two degrees of congestion. During the first, less severe level, the number of message units sent in each window is decreased by 1 until congestion clears. The sending subarea node can continue to send message units until the pacing count reaches 0. Then the subarea node must wait until it receives a pacing response before sending any additional message units along the virtual route.

During the second, more severe level of congestion, the window size is immediately decreased to its minimum value. The receiving subarea node withholds a pacing response until congestion along the virtual route lessens. The sending subarea node must avoid severely depleting its buffers by accepting too many message units that need to be sent over the congested virtual route. You can limit the number of additional message units the sending subarea will receive, and therefore control the amount of buffer depletion, by specifying congestion parameters for access methods and network control programs.

You can specify minimum and maximum window values when you specify flow-control-threshold parameters for system generation, or by writing an access-method exit routine. The minimum window size default is equal to the number of transmission groups that make up the explicit route that underlies the virtual route. The maximum window size default is equal to three times the minimum window value.

For detailed information on pacing protocols, refer to either the *SNA Format and Protocol Reference Manual: Architectural Logic* or the *SNA Format and Protocol Reference Manual: Architecture Logic for LU Type 6.2*.

## Summary

The network initially transmits pacing windows with the maximum pacing window values. You specify these values in the flow-control-threshold parameters. If network congestion results, pacing window sizes decrease. Pacing moderates the data exchanged between two logical units, prevents buffer overruns, and controls congestion within the network.





### Data Formats

This chapter identifies the different kinds of message units that flow through an SNA network.

### Contents

Requests and Responses	99
Message Unit Formats	99
Basic Information Unit	100
Request Header	100
Request Unit	100
Response Header	100
Response Unit	101
Path Information Unit	101
Transmission Header Formats	102
Basic Link Unit	102



## Requests and Responses

Message units flowing through the network contain either a request or a response. **Requests** are message units that contain (1) end-user data or (2) network commands. **Responses** are message units that acknowledge the receipt of a request.

Requests that contain end-user data are **data requests**. Examples of end-user data include payroll data, personnel data, insurance policy data, and inventory data. Requests that contain network commands are **command requests**. Network commands establish sessions, terminate sessions, and control communication between network addressable units (NAUs).

Responses are either positive or negative. **Positive responses** indicate that a request was received and is acceptable. **Negative responses** indicate that a request was received, but is unacceptable. Negative responses contain error codes that explain why the request is unacceptable.

## Message Unit Formats

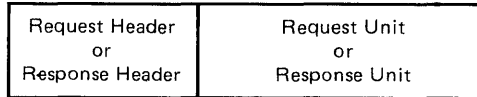
Network addressable units, path control elements, and data link control elements all use different message-unit formats to exchange information with other network addressable units, path control elements, and data link control elements in the network. This section explains the different message-unit formats that network resources use to exchange information and the type of information that each message unit contains.

SNA defines the following message-unit formats for NAUs, path control elements, and data link control elements to use:

- Network addressable units use basic information units (BIUs)
- Path control elements use path information units (PIUs)
- Data link control elements use basic link units (BLUs).

## Basic Information Unit

Network addressable units use **basic information units (BIUs)** to exchange requests and responses with other network addressable units. Figure 39 shows the format of a BIU.



**Figure 39. Basic Information Unit (BIU) Format**

Basic information units that carry requests contain both a request header and a request unit. Basic information units that carry responses consist of (1) both a response header and a response unit or (2) only a response header.

### Request Header

Each request that an NAU sends begins with a **request header (RH)**. A request header is a 3-byte field that identifies the type of data in the associated request unit. The request header also provides information about the format of the data and specifies protocols for the session.

### Request Unit

Each request that an NAU sends also contains a **request unit (RU)**. A request unit is a field of variable length that contains either end-user data (data RUs) or an SNA command (command RUs). Data RUs contain information to be exchanged between end users. Command RUs control the operation of the network.

### Response Header

Each response that an NAU sends includes a **response header (RH)**. Like a request header, a response header is a 3-byte field that identifies the type of data in the associated response unit. A bit called the request/response indicator (RRI) distinguishes a response header from a request header.

The receiving NAU indicates whether the response being returned to the request sender is positive or negative by setting a single bit, called the response type indicator (RTI), in the response header.

## Response Unit

A **response unit (RU)** contains information about the request. Positive responses to command requests generally contain a one-to-three byte response unit that identifies the command request. Positive responses to data requests contain response headers, but no response unit. Negative response units are four-to-seven bytes long and are always returned with a negative response. Response units are identified as response RUs.

The receiving NAU would return a negative response to the request sender if:

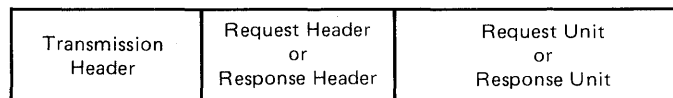
- The sender violates an SNA protocol
- The receiver does not understand the transmission
- An unusual condition, such as a path outage, occurs.

The receiving NAU returns a four-to-seven byte negative response unit to inform the request sender why a request is unacceptable. The first four bytes of the response unit contain sense data that explains why the request is unacceptable. The receiving NAU sends up to three additional bytes that identify the rejected request.

For additional information on basic information units, refer to the *SNA Format and Protocol Reference Manual: Architectural Logic*.

## Path Information Unit

The message-unit format used by NAUs is a basic information unit; the message-unit format used by path control elements is a **path information unit (PIU)**. Path control elements form a PIU by adding a transmission header to a basic information unit. Figure 40 shows the format of a PIU.



**Figure 40. Path Information Unit (PIU) Format**

Path control uses the **transmission header (TH)** to route message units through the network. The transmission header contains information such as the addresses of the origin and destination NAUs, an explicit route number, and a virtual route number.

## Transmission Header Formats

SNA defines different transmission header formats and identifies the different formats by a format identification (FID) type. Transmission headers vary in length according to their FID type. Path control uses the different FID types to route data between different types of nodes.<sup>5</sup>

**FID 0:** Path control uses this format to route data between adjacent subarea nodes for non-SNA devices. Few networks still use the FID 0; now a bit is set in the FID 4 transmission header to indicate whether the device is an SNA device or a non-SNA device.

**FID 1:** Path control uses this format to route data between adjacent subarea nodes if one or both of the subarea nodes do not support explicit and virtual route protocols.

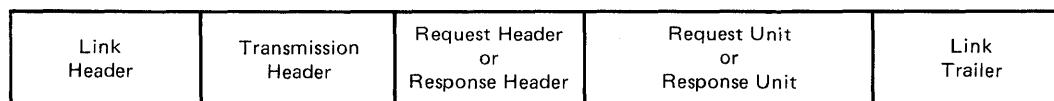
**FID 2:** Path control uses this format to route data between a subarea node and an adjacent type 2.0 or type 2.1 peripheral node. Path control also uses this format to route data between directly-connected type 2.1 nodes.

**FID 4:** Path control uses this format to route data between adjacent subarea nodes if both of the subarea nodes support explicit and virtual route protocols.

For additional information on path information units, refer to the *SNA Format and Protocol Reference Manual: Architectural Logic* or the *SNA Reference Summary*.

## Basic Link Unit

Data link control uses a message-unit format called a **basic link unit (BLU)** to transmit data across a link. Data link control forms a BLU by adding a **link header (LH)** and a **link trailer (LT)** to a path information unit. For an explanation of SDLC link headers and link trailers, refer to the *IBM SDLC General Information* manual. Figure 41 shows the format of a BLU.



**Figure 41. Basic Link Unit (BLU) Format**

Figure 42 reviews the format of the different message units and shows how NAUs, path control, and data link control use these different formats to route a request from end user D to end user A.

---

<sup>5</sup> SNA defines two additional FID types: FID 3 and FID F. Path control used FID 3 to route data between a subarea node and a type 1 node. For information about FID F, refer to the *SNA Format and Protocol Reference Manual: Architectural Logic*.

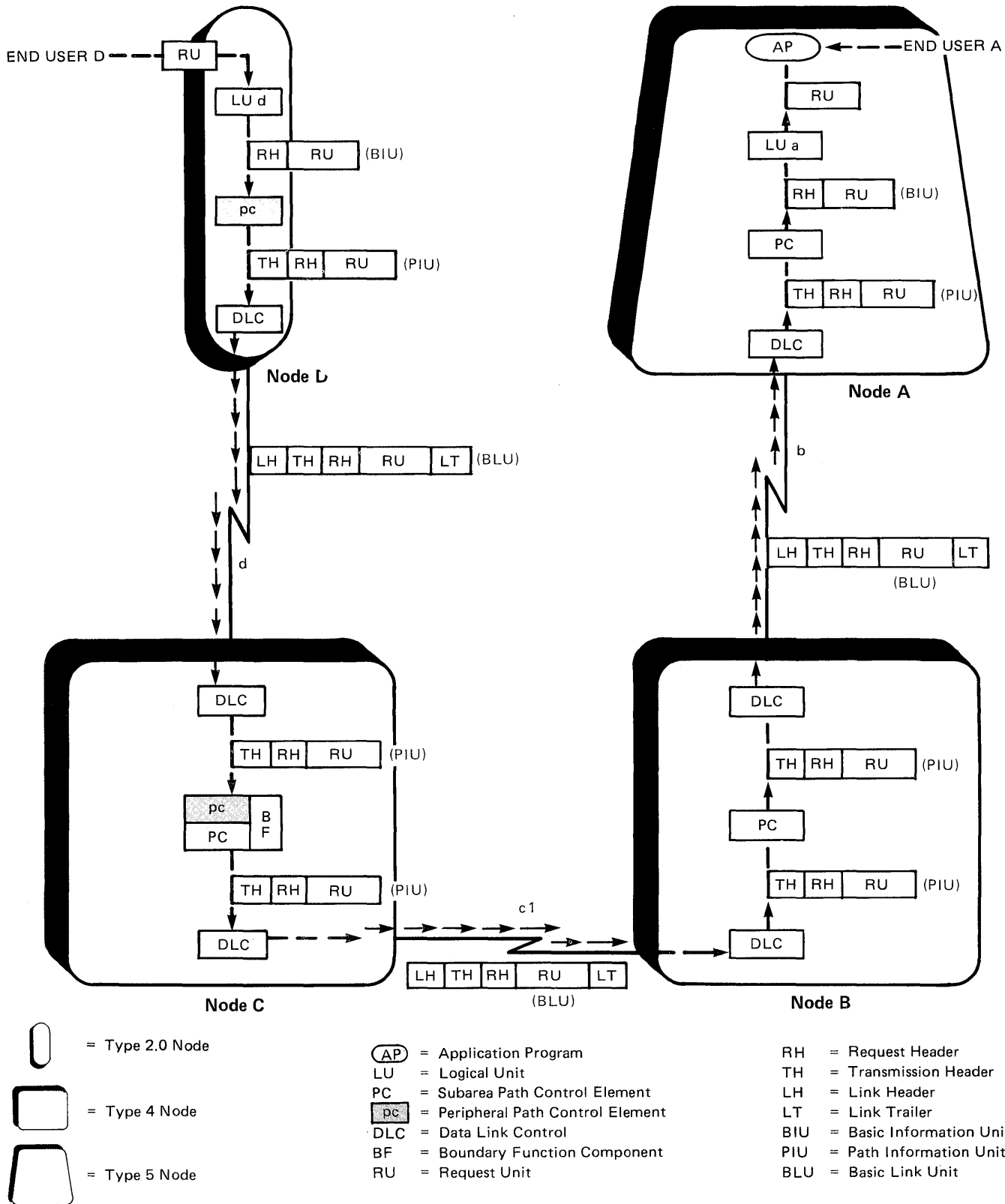


Figure 42. Use of Data Formats



## Node D

1. End user D gives the message unit that contains end-user data to LU *d*.

Because end-user data is called a request, this message unit is called a request unit (RU).

2. LU *d* appends a header to the RU; this forms a basic information unit (BIU).

The header that the logical unit appends to the RU is called a request header (RH). Request headers describe the kind of data in the RU and some of the ways in which the two NAUs are to communicate. Only NAUs use request header information.

3. LU *d* gives the BIU to peripheral path control.

4. Path control appends a header to the BIU; this forms a path information unit (PIU).

The header that path control appends to the BIU is called a transmission header (TH). Transmission headers contain routing information for the path control network.

5. Path control gives the PIU to a data link control element.

6. Data link control appends both a header and a trailer to the PIU; this forms a basic link unit (BLU).

The header that data link control appends to the PIU is called a link header (LH). The trailer that data link control appends to the PIU is called a link trailer (LT). Link headers and link trailers contain link control information that manages the transmission of a message unit across a link. Only data link control elements use link header and link trailer information.

7. Data link control transmits the BLU over link *d* to node C.

## Node C

1. The data link control element that receives the BLU removes the link header and link trailer, then gives the PIU to path control.

2. Path control uses the information in the transmission header to determine where to route the message unit next. Then path control gives the PIU to another data link control element.

3. Data link control appends a link header and link trailer to the PIU.

4. Data link control transmits the BLU over link *c1* to node B.

### **Node B**

1. The data link control element that receives the BLU removes the link header and link trailer, then gives the PIU to path control.
2. Path control uses the information in the transmission header to determine where to route the message unit next. Then path control gives the PIU to another data link control element.
3. Data link control appends a link header and link trailer to the PIU.
4. Data link control transmits the BLU over link *b* to node A.

### **Node A**

1. The data link control element that receives the BLU removes the link header and link trailer, then gives the PIU to path control.
2. Path control uses the destination subarea address in the message unit to determine that node A is the destination subarea node. Path control removes the transmission header, then delivers the BIU to the destination NAU.
3. LU *a* removes the request header and gives the RU to end user A.



### SNA Protocols

This chapter introduces the SNA protocols that govern communication between two network addressable units.

### Contents

Path Control Network Transmission Protocols	109
Sequencing	109
Blocking	110
Segmenting	111
BIND Protocols	112
Response Protocols	112
Definite response	112
Exception response	113
No response	113
Chaining Protocols	113
Definite Response Chain	114
Exception Response Chain	114
No-Response Chain	114
Bracket Protocols	114
Sequencing Protocols	115
Request and Response Mode Protocols	115
Immediate Request Mode	115
Delayed Request Mode	115
Immediate Response Mode	116
Delayed Response Mode	116
Send and Receive Mode Protocols	116
Full Duplex	116
Half-Duplex Contention	116
Half-Duplex Flip-Flop	117
Comparison of Send and Receive Mode Protocols and Transmission Medium Protocols	117
Data Security Protocols	117
Passwords and User IDs	117
Session Cryptography	118
LU-LU Session Passwords	118
Function Management Headers	119



## Path Control Network Transmission Protocols

This section explains three transmission protocols that the path control network uses when transmitting data between network addressable units. The three protocols are:

- Sequencing
- Blocking
- Segmenting.

Path control uses sequencing protocols to reorder out-of-sequence message units that it receives over a transmission group. Path control uses the blocking and segmenting protocols to modify the path information unit (PIU) that data link control transmits over a link.

The format of the modified PIU is called a basic transmission unit. A **basic transmission unit (BTU)** is the information that data link control transmits over a link. A BTU can consist of one PIU, multiple PIUs, or a part of one PIU.

### Sequencing

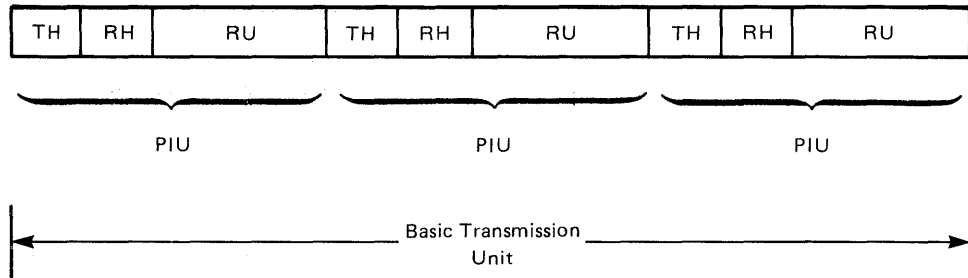
The architecture specifies that requests and responses must arrive at their destination in the same order in which the sender transmits the message units. Because data link control can route related PIUs over different links within a transmission group, path control in another node might not receive the PIUs in the order in which they were sent.

The path control network assigns transmission group sequence numbers to PIUs before transmitting them across a transmission group. Then path control on the other side of the transmission group uses these sequence numbers to reorder any out-of-sequence PIUs before continuing to route the data through the network.

## Blocking

**Blocking** is the combination of multiple PIUs into one basic transmission unit (BTU). Path control uses blocking protocols to combine multiple PIUs before giving the message units to data link control. Data link control then transmits one BTU (containing multiple PIUs) over a link. You specify a maximum BTU length (the number of PIUs) for the network during the system definition process.

Blocking increases the amount of information that data link control can transmit over a link at one time. Path control can implement blocking only for a System/370 data channel. Figure 43 shows a basic transmission unit that contains more than one PIU.



**Legend:**

- PIU = Path Information Unit
- TH = Transmission Header
- RH = Request Header or Response Header
- RU = Request Unit or Response Unit

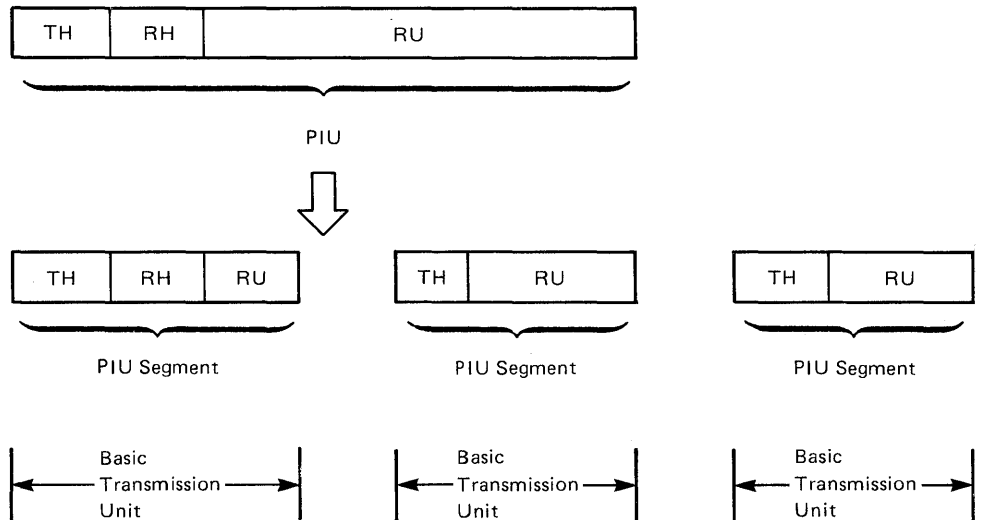
**Figure 43. Blocking of Path Information Units**

## Segmenting

**Segmenting** is the division of one PIU into multiple PIUs. Path control uses segmenting protocols to divide a single PIU into multiple PIU segments before giving the message unit to data link control. Data link control then transmits each BTU (one PIU segment) separately over a link.

Path control implements segmenting when a peripheral node cannot accept or process the amount of data contained in a single PIU. Figure 44 shows a basic transmission unit that contains a segmented PIU.

Each PIU segment contains the transmission header and a portion of the BIU. The transmission header is part of each PIU segment because it contains routing information that path control in the peripheral node requires.



**Legend:**

PIU = Path Information Unit  
RH = Request Header or Response Header  
RU = Request Unit or Response Unit  
TH = Transmission Header

**Figure 44. Segmenting of Path Information Units**



## BIND Protocols

Recall from “Chapter 4. LU-LU Sessions” that parameters in the BIND command specify protocols for an LU-LU session. Some BIND parameters specify how the LUs are to transmit data to one another; some specify how the LUs are to present data to the end user. This section explains the following protocols:

- Response Protocols
- Chaining Protocols
- Bracket Protocols
- Sequencing Protocols
- Request and Response Mode Protocols
- Send and Receive Mode Protocols
- Data Security Protocols
- Function Management Headers.

### Response Protocols

When a logical unit sends a request to its session partner, it sometimes needs to know if its session partner received the information. A session partner acknowledges the receipt of a request by returning a response to the sender. Indicators in the request header (RH) identify one of three SNA response protocols: definite response, exception response, or no response.

#### Definite response

If a request specifies a **definite response**, the request receiver must return either (1) a positive response to accept the request or (2) a negative response to reject the request. The request sender sets definite response indicators (DR1I, DR2I) in the request header to identify the definite response protocol.

Recall from the discussion of “Sync Points” on page 71 that transaction programs issue verbs to invoke the sync point function. The LU type 6.2 uses the DR2 indicator in a request header to notify its session partner that its transaction program has requested a sync point. Similarly, the remote LU type 6.2 uses the DR2 indicator in a response header when it responds to the sync point request. For additional information on the use of the DR2 indicator for sync point functions, refer to the *SNA Format and Protocol Reference Manual: Architecture Logic for LU Type 6.2*.

## Exception response

If a request specifies an **exception response**, the request receiver returns a negative response for any unacceptable requests. The receiver does not return any positive responses. The request sender sets the exception response indicator (ERI) in the request header to identify the exception response protocol.

## No response

If a request specifies **no response**, the request receiver returns neither a positive nor a negative response.

BIND parameters specify which type of response protocols logical units will abide by while communicating over an LU-LU session. If definite response protocols are specified for all LU-LU sessions, unnecessary link traffic may result. If an LU-LU session requires transmission accuracy, the session should not use the no-response protocol. Using the no-response protocol, the request sender does not know if errors have occurred or if data has been lost. There is no response to indicate whether the request was acceptable or unacceptable.

## Chaining Protocols

An RU-size parameter in the BIND command limits the size of a request unit (RU) that two logical units can send to each other. In order to send a request that contains more information than will fit into one request unit, logical units divide the information into a series of separate requests. This series of related requests is called a chain. A **chain** is a sequence of BIUs that constitute a single, unidirectional transfer of data. Even though a chain is a series of separately transmitted requests, the receiving logical unit treats the chain as a single request.

Logical units use chains to:

- Divide data when the request unit is extremely lengthy
- Transmit related request units as a single entity
- Establish a series of requests as a unit of error recovery.

There are both single-element and multiple-element chains. Single-element chains are either (1) command RUs, (2) a single data RU, or (3) a response RU. Multiple-element chains are multiple data RUs. Every RU belongs to a chain.

A logical unit sets indicators in request headers to identify the beginning and end of a chain for the receiving logical unit. A logical unit sets the begin chain indicator (BCI) in the request header of the first request to identify the beginning of a chain; it sets the end chain indicator (ECI) in the request header of the last request to identify the end of a chain. A logical unit sets both the BCI and ECI in the request header of a single-element chain.

Because the receiving logical unit treats a chain as a single request, it can return only one response to the sending logical unit. The following text explains how the sending logical unit specifies a response protocol (definite response, exception response, or no response) for a multi-element chain and how the receiving logical unit returns the specified response to its session partner.

### **Definite Response Chain**

The sending logical unit sets the exception response indicator in the request headers of all but the last request in the chain; it sets the definite response indicator in the request header of the last request in the chain.

After receiving the last element of the chain, the receiving logical unit returns a positive response to acknowledge that the entire chain is acceptable. If any one of the chain elements is unacceptable, the receiving logical unit returns a negative response for the entire chain.

### **Exception Response Chain**

The sending logical unit sets the exception response indicator in the request headers of all the requests in the chain.

The receiving logical unit never returns a positive response for the chain. However, it will return a negative response for the entire chain if any of the requests in the chain is unacceptable.

### **No-Response Chain**

The receiving logical unit never returns a response for the chain.

## **Bracket Protocols**

Logical units use chains to send related requests as a single message; they use brackets to separate groups of related chains and responses from other groups of related chains and responses. A **bracket** is a sequence of request chains and responses exchanged in either, or both, directions between two logical units. The primary use of brackets is to delimit conversations between logical units.

A bracket includes the first request header through the last response unit of related chains and their responses. A bracket can be a monolog (one session partner sends a series of chains), or it can be a dialog (both session partners exchange a series of chains). In both cases, a bracket identifies a sequence of related requests and responses that flow between session partners.

A type 6.2 logical unit sets begin bracket indicators (BBIs) and conditional end bracket indicators (CEBIs) in request headers to identify the beginning and end of a bracket. Logical units other than type 6.2 use end bracket indicators (EBIs), not CEBIs, to identify the end of a bracket. To identify the beginning of a chain, a logical unit sets the BBI in the request header of the first chain element of the bracket. A type 6.2 logical unit sets the CEBI in the request header of the first request in the last chain of the bracket to identify the end of a chain. Other logical units set the EBI in the request header of the last chain element of the bracket to identify the end of a chain.

The BIND parameters specify (1) whether brackets will be used, (2) which logical unit will be the “first speaker,” and (3) which logical unit will be the “bidder.” The first speaker can begin a bracket without asking for permission from its session partner, the bidder. The bidder must ask for and receive permission from the first speaker to begin a bracket. For additional information on the protocols that govern the use of bracket indicators (BBI, CEBI, and EBI), refer to the *SNA Format and Protocol Reference Manual: Architecture Logic for LU Type 6.2* and the *SNA Format and Protocol Reference Manual: Architectural Logic*.

## Sequencing Protocols

SNA session protocols require that a logical unit receive related requests and responses in the same order in which they were sent by its session partner. Logical units assign sequence numbers to each request that they send over an LU-LU session. Session partners assign sequence numbers independently of one another.

A logical unit assigns the sequence number 1 to the first request that it sends after session activation. Then the logical unit increases the sequence number by 1 for every subsequent request that it sends to its session partner. The receiving logical unit assigns responses the same sequence number as their associated requests. Sequence numbers enable logical units to identify which response is associated with which request.

## Request and Response Mode Protocols

Request and response mode protocols control when logical units can send and receive requests and responses over an LU-LU session. The BIND command specifies request and response modes. The modes that the BIND indicates for traffic in one direction are independent of the modes that it indicates for traffic in the reverse direction. SNA defines four request and response modes: immediate request mode, delayed request mode, immediate response mode, and delayed response mode.

### Immediate Request Mode

The **immediate request mode** requires that the sending logical unit wait for a response before sending any additional requests. If the sender groups the request units into chains, it must wait for the receiving logical unit to return a response to the chain before it can send any additional requests.

The immediate request mode applies only to requests that specify definite responses. It does not apply to a request that specifies an exception response or no response.

### Delayed Request Mode

The **delayed request mode** allows the sending logical unit to send additional requests without waiting for any responses.

## Immediate Response Mode

The **immediate response mode** requires that the receiving logical unit return responses in the same order in which it received the requests. For example, if request A is received before request B, the response to request A must be returned before the response to request B. Responses need not follow the request immediately, but they must be returned in the same sequence as the requests were received.

## Delayed Response Mode

The **delayed response mode** allows the receiving logical unit to return responses in any order. For example, if request A is received before request B, the response to either request can be returned first. Responses need not follow the request immediately and they can be returned to the sender in any sequence.

## Send and Receive Mode Protocols

Send and receive mode protocols determine when session partners can send and receive requests. The BIND parameters specify whether both session partners are allowed to send and receive requests simultaneously (full duplex) or whether session partners must take turns sending and receiving (half duplex). SNA defines three send and receive modes: full duplex, half-duplex contention, and half-duplex flip-flop.

### Full Duplex

Logical units using a **full-duplex** send and receive mode can send and receive requests simultaneously. Session traffic in one direction is independent of traffic in the other direction.

### Half-Duplex Contention

Logical units using a **half-duplex contention** send and receive mode can both send requests to each other, but not at the same time. Contention and lost data occur if both logical units try to send requests at the same time.

Parameters in the BIND resolve the contention by identifying one of the LUs as the "contention winner." The contention winner has the right to send its request first; the other logical unit must try to send its request later.

When specifying BIND parameters, one of the items you need to consider in the selection of a contention winner is each logical unit's processing capacity. For example, you would normally choose a workstation over a program as the contention winner; the program can buffer the data to be sent to the workstation, but the workstation normally cannot buffer the data to be sent to the program.

## Half-Duplex Flip-Flop

Logical units using a **half-duplex flip-flop** send and receive mode alternate sending requests to one another. Parameters in the BIND identify one of the logical units as “first speaker.” The first speaker begins in the send state, and the other logical unit begins in the receive state. The first speaker allows the other logical unit to become the sender by setting the change direction indicator (CDI) in the request header of the last request unit sent. A logical unit sets the CDI to switch from sender status to receiver status. The two logical units continue to switch between send and receive states until the session is deactivated.

*Note:* Conversations between type 6.2 logical units always use half-duplex flip-flop protocols.

## Comparison of Send and Receive Mode Protocols and Transmission Medium Protocols

Do not confuse send and receive mode protocols with transmission medium protocols. Transmission media are either full duplex or half duplex. A full-duplex transmission medium refers to the capability of the medium to transmit data in two directions simultaneously. A half-duplex transmission medium refers to the capability of the medium to transmit data in only one direction at a time. Send and receive modes are independent of the physical properties of the network.

For example, requests sent between two LUs might travel over several links, some of which might be half-duplex transmission media. The fact that requests can flow in only one direction at a time over the half-duplex links does not mean that the two session partners cannot simultaneously send requests to one another (full-duplex send and receive mode). The requests flowing in opposite directions can pass each other at buffered points in the network.

## Data Security Protocols

Logical units provide three functions that help your network provide security for end-user data: passwords and user IDs, session cryptography, and LU-LU session passwords.

### Passwords and User IDs

To help prevent unauthorized end users from gaining access to local transaction programs, logical units can verify the identity of end users. Logical units use **passwords** and **user IDs** to verify the authority of remote end users prior to allowing them access to a transaction program (over an existing LU-LU session). A request unit carries a password and user ID in an Attach function management header. How passwords and user IDs are verified and enforced is implementation-defined.

## Session Cryptography

To help protect data while it is being transmitted across a link, logical units can provide **session cryptography**. Logical units use the Data Encryption Standard algorithm to provide session cryptography. Each LU-LU session uses a randomly-generated cryptography key to encipher and decipher data.

Type 6.2 logical units can provide either mandatory data encryption or no data encryption. Other types of logical units can provide mandatory data encryption, selective data encryption, or no data encryption. In **mandatory cryptographic** sessions, logical units encipher all outbound data RUs and decipher all inbound data RUs. In **selective cryptographic** sessions, logical units encipher only data RUs that have the enciphered data indicator (EDI) set in the request header.

For additional information about cryptography, refer to the *Data Security through Cryptography* manual.

## LU-LU Session Passwords

To help ensure session security, logical units can verify a password before they activate an LU-LU session. Logical units use **LU-LU session passwords** to provide this LU-LU session verification. Individuals wishing to provide session security must agree upon the password prior to session initiation.

LU-LU session verification proceeds as follows:

1. The two end users that wish to communicate over the LU-LU session each enter the mutually-agreed-upon password at their workstations.
2. The primary LU sends a BIND request that contains random data to the secondary LU.<sup>6</sup>
3. The secondary LU uses its LU-LU session password to encipher the random data in the BIND. It then returns the enciphered random data to the primary LU in a BIND response.

The secondary LU also returns additional random data for the primary LU in this BIND response.

4. The primary LU uses its password to encipher the random data that it sent in the BIND. Then it matches the resulting information with the enciphered random data that it received from the secondary LU in the BIND response.

If the information matches, the primary LU knows that the secondary LU is using the same LU-LU session password that it is using. The primary LU thus verifies that the requested LU-LU session is authorized.

5. The primary LU then uses its LU-LU session password to encipher the random data that the secondary LU returned in the BIND response. It sends this enciphered data to the secondary LU in a Security function management header.

---

<sup>6</sup> A user-data subfield in the BIND contains this random data.

6. The secondary LU uses its password to encipher the random data that it sent in the BIND response. Then it matches the resulting information with the enciphered random data that it received from the primary LU in the Security function management header.

If the information matches, the secondary LU also knows that its session partner is using the same LU-LU session password. The two end users can now communicate over the LU-LU session.

## Function Management Headers

A **function management (FM) header** is an optional field at the beginning of a request unit that carries certain control information for logical units. Different types of logical units use different types of FM headers.

Type 6.2 logical units use three types of FM headers:

- An Attach FM header to specify the name and required characteristics of a target transaction program
- An Error-description FM header to describe a transaction program error or an attach failure
- A Security FM header to carry LU-LU session password verification data.

Other types of logical units use FM headers to tell the receiving logical unit how to format a data RU for presentation to the end user. For example, FM headers tell a logical unit how to take data that it receives in brackets, chains, and individual request units, and transform it into lines of printed data. These FM headers alter the way information is presented in order to match the needs (and language) of each end user.

Parameters in the BIND command indicate whether FM headers will be used during the session and if there are limitations on their use. A format indicator (FI) in the request header identifies whether the request unit contains any FM headers.

In the BIND parameters, you can specify (1) whether a logical unit has full capability to use headers, (2) whether a logical unit has limited capability to use headers, or (3) whether a logical unit cannot use headers. The option you select depends on whether the LU type supports FM headers and on the amount of header capability that you need to format end-user data.

For additional information about the use of FM headers by type 6.2 logical units, refer to the *SNA Format and Protocol Reference Manual: Architecture Logic for LU Type 6.2*. For additional information about the use of FM headers by other types of logical units, refer to the *SNA Sessions Between Logical Units* manual.





# Relationship of SNA Layers to Network Operation

This chapter relates the different SNA functions and protocols to the seven architectural layers. The chapter also compares the Systems Network Architecture with the Open Systems Interconnection (OSI) architecture.

## Contents

Architectural Definitions	123
Network Addressable Unit Functions	124
Path Control Network Functions	124
SNA Layers	125
Transaction Services Layer	126
Configuration Services	126
Session Services	126
Management Services	127
Presentation Services Layer	127
Data Flow Control Layer	127
Transmission Control Layer	128
Path Control Layer	128
Data Link Control Layer	128
Physical Control Layer	129
Peer-to-Peer Communication between SNA Layers	129
SNA Layer Management	131
Open Systems Interconnection	132
OSI Physical Layer	132
OSI Data Link Layer	132
OSI Network Layer	132
OSI Transport Layer	133
OSI Session Layer	133
OSI Presentation Layer	133
OSI Application Layer	133
Comparison of SNA and OSI	134



## Architectural Definitions

Recall that Systems Network Architecture consists of seven layers, each layer performing a specific function. The upper four layers of the architecture provide the network addressable unit functions and the boundary function; the lower three layers provide the path control network functions. Figure 45 illustrates this concept.

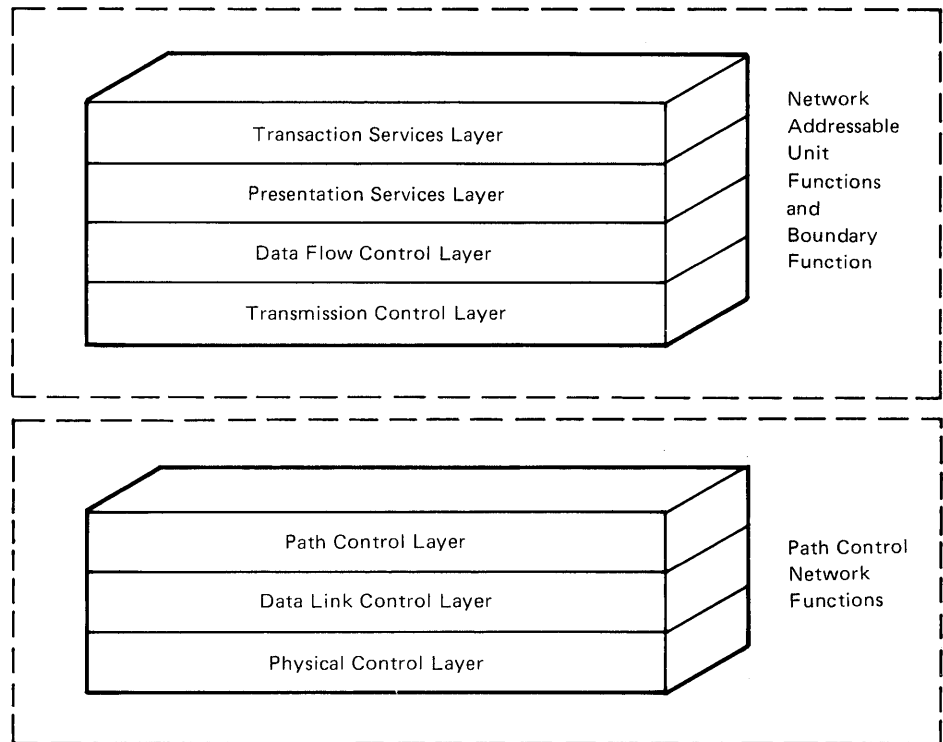
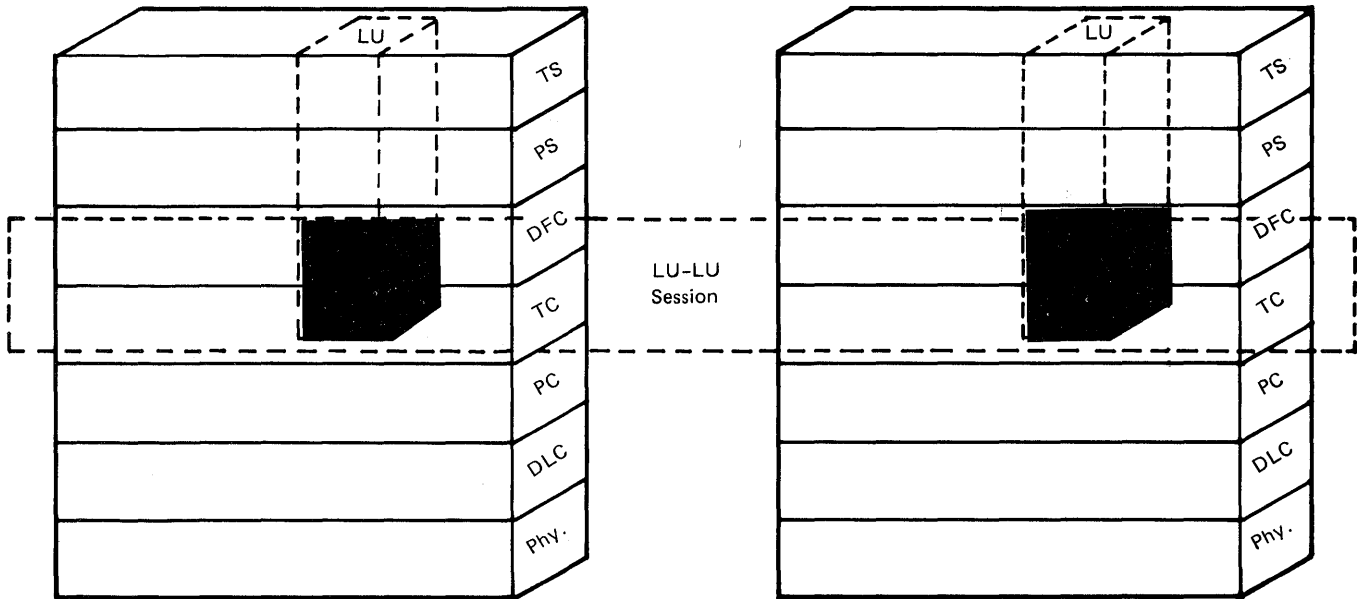


Figure 45. Relating SNA Layers to SNA Functions

## Network Addressable Unit Functions

Network addressable unit (NAU) functions enable end users to send and receive data through the network. Recall that an LU-LU half-session identifies the resources that a logical unit allocates to support a session. The upper four layers of the architecture define the logical unit (and other NAU) functions in each node. The architecture defines the half-session resources in the two lower layers of the logical unit function, as Figure 46 illustrates.



**Legend:**

- TS = Transaction Services Layer
- PS = Presentation Services Layer
- DFC = Data Flow Control Layer
- TC = Transmission Control Layer
- PC = Path Control Layer
- DLC = Data Link Control Layer
- Phy. = Physical Control Layer
- LU = Logical Unit

**Figure 46. Definition of a Half-Session**

## Path Control Network Functions

The path control network routes and transmits data between NAUs. Whereas the upper four layers within each node define the NAUs, the combination of the path control, data link control, and physical control layers within all the nodes in the network defines the path control network.

# SNA Layers

Systems Network Architecture is a hierarchical, seven-layered structure. Each well-defined layer performs a specific SNA function. Because each of the layers is designed to be functionally self-contained, changes to one layer do not affect the others.

Figure 47 reviews the seven SNA layers:

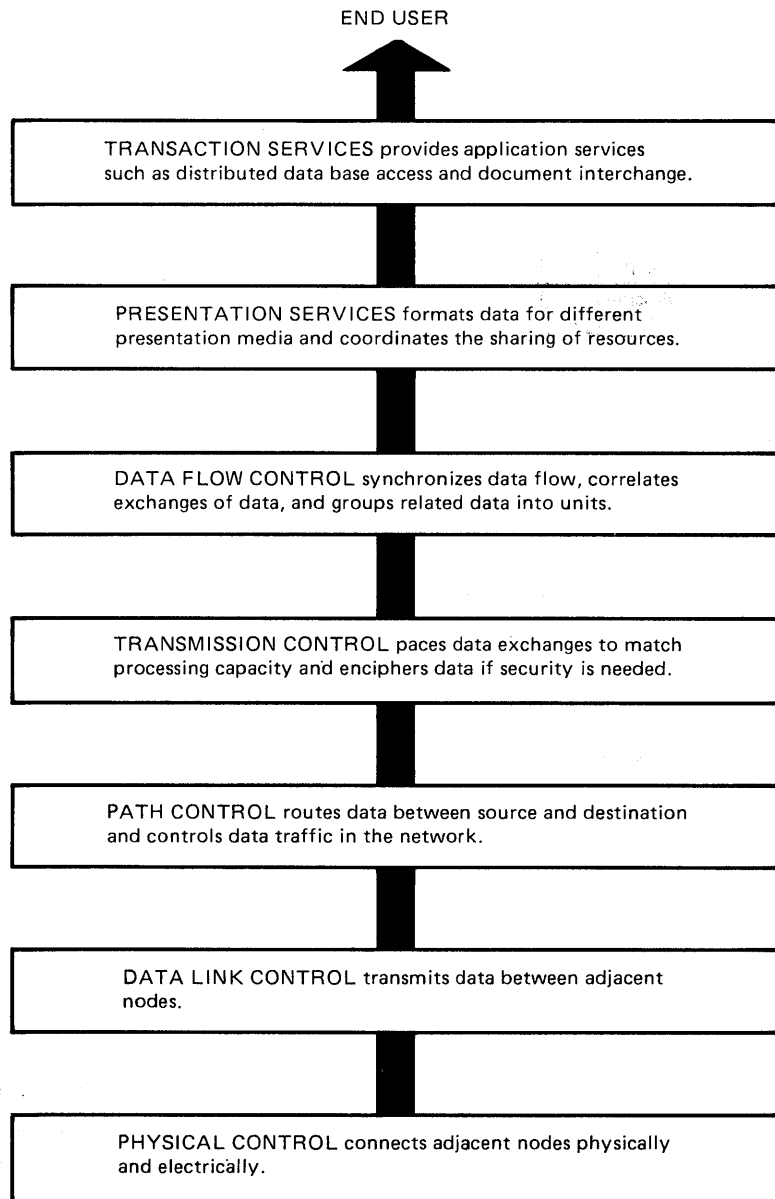


Figure 47. The Seven SNA Layers

## Transaction Services Layer

The **transaction services layer** implements service transaction programs in an SNA network. Service transaction programs provide the following services for end users of an SNA network:

- Operator control of LU-LU session limits
- Document Interchange Architecture (DIA) for document distribution between office systems
- SNA Distribution Services (SNADS) for asynchronous data distribution between distributed applications and office systems.

The transaction services layer also provides the following services to control the network's operation:

- Configuration services
- Session services
- Management services.

## Configuration Services

SSCP-PU sessions use configuration services to control resources associated with the physical configuration. Configuration services:

- Activate and deactivate links
- Load same-domain software
- Assign network addresses during dynamic reconfiguration.

## Session Services

SSCP-SSCP and SSCP-LU sessions use session services to establish LU-LU sessions. Session services:

- Translate network names to network addresses
- Verify user passwords and user access authority
- Select session parameters.

## Management Services

SSCP-PU and SSCP-LU sessions use management services to help control the operation of the network. Management services:

- Perform monitoring
- Perform testing
- Provide traces
- Record network resource statistics.

## Presentation Services Layer

The **presentation services layer** defines protocols for program-to-program communication and controls conversation-level communication between transaction programs by:

- Loading and invoking transaction programs
- Maintaining conversation send and receive mode protocols
- Enforcing correct verb parameter usage and sequencing restrictions
- Processing transaction program verbs.

## Data Flow Control Layer

The **data flow control layer** provides flow control services for an LU-LU session by:

- Assigning sequence numbers
- Correlating requests and responses
- Grouping related request units into chains
- Grouping related series of chains into brackets
- Enforcing session request and response mode protocols
- Coordinating session send and receive modes.



## Transmission Control Layer

The **transmission control layer** provides basic control of the transmission resources in the network by:

- Verifying received sequence numbers
- Enciphering and deciphering data
- Managing session-level pacing
- Providing boundary function support for peripheral nodes.

## Path Control Layer

The **path control layer** provides protocols to route message units through the network. All types of sessions use the path control layer to:

- Select paths through the network
- Route data through the network
- Segment and reassemble message units
- Control virtual routes, including virtual route pacing
- Control explicit routes.

The path control layer in subarea nodes consists of three sublayers: (1) transmission group control, (2) explicit route control, and (3) virtual route control.

Transmission group control, the inner sublayer, provides one or more transmission group connections between adjacent subarea nodes. Explicit route control, the middle sublayer, provides an explicit route connection between the two end subarea nodes of a path. Virtual route control, the outer sublayer, provides a virtual route connection between half-sessions.

## Data Link Control Layer

The **data link control layer** provides protocols for (1) transferring message units across a link connection and (2) performing link-level flow and error recovery.

The data link control layer:

- Transmits message units across links
- Manages link-level flow
- Manages error recovery procedures for transmission errors.

The data link control layer supports both SDLC and System/370 data channel protocols.

## Physical Control Layer

The **physical control layer** provides a physical interface for any transmission medium that is attached to it. This layer defines the electrical and transmission (signaling) characteristics needed to establish, maintain, and terminate physical connections.

## Peer-to-Peer Communication between SNA Layers

SNA defines peer-to-peer communication between layers that permits equivalent layers (layers at the same level within the hierarchy) to communicate with one another. Each layer performs services for the next higher layer, requests services from the next lower layer, and communicates with equivalent layers. Peer-to-peer communication requires that each layer be isolated from the internal procedures that another layer follows.

Figure 48 illustrates the communication between layers.

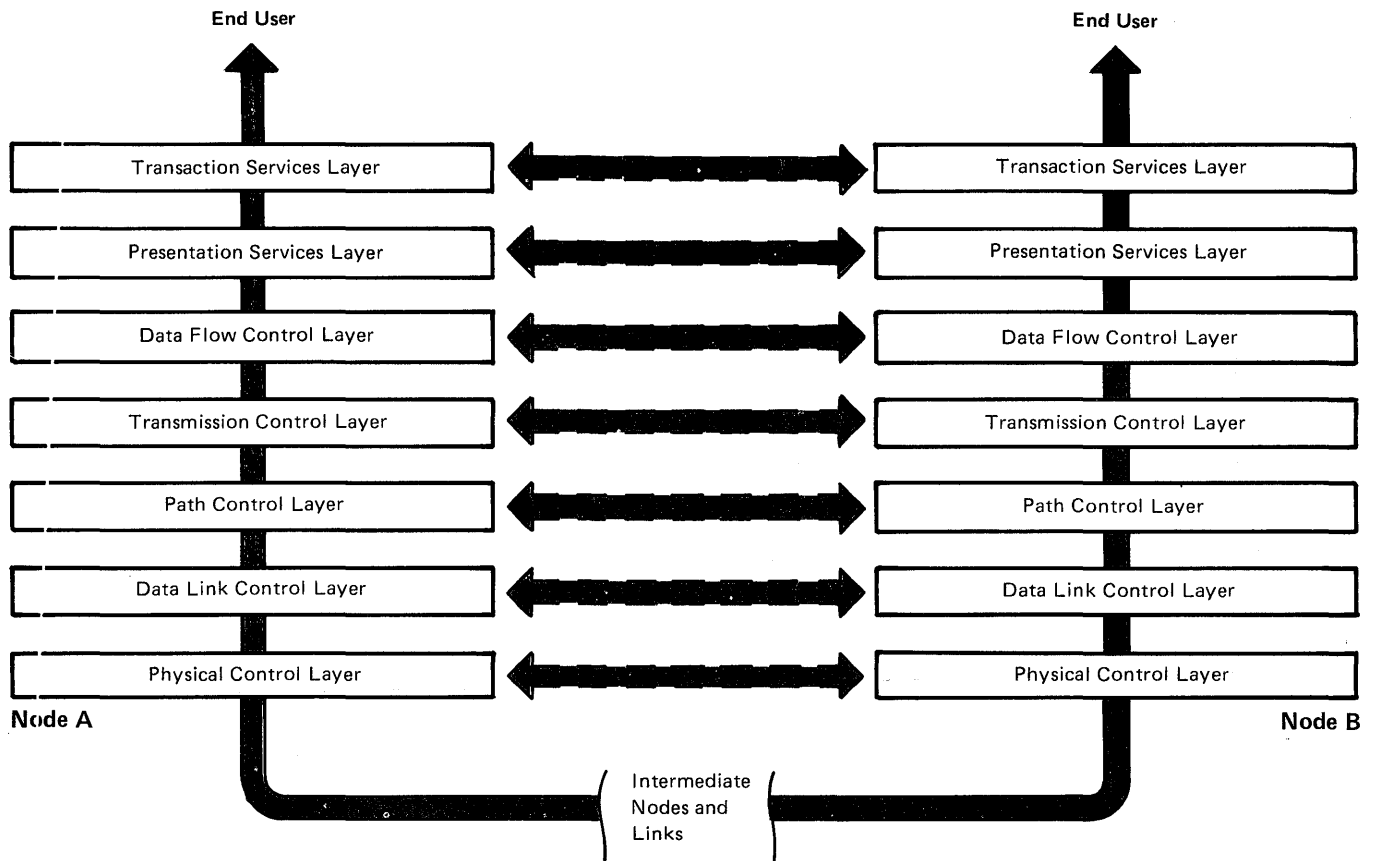
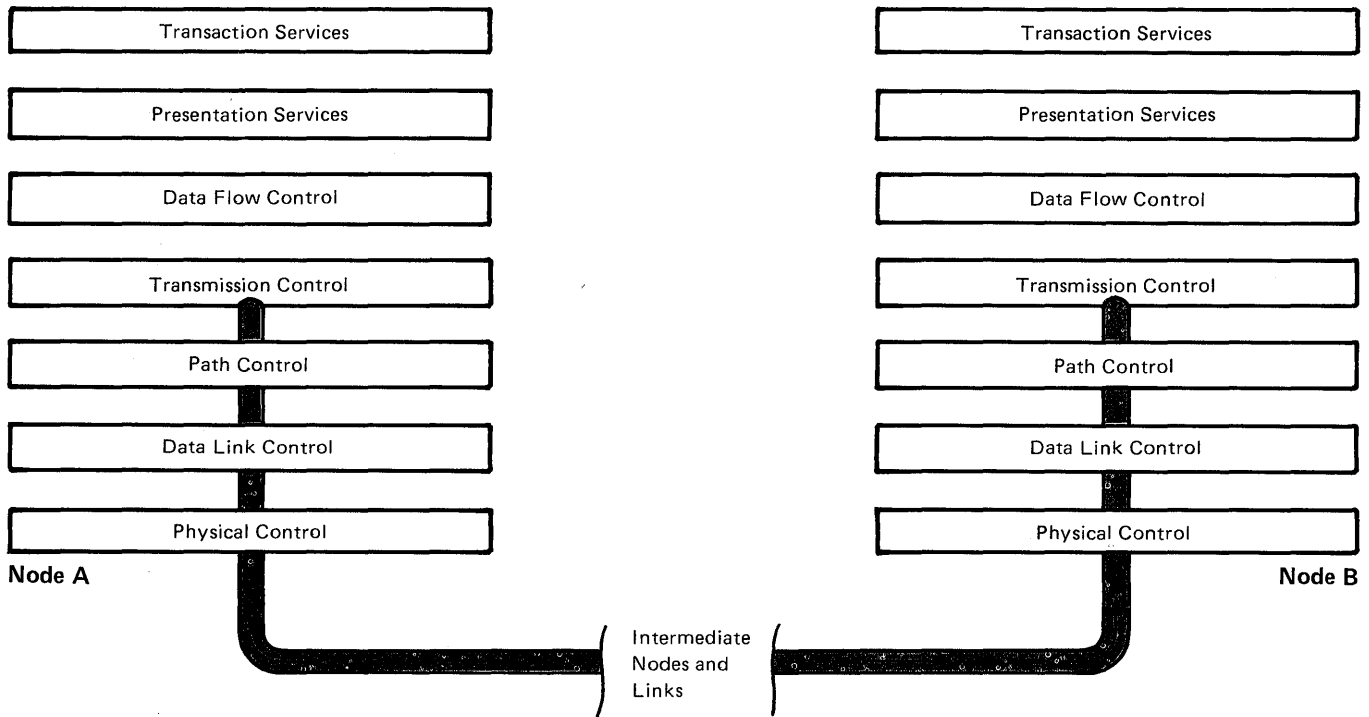


Figure 48. Communication between SNA Layers

For example, the transmission control layer performs services for the data flow control layer, requests services from the path control layer, and communicates with other transmission control layers in other nodes. The transmission control layer functions independently of any other layer in the architecture.

To further illustrate this concept, consider end-user data that requires encryption. Figure 49 illustrates how the two transmission control layers communicate.



**Figure 49. Communication between Two Transmission Control Layers**

The two transmission control layers encipher and decipher the data independently of the functions that any other layer performs. The transmission control layer in the originating node enciphers the data that it receives from the data flow control layer. Then it requests that the path control layer route the enciphered data to the destination node. The transmission control layer in the destination node decipheres the data that the path control layer delivered. Then it requests that the data flow control layer give the deciphered data to the destination end user.

Peer-to-peer communication means that the layers are defined to be functionally independent of one another. This allows SNA networks to grow, change, and adapt to new technologies. Any one layer can be enhanced or modified without necessarily disrupting the functions that any other layer provides.

## SNA Layer Management

**NAU services managers** are responsible for managing the upper four layers of the architecture. The NAU services managers are divided into (1) SSCP services managers, (2) PU services managers, and (3) LU services managers. The functions of the services managers include activating sessions, managing network resources, and controlling error recovery and restart management.

In addition to the above functions, the **LU services manager** controls the sending and receiving of end-user data. It has two components: a resources manager and a network services component.

The **resources manager** allocates conversation resources and assigns conversations to sessions. Additionally, the resources manager maintains queues of available sessions and allocation requests.

The **network services component** coordinates a logical unit's session-initiation request with the SSCP by supplying and negotiating BIND session parameters. Additionally, the network services component notifies the SSCP of a logical unit's characteristics and conditions when the SSCP activates the LU.

For additional information, refer to the *SNA Format and Protocol Reference Manual: Architecture Logic for LU Type 6.2* and the *SNA Format and Protocol Reference Manual: Architectural Logic*.

# Open Systems Interconnection

SNA is IBM's architecture for a data communication network. Other companies and organizations also have developed, or are developing, other architectures for data communication networks. Therefore, the International Standards Organization (ISO) is developing a standard set of protocols for exchanging information between different architectures. This section summarizes the basic functions of the seven OSI layers.

## OSI Physical Layer

Layer 1, the physical layer, defines the electrical characteristics and signaling needed to establish, maintain, and disconnect physical connections. Current examples of this layer's standards include CCITT recommendations V.35 and V.24 for analog transmission, and X.21 for digital transmission.

## OSI Data Link Layer

Layer 2, the data link layer, provides transmission over a single data link. The data link layer services include data link activation and deactivation, as well as data link error detection, recovery, and notification.

Examples of layer 2 protocols are the ANSI Advanced Data Communications Control Procedures (ADCCP) and ISO High-Level Data Link Control (HDLC).

## OSI Network Layer

Layer 3, the network layer, provides control between two adjacent nodes. The network layer's services include:

- Network addressing
- Segmenting, blocking, and sequencing of data units
- Local flow control
- Optional expedited data transfer
- Error detection, recovery, and notification.

The CCITT Recommendation X.25 includes the above network layer functions.

## OSI Transport Layer

Layer 4, the transport layer, provides end-to-end control between two user nodes. OSI defines three phases of operation within this layer. The three phases are:

- Establishment

The establishment phase establishes transport connections to lower layer connections.

- Data transfer

The data transfer phase transports data between layers.

- Release.

The release phase disconnects transport connections.

## OSI Session Layer

Layer 5, the session layer, establishes, maintains, and terminates sessions. The session layer establishes, releases, synchronizes, and resynchronizes session connections. It also manages normal and expedited data exchange.

## OSI Presentation Layer

Layer 6, the presentation layer, provides the protocols that allow systems to determine and keep track of the representation (for example, syntax) of user data stream information.

## OSI Application Layer

Layer 7, the application layer, contains all the protocols between systems that have not been included in any other OSI layer. The OSI application layer consists of the:

- Common Application Services
- Specific Application Services
- User Element.

The Common Application Services are those services useful to all communicants. Common Application Services provide the protocols that select and control the type of conversations to be held between users (for example, a library bibliographic list exchange) and the structure of that conversation (for example, a file transfer structure).

The Specific Application Services contain the protocols of all user information exchanges regardless of whether they are recognized international standards or private protocols within a single communication.

The User Element represents the final origination and destination of user information, which precludes the need to define a user interface above the Application Layer.

## Comparison of SNA and OSI

The structure of the OSI model is similar to that of SNA, but the purpose of each is different. OSI is designed for the exchange of information between autonomous systems. The intent of the OSI model is to standardize protocols, thereby allowing communication between different architectures. SNA is designed for exchanges among nodes that conform to a single architecture; its intent is to define a single architecture for IBM's product offerings. While there is not a one-to-one correlation between SNA layers and OSI layers, the functional layering of SNA and OSI is quite similar:

- SNA's physical control layer is functionally equivalent to OSI's physical layer.
- SNA's data link control layer can be implemented to use SDLC or System/370 data channels. SDLC is a subset of HDLC, which can be used in OSI's data link layer. In addition, SNA's data link control layer can support the X.25 interface.
- SNA's path control layer encompasses functions similar to those in OSI's network and transport layers.
- SNA's data flow control and transmission control layers provide functions similar to those in OSI's transport and session layers.
- SNA's presentation services and transaction services layers perform functions similar to those in OSI's presentation layer and the Common Application Services in OSI's application layer.
- The exchanges that SNA considers end-user exchanges are considered Specific Application Services in OSI's application layer.

# Appendixes

## Contents

Appendix A. Summary of LU Types and Representative Products	137
Appendix B. Data Streams	139
Appendix C. Sequence Charts	143
Appendix D. Specifying Parameters for System Definition	175





### Summary of LU Types and Representative IBM Products

Listed below are the LU types that SNA currently defines and the kind of configuration or application that each type represents. Also mentioned are hardware or software products that typically use each type of logical unit.

#### **LU type 1**

An LU type 1 is for an application program that communicates with single- or multiple-device data processing workstations in an interactive, batch data transfer, or distributed data processing environment. The data stream conforms to the SNA character string or Document Content Architecture (DCA). An example of an LU type 1 is an application program that uses IMS/VS and communicates with an IBM 8100 Information System, at which the workstation operator is correcting a data base that the application program maintains.

#### **LU type 2**

An LU type 2 is for an application program that communicates with a single display workstation in an interactive environment, using the SNA 3270 data stream. Type 2 LUs also use the SNA 3270 data stream for file transfer. An example of an LU type 2 is an application program that uses IMS/VS and communicates with an IBM 3719 Display Station, at which the 3719 operator is creating and sending data to the application program.

#### **LU type 3**

An LU type 3 is for an application program that communicates with a single printer, using the SNA 3270 data stream. An example of an LU type 3 is an application program that uses CICS/VS and sends data to an IBM 3278 Printer that is attached to an IBM 3274 Controller.

#### **LU type 4**

An LU type 4 is for: (1) an application program that communicates with a single- or multiple-device data processing or word processing workstation in an interactive, batch data transfer, or distributed data processing environment (for example, an LU for an application program that uses CICS/VS and communicates with an IBM 6670 Information Distributor); or (2) logical units in peripheral nodes (for example, two 6670s) that communicate with each other. The data stream is the SNA character string (SCS) for data processing environments and Office Information Interchange (OII) Level 2 (a precursor of DCA) for word processing environments.

**LU type 6.1**

An LU type 6.1 is for an application subsystem that communicates with another application subsystem in a distributed data processing environment. An example of an LU type 6.1 is an application program that uses CICS/VS and communicates with an application program that uses IMS/VS.

**LU type 6.2**

An LU type 6.2 supports sessions between two applications in a distributed data processing environment. The data stream is either the SNA general data stream (GDS), which is a structured-field data stream, or a user-defined data stream. LU 6.2 sessions provide communication between two type 5 nodes, a type 5 node and a type 2.1 node, and two type 2.1 nodes. Examples of an LU type 6.2 are (1) an application program that uses CICS/VS and communicates with another application program that uses CICS/VS, (2) a DISOSS/370 application that uses CICS/VS and communicates with a Displaywriter System, or (3) an application program in a System/36.

**LU type 7**

An LU type 7 is for an application program and a single display workstation in an interactive environment. An example of an LU type 7 is an application program in a System/34 that communicates with an IBM 5251 Display Station, at which the 5251 operator is creating data and sending it to the application program. The data stream is the 5250 data stream.

### Data Streams

#### SNA Character String

SNA character string controls are EBCDIC control codes that define a data stream. Their primary function is to format a visual presentation medium such as a printed page or an alphanumeric display screen. These control codes can also set modes of device operation, define data to be used in a unique fashion, or be used for communication between a device operator and an application program (where the specific function associated with the code is defined in a protocol established between a program and an operator).

An SNA character string data stream consists of a sequential string of SNA character string control codes and data characters. Control codes can be intermixed with graphic data characters. SNA character string control codes are in the range X'00' through X'3F' plus X'FF'. Graphic codes are in the range X'40' through X'FE'. Other data types (such as binary and packed decimal) are permitted, but only with certain specific SNA character string control codes. Some codes also permit you to use one-byte parameters to specify functions or binary values.

SNA character string control codes and data appear within the request unit (RU). They may be preceded or separated by other control information in the RU, such as function management (FM) headers and string control bytes for functions such as selecting destinations, managing data, and compressing or compacting data.

SNA character string functions do not include data flow control functions, even though they may be available to a keyboard operator through keys on the keyboard. Cancel, for example, is a data flow control request that may be initiated by a key on the keyboard.

An SNA character string control and parameter sequence may be contained entirely within a single RU or it may span two or more RUs; however, it must be entirely contained within one RU chain.

Examples of SNA character string controls are Backspace, Carriage Return, Form Feed, Horizontal Tab, Indent Tab, Presentation Position, Select Left Platen, Set Horizontal Format, Superscript, and Word Underscore.

LU types 1, and 4 can use SNA character string controls.

## SNA 3270 Data Stream

The SNA 3270 data stream consists of user-provided data and commands that logical units transmit over an LU-LU session. Logical units also transmit control information that governs the way the data is handled and formatted.

The SNA 3270 data stream is the only data stream that LU types 2 and 3 use. It is an optional data stream for LU types 6.1 and 6.2. The data stream supports file-to-file transfer, display applications, and printer applications.

An application program communicates with a display operator using one of two methods. In the first method, the application program leaves the display surface unformatted, and the operator uses it in a free-form manner. In the second method, the application program completely or partially formats the display surface (that is, organizes or arranges it into fields) and the operator enters data into the fields.

The SNA 3270 data stream allows the application programmer to divide the display surface into one active area and, optionally, one or more reference areas. Each area is called a partition. The partition that is "active" contains a cursor and is the only partition in which the operator can enter data or requests.

Specific information on the 3270 functions that you can specify in an SNA 3270 data stream appears in *IBM 3270 Data Stream Programmer's Reference*.

## General Data Stream

The general data stream consists of data and commands that are defined by length (LL) and identification (ID) bytes. The length field is a descriptive prefix that indicates the length of the general data stream variable. The identification field determines the formats of the fields that follow (character and binary formats). General data stream variables enable transaction programs to interpret the records they transfer in the same way. Service transaction programs use general data stream variables to represent the data they transfer to one another. For additional information on the general data stream, refer to the *SNA Format and Protocol Reference Manual: Architecture Logic for LU Type 6.2* and the *SNA Reference Summary*.

## Document Content Architecture

Document Content Architecture (DCA) defines a uniform interface for interchanging information or documents in an office environment and specifies how documents are transported along a route. DCA deals with the functions that are performed on document contents such as pagination, highlighting, headings, footings, and centering.

Different levels of DCA are defined, including revisable form and final form. Revisable form defines the structure of the document that is to be edited or formatted. Final form defines the structure of a formatted document or a final form device-independent document.



Sequence Charts

Typical Request Unit Sequences for Activating and Deactivating Network Resources

Figure 51 through Figure 56 present typical request unit sequences for activating and deactivating various portions of a network. Figure 50 gives the meaning of each of the symbols and abbreviations that appear in these request unit sequences. "Chapter 3. Network Activation" explains how control points activate and deactivate network resources.

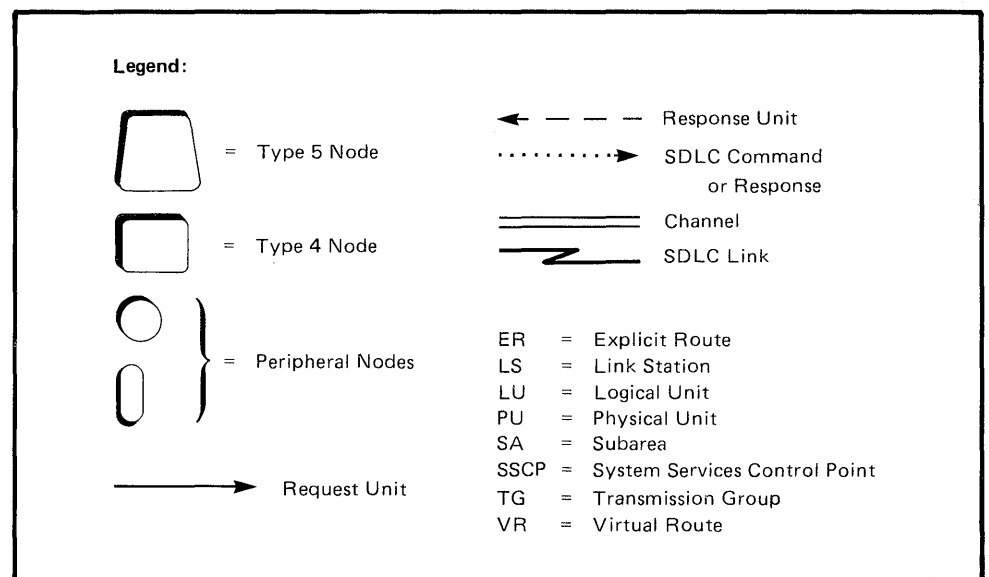
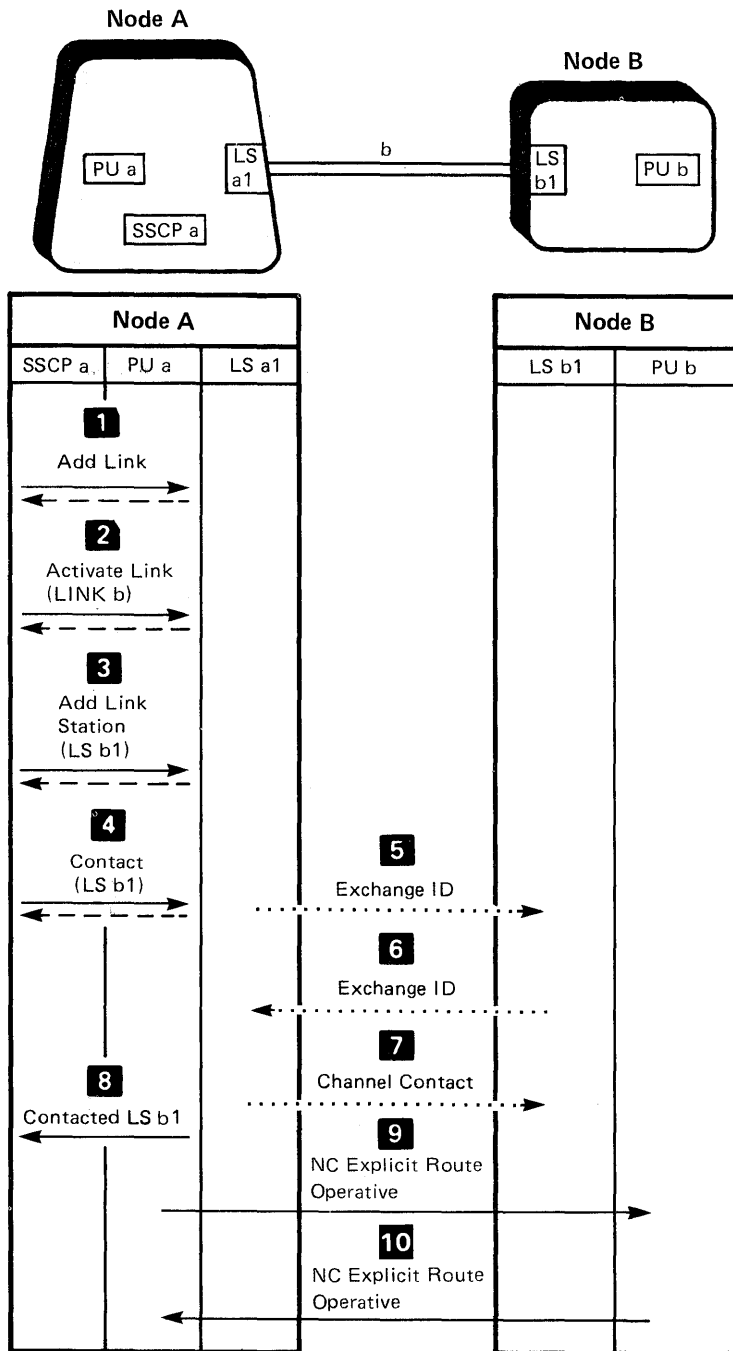


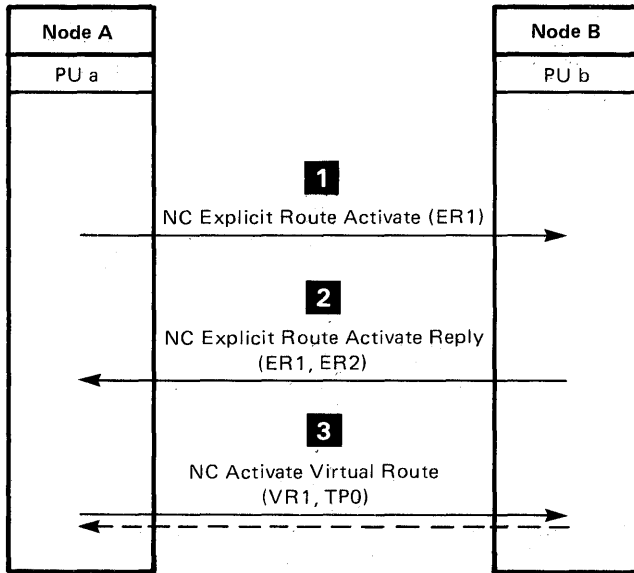
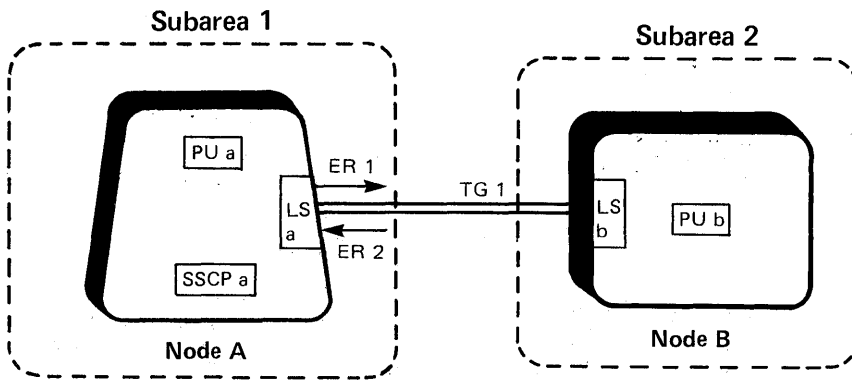
Figure 50. Symbols and Abbreviations for Figures 51 through 56





- 1** SSCP a requests PU a to furnish SSCP a with a network address for the designated link. PU a does so.
- 2** SSCP a tells PU a to activate LINK b and to prepare to issue and receive data link control commands and responses for the link.
- 3** SSCP a requests PU a to furnish SSCP a with a network address for the designated link station. PU a does so.
- 4** SSCP a tells PU a to contact the adjacent link station LS b1. The representation of the link station in Node A has network address of a1.
- 5** PU a sends PU b information about Node A, including the maximum number of bytes that Node A will accept across the channel at one time.
- 6** PU b informs PU a that the parameters sent by PU a are acceptable, and sends PU a information about Node B.
- 7** PU a completes the activation of LINK b by accepting the parameters sent by PU b.
- 8** PU a informs SSCP a that message units can now be sent to PU b through link station LS a1.
- 9** PU a tells PU b which subareas can be reached from Node A, and which explicit routes are used to reach these subareas.
- 10** PU b tells PU a which subareas can be reached from Node B, and which explicit routes are used to reach these subareas.

Figure 51. Activating a Host Node, a Channel-Attached Subarea Node, and the Channel between Them



- 1** PU a initiates the activation of an explicit route between subarea 1 and subarea 2. This route has an explicit route number of 1. Activation of an operative but inactive explicit route is initiated when the route is needed to satisfy a session activation request.
- 2** PU b completes activation of the explicit route between subarea 1 and subarea 2, replying that the reverse explicit route number for this explicit route is 2. (ER1 and ER2 in this case refer to the same explicit route; an explicit route is known by two explicit route numbers – one for each direction.) PU b indicates to PU a that the explicit route has a length of one transmission group. This length is used in determining the pacing group size for virtual route pacing.
- 3** PU a activates a virtual route between sub-areas 1 and 2. This virtual route, which has a virtual route number of 1 and a transmission priority of 0, uses the explicit route identified by explicit route numbers 1 (in the Node A-to-Node B direction) and 2 (in the reverse direction). An inactive virtual route is activated when it is needed to satisfy a session activation request.

**Figure 52. Activating Explicit and Virtual Routes between Adjacent Subarea Nodes**

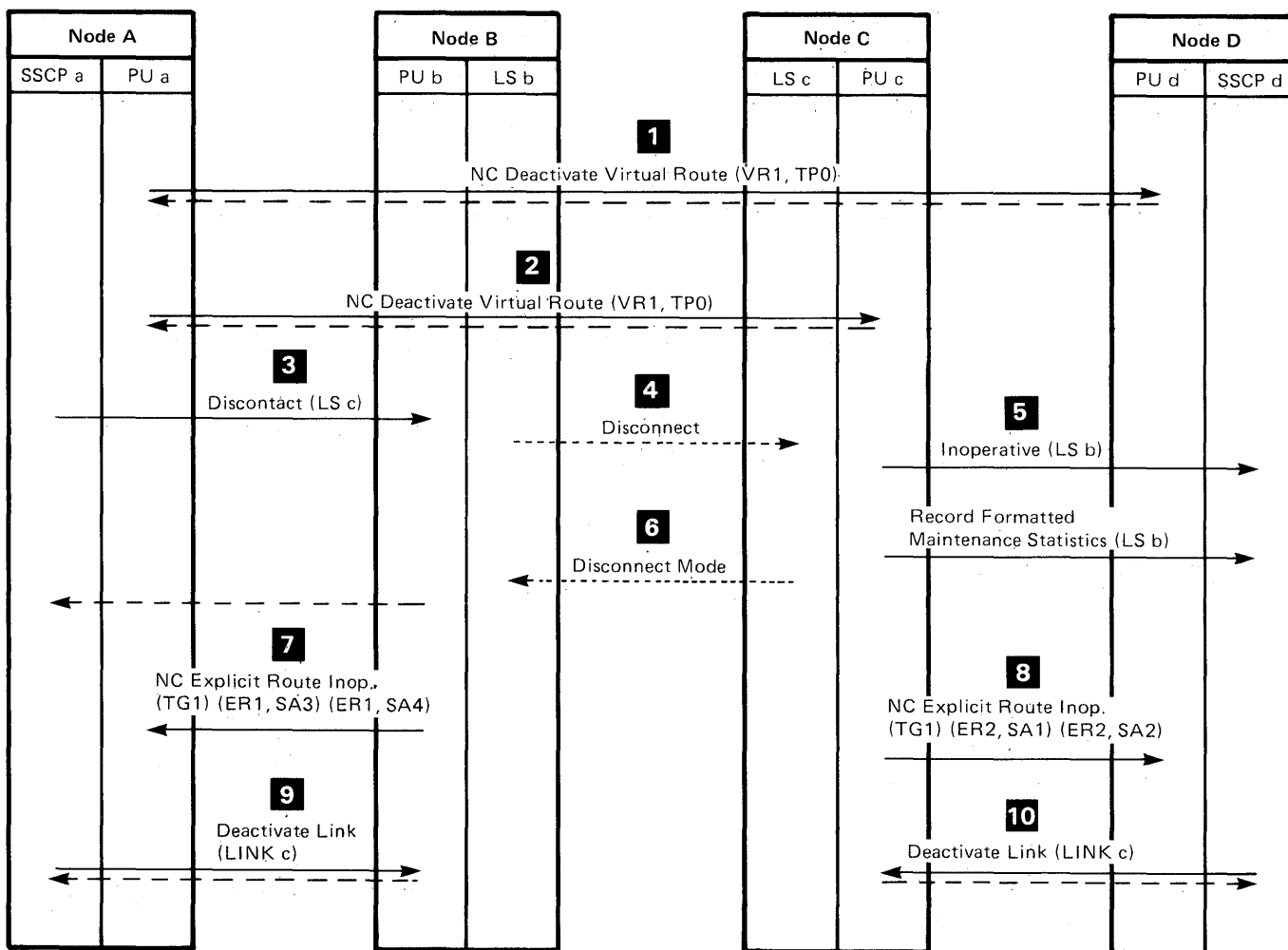
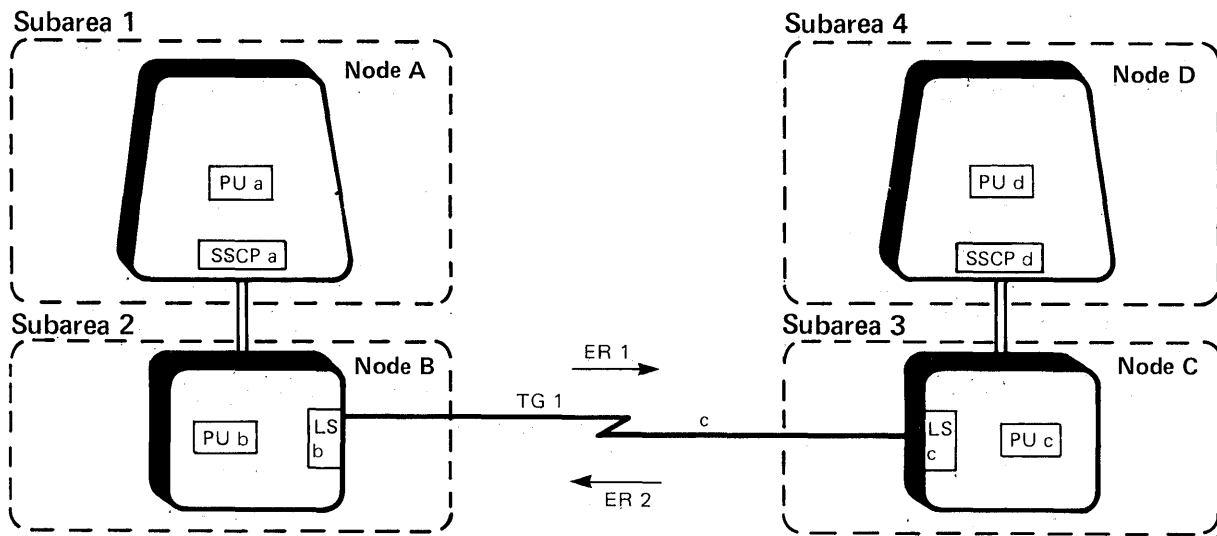
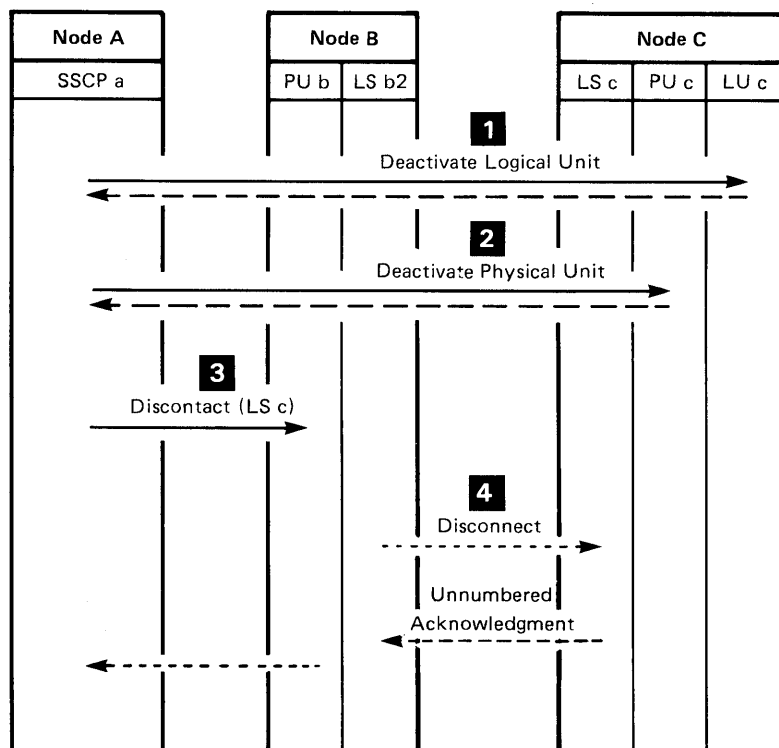
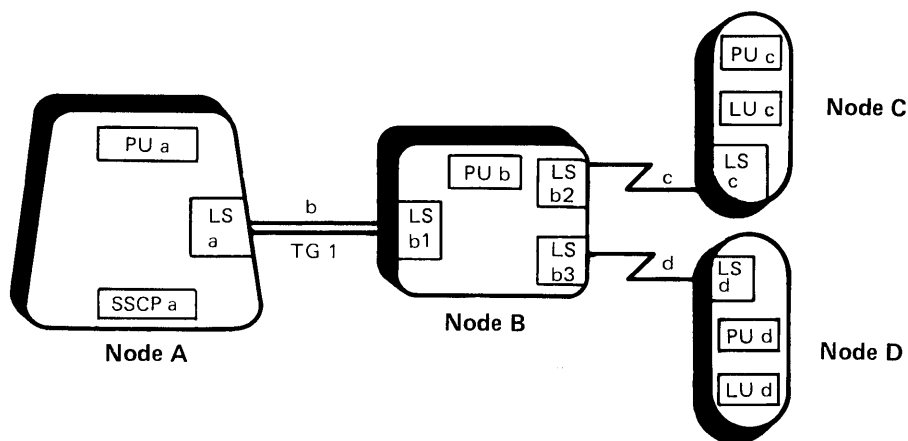


Figure 53 (Part 1 of 2). Deactivating Virtual Routes, Explicit Routes, and SDLC Links

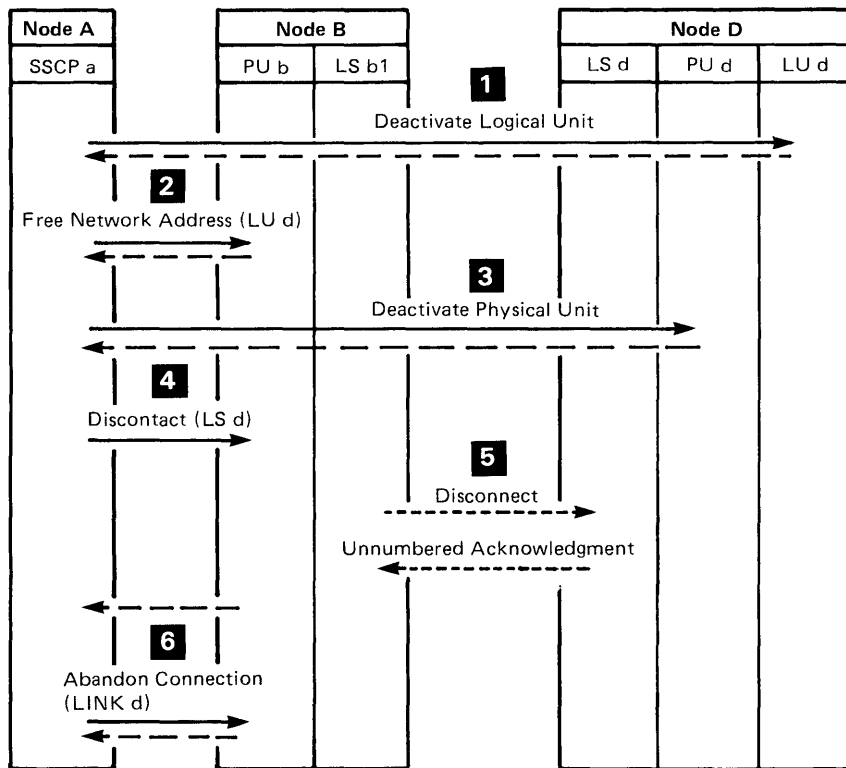
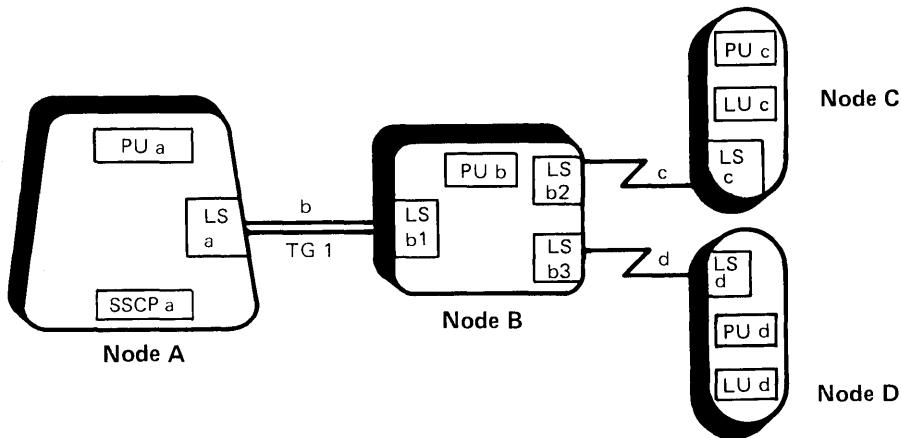
- 1** PU a deactivates the virtual route having a virtual route number of 1 and a transmission priority of 0 between subarea 1 and subarea 4, because the last session assigned to this virtual route has been deactivated.
- 2** PU a deactivates the virtual route having a virtual route number of 1 and a transmission priority of 0 between subarea 1 and subarea 3, because the last session assigned to this virtual route has been deactivated.
- 3** SSCP a tells PU b to break contact with link station LS c. The representation of the LS in Node C has a network address of c.
- 4** PU b causes local link station LS b to initiate data link control procedures to break contact with link station LS c. LS b tells LS c to go into disconnect mode.
- 5** PU c informs SSCP d that link station LS b is inoperative, and sends SSCP d maintenance statistics relating to LS b. The representation of the LS in Node C has a network address of b.
- 6** LS c informs LS b that LS c has gone into disconnect mode.
- 7** PU b informs PU a that TG 1 has had a routing interruption that rendered inoperative ER1 to subarea 3 and ER1 to subarea 4.
- 8** PU c informs PU d that TG 1 has had a routing interruption that rendered inoperative ER2 to subarea 1 and ER2 to subarea 2.
- 9** SSCP a tells PU b to deactivate Node B's end of link c between Node B and Node C.
- 10** SSCP d tells PU c to deactivate Node C's end of link c between Node B and Node C.

**Figure 53 (Part 2 of 2). Deactivating Virtual Routes, Explicit Routes, and SDLC Links**



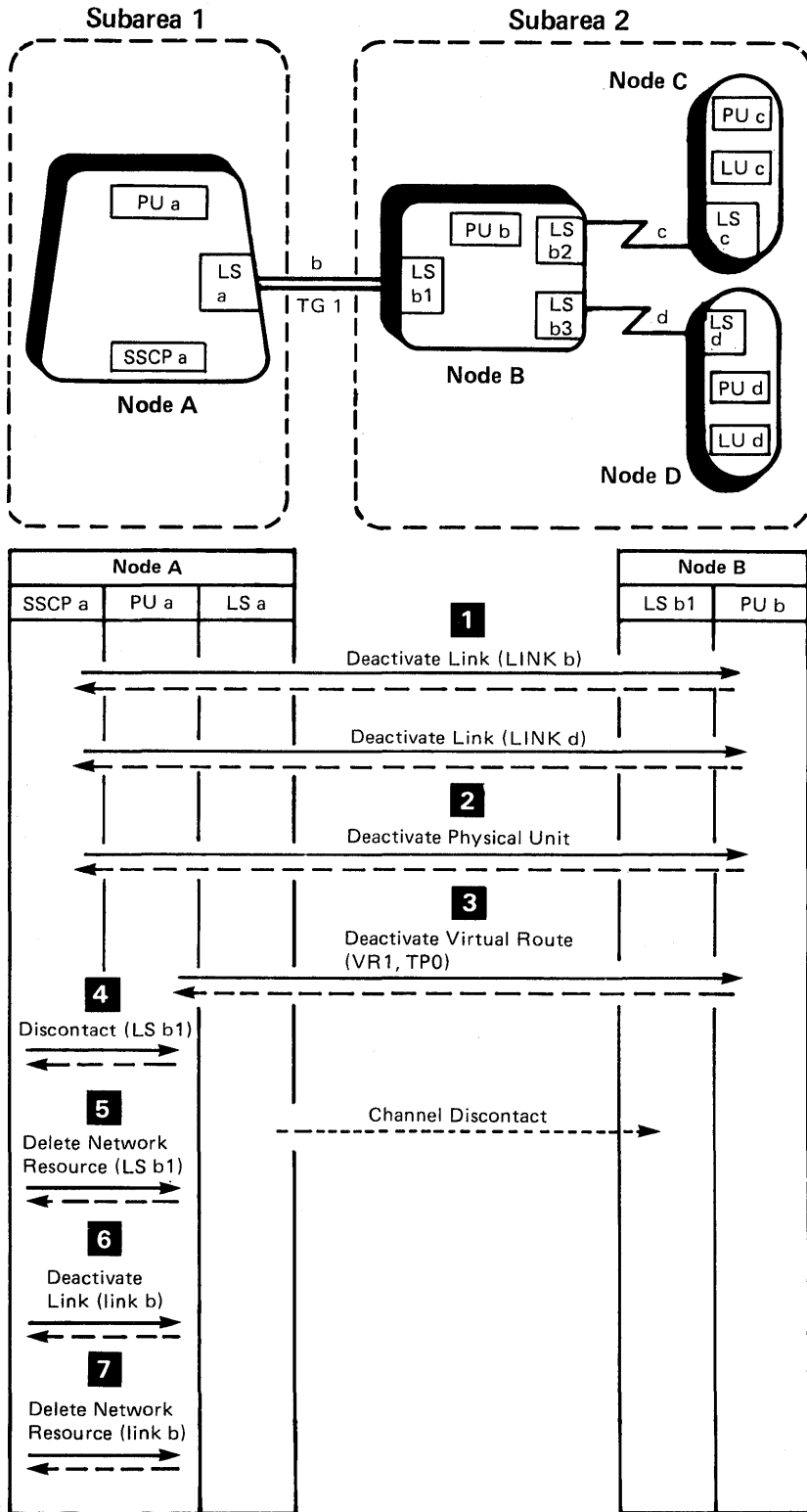
- 1** SSCP a relinquishes control of LU c by deactivating the SSCP-LU session between itself and LU c.
- 2** SSCP a relinquishes control of Node C by deactivating the SSCP-PU session between itself and PU c.
- 3** SSCP a tells PU b to break contact with link station LS c.
- 4** PU b causes local link station LS b2 to initiate data link control procedures to break contact with link station LS c. LS b2 tells LS c to go into disconnect mode, and LS c responds affirmatively.

Figure 54. Deactivating a Peripheral Node Attached by a Nonswitched SDLC Link



- 1** SSCP a relinquishes control of LU d by deactivating the SSCP-LU session between itself and LU d.
- 2** SSCP a tells PU d to disassociate LU d's network address from LU d. The freed network address is returned to a pool in Node B, from which it may be reassigned.
- 3** SSCP a relinquishes control of Node D by deactivating the SSCP-PU session between itself and PU d.
- 4** SSCP a tells PU d to break contact with link station LS d. The representation of LS d in Node B has a network address of d.
- 5** PU b causes local link station LS b to initiate data link control procedures to break contact with link station LS d. LS b tells LS d to go into disconnect mode, and LS d responds affirmatively.
- 6** SSCP a tells PU b to break the switched connection between Node B and Node D (link d). PU b does so.

**Figure 55. Deactivating a Peripheral Node Attached by a Switched SDLC Link**

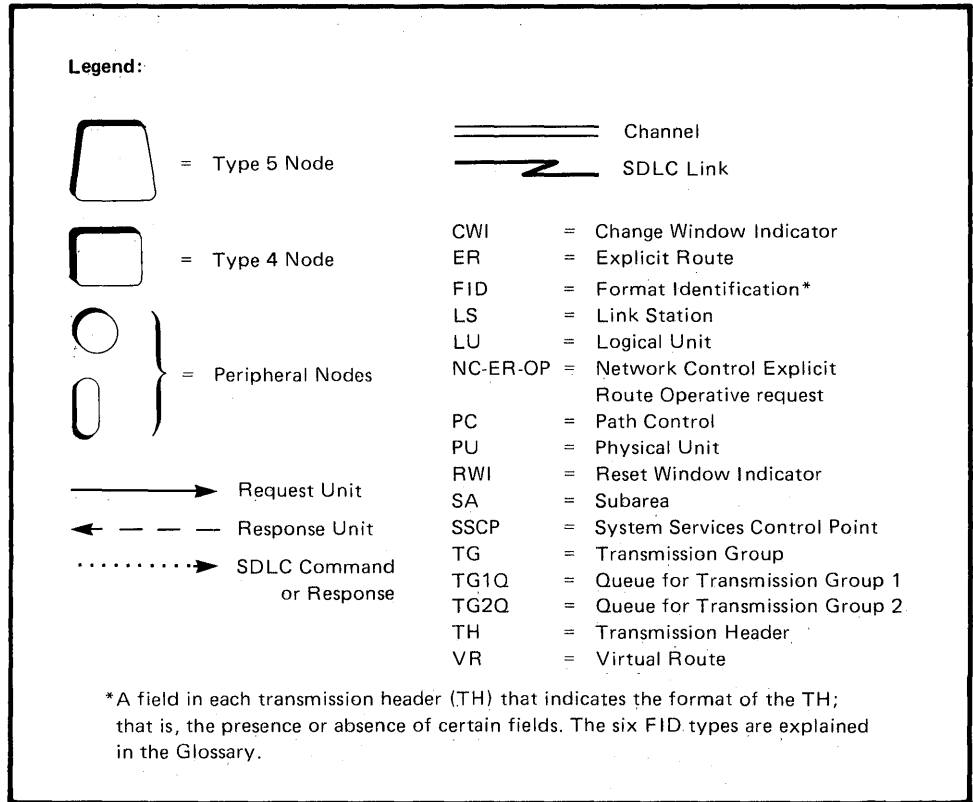


- 1 SSCP a causes PU b to deactivate all links to peripheral nodes in subarea 2.
- 2 SSCP a relinquishes control of Node B by deactivating the SSCP-PU session between itself and PU b.
- 3 PU a deactivates the virtual route having a virtual route number of 1 and a transmission priority of 0 between subarea 1 and subarea 2, because the last session assigned to this virtual route has been deactivated.
- 4 SSCP a initiates a discontact procedure that results in link station LS a breaking contact with link station LS b1.
- 5 SSCP a tells PU a to break the association with the adjacent link station in Node B, thereby rendering its network address available for reassignment.
- 6 SSCP a tells PU a to deactivate Link b.
- 7 SSCP a tells PU a to break the association between Link b and Link b's network address, thereby rendering the network address available for reassignment.

Figure 56. Deactivating a Channel-Attached Subarea Node and Associated Resources

# Typical Request Unit Sequences for Routing

Figure 58 through Figure 59 present typical request unit sequences for routing data through the network. Figure 57 gives the meaning of each of the symbols and abbreviations that appear in these sequence charts. "Chapter 5. Route Design" explains how the path control network routes data between network addressable units.



**Figure 57. Symbols and Abbreviations for Figures 58 and 59**



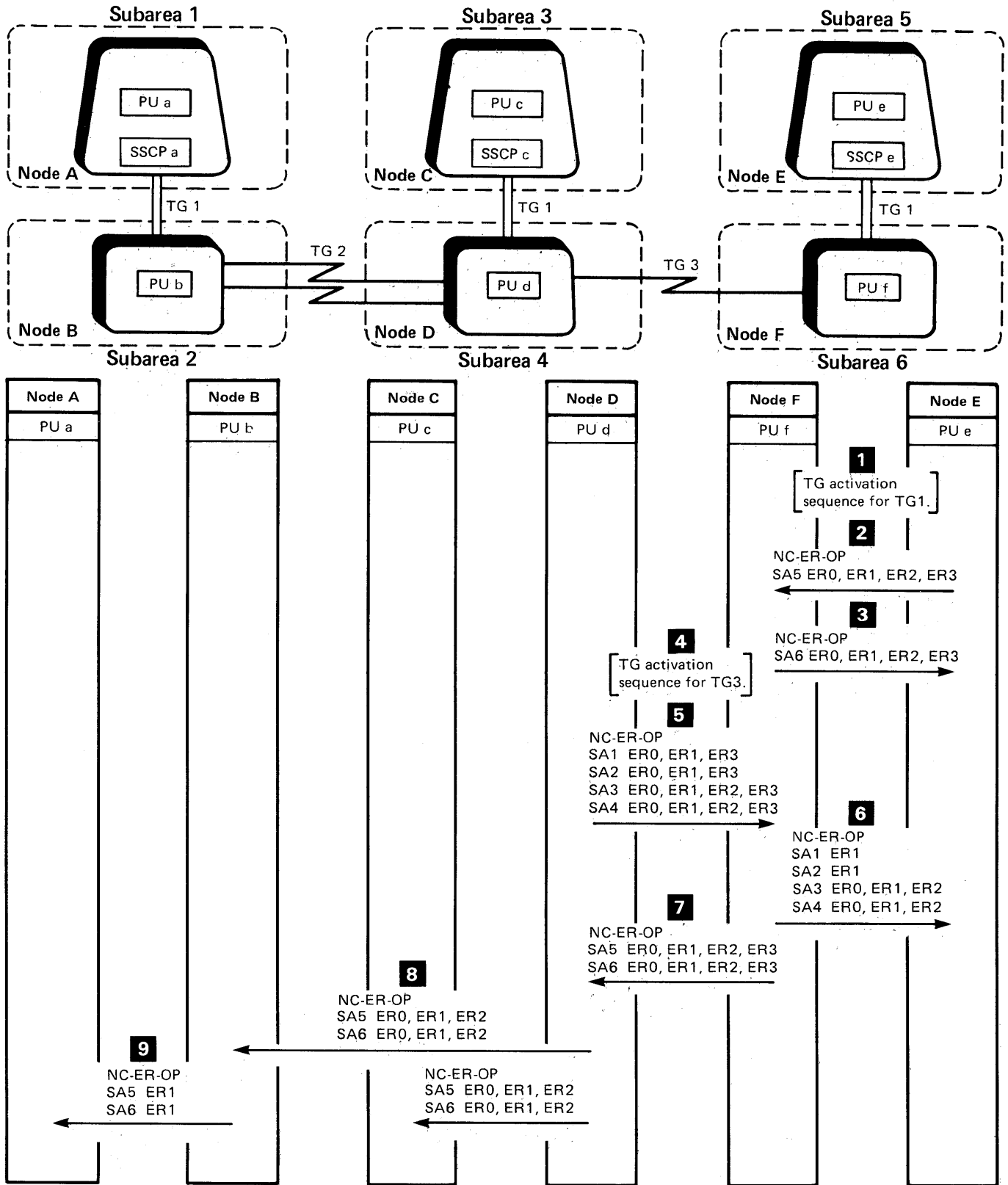
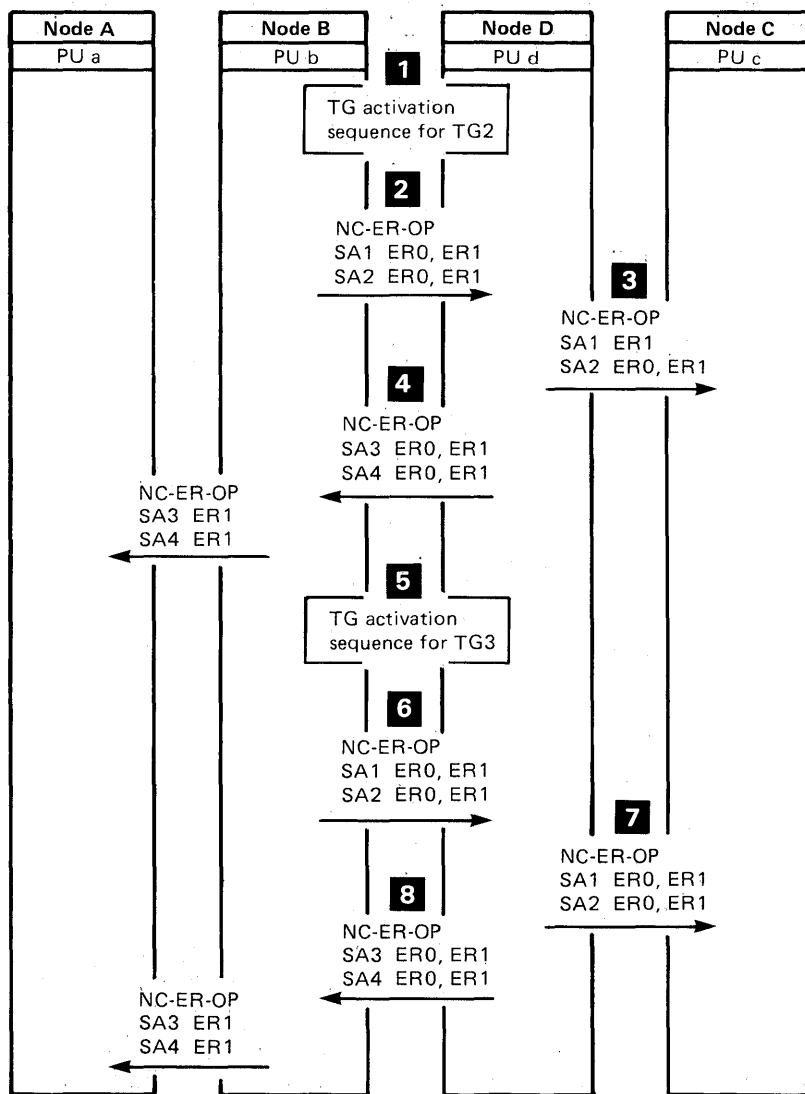
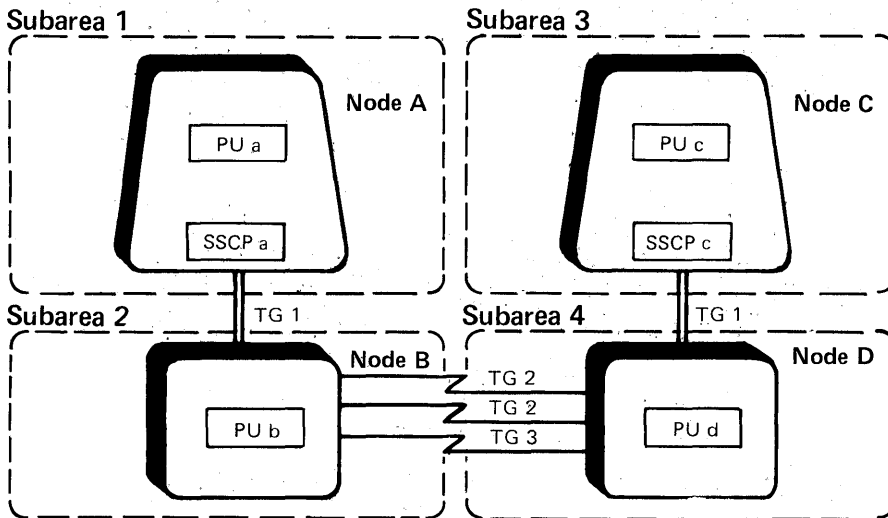


Figure 58 (Part 1 of 2). Propagation of Explicit Route Operative (NC-ER-OP) Requests

This sequence of request units assumes that all transmission groups are active except for TG 1 between Node E and Node F, and for TG 3, which are inactive.

- 1** Transmission group TG 1 is activated.
- 2** PU e tells PU f that PU e can handle data routed to subarea 5 over explicit routes 0, 1, 2, and 3, once these routes are activated. PU f uses this information in deciding which explicit routes to activate.
- 3** PU f tells PU e that PU f can handle data routed to subarea 6 over explicit routes 0, 1, 2, and 3.
- 4** Transmission group TG 3 is activated.
- 5** PU d tells PU f that PU d can handle data routed to the following subareas over the following explicit routes once these routes are activated:
  - Subareas 1 and 2 via explicit routes 0, 1, and 3
  - Subareas 3 and 4 via explicit routes 0, 1, 2, and 3
- 6** PU f tells PU e that PU f can handle data routed to the following subareas over the following explicit routes:
  - Subareas 1 and 2 via explicit route 1
  - Subareas 3 and 4 via explicit routes 0, 1, and 2
- 7** PU f tells PU d that PU f can handle data routed to subareas 5 and 6 over explicit routes 0, 1, 2, and 3.
- 8** After modifying the information from PU f to reflect the information in its own routing table, PU d propagates this information to the PUs in the other subarea nodes connected to it – that is, PU b in Node B and PU c in Node C.
- 9** PU b modifies the information it received from PU d and propagates it to PU a.

**Figure 58 (Part 2 of 2). Propagation of Explicit Route Operative (NC-ER-OP) Requests**



This sequence of request units assumes that transmission groups TG 2 and TG 3 are active and that both TG 1s are inactive.

- 1** TG 2 is activated.
- 2** PU b tells PU d that PU b can reach subareas 1 and 2 over explicit routes 0 and 1.
- 3** PU d tells PU c that PU d can reach subarea 1 over explicit route 1 and subarea 2 over explicit routes 0 and 1. PU d uses TG 3 to reach subarea 1 over explicit route 0; since TG 3 has not yet been activated, explicit route 0 is not yet available.
- 4** PU d sends the appropriate information on connectivity to PU b, which modifies the information to reflect the contents of its own routing tables and then propagates the information to PU a.
- 5** TG 3 is activated.
- 6** PU b sends PU d the same information that it sent in Step 2.
- 7** PU d in turn propagates this information to PU c. Because TG 3 is now active, PU d tells PU c that PU d can now reach subarea 1 via explicit route 0, as well as via explicit route 1.
- 8** PU d sends PU b the same information that it sent in Step 4. PU b propagates this information to PU a.

**Figure 59. Propagation of Routing Information Following Activation of Multiple Transmission Groups between the Same Subareas**

## Typical Request Unit Sequences for Activating Sessions, Deactivating Sessions, and Transferring Data

Figure 61 through Figure 71 present typical request unit sequences for activating and deactivating sessions between network addressable units and for transferring data over a session. Figure 60 gives the meaning of each of the symbols and abbreviations that appear in these sequence charts.

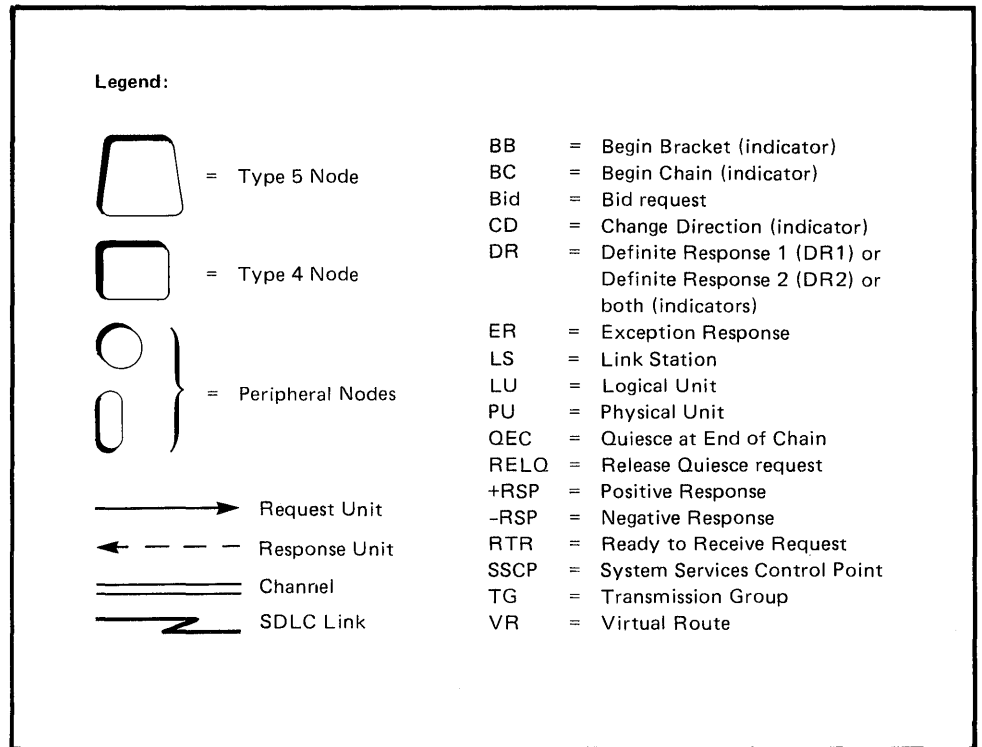
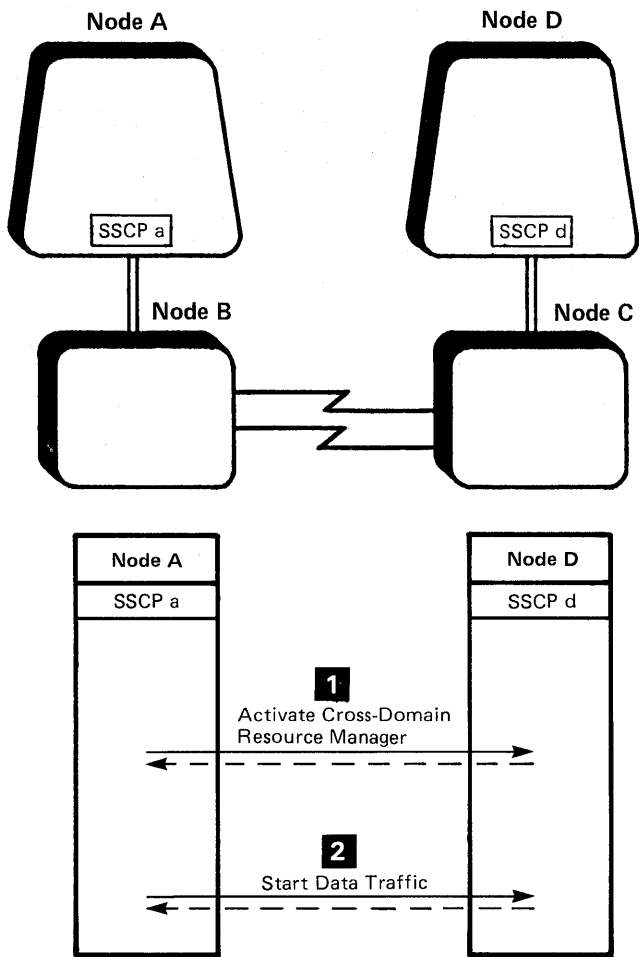


Figure 60. Symbols and Abbreviations for Figures 61 through 72



- 1** SSCP a activates a session with SSCP d.
- 2** SSCP a enables the flow of message units over its SSCP-SSCP session with SSCP d.

Figure 61. Activating an SSCP-SSCP Session

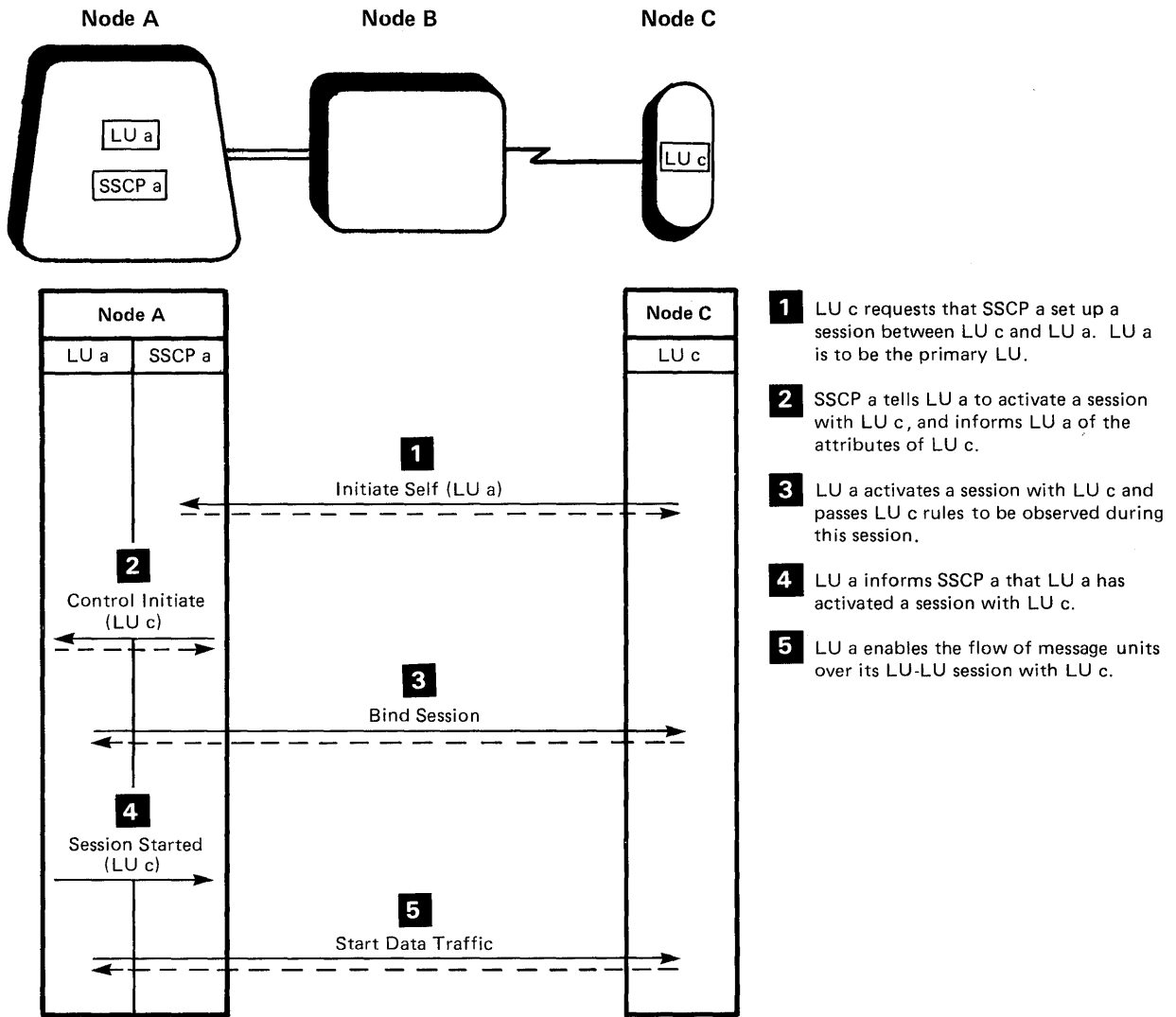


Figure 62. Activating a Same-Domain LU-LU Session

**This page intentionally left blank.**

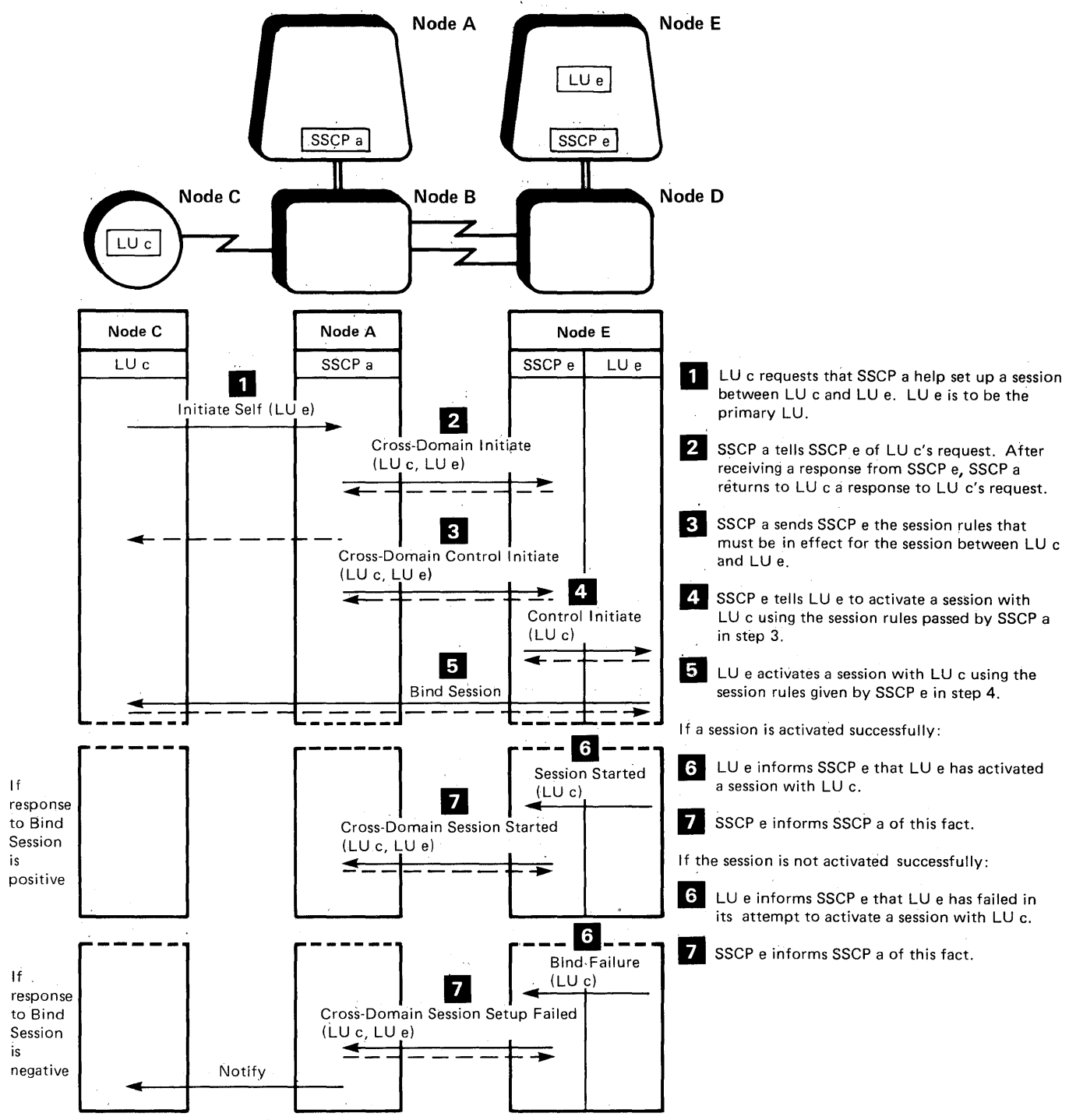


Figure 63. Activating a Cross-Domain LU-LU Session



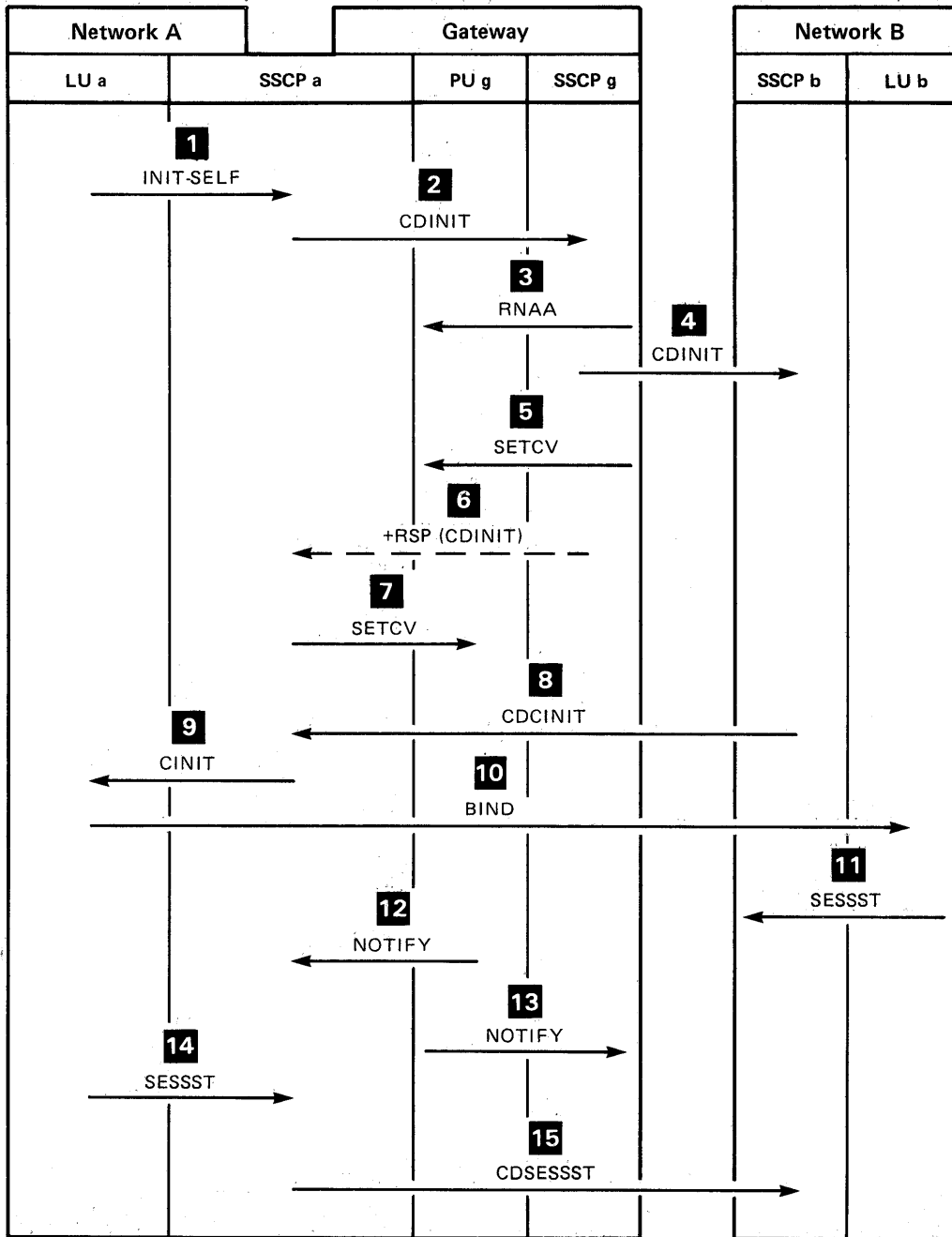
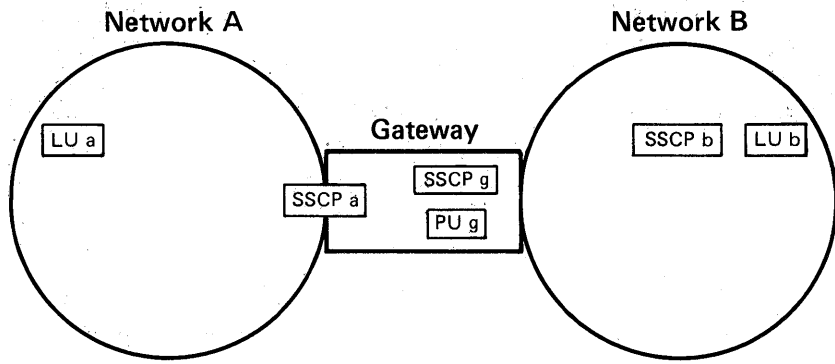
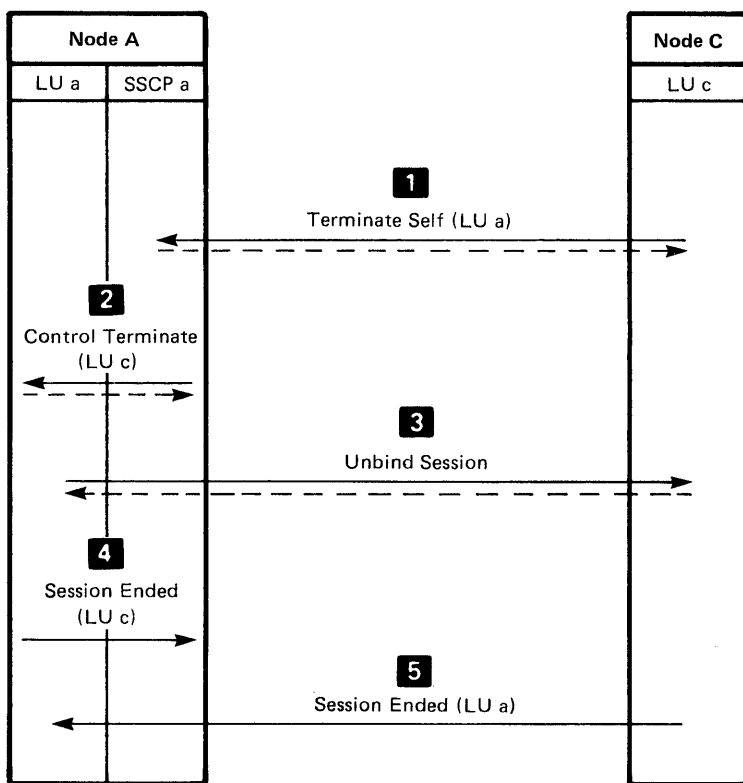
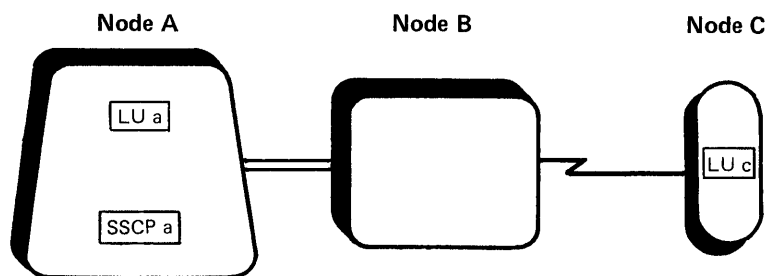


Figure 64 (Part 1 of 2). Activating a Cross-Network LU-LU Session

- 1** LU a sends the SSCP in its domain an Initiate-Self (INIT-SELF) request asking for a session with LU b. LU a is unaware that its SSCP also functions as a gateway SSCP.
- 2** SSCP a sends a Cross-Domain Initiate (CDINIT) request to SSCP g. Between gateway SSCPs, this request carries control vectors.
- 3** Gateway SSCP g sends a Request Network Address Assignment (RNAA) request to PU g, requesting that it assign alias addresses for the LU-LU session. PU g returns the alias addresses.
- 4** SSCP g reroutes SSCP a's CDINIT request to SSCP b, using the alias address that PU g provided in Step 3.
- 5** SSCP g sends PU g a Set Control Vector (SETCV) request to inform it of the real address of LU b in Network B.
- 6** SSCP g reroutes SSCP b's +RSP (CDINIT) to SSCP a.
- 7** SSCP a resolves the class of service (COS) name for LU a to a VR id list, and sends the list and the LU name translation to PU g in a SETCV request.
- 8** SSCP b sends a Cross-Domain Control Initiate (CDCINIT) request containing parameters for the Bind image to SSCP a.
- 9** SSCP a sends a Control Initiate (CINIT) request that contains the Bind image to LU a.
- 10** LU a sends LU b a Bind Session (BIND) request to activate the LU-LU session.
- 11** LU b sends SSCP b a Session Started (SESSST) request to notify it of the successful LU-LU session activation.
- 12 – 13** The gateway node notifies all gateway SSCPs in its gateway of the LU-LU session activation.
- 14** LU a sends SSCP a a SESSST request to inform it of the successful LU-LU session activation.
- 15** SSCP a sends a Cross-Domain Session Started (CDSESSST) request to SSCP b.

**Figure 64 (Part 2 of 2). Activating a Cross-Network LU-LU Session**

This page intentionally left blank.



**Figure 65. Deactivating a Same-Domain LU-LU Session**

- 1** LU c requests that SSCP a help terminate the session between LU a and LU c. LU a is the primary LU.
- 2** SSCP a tells LU a to deactivate its session with LU c.
- 3** LU a deactivates its session with LU c.
- 4** LU a informs SSCP a that LU a has deactivated its session with LU c.
- 5** The boundary function of LU c informs SSCP a that the session between LU c and LU a has been deactivated. (Although the RU bearing this information has the network address of LU c in the origin field of its TH, this RU actually comes from the LU c boundary function in Node B.)

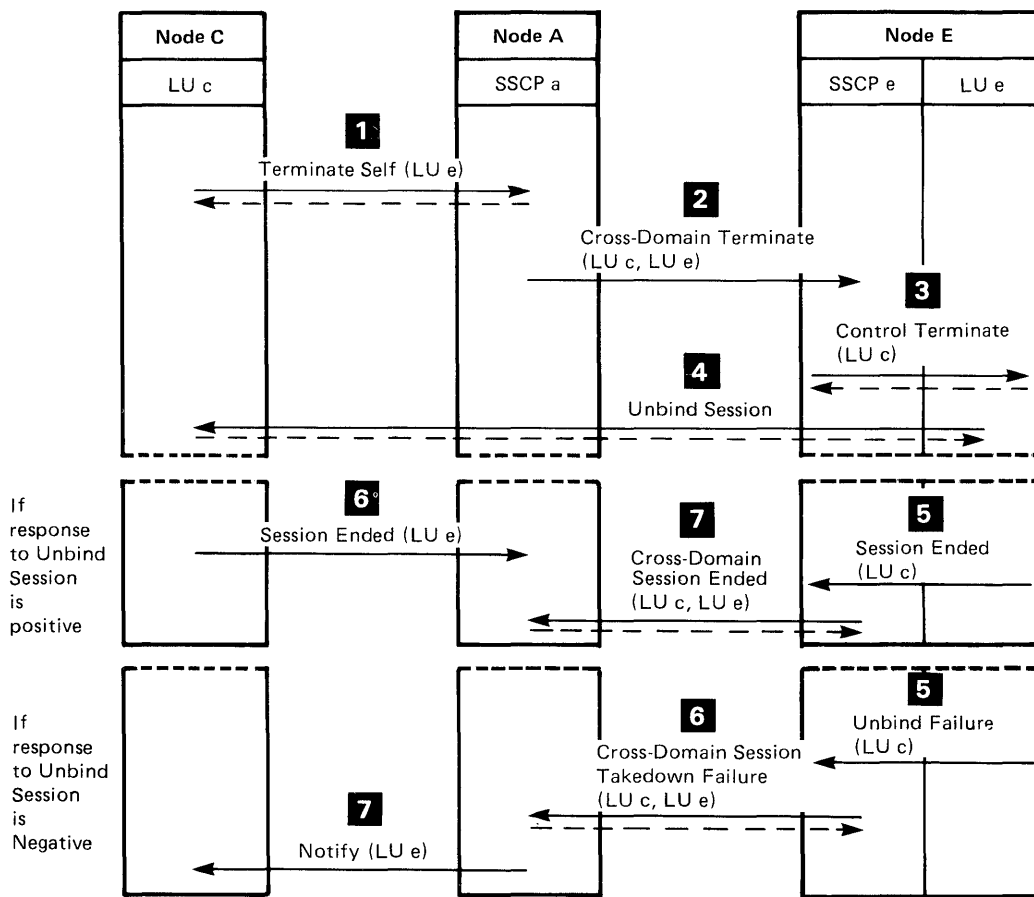
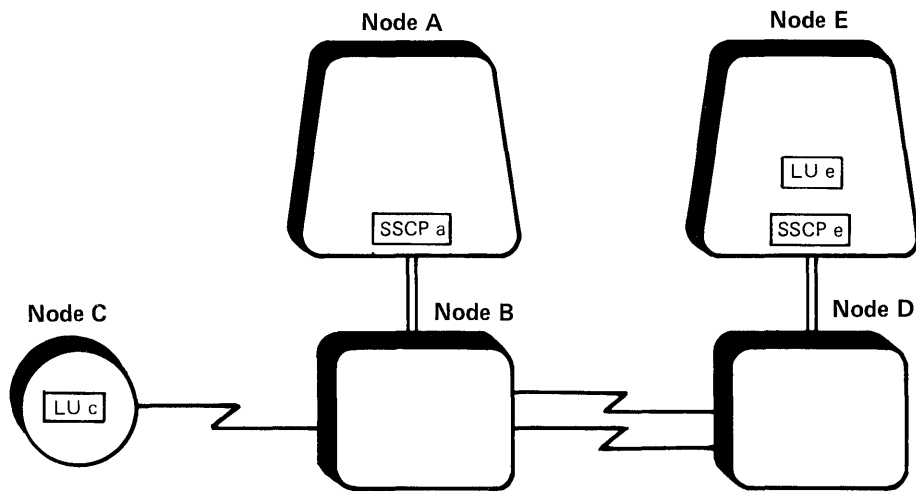


Figure 66 (Part 1 of 2). Deactivating a Cross-Domain LU-LU Session

- 1** LU c requests that SSCP a help deactivate the session between LU c and LU e. LU e is the primary LU.
- 2** SSCP a tells SSCP e of LU c's request.
- 3** SSCP e tells LU e to deactivate its session with LU c.
- 4** LU e deactivates its session with LU c.

If the session is deactivated successfully:

- 5** LU e informs SSCP e that LU e has deactivated its session with LU c.
- 6** The boundary function of LU c informs SSCP a that the session between LU c and LU e has been deactivated. (Although the RU bearing this information has the network address of LU c in the origin field of its TH, this RU actually comes from the LU c boundary function in Node B.)
- 7** SSCP e informs SSCP a that the cross-domain session has been deactivated.

If the session is not deactivated successfully:

- 5** LU e informs SSCP e that LU e has failed in its attempt to deactivate a session with LU c.
- 6** SSCP e informs SSCP a that the attempt to deactivate the session between LU c and LU e has failed.
- 7** SSCP a informs LU c that the LU e believes that LU e's attempt to deactivate the session has failed.

**Figure 66 (Part 2 of 2). Deactivating a Cross-Domain LU-LU Session**

This page intentionally left blank.

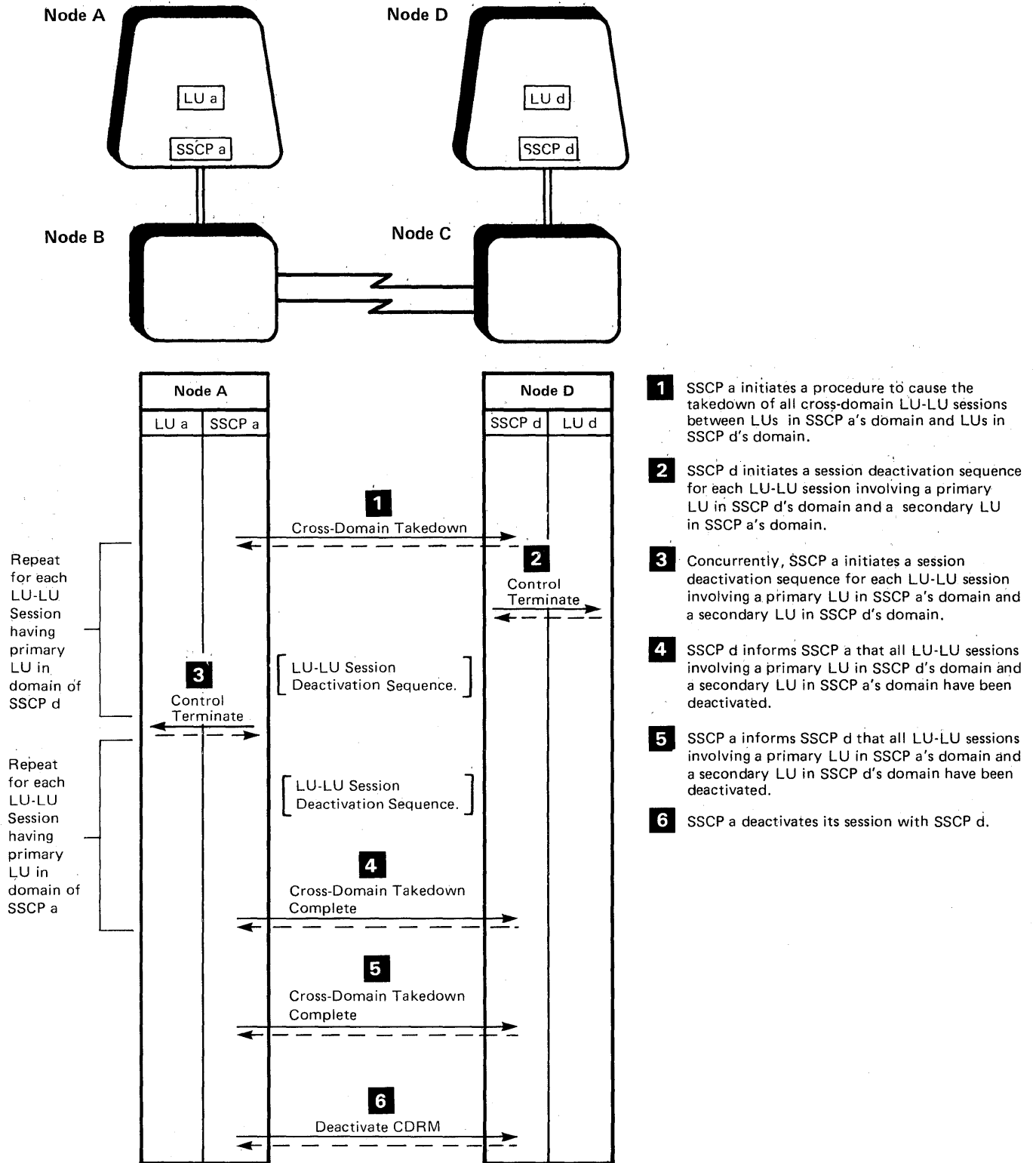
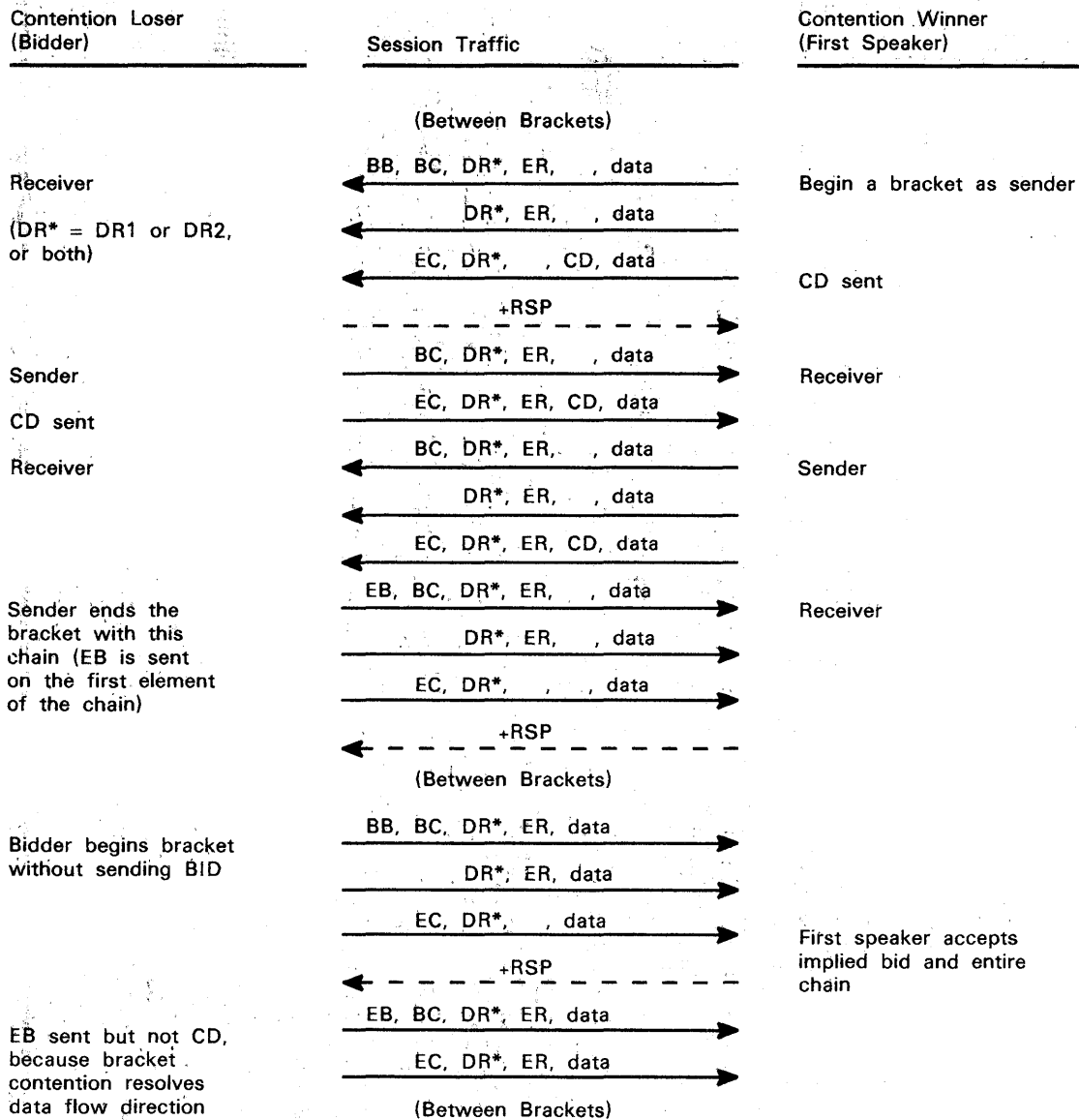


Figure 67. Cross-Domain Takedown Sequence





**Figure 68 (Part 1 of 2). Communication Using Brackets in a Half-Duplex Flip-Flop Mode**

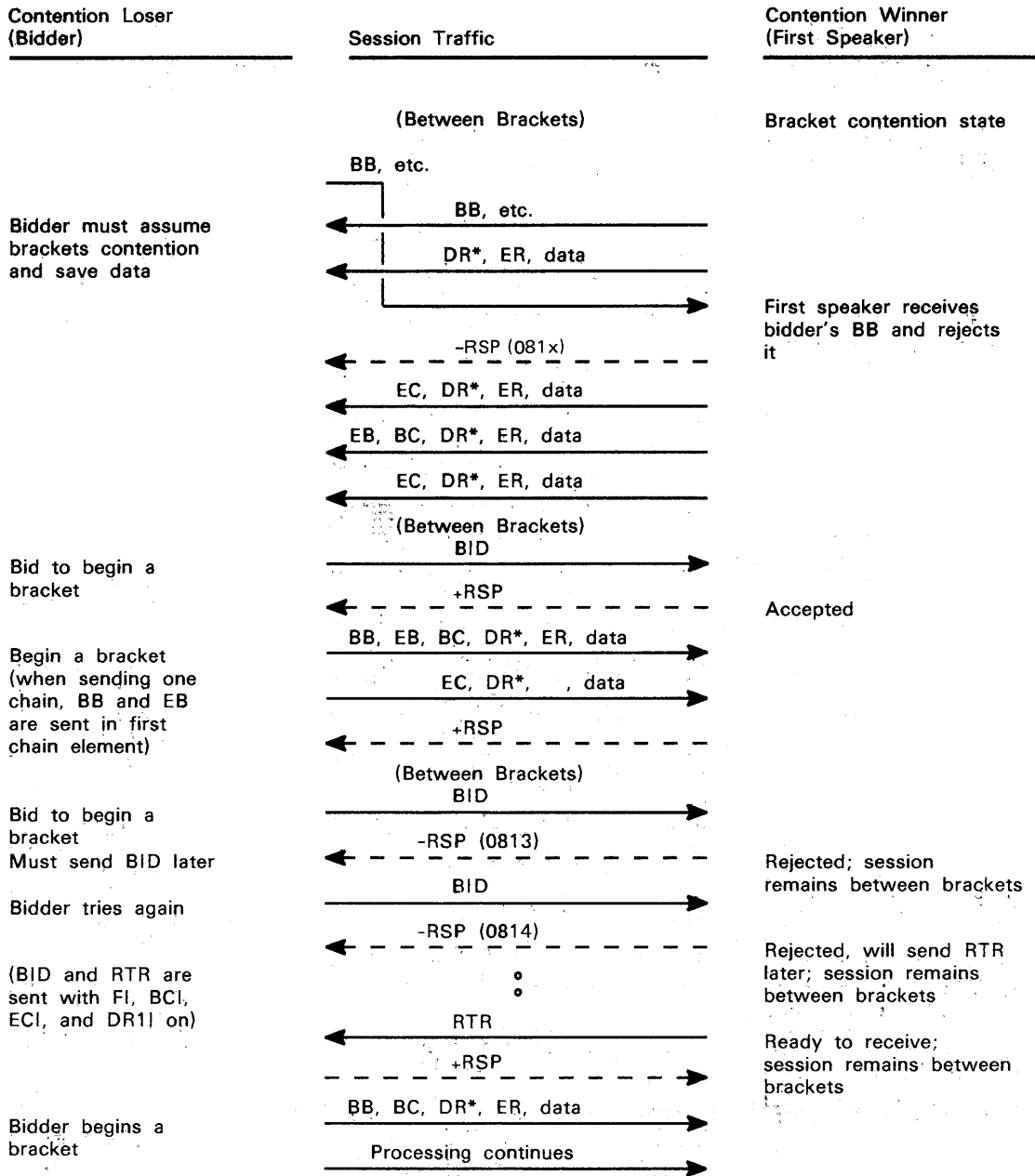
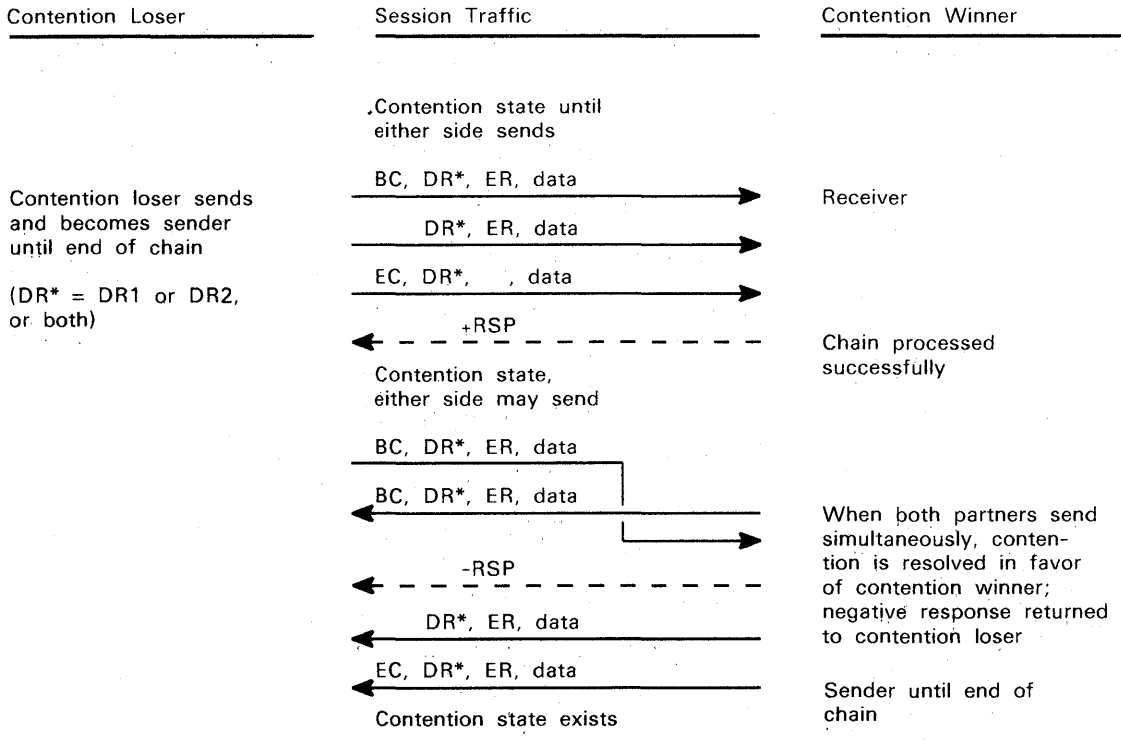
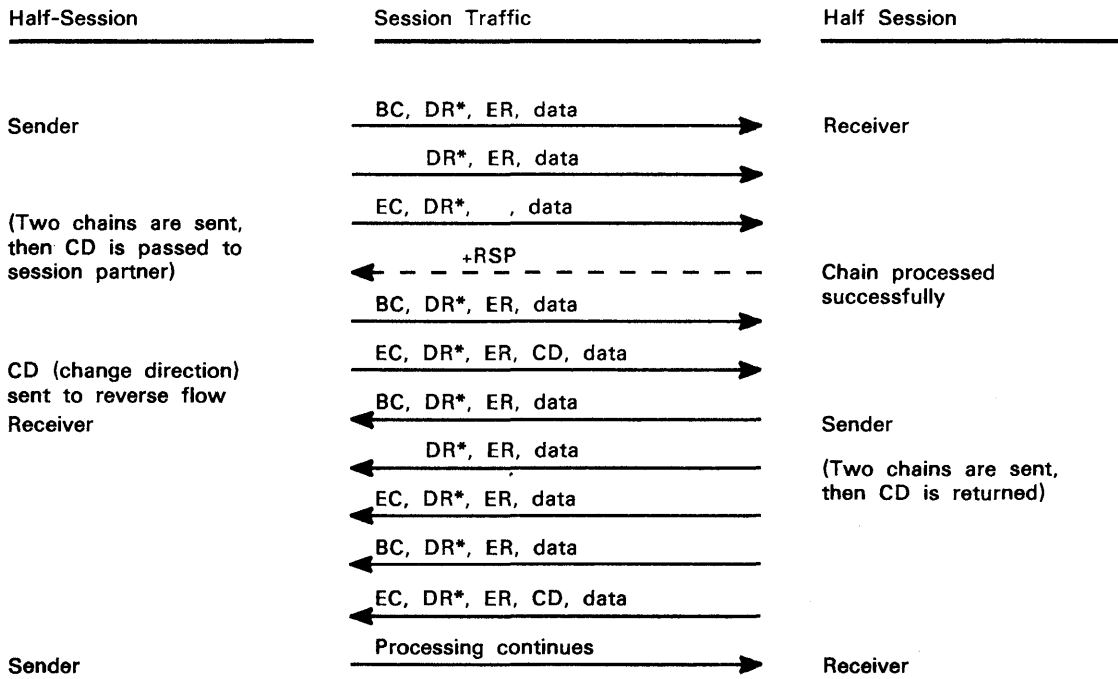


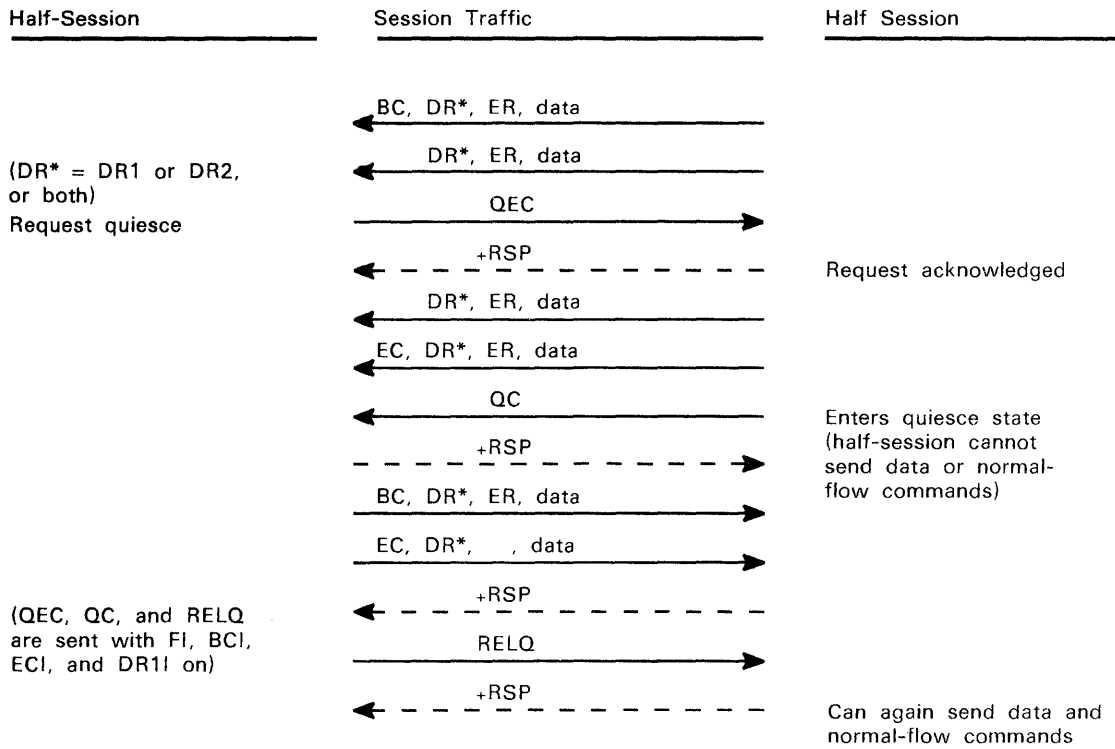
Figure 68 (Part 2 of 2). Communication Using Brackets in a Half-Duplex Flip-Flop Mode



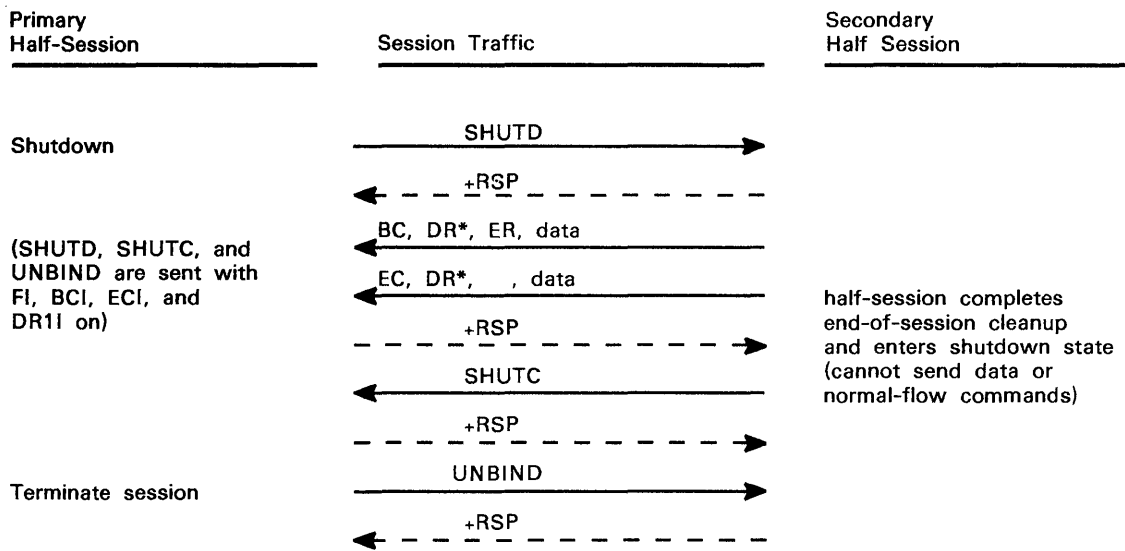
**Figure 69. Communication Using Half-Duplex Contention Protocols**



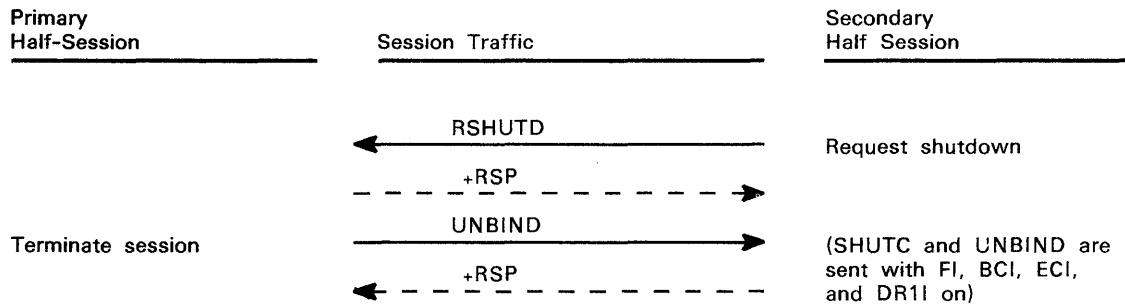
**Figure 70. LU-LU Communication Using Half-Duplex Flip-Flop Protocols**



**Figure 71. Protocols for Quiescing Data Flow**



(a) Orderly shutdown initiated by primary half-session.



(b) Orderly shutdown initiated by secondary half-session.

**Figure 72. Protocols for Deactivating LU-LU Sessions**



### Specifying Parameters for System Definition

#### Specifying Activation, Deactivation, and Control Options

This section explains how system programmers specify activation, deactivation, and control options to ACF/VTAM.

##### ACF/VTAM Options

In order for ACF/VTAM to assume control of an SNA resource, the resource must be defined to it. A PU or LU definition statement defines a channel-attached resource; the appropriate NCP macro instruction from the following list defines an NCP-controlled resource:

- GROUP
- LINE
- SERVICE
- PU
- LU

For NCP-controlled resources, ACF/VTAM must also be able to access an NCP resource-resolution table that contains entries for the appropriate resources. This table is created during NCP generation as a result of the system programmer coding GROUP, LINE, PU and LU macros for the appropriate resources.

The system programmer specifies the initial activation status of an SNA resource in one of two ways, depending on whether the resource is attached directly to a channel or is under the control of an NCP.

The system programmer specifies the initial activation status of a channel-attached resource via the ISTATUS operand of the definition statement for the resource. The system programmer specifies the initial activation status of an NCP-controlled resource via the ISTATUS operand of the NCP macro instruction that defines the resource.

The network operator can use the VARY operator command to alter the activation status of an SNA resource.



## Specifying Links and Associated Resources to SNA Products

Associated with each secondary link station in SNA networks are two link-level address parameters: (1) a set of receive addresses (addresses to which the secondary link station will respond) and (2) a unique send address. Associated with each primary link station is a single primary link station address. The connection between the primary station and all secondary stations is a link connection.

### Defining SDLC Links, Link Stations, and Transmission Groups

Access methods in host processors and network control programs in communication controllers define SDLC links, link stations, and transmission groups. Each program product (ACF/VTAM and ACF/NCP) has a specific way of defining SDLC links.

#### ACF/NCP Definition

To define SDLC links for ACF/NCP, a programmer codes the ADDRESS and SPEED operands of LINE macros. The ADDRESS operand specifies the line interface address of the SDLC link. The SPEED operand specifies the data rate on the link in bits per second.

To define a link station for ACF/NCP, a programmer codes the TYPE operand of the PU macro.

To define a transmission group for ACF/NCP, a programmer specifies its transmission group number in the TGN operand of the PU macro.

#### ACF/VTAM Definition

ACF/VTAM relies on, and refers to, macros and operands coded in ACF/NCP for the definitions of SDLC links, link stations, and transmission groups.

## Specifying Routes

To specify routes between subareas, the system programmer must define:

- Links between adjacent subareas
- Transmission groups in which the links reside
- Explicit routes between end subareas
- Virtual routes associated with each explicit route
- The correspondence between the class of service requested at session initiation and the particular virtual routes that session can use.

"Chapter 5. Route Design" explains how to define links between adjacent subarea nodes.

Transmission groups and explicit routes are defined to ACF/NCP. ACF/VTAM provides statements that allow the system programmer to resolve potential

routing conflicts. Virtual routes and class of service are defined primarily to ACF/VTAM.

To define explicit routes among subareas using the facilities described below can be tedious for large networks. IBM provides a field-developed program, called the Routing Table Generator (RTG), that uses the physical network topology to select routes between SNA subareas, assign explicit-route numbers to the routes, and generate appropriate macros to define routing tables in nodes along the routes.

## Specifying Routes to ACF/NCP

To associate a link with a transmission group, the system programmer codes the transmission group number in the TGN operand of the PU macro that represents the adjacent link station for the link to the NCP.

The system programmer defines explicit routes by coding PATH macros in each NCP that serves as a node on the route. These PATH macros tell the NCP where to send message units that specify a particular destination subarea and that are assigned to a particular explicit route. For each combination of destination subarea and explicit route, the macros specify which adjacent subarea to send the message units to and which transmission group to use in sending them to that subarea. Using information that the PATH macros provide, each NCP builds an explicit-routing table similar to that shown in Figure 37.

Although NCP macros are not used to define virtual routes, three operands of the NCP BUILD macro are used to reserve storage that the NCP needs to handle virtual routes that have one end in the NCP node:

- The VRPOOL operand specifies the number of virtual routes, ending at the NCP, that can be concurrently active.
- The NUMHSAS operand specifies the number of origin subareas that can concurrently communicate with the NCP as a destination subarea.
- The MAXSSCP operand specifies the maximum number of SSCPs that can concurrently maintain active SSCP-PU sessions with the NCP.

## Specifying Routes to ACF/VTAM

ACF/VTAM builds its routing tables using information contained in one or more path definition sets filed in the ACF/VTAM definition library. A path definition set consists of one or more ACF/VTAM PATH statements. These statements are similar in format to the ACF/NCP PATH macros. In addition to telling ACF/VTAM to which adjacent subarea to send message units, and which transmission group to use in doing so, the PATH statement also assigns virtual routes to explicit routes. VRN operands in the PATH statements are used for this purpose.

ACF/VTAM provides a user-replaceable module to calculate minimum and maximum pacing-group sizes for virtual-route pacing. The system programmer can replace the VTAM-supplied module with a different module if necessary.

ACF/VTAM provides the COSTAB, COS, and COSEND macros to define a class-of-service table that associates a set of virtual routes with a class of service. An LU specifies a class of service when it initiates a session; when activated, the session is assigned to the first available virtual route that provides its class of service. (A virtual route is available if it is either active or currently capable of being activated.) ACF/VTAM provides a default algorithm for assigning sessions to virtual routes when no class-of-service table is provided.

The system programmer can associate a class of service with a log-on mode table entry by coding the COS operand of the MODEENT macro that defines the table entry.

ACF/VTAM provides a virtual-route selection exit for which the system programmer can provide a routine to examine and modify ACF/VTAM's virtual-route selection process. ACF/VTAM invokes this exit routine whenever a session between a primary LU in the ACF/VTAM subarea and an LU in another subarea is about to be activated.

If none of the virtual routes listed in a specified class of service (or in an exit routine) is available for an LU-LU session, ACF/VTAM rejects the request to activate a session and informs the ACF/VTAM operator of the situation.

**List of Abbreviations**  
**Glossary**  
**Index**

## **Contents**

List of Abbreviations 181  
Glossary 183  
Index 195



## List of Abbreviations

<b>ACF</b>	Advanced Communications Function	<b>GDS</b>	general data stream
<b>ACF/NCP</b>	Advanced Communications Function for the Network Control Program	<b>ISO</b>	International Standards Organization
<b>ACF/VTAM</b>	Advanced Communications Function for the Virtual Telecommunications Access Method	<b>LH</b>	link header
<b>BBI</b>	begin bracket indicator	<b>LT</b>	link trailer
<b>BCI</b>	begin chain indicator	<b>LU</b>	logical unit
<b>BIU</b>	basic information unit	<b>NCP</b>	Network Control Program
<b>BLU</b>	basic link unit	<b>OII</b>	Office Information Interchange
<b>BTU</b>	basic transmission unit	<b>OSI</b>	Open Systems Interconnection
<b>CCITT</b>	International Telegraph and Telephone Consultative Committee	<b>PIU</b>	path information unit
<b>CDI</b>	change direction indicator	<b>PNCP</b>	peripheral node control point
<b>CEBI</b>	conditional end bracket indicator	<b>PU</b>	physical unit
<b>COS</b>	class of service	<b>PUCP</b>	physical unit control point
<b>DIA</b>	Document Interchange Architecture	<b>RH</b>	request header or response header
<b>DR1I</b>	definite response 1 indicator	<b>RRI</b>	request/response indicator
<b>DR2I</b>	definite response 2 indicator	<b>RTI</b>	response type indicator
<b>EBI</b>	end bracket indicator	<b>RU</b>	request unit or response unit
<b>ECI</b>	end chain indicator	<b>SDLC</b>	Synchronous Data Link Control
<b>EDI</b>	enciphered data indicator	<b>SNA</b>	Systems Network Architecture
<b>ERI</b>	exception response indicator	<b>SNADS</b>	SNA Distribution Services
<b>FID</b>	format identification	<b>SSCP</b>	system services control point
<b>FM</b>	function management	<b>TH</b>	transmission header
		<b>VTAM</b>	Virtual Telecommunications Access Method



# Glossary

This glossary defines the Systems Network Architecture (SNA) terms and abbreviations that this manual uses. If you do not find the term that you are looking for, refer to the Index or to the *IBM Vocabulary for Data Processing, Telecommunications and Office Systems*, GC20-1699.

## A

---

**ACF.** Advanced Communications Function.

**adjacent link station.** A link station that is directly connected to a given node by a link connection, over which network traffic can be carried.

**adjacent nodes.** Two nodes that are connected by one or more links with no intervening nodes.

**Advanced Communications Function (ACF).** A group of IBM program products (principally ACF/VTAM and ACF/NCP) that conform to Systems Network Architecture (SNA).

**Advanced Communication Function for the Network Control Program (ACF/NCP).** A program product that provides communication controller support in an SNA network.

**Advanced Communications Function for the Virtual Telecommunications Access Method (ACF/VTAM).** A program product that provides host processor support in an SNA network.

**alias address.** A network address that identifies in one network a logical unit (LU) or a system services control point (SSCP) that resides in an interconnected network. Contrast with *real address*.

**alias name.** A name that identifies in one network a logical unit (LU) or system services control point (SSCP) that resides in an interconnected network. Contrast with *real name*.

**application program.** A program written for or by a user to perform the user's work; in an SNA network, an end user.

**application transaction program.** A program written for or by a user to process the user's application; in an SNA network, an end user of a type 6.2 logical unit. See also *transaction program*. Contrast with *service transaction program*.

## B

---

**basic information unit (BIU).** The unit of data and control information that is passed between half-sessions. It consists of a request/response header (RH) followed by a request/response unit (RU).

**basic link unit (BLU).** The unit of data and control information transmitted over a link by data link control.

**basic transmission unit (BTU).** The unit of data and control information passed between path control components. A BTU can consist of one or more path information units (PIUs). See also *blocking of PIUs*, *segmenting of PIUs*.

**begin bracket.** The value (binary 1) of the begin-bracket indicator in the request header (RH) of the first request in the first chain of a bracket; the value denotes the start of a bracket. Contrast with *conditional end bracket*, *end bracket*.

**bidder.** The LU-LU half-session defined at session activation as having to request and receive permission from the other LU-LU half-session in order to begin a bracket. Contrast with *first speaker*.

**binary synchronous communication (BSC).** A data link control procedure, using a uniform set of control characters and control character sequences for synchronous transmission of binary-coded data between link stations.

**BIND.** Bind Session.

**Bind image.** The session parameters that the system services control point (SSCP) sends to the primary logical unit (PLU) and the PLU sends in the BIND request to the secondary logical unit (SLU); these parameters specify the proposed protocol options for an LU-LU session.



**Bind Session (BIND).** A request to activate a session between two logical units. See also *session-activation request*. Contrast with *Unbind Session (UNBIND)*.

**BIU.** Basic information unit.

**BIU segment.** The portion of a basic information unit (BIU) that is contained within a path information unit (PIU). A BIU segment consists of either a request/response header (RH), followed by all or part of a request/response unit (RU), or of only a part of an RU.

**blocking of PIUs.** An optional function of path control that combines multiple path information units (PIUs) into a single basic transmission unit (BTU).

**BLU.** Basic link unit.

**boundary function.** (1) A capability of a subarea node to provide protocol support for attached peripheral nodes, such as: (a) interconnecting subarea path control and peripheral path control elements, (b) performing session sequence numbering for low-function peripheral nodes, and (c) providing session-level pacing support. (2) The component that provides these capabilities. See also *gateway function*, *intermediate routing function*.

**boundary node.** A subarea node with boundary function.

**bracket.** One or more chains of request units (RUs) and their responses that are exchanged between two LU-LU half-sessions and that represent a transaction between them. One bracket must be completed before another one can be started. Examples of brackets are data base inquiries/replies, update transactions, and remote job entry output sequences to workstations. See also *begin bracket*, *conditional end bracket*, *end bracket*, *RU chain*.

**bracket protocol.** A data flow control protocol in which exchanges between the two LU-LU half-sessions are controlled through the use of brackets; one LU is designated at session activation as the first speaker and the other LU as the bidder. The bracket protocol involves bracket initiation and termination rules. See also *bidder*, *conversation*, *first speaker*.

**BSC.** Binary synchronous communication.

**BTU.** Basic transmission unit.

## C

---

**chain.** See *RU chain*.

**channel.** See *data channel*.

**class of service.** A designation of the path control network characteristics, such as path security, transmission priority, and bandwidth, that apply to a particular session. At session initiation a class of service designation is mapped into a list of virtual routes; any one of the virtual routes can be selected for the session to provide the requested class of service.

**cluster controller.** A peripheral node that can control a variety of devices.

**command.** (1) Any field set in the transmission header (TH), request header (RH), and sometimes portions of a request unit, that initiates an action or that begins a protocol; for example: (a) Bind Session (session-control request unit), a command that activates an LU-LU session, (b) the change-direction indicator in the RH of the last RU of a chain, (c) the virtual-route reset window indicator in a FID4 transmission header. (2) Loosely, a request unit. (3) In SDLC, the control information (in the C-field of the link header) sent from the primary station to the secondary station.

**communication controller.** A type of communication control unit whose operations are controlled by one or more programs stored and executed in the unit, for example, the IBM 3725 Communications Controller.

**communication controller node.** A communication controller that contains a network control program.

**conditional end bracket.** The value (binary 1) of the conditional end bracket indicator in the request header (RH) of the first request in the last chain of a bracket; the value denotes the conditional end of the bracket. Contrast with *begin bracket*, *end bracket*.

**configuration.** (1) The arrangement of a computer system or network as defined by the nature, number, and chief characteristics of its functional units. The term can refer to both hardware and software configurations. (2) The machines and programs that make up a network.

**configuration services.** One of the types of network services in the system services control point (SSCP) and in the physical unit (PU); configuration services activate, deactivate, and maintain the status of physical units, links, and link stations. Configuration services also shut down and restart network elements and modify path-control routing tables and address-transformation tables. See also *management services*, *session services*.

**control point.** A system services control point (SSCP), a peripheral node control point (PNCP), or a physical unit control point (PUCP).

**conversation.** The logical connection between a pair of transaction programs for serially sharing a session between type 6.2 logical units from transaction to transaction. While a conversation is active, it has exclusive use of an LU-LU session as delimited by a distinct bracket; successive conversations may use the same session.

**cross-domain.** Pertaining to control or resources involving more than one domain.

**cross-domain LU-LU session.** A session between logical units (LUs) in different domains.

**cross-network LU-LU session.** A session between logical units (LUs) in different networks.

**cryptography.** The transformation of data to conceal its meaning.

**cryptography key.** A binary value that is used as a data encrypting key to encipher and decipher end-user data that is transmitted over an LU-LU session that uses cryptography. See also *Data Encryption Standard (DES) algorithm, session cryptography.*

## D

---

**data channel.** A device that connects a processor and main storage with I/O control units.

**Data Encryption Standard (DES) algorithm.** A cryptographic algorithm designed to encipher and decipher data using a 64-bit cryptography key, as specified in the Federal Information Processing Standard Publication 46, January 15, 1977.

**data flow control (DFC) layer.** The layer within a half-session that (1) controls whether the half-session can send, receive, or concurrently send and receive request units (RUs); (2) groups related RUs into RU chains; (3) delimits transactions via the bracket protocol; (4) controls the interlocking of requests and responses in accordance with control modes specified at session activation; (5) generates sequence numbers; and (6) correlates requests and responses.

**data link.** Synonym for link.

**data link control (DLC) layer.** The layer that consists of the link stations that schedule data transfer over a link between two nodes and perform error control for the link. Examples of data link control are SDLC for serial-by-bit link connection and data link control for the System/370 channel.

**data stream.** A continuous stream of data elements being transmitted, or intended for transmission, in character or binary-digit form, using a defined format.

**decipher.** To return enciphered data to its original form. Contrast with *encipher.*

**definite response.** A protocol requested in the form-of-response-requested field of the request header that directs the receiver of the request to return a response unconditionally, whether positive or negative, to that request chain. Contrast with *exception response, no response.*

**delayed-request mode.** An operational mode in which the sender may continue sending request units after sending a definite-response request chain on that flow. The sender does not have to wait for a response to that chain. Contrast with *immediate-request mode.*

**delayed-response mode.** An operational mode in which the receiver of request units can return responses to the sender in a sequence different from that in which the corresponding request units were sent. Contrast with *immediate-response mode.*

**DFC.** Data flow control.

**DIA.** Document Interchange Architecture.

**directory services.** Services for resolving user identifications of network components to network routing information.

**DLC.** Data link control.

**Document Interchange Architecture (DIA).** Protocols within the transaction services layer by which distributed office-application processes interchange data.

**domain.** A system services control point (SSCP) and the physical units (PUs), logical units (LUs), links, link stations, and all the associated resources that the SSCP has the ability to control by means of activation requests and deactivation requests. See also *shared control.*

**dynamic reconfiguration.** The process of changing the network configuration (peripheral PUs and LUs) associated with a boundary node, without regenerating the boundary node's complete configuration tables.

## E

---

**element address.** A value in the network address that identifies a particular resource within a subarea. See also *subarea address.*

**ENA.** Extended network addressing.

**encipher.** To scramble data or convert it, prior to transmission, in order to mask the meaning of the data to any unauthorized recipient. Contrast with *decipher*.

**end bracket.** The value (binary 1) of the end bracket indicator in the request header (RH) of the last chain element in a bracket; the value denotes the end of the bracket. Contrast with *begin bracket*, *conditional end bracket*.

**end user.** The ultimate source or destination of data flowing through an SNA network. An end user can be an application program or a workstation operator.

**ER.** Explicit route.

**exception response.** A protocol requested in the form-of-response-requested field of a request header that directs the receiver to return a response only if the request is unacceptable as received or cannot be processed; that is, a negative response, but not a positive response, may be returned. Contrast with *definite response*, *no response*.

**Exchange Identifications (XID).** A data link control command and response passed between adjacent link stations that allow the two link stations to exchange identification and other information that is necessary for further operation over the data link.

**explicit route (ER).** The path control network components, including a specific set of one or more transmission groups and any intermediate nodes, that connect two subarea nodes. An explicit route is identified by an origin subarea address, a destination subarea address, an explicit-route number, and a reverse explicit-route number. See also *path*, *route extension*, *virtual route (VR)*.

**extended network addressing.** The network addressing system that extends the network address from 16 bits to 23 bits. The subarea portion of the address uses a eight (8) bits to address subarea nodes. The element portion uses fifteen (15) bits to address resources within each subarea.

## F

---

**FID.** Format identification.

**first speaker.** The LU-LU half-session defined at session activation as able to begin a bracket without requesting permission from the other LU-LU half-session to do so. Contrast with *bidder*.

**flow control.** The process of managing the rate at which data traffic passes between components of the network. Flow control optimizes the rate of flow of message units with minimum congestion in the

network. Flow control allows the receiver to receive message units at the rate at which it can process them without overflowing its buffers or the buffers at intermediate routing nodes. See also *pacing*, *session-level pacing*, *virtual-route (VR) pacing*.

**FMH.** Function management header.

**format identification (FID) field.** A field in each transmission header (TH) that indicates the format of the TH; that is, the presence or absence of certain fields. Transmission header formats differ according to the types of nodes between which they pass.

*Note: There are six FID types:*

- *FID 0, used for traffic involving non-SNA devices between adjacent subarea nodes when either or both nodes do not support explicit-route and virtual-route protocols.*
- *FID 1, used for traffic between adjacent subarea nodes when either or both nodes do not support explicit-route and virtual-route protocols.*
- *FID 2, used for traffic between a subarea node and an adjacent type 2.0 or type 2.1 peripheral node and between adjacent type 2.1 peripheral nodes.*
- *FID 3, used for traffic between a subarea node and an adjacent type 1 peripheral node.*
- *FID 4, used for traffic between adjacent subarea nodes when both nodes support explicit-route and virtual-route protocols.*
- *FID F, used for certain commands (for example, for transmission-group control) sent between adjacent subarea nodes when both nodes support explicit-route and virtual-route protocols.*

**function management (FM) header.** Optional control information, present in the leading request units (RUs) of an RU chain, that allows one half-session in an LU-LU session to: (1) carry a request that a conversation be established between two transaction programs, (2) carry information that relates to an error on the session or conversation, (3) carry LU-LU password verification data, (4) select a destination as the session partner and control the way the end-user data it sends is handled at the destination, (5) change the destination or the characteristics of the data during the session, and (6) transmit status or user information about the destination (for example, whether it is a program or a device) between session partners.

## G

---

**gateway.** The combination of a gateway node and one or more gateway SSCPs that provides the name translation, network address translation, and SSCP rerouting functions between connected networks.

**gateway function.** (1) A capability of a subarea node to interconnect subarea path control elements that reside in different networks. (2) the component that provides this capability. See also *boundary function*, *intermediate routing function*.

**gateway node.** A subarea node with gateway function.

**gateway SSCP.** The SSCP that is involved with cross-network session initiation, termination, takedown, and outage notification. The gateway SSCP provides name translation and cooperates with the gateway node in setting up alias addresses for cross-network sessions.

## H

---

**half-session.** A component that provides data flow control and transmission control at one end of a session.

**host node.** A subarea node that contains a telecommunication access method.

## I

---

**immediate-request mode.** An operational mode in which the sender stops sending request units (RUs) after sending a definite-response request chain until that chain has been responded to. Contrast with *delayed-request mode*. See also *immediate-response mode*.

**immediate-response mode.** An operational mode in which the receiver responds to request units (RUs) in the order in which it receives them; that is, in a first-in, first-out sequence. Contrast with *delayed-response mode*. See also *immediate-request mode*.

**intermediate routing function.** A path control capability in a subarea node that receives and routes path information units (PIUs) that neither originate in nor are destined for network addressable units (NAUs) in that subarea node. See also *boundary function*, *gateway function*.

**intermediate routing node.** A subarea node with intermediate routing function.

## L

---

**layer.** A grouping of related functions that are logically separate from the functions in other layers; the implementation of the functions in one layer can be changed without affecting functions in other layers. See also *transaction services (TS) layer*, *presentation services (PS) layer*, *data flow control (DFC) layer*, *transmission control (TC) layer*, *path control (PC) layer*, *data link control (DLC) layer*, *physical control layer*.

**link.** The combination of the link connection and the link stations that joins adjacent nodes in a network; for example: (1) a System/370 channel and its associated protocols, (2) a serial-by-bit connection under the control of synchronous data link control (SDLC). Synonymous with *data link*.

*Note:* A link connection is the physical medium of transmission; for example, a telephone wire or a microwave beam. A link includes the physical medium of transmission, the protocol, and associated communication devices and programming; it is both logical and physical.

**link connection.** The physical equipment that provides two-way communication between link stations; for example, a communication line and data circuit terminating equipment.

**link header.** Control information for data link control at the beginning of a basic link unit (BLU).

**link station.** The combination of hardware and software that allows a node to attach to, and provide control for, a link. See also *adjacent link station*.

**link trailer.** Control information for data link control at the end of a basic link unit (BLU).

**local address.** An address used in a peripheral node in place of a network address and paired with a network address by the boundary function in a subarea node. See also *network address*.

**logical unit (LU).** A port through which an end user accesses the SNA network in order to communicate with another end user and through which the end user accesses the functions provided by system services control points (SSCPs). An LU can support at least two sessions—one with an SSCP, and one with another logical unit—and may be capable of supporting many sessions with other logical units. See also *network addressable unit (NAU)*, *physical unit (PU)*, *primary logical unit*, *secondary logical unit*, *system services control point (SSCP)*.

**LU.** logical unit.

**LU-LU session.** A session between two logical units (LUs) that supports communication between two end users.

**LU-LU session initiation.** The process that begins with a session-initiation request from a logical unit (LU) to a control point, and culminates in the activation of an LU-LU session. See also *session activation*.

**LU-LU session termination.** The process that begins with a session-termination request from a logical unit (LU) to a control point, and culminates in the deactivation of an LU-LU session.

**LU services manager.** A component that provides a logical unit (LU) with network services and session management services. The LU services manager provides services for all half-sessions within the LU.

**LU type 0.** The architecture does not define LU type 0. It is an implementation-defined LU that uses SNA-defined protocols for transmission control and data flow control, but may also use end-user or product-defined protocols to augment or replace higher layer protocols. For example, an LU for an application program using IMS/VS communicating with an IBM 4700 Finance Communication System at which the operator of the 3600 workstation is updating the passbook balance for a customer's savings account.

**LU type 1.** A type of LU for an application program that communicates with single- or multiple-device data processing workstations in an interactive, batch data transfer, or distributed processing environment. For example, an LU for an application program using IMS/VS that communicates with an IBM 8100 Information System at which the workstation operator is correcting a data base that the application program maintains. The data stream conforms to SNA character string or Document Content Architecture (DCA).

**LU type 2.** A type of LU for an application program that communicates with a single display workstation in an interactive environment, using the SNA 3270 data stream. Type 2 LUs also use the SNA 3270 data stream for file transfer. For example, an LU for an application program that uses IMS/VS and communicates with an IBM 3719 Display Station, at which the 3719 operator is creating and sending data to the application program.

**LU type 3.** A type of LU for an application program that communicates with a single printer, using the SNA 3270 data stream. For example, an LU for an application program that uses CICS/VS to send data to an IBM 3278 Printer attached to an IBM 3274 Controller.

**LU type 4.** A type of LU for: (1) an application program that communicates with a single- or multiple-device

data processing or word processing workstation in an interactive, batch data transfer, or distributed processing environment (for example, an LU for an application program that uses CICS/VS to communicate with an IBM 6670 Information Distributor); or (2) logical units in peripheral nodes (for example, two 6670s) that communicate with each other. The data stream is the SNA character string for data processing environments and Office Information Interchange (OII) Level 2 (a precursor of DCA) for word processing environments.

**LU type 6.1.** A type of LU for an application subsystem that is to communicate with another application subsystem in a distributed data processing environment. For example, an LU for an application program that uses CICS/VS to communicate with an application program that uses IMS/VS.

**LU type 6.2.** A type of LU that supports sessions between two applications in a distributed data processing environment using the SNA general data stream, which is a structured-field data stream, or a user-defined data stream. LU 6.2 sessions provide communication between two type 5 nodes, a type 5 and a type 2.1 node, and two type 2.1 nodes. For example, an application program running on CICS/VS communicating with an application program running on another CICS/VS, a DISOSS/370 application on CICS/VS communicating with a Displaywriter System, or an application program running on a System/36.

**LU type 7.** A type of LU for an application program and a single display workstation in an interactive environment. For example, an application program in a System/34 communicating with an IBM 5251 Display Station, where the 5251 operator is creating data and sending it to the application program. The data stream is the 5250 data stream.

## M

---

**management services.** One of the types of network services in system services control points (SSCPs) and logical units (LUs). Management services provide facilities for problem determination and performance monitoring and forward requests for network data, such as error statistics, and deliver the data in reply. See also *configuration services*, *session services*.

**mandatory cryptographic session.** A session in which all outgoing data is enciphered and all incoming data is deciphered. Contrast with *selective cryptographic session*. See also *session cryptography*.

**message unit.** A generic term for the unit of data processed by any layer. Basic information units (BIUs), path information units (PIUs), and request/response units (RUs) are examples of message units.

**mode name.** An identifier of a set of session parameters for an LU-LU session; the mode name is used as an index into a mode table.

**mode table.** A set of entries for one or more mode names. Each entry specifies the characteristics of a session between two logical units.

**multiple-domain network.** A network with more than one system services control point (SSCP). Contrast with *single-domain network*.

## N

---

**NAU.** Network addressable unit.

**NCP.** Network control program.

**negative response.** A response indicating that a request did not arrive successfully or was not processed successfully by the receiver. Contrast with *positive response*. See also *exception response*.

**negotiable BIND.** A capability that allows two LU-LU half-sessions to negotiate the parameters of a session when the session is being activated.

**network address.** An address, consisting of subarea and element fields, that identifies a link, a link station, or a network addressable unit. Subarea nodes use network addresses; peripheral nodes use local addresses. The boundary function in the subarea node to which a peripheral node is attached pairs local addresses with network addresses and vice versa. See also *extended network addressing*, *local address*, *network name*.

**network addressable unit (NAU).** A logical unit, a physical unit, or a system services control point. It is the origin or the destination of information transmitted by the path control network. See also *network name*, *network address*, *path control network*.

*Note:* Each NAU has a network address that represents it to the path control network. (LUs may have multiple addresses for parallel LU-LU sessions.) The path control network and the NAUs together constitute the SNA network.

**network name.** The symbolic identifier by which end users refer to a network addressable unit (NAU), a link station, or a link. See also *network address*.

**network operator.** A person or program responsible for controlling the operation of all or part of a network.

**node.** An endpoint of a link, or a junction common to two or more links in a network. Nodes can be processors, controllers, or workstations. Nodes vary in

routing and other functional capabilities. See also *node type*, *peripheral node*, *subarea node*.

**node type.** A designation of a node according to the protocols it supports and the network addressable units (NAUs) that it can contain. Five types are defined: 1, 2.0, 2.1, 4, and 5. Type 1, type 2.0, and type 2.1 nodes are peripheral nodes; type 4 and type 5 nodes are subarea nodes.

**nonswitched link.** A connection between two nodes that does not have to be established by dialing. Contrast with *switched link*.

**no response.** A protocol requested in the form-of-response-requested field of the request header (RH) that directs the receiver of the request not to return any response, regardless of whether or not the request is received and processed successfully. Contrast with *definite response*, *exception response*.

## P

---

**padding.** A technique by which a receiving component controls the rate of transmission by a sending component to prevent overrun or congestion. See also *flow control*, *receive pacing*, *send pacing*, *session-level pacing*, *virtual-route (VR) pacing*.

**pacing group.** (1) The path information units (PIUs) that can be transmitted on a virtual route before a virtual-route pacing response is received, indicating that the virtual-route receiver is ready to accept more PIUs on the route. (2) The requests that can be transmitted in one direction over a session before a session-level pacing response is received, indicating that the receiver is ready to accept the next group of requests. (3) Synonymous with *window*.

**pacing-group size.** (1) The number of path information units (PIUs) in a virtual route pacing group. The pacing-group size varies according to traffic congestion along the virtual route. (2) The number of requests in a session-level pacing group. The pacing-group size is set at session activation. (3) Synonymous with *window size*.

**pacing response.** An indicator that signifies a receiving component's readiness to accept another pacing group; the indicator is carried in a response header (RH) for session-level pacing, and in a transmission header (TH) for virtual-route pacing.

**parallel links.** Two or more links between adjacent subarea nodes.

**parallel sessions.** Two or more concurrently active sessions between the same two logical units (LUs)

using different pairs of network addresses. Each session can have independent session parameters.

**path.** The series of path control network components traversed by the information exchanged between two network addressable units (NAUs). A path consists of a series of path control elements, data link control elements, and links. See also *explicit route*.

**path control (PC) layer.** The layer that manages the sharing of link resources of the SNA network and routes basic information units (BIU) through it. Path control routes message units between network addressable units (NAUs) in the network and provides the paths between them. It converts the BIUs from transmission control (possibly segmenting them) into path information units (PIU) and exchanges basic transmission units (BTUs)—one or more PIUs—with data link control. See also *BIU segment*, *blocking of PIUs*, *segmenting of PIUs*, *data link control layer (DLC)*, *transmission control (TC) layer*.

**path control network.** The part of the SNA network that includes the data link control, path control, and physical control layers.

**path information unit (PIU).** A message unit consisting of a transmission header (TH) alone, or of a TH followed by a basic information unit (BIU) or a BIU segment. See also *transmission header*.

**PC.** Path control.

**peripheral link.** A link that connects a peripheral node to a subarea node. See also *route extension*.

**peripheral node.** A node that uses local addresses for routing and therefore is not affected by changes in network addresses. A peripheral node requires boundary-function assistance from an adjacent subarea node. See also *node type*, *peripheral link*.

**peripheral node control point (PNCP).** A control point in a type 2.1 node that provides a subset of the system services control point (SSCP) functions for (1) mediating LU-LU session-initiation requests between logical units residing in adjacent type 2.1 nodes, (2) activating resources local to its node.

**physical control layer.** The layer that provides a physical interface for any transmission medium that is attached to it. This layer defines the electrical and transmission (signaling) characteristics needed to establish, maintain, and terminate physical connections.

**physical unit (PU).** The component that manages and monitors the resources (such as attached links and

adjacent link stations) of a node, as requested by an SSCP via an SSCP-PU session. Each node of an SNA network contains a physical unit.

**physical unit control point (PUCP).** A control point in a type 1, type 2, and type 4 node that provides a subset of the system services control point (SSCP) functions for activating the links that are attached to its node.

**PIU.** Path information unit.

**PLU.** Primary logical unit.

**PNCP.** Peripheral node control point.

**positive response.** A response indicating that a request was successfully received and processed. Contrast with *negative response*.

**presentation services (PS) layer.** The layer that provides services for transaction programs, such as controlling conversation-level communication between them.

**primary logical unit.** The logical unit (LU) that sends the Bind Session (BIND) request for a particular LU-LU session. Contrast with *secondary logical unit*.

**protocol.** The meanings of, and the sequencing rules for, requests and responses used for managing the network, transferring data, and synchronizing the states of network components.

**PS.** Presentation services.

**PU.** Physical unit.

**PUCP.** Physical unit control point.

**PU services manager.** A component that provides network services for all half-sessions within the physical unit (PU).

**PU type 5.** A physical unit that resides in a type 5 node.

**PU type 4.** A physical unit that resides in a type 4 node.

**PU type 2.1.** A physical unit that resides in a type 2.1 node.

**PU type 2.0.** A physical unit that resides in a type 2.0 node.

**PU type 1.** A physical unit that resides in a type 1 node.

## R

---

**real address.** A network address that identifies a resource in the network in which that resource resides. Contrast with *alias address*.

**real name.** A name that identifies a resource in the network in which that resource resides. Contrast with *alias name*.

**receive pacing.** The pacing of message units that a component is receiving. See also *send pacing*.

**Recommendation X.25 (Geneva 1980).** A Consultative Committee on International Telegraph and Telephone (CCITT) recommendation for the interface between data terminal equipment and packet-switched data networks.

**reply.** A request unit sent only in reaction to a received request unit. For example, Quiesce Complete is the reply sent after receipt of Quiesce At End of Chain. Synonymous with *reply request*.

**reply request.** Synonym for *reply*.

**request.** A message unit that signals initiation of a particular action or protocol. For example, Initiate-Self (INIT-SELF) is a request for activation of an LU-LU session.

**request header (RH).** A request unit (RU) header preceding a request unit.

**request unit (RU).** A message unit that contains control information such as a request code, or function management (FM) headers, end-user data, or both.

**request/response header (RH).** Control information, preceding a request/response unit (RU), that specifies the type of RU (request unit or response unit) and contains control information associated with that RU.

**request/response unit (RU).** A generic term for a request unit or a response unit.

**resource-definition statement.** (1) In a telecommunication access method, the means of describing a resource of the network. (2) In a network control program, macro instructions that define a resource to the network control program. See also *system definition*, *system generation*.

**response.** (1) A message unit that acknowledges receipt of a request; a response consists of a response header (RH), a response unit (RU), or both. (2) In SDLC, the control information (in the C-field of the link header) sent from the secondary station to the primary station.

**response header (RH).** A header, optionally followed by a response unit (RU), that indicates whether the response is positive or negative and that may contain a pacing response. See also *negative response*, *pacing response*, *positive response*.

**response unit (RU).** A message unit that acknowledges a request unit; it may contain prefix information received in a request unit. If positive, the response unit may contain additional information (such as session parameters in response to BIND SESSION), or if negative, contains sense data defining the exception condition.

**RH.** Request/response header.

**route.** See *explicit route*, *virtual route*.

**route extension.** The path control network components, including a peripheral link, that make up the portion of a path between a subarea node and a network addressable unit (NAU) in an adjacent peripheral node. See also *path*, *explicit route (ER)*, *virtual route (VR)*.

**routing.** The forwarding of a message unit along a particular path through a network as determined by parameters carried in the message unit, such as the destination network address in a transmission header.

**RU.** Request/response unit.

**RU chain.** A set of related request/response units (RUs) that are consecutively transmitted in one direction over a session. Each RU belongs to only one chain, which has a beginning and an end indicated by control bits in request/response headers within the RU chain.

## S

---

**SCS.** SNA character string.

**SDLC.** Synchronous Data Link Control.

**secondary logical unit.** The logical unit (LU) that receives the Bind Session (BIND) request for a particular LU-LU session. Contrast with *primary logical unit*.

**segmenting of BIUs.** An optional function of path control that divides a basic information unit (BIU) received from transmission control into two or more path information units (PIUs). The first PIU contains the request header (RH) of the BIU and usually part of the RU; the remaining PIU or PIUs contain the remaining parts of the RU.



**segmenting of PIUs.** An optional function of the path control network that divides a path information unit (PIU) into two or more basic transmission units (BTUs).

**selective cryptographic session.** A cryptographic session in which an application program is allowed to specify the request units to be enciphered. Contrast with *mandatory cryptographic session*.

**send pacing.** Pacing of message units that a component is sending. See also *receive pacing*.

**service transaction programs.** IBM-supplied programs that are defined by SNA for providing transaction services. See also *transaction program*. Contrast with *application transaction program*.

**session.** A logical connection between two network addressable units (NAUs) that can be activated, tailored to provide various protocols, and deactivated, as requested. The session-activation request and response can determine options relating to such things as the rate and concurrency of data exchange, the control of contention and error recovery, and the characteristics of the data stream. See also *half-session*, *LU-LU session*, *SSCP-LU session*, *SSCP-PU session*, *SSCP-SSCP session*.

**session activation.** The process of exchanging a session-activation request and a positive response between network addressable units (NAUs). See also *LU-LU session initiation*. Contrast with *session deactivation*.

**session-activation request.** A request that activates a session between two network addressable units (NAUs) and specifies session parameters that control various protocols during session activity; for example, Bind Session (BIND) and Activate Physical Unit (ACTPU). Contrast with *session-deactivation request*.

**session cryptography.** The process of providing security for end-user data by enciphering and deciphering the data. See also *cryptography key*, *Data Encryption Standard (DES) algorithm*, *mandatory cryptographic session*, *selective cryptographic session*.

**session deactivation.** The process of exchanging a session deactivation request and response between network addressable units (NAUs). Contrast with *session activation*.

**session-deactivation request.** A request that deactivates a session between two network addressable units (NAUs); for example, Unbind Session (UNBIND) and Deactivate Physical Unit (DACTPU). Contrast with *session-activation request*.

**session initiation.** See *LU-LU session initiation*.

**session-initiation request.** An Initiate (INIT) or logon request from a logical unit (LU) to a control point that asks for the LU-LU session to be activated.

**session-level pacing.** A flow control technique that permits a receiving half-session to control the data transfer rate (the rate at which it receives request units). It is used to prevent overloading a receiver with unprocessed requests when the sender can generate requests faster than the receiver can process them. See also *pacing*, *virtual-route (VR) pacing*.

**session limit.** The maximum number of concurrently active LU-LU sessions a particular logical unit (LU) can support.

**session parameters.** The parameters in a session-activation request that specify or constrain the protocols (such as bracket protocol and pacing) for a session between two network addressable units (NAUs).

**session partner.** One of the two network addressable units (NAUs) having an active session.

**session services.** One of the types of network services in the system services control point (SSCP) and in a logical unit (LU). These services provide facilities for a logical unit (LU) or a network operator to request that the SSCP initiate or terminate sessions between logical units. See also *configuration services*, *management services*.

**session termination.** See *LU-LU session termination*.

**session-termination request.** A Terminate-Self (TERM-SELF) or Shutdown (SHUTD) request from a logical unit (LU) to a control point or session partner, respectively, that asks for the LU-LU session to be deactivated.

**shared control.** Serial or concurrent control of network resources—physical units (PUs), logical units (LUs), links, link stations—by two or more control points. See also *share limit*.

**share limit.** The maximum number of control points that can concurrently control a network resource. See also *shared control*.

**single-domain network.** A network with one system services control point (SSCP). Contrast with *multiple-domain network*.

**SLU.** Secondary logical unit.

**SNA.** Systems Network Architecture.

**SNA Distribution Services (SNADS).** Service transaction programs that allow processors and

workstations to asynchronously exchange files and documents.

**SNA network.** The part of a user-application network that conforms to the formats and protocols of Systems Network Architecture. It enables reliable transfer of data among end users and provides protocols for controlling the resources of various network configurations. The SNA network consists of network addressable units (NAUs), boundary-function components, and the path control network.

**SNA network interconnection.** A facility that provides cross-network communication between two or more independent SNA networks.

**SNI.** SNA network interconnection.

**SSCP.** System services control point.

**SSCP-LU session.** A session between a system services control point (SSCP) and a logical unit (LU); the session enables the LU to request the SSCP to help initiate LU-LU sessions.

**SSCP-PU session.** A session between a system services control point (SSCP) and a physical unit (PU); SSCP-PU sessions allow SSCPs to send requests to and receive status information from individual nodes in order to control the network configuration.

**SSCP rerouting.** The transfer of requests by a gateway SSCP between SSCPs in different networks.

**SSCP services manager.** A component that provides network services for all the half-sessions of the system services control point (SSCP).

**SSCP-SSCP session.** A session between two system services control points (SSCPs), used to initiate and terminate sessions between LUs that are either in (1) different domains, or (2) different networks.

**start/stop.** Asynchronous transmission such that a group of signals representing a character is preceded by a start element and followed by a stop element.

**subarea.** A portion of the SNA network consisting of a subarea node, any attached peripheral nodes, and their associated resources. Within a subarea node, all network addressable units (NAUs), links, and adjacent link stations that are addressable within the subarea share a common subarea address and have distinct element addresses.

**subarea address.** A value in the network address that identifies a particular subarea. See also *element address*.

**subarea node.** A node that uses network addresses for routing and whose routing tables are therefore affected by changes in the configuration of the network. Subarea nodes can provide boundary function, gateway function, and intermediate routing function. See also *node type*.

**switched link.** A link between two nodes that is established by dialing. Contrast with *nonswitched link*.

**Synchronous Data Link Control (SDLC).** A discipline for managing synchronous, code-transparent, serial-by-bit, information transfer over a link connection. Transmission exchanges may be duplex or half-duplex over switched or nonswitched links. The configuration of the link connection may be point-to-point, multipoint, or loop. SDLC conforms to subsets of the Advanced Data Communication Control Procedures (ADCCP) of the American National Standards Institute and High-level Data Link Control (HDLC) of the International Standards Organization.

**system definition.** The process of coding and loading resource-definition statements and macro instructions that describe the network's configuration and operation to the network's software. The system definition process includes system generations.

**system generation.** The process of selecting optional parameters for software and tailoring the software to the requirements of the network. See also *system definition*.

**system services control point (SSCP).** A focal point within an SNA network for managing the configuration, coordinating network operator and problem determination requests, and providing directory services and other session services for end users of a network. Multiple SSCPs, cooperating as peers with one another, can divide the network into domains of control, with each SSCP having a hierarchical control relationship to the physical units and logical units within its own domain. See also *gateway SSCP*, *peripheral node control point (PNCP)*, *physical unit control point (PUCP)*.

**Systems Network Architecture (SNA).** The description of the logical structure, formats, protocols, and operational sequences for transmitting information units through and controlling the configuration and operation of networks.

---

## T

**TC.** Transmission control.

**termination.** See *LU-LU session termination*.

**TG.** Transmission group.

**TH.** Transmission header.

**transaction.** An exchange between (1) a workstation and a program, (2) two workstations, or (3) two programs, that accomplishes a particular action or result; for example, the entry of a customer's deposit and the updating of the customer's balance.

**transaction program.** A program that processes transactions in an SNA network. There are two kinds of transaction programs: application transaction programs and service transaction programs. See also *conversation*.

**transaction services (TS) layer.** The layer that includes service transaction programs, and provides configuration services, session services, and management services.

**transmission control (TC) layer.** The layer within a half-session that synchronizes and paces session-level data traffic, checks session sequence numbers of requests, and enciphers and deciphers end-user data.

**transmission group.** A group of links between adjacent subarea nodes, appearing as a single logical link for routing of messages. A transmission group may consist of one or more SDLC links (parallel links) or of a single System/370 channel.

**transmission header (TH).** Control information, optionally followed by a basic information unit (BIU) or a BIU segment, that is created and used by path control to route message units and to control their flow within the network. See also *path information unit*.

**transmission priority.** A rank assigned to a path information unit (PIU) that determines its precedence for being selected by the transmission-group control component of path control for forwarding to the next subarea node along the route traversed by the PIU.

**TS.** Transaction services.

---

## U

**UNBIND.** Unbind Session.

**Unbind Session (UNBIND).** A request to deactivate a session between two logical units (LUs). See also *session-deactivation request*. Contrast with *Bind Session (BIND)*.

---

## V

**virtual route (VR).** A logical connection (1) between two subarea nodes that is physically realized as a particular explicit route, or (2) that is contained wholly within a subarea node for intra-node sessions. A virtual route between distinct subarea nodes imposes a transmission priority on the underlying explicit route, provides flow control through virtual-route pacing, and provides data integrity through sequence numbering of path information units (PIUs). See also *explicit route (ER)*, *path*, *route extension*.

**virtual-route (VR) pacing.** A flow control technique used by the virtual-route control component of path control at each end of a virtual route. VR pacing controls the rate at which path information units (PIUs) flow over the virtual route. VR pacing can be adjusted according to traffic congestion in any of the nodes along the route. See also *pacing*, *session-level pacing*.

**VR.** Virtual route.

---

## W

**window.** Synonym for pacing group.

**window size.** Synonym for pacing-group size.

---

## X

**XID.** Exchange Identifications.

**X.25.** See *Recommendation X.25*.

# Index

## A

access method 34  
  defining network resources to 25, 175, 176  
  specifying routes to 177  
  telecommunication 4, 7

ACF/NCP  
  See network control program

ACF/VTAM  
  See access method

acknowledgment data 99

Activate Link (ACTLINK) request 47, 49, 50, 52, 53

Activate Logical Unit (ACTLU) request 46, 52

Activate Physical Unit (ACTPU) request 45, 52

activation  
  hierarchy of 44  
  of links 16, 44, 47, 50  
  of LU-LU sessions 68  
  of network resources 25, 143-145  
  of routes 86  
  of SSCP-LU sessions 43, 46  
  of SSCP-PU sessions 43, 45  
  of SSCPs 43

ACTLINK  
  See Activate Link (ACTLINK) request

ACTLU  
  See Activate Logical Unit (ACTLU) request

ACTPU  
  See Activate Physical Unit (ACTPU) request

addresses  
  alias address 38  
  extended network addressing 32  
  local address 33  
  network address 30, 32  
    element address field 30, 32  
    subarea address field 30, 31

adjacent nodes 10

alias  
  address 38  
  name 38

ALLOCATE verb 71

application transaction programs 69  
  See also conversations,  
  See also service transaction programs, verbs

Attach FM header 71, 119

automatic network shutdown 59

## B

BACKOUT verb 71

basic information unit (BIU) 99, 100, 104, 111, 113  
  request header 100  
  request unit 100  
  response header 100  
  response unit 101

basic link unit (BLU) 99, 102, 104  
  link header 102  
  link trailer 102

basic transmission unit (BTU) 109

BBI  
  See begin bracket indicator (BBI)

BCI  
  See begin chain indicator (BCI)

begin bracket indicator (BBI) 114

begin chain indicator (BCI) 113

BF  
  See boundary function

Binary Synchronous Communication (BSC)  
  protocols 10

BIND  
  See Bind Session (BIND) request

Bind Failure (BINDF) request 68

Bind image 64

Bind Session (BIND) request 68  
  negotiable BIND 68  
  nonnegotiable BIND 68  
  parameters 68, 112, 113, 114, 116, 117, 119  
  protocols 112  
    bracket protocols 114  
    chaining protocols 113  
    data security protocols 117  
    function management headers 119  
    request and response mode protocols 115  
    response protocols 112  
    send and receive mode protocols 116  
    sequencing protocols 115

BIND table 64

BINDF  
  See Bind Failure (BINDF) request

BIS  
  See Bracket Initiation Stopped (BIS) request

BIU  
  See basic information unit (BIU)

blocking of PIUs 109, 110

BLU  
  See basic link unit (BLU)

boundary function 33, 92, 123

Bracket Initiation Stopped (BIS) request 72

bracket protocols 114, 169

BSC  
  See Binary Synchronous Communication (BSC)  
  protocols

BTU  
See basic transmission unit (BTU)

## C

cascaded  
activation 58  
deactivation 58  
CDCINIT  
See Cross-Domain Control Initiate (CDCINIT) request  
CDI  
See change direction indicator (CDI)  
CDINIT  
See Cross-Domain Initiate (CDINIT) request  
CEBI  
See condition end bracket indicator (CEBI)  
chains 113  
definite response chains 114  
exception response chains 114  
no-response chains 114  
change direction indicator (CDI) 117  
CINIT  
See Control Initiate (CINIT) request  
class of service 84  
COS table 84  
command RUs 100, 113  
Activate Link (ACTLINK) 47  
Activate Logical Unit (ACTLU) 46  
Activate Physical Unit (ACTPU) 45  
Bind Failure (BINDF) 68  
Bind Session (BIND) 68  
Bracket Initiation Stopped (BIS) 72  
Contact (CONTACT) 47  
Contacted (CONTACTED) 47  
Control Initiate (CINIT) 64, 68  
Control Terminate (CTERM) 72  
Cross-Domain Control Initiate (CDCINIT) 64, 66  
Cross-Domain Initiate (CDINIT) 64, 66  
Explicit Route Activate (NC-ER-ACT) 86  
Request Network Address Assignment (RNAA) 39  
Session Started (SESSST) 68  
Set Control Vector (SETCV) 39  
Shutdown (SHUTD) 72  
Shutdown Complete (SHUTC) 72  
Terminate Self (TERM-SELF) 72  
Unbind Session (UNBIND) 72  
communication controller subarea nodes 7  
concurrent sharing of resources 26  
share limit 28  
condition end bracket indicator (CEBI) 114  
configuration  
See network, configuration of  
Contact (CONTACT) request 47, 52  
Contacted (CONTACTED) request 47  
Control Initiate (CINIT) request 64, 68  
control points

See system services control point (SSCP),  
peripheral node control point (PNCP), physical unit  
control point (PUCP)  
Control Terminate (CTERM) request 72  
controllers 4, 7  
conversations 71  
COS  
See class of service  
Cross-Domain Control Initiate (CDCINIT) request 64,  
66  
Cross-Domain Initiate (CDINIT) request 64, 66  
cryptography 118  
CTERM  
See Control Terminate (CTERM) request

## D

data channels 10, 110, 128, 134  
data flow control layer 3, 123, 124, 125, 127, 134  
data link control elements 16  
See also link stations  
data link control layer 3, 123, 124, 125, 128, 134  
data link control protocols 10  
Binary Synchronous Communication (BSC) 10  
start/stop 10  
Synchronous Data Link Control (SDLC) 10  
System/370 data channels 10  
data RUs 100, 113  
data security protocols 117, 118, 119  
data streams 139, 141  
Document Content Architecture (DCA) 137, 141  
general data stream (GDS) 138, 140  
Office Information Interchange (OII) Level 2 137  
SNA character string 139  
SNA character string (SCS) 137  
3270 data stream 137, 140  
5250 data stream 138  
DCA  
See Document Content Architecture (DCA)  
deactivation  
cascaded 58  
hierarchy of 54  
of network resources 54, 146-150  
defining network resources 25  
definite response indicators (DR1I, DR2I) 112  
definite responses 112  
DIA  
See Document Interchange Architecture (DIA)  
Document Content Architecture (DCA) 137, 141  
Document Interchange Architecture (DIA) 69, 126  
domains 13  
identifying 25  
multiple-domain network 13  
single-domain network 13  
DR1I  
See definite response indicators (DR1I, DR2I)  
DR2I

See definite response indicators (DR11, DR21)  
dynamic reconfiguration 59

## E

EBI  
See end bracket indicator (EBI)  
ECI  
See end chain indicator (ECI)  
EDI  
See enciphered data indicator (EDI)  
element address 32  
ENA  
See extended network addressing  
enciphered data indicator (EDI) 118  
end bracket indicator (EBI) 114  
end chain indicator (ECI) 113  
end users 11, 69, 117  
end-user data 99  
ER  
See explicit routes  
ERI  
See exception response indicator (ERI)  
Error description FM header 119  
exception response indicator (ERI) 113  
exception responses 113  
Exchange Identification (XID) command 50  
Explicit Route Activate (NC-ER-ACT) request 86  
explicit routes 80  
deactivating 147  
defining 80  
relationship to virtual routes 84  
extended network addressing 32

## F

FI  
See format indicator (FI)  
FID  
See format identification (FID) types  
flow control algorithms  
global 91  
local 91  
FMH  
See function management (FM) header  
format identification (FID) types 102  
See also transmission header (TH)  
format indicator (FI) 119  
formatted systems services (FSS) 64  
FSS  
See formatted systems services (FSS)  
full-duplex send/receive mode 116  
function management (FM) header 119  
Attach 71, 117, 119  
Error description 119

Security 118, 119

## G

gateways 36, 39, 40  
gateway nodes 38, 39  
gateway function component 38  
gateway SSCPs 38, 39  
GDS  
See general data stream (GDS)  
general data stream (GDS) 138, 140  
global flow control 91

## H

half-duplex contention send and receive mode 116  
half-duplex contention send/receive mode 170  
half-duplex flip-flop send and receive mode 117  
half-duplex flip-flop send/receive mode 169, 171  
half-session 69  
architectural definition of 124  
headers  
function management headers 119  
link headers 102  
request headers 100  
response headers 100  
transmission headers 101  
host subarea node 7

## I

interconnection of SNA networks 36  
intermediate routing function 33

## L

layers 3, 121-134  
See also path control layer, data link control layer,  
physical control layer  
See also transaction services layer, presentation  
services layer, data flow control layer,  
transmission control layer  
management of 131  
overview of 3, 4, 125  
peer-to-peer communication between 129  
LH  
See link header (LH)  
link connection 10, 16, 22  
See also links

- link header (LH) 102
- link stations 10, 16, 22
  - See also data link control elements
  - See also links
- link trailer (LT) 102
- links 10
  - See also data channels, Synchronous Data Link Control (SDLC)
  - activation of 44
  - deactivating 147
  - defining 176
    - to ACF/NCP 176
    - to ACF/VTAM 176
  - parallel links 75
  - shared control of
    - concurrent sharing 26
    - serial sharing 28
  - types of 10
- local addresses 33
- local flow control 91
- logical unit (LU) 11, 12, 13, 22, 30, 34
  - See also LU-LU session, SSCP-LU session, PNCP-LU session
  - LU-LU sessions 11
  - primary LU 68, 72
  - secondary LU 68, 72
  - services manager 131
    - network services component 131
    - resources manager 131
  - shared control of 28
    - share limit 28
  - types of 12, 138
- LT
  - See link trailer (LT)
- LU
  - See logical unit (LU)
- LU-LU session 11, 61-72
  - activating 68, 157
    - Bind Session (BIND) request 68
  - Bracket Initiation Stopped (BIS) request 72
  - cross-domain 64, 164, 165
  - cross-network 66, 160, 161
  - deactivating 72
    - Unbind Session (UNBIND) request 72
  - half-session 69
  - initiating 63
  - parallel sessions 11
  - security 118
  - session-initiation requests 63
    - formatted requests 64
    - unformatted requests 64
  - terminating 72, 163, 173
    - Unbind Session (UNBIND) request 72
- LU-LU sessions 85

## M

- message units 18, 99
  - basic information unit (BIU) 100
  - basic link unit (BLU) 102
  - basic transmission unit (BTU) 109
  - formats of 99
  - path information unit (PIU) 101
- mode name 64, 84
  - entry into a mode table 64
- mode table 64

## N

- NAUs
  - See network addressable units (NAUs)
- NC-ER-ACT
  - See Explicit Route Activate (NC-ER-ACT) request
- network
  - activation 43-55
    - cascaded 58
    - controlling 58
    - hierarchy of 44
    - specifying to ACF/VTAM 175
  - addresses
    - alias address 38
    - element address field 32
    - extended network addressing 32
    - subarea address field 31
  - commands 99
  - components of 4
  - configuration of 4
  - configuring 58
    - scheduled changes 58
    - unscheduled changes 59
  - deactivation 58
    - cascaded 58
    - hierarchy of 54
    - specifying to ACF/VTAM 175
  - dynamic reconfiguration of 59
  - interconnection of, 36
  - multiple-domain network 13
  - names 34
    - alias name 38
  - resources 25
    - defining 25, 175-178
    - identifying 30
    - shared control of 25
  - single-domain network 13
- network addressable units (NAUs) 11, 13, 124
  - architectural definition of 124
  - logical units 11
  - physical units 12
  - service managers 131
    - LU services managers 131

- PU services managers 131
- SSCP services managers 131
- system services control point 13
- network control program 4, 7
  - defining network resources to 25, 176
  - specifying routes to 177
- nodes 7
  - peripheral nodes 7
  - subarea nodes
    - communication controller subarea nodes 7
    - host subarea nodes 7
  - types of 20

**O**

- Office Information Interchange (OII) Level 2 137
- OII
  - See Office Information Interchange (OII) Level 2
- Open Systems Interconnection (OSI) 132
  - application layer 133
  - comparison with SNA 134
  - data link layer 132
  - network layer 132
  - physical layer 132
  - presentation layer 133
  - session layer 133
  - transport layer 133
- OSI
  - See Open Systems Interconnection (OSI)

**P**

- padding 91
  - inbound 92
  - outbound 92
  - response 92, 94
  - session-level 92
  - virtual-route 92, 94
  - window 92, 94
  - window size 92, 94
- parallel links 75, 76, 80, 87
- parallel LU-LU sessions 11
- passwords
  - end-user 117
  - LU-LU session 118
- path 77
  - defining 79
- path control 18
  - elements of 18
    - peripheral 18
    - subarea 18
- path control layer 3, 123, 124, 125, 128, 134
- path control network 16, 124
  - architectural definition of 124
  - data link control elements 16

- elements of 16
- path control 18
- transmission protocols 109
- path information unit (PIU) 99, 101, 104, 109
  - blocking of 110
  - segmenting of 111
  - sequencing of 109
  - transmission header (TH) 101
    - format identification (FID) types 102
- peripheral node control point (PNCP) 43, 44, 53, 54, 63
  - See also PNCP-LU session, PNCP-PU session
- peripheral nodes 7, 33
- peripheral path control elements 18
- physical control layer 3, 4, 123, 124, 125, 129, 134
- physical unit (PU) 11, 12, 22, 30, 34
  - See also SSCP-PU session, PNCP-PU session
- services manager 131
  - shared control of
    - concurrent sharing 26
    - serial sharing 28
    - share limit 28, 38
  - types 21
- physical unit control point (PUCP) 44, 47
- PIU
  - See path information unit (PIU)
- PNCP
  - See peripheral node control point (PNCP)
- PNCP-LU session 43, 63
- PNCP-PU session 43
- presentation services layer 3, 123, 124, 125, 127, 134
- processors 4, 7
- PU
  - See physical unit (PU)
- PUCP
  - See physical unit control point (PUCP)

**R**

- request 99
  - See also request header (RH), request unit (RU)
- modes
  - delayed 115
  - immediate 115
- request and response mode protocols 115
  - delayed request mode 115
  - delayed response mode 116
  - immediate request mode 115
  - immediate response mode 116
- request header (RH) 100
- Request Network Address Assignment (RNAA)
  - request 39
- request unit (RU) 100
  - command RUs 100
  - data RUs 100
  - sequences 143-173
    - for activating and deactivating network resources 143



- for activating and deactivating sessions 155
- for routing 151
- for transferring data over a session 155
- request/response indicator (RRI) 100
- resource
  - definition 25
  - shared control of 25
    - concurrent sharing 26
    - serial sharing 28
- response 99
  - See also response header (RH), response unit (RU)
  - modes 115
    - delayed 116
    - immediate 116
  - negative 99, 101
  - positive 99
  - protocols 112
    - definite response 112
    - exception response 113
    - no response 113
- response header (RH) 100
- response type indicator (RTI) 100
- response unit (RU) 101
- RH
  - See request header (RH), response header (RH)
- RNAA
  - See Request Network Address Assignment (RNAA)
  - request
- route extension 84
- routes
  - activating and deactivating 86
  - explicit route 80, 152, 153
  - specifying 176
    - to ACF/NCP 177
    - to ACF/VTAM 177
  - virtual route 84
- routing tables 82
- RRI
  - See request/response indicator (RRI)
- RTI
  - See response type indicator (RTI)
- RU
  - See request unit, response unit

S

- SDLC
  - See Synchronous Data Link Control (SDLC)
- security
  - See data security protocols
- Security FM header 119
- segmenting of PIUs 109, 111
- send and receive mode protocols 116
  - comparison with transmission medium protocols 117
  - full duplex 116
  - half-duplex contention 116

- half-duplex flip-flop 117
- sense data 101
- sequencing of PIUs 109
- sequencing protocols
  - for BIUs 115
  - for PIUs 109
- serial sharing of resources 28
  - share limit 28
- service transaction programs 69, 126
- Session Started (SESSST) request 68
- session-level pacing 92
  - one-stage 93
  - two-stage 93
- sessions 11
  - cryptography 118
  - LU-LU session 11
  - PNCP-LU session 43, 63
  - PNCP-PU session 43
  - SSCP-LU session 43, 46, 63, 72
  - SSCP-PU session 43, 44, 45
  - SSCP-SSCP session 64, 66
- SESSST
  - See Session Started (SESSST) request
- Set Control Vector (SETCV) request 39
- Set Normal Response Mode (SNRM) command 50
- SETCV
  - See Set Control Vector (SETCV) request
- share limit 28
- shared control 25
  - concurrent sharing 26
  - serial sharing 28
  - share limit 28
- SHUTC
  - See Shutdown Complete (SHUTC) request
- SHUTD
  - See Shutdown (SHUTD) request
- Shutdown (SHUTD) request 72
- Shutdown Complete (SHUTC) request 72
- single-domain network 13
- SNA
  - See Systems Network Architecture (SNA)
- SNA character string 139
- SNA character string (SCS) 137
- SNA Distribution Services (SNADS) 69, 126
- SNA network interconnection 36
  - gateways 36
    - gateway nodes 38
    - gateway SSCPs 38
- SNADS
  - See SNA Distribution Services (SNADS)
- SNI
  - See SNA network interconnection
- SSCP-LU session 43, 46, 63, 72, 85
  - Activate Logical Unit (ACTLU) request 46
  - activation of 43
  - Control Initiate (CINIT) request 64, 68
  - Control Terminate (CTERM) request 72
  - Session Started (SESSST) request 68
  - Terminate Self (TERM-SELF) request 72

SSCP-PU session 43, 44, 45, 85  
 Activate Link (ACTLINK) request 47  
 Activate Physical Unit (ACTPU) request 45  
 activation of 43  
 Contact (CONTACT) request 47  
 Contacted (CONTACTED) request 47  
 Request Network Address Assignment (RNAA) request 39  
 Set Control Vector (SETCV) request 39  
 SSCP-SSCP session 64, 66, 85  
 activating 156  
 Cross-Domain Control Initiate (CDCINIT) request 64  
 Cross-Domain Initiate (CDINIT) request 64  
 start/stop protocols 10  
 subarea  
 address field 31  
 definition of 9  
 nodes 7  
 path control elements 18  
 sync point 71, 112  
 Synchronous Data Link Control (SDLC) 10, 128, 134  
 SYNCPT verb 71  
 system definition 23, 82, 175  
 system generation 25, 58, 94, 110  
 system services control point (SSCP) 11, 13, 16, 22, 25, 26, 28, 30, 40, 44, 58, 59, 63, 64  
 See also SSCP-LU session, SSCP-LU session, SSCP-SSCP session  
 activation of 43  
 defining resources to 25  
 directory services 30, 34, 40, 64  
 domain of control 13, 25  
 gateway SSCP 36, 38, 40, 66  
 rerouting 66  
 services manager 131  
 Systems Network Architecture (SNA) 3  
 comparison with OSI 134  
 layers of 3, 125

**T**

TERM-SELF  
 See Terminate Self (TERM-SELF) request  
 Terminate Self (TERM-SELF) request 72  
 TG  
 See transmission group  
 TH  
 See transmission header (TH)  
 TP  
 See transmission priority  
 transaction programs 69  
 See also transaction services layer  
 application transaction programs 69  
 conversations 71  
 invoking 71

service transaction programs 69  
 Document Interchange Architecture (DIA) 69, 70  
 SNA Distribution Services (SNADS) 69, 70  
 use of verbs 70  
 transaction services layer 3, 4, 123, 124, 125, 126, 134  
 transmission  
 group 76, 80, 83, 85, 86, 87, 94, 109, 128, 176, 177  
 defining to ACF/NCP 176  
 defining to ACF/VTAM 176  
 numbers 177  
 sequence numbers 109  
 medium 117, 129  
 priority 84, 85  
 transmission control layer 3, 123, 124, 125, 128, 130, 134  
 transmission header (TH) 101, 104  
 See also format identification (FID) types

**U**

UNBIND  
 See Unbind Session (UNBIND) request  
 Unbind Session (UNBIND) request 72  
 unformatted systems services (USS) 64  
 Unnumbered Acknowledgment (UA) response 50  
 user IDs 117  
 USS  
 See unformatted systems services (USS)

**V**

verbs 70, 71, 112  
 ALLOCATE 71  
 BACKOUT 71  
 SYNCPT 71  
 virtual routes 84  
 deactivating 147  
 defining 84  
 relationship to explicit routes 84  
 route extension 84  
 selecting 85  
 transmission priority 84  
 virtual-route pacing 92, 94  
 VR  
 See virtual routes

**W**

workstations 4, 7

**X**

X.25 interface 10, 132, 134

**Numerics**

3270 data stream 137, 140  
5250 data stream 138

This manual is part of a library that serves as a reference source for systems analysts, programmers, and operators of IBM systems. You may use this form to communicate your comments about this publication, its organization, or subject matter, with the understanding that IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

**Note:** Copies of IBM publications are not stocked at the location to which this form is addressed. Please direct any requests for copies of publications, or for assistance in using your IBM system, to your IBM representative or to the IBM branch office serving your locality.

Possible topics for comment are:

Clarity    Accuracy    Completeness    Organization    Coding    Retrieval    Legibility

If you wish a reply, give your name, company, mailing address, and date:

---

---

---

---

What is your occupation? \_\_\_\_\_

Number of latest Newsletter associated with this publication: \_\_\_\_\_

Thank you for your cooperation. No postage stamp necessary if mailed in the U.S.A. (Elsewhere, an IBM office or representative will be happy to forward your comments or you may mail directly to the address in the Edition Notice on the back of the title page.)

**Reader's Comment Form**

Fold and tape

**Please Do Not Staple**

Fold and tape



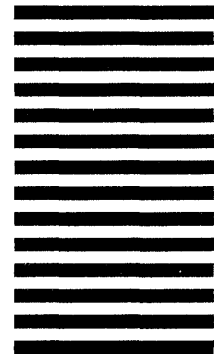
**BUSINESS REPLY MAIL**

FIRST CLASS PERMIT NO. 40 ARMONK, N.Y.

POSTAGE WILL BE PAID BY ADDRESSEE

International Business Machines Corporation  
Dept. E01  
P.O. Box 12195  
Research Triangle Park, N.C. 27709-2195

NO POSTAGE  
NECESSARY  
IF MAILED  
IN THE  
UNITED STATES



Fold and tape

**Please Do Not Staple**

Fold and tape



File No. GENL-30 (SNA)  
SLSS No. 5743-SNA

GC30-3073-1

Printed in USA.

GC30-3073-01

