

# **Systems Network Architecture**

---

## **Technical Overview**

GC30-3073-0

File No. S370/4300/8100-30

SLSS No. 5743-SNA

#### First Edition (March 1982)

Changes are made periodically to the information herein; before using this publication in connection with the operation of IBM systems, consult the latest IBM System/370 and 4300 Processors Bibliography, GC20-0001, for the editions that are applicable and current.

Any reference to an IBM program product in this document is not intended to state or imply that only IBM's program product may be used. Any functionally equivalent program may be used instead.

It is possible that this material may contain references to, or information about, IBM products (machines and programs), programming, 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 products, programming, or services in your country.

Publications are not stocked at the address 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, P.O. Box 12195, Research Triangle Park, North Carolina 27709, U.S.A. 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 publication presents detailed information on the major functions of Systems Network Architecture<sup>1</sup> (SNA) for individuals responsible for designing, installing, programming, administering, and maintaining SNA networks. The book relates the architecture to major products that implement it, and is meant to be used with the product documentation for the SNA products that make up any particular network.

The book contains annotated sequences of request/response units illustrating important SNA functions.

This book assumes that the reader is familiar with the SNA concepts presented in Systems Network Architecture Concepts and Products, GC30-3072.

**Note:** The term "network" has at least two meanings. A public network is a network established and operated by common carriers or telecommunications Administrations for the specific purpose of providing circuit-switched, packet-switched, and leased-circuit services to the public. A user application network is a configuration of data processing products, such as processors, controllers, and terminals, established and operated by users for the purpose of data processing or information exchange, which may use transport services offered by common carriers or telecommunications Administrations. "Network," as used in this publication, refers to a user application network.

The term "end user," as used in this publication, refers to the ultimate source or destination of application data flowing through an SNA network. An end user may be an application program or a terminal operator.

Chapter 1 presents the structure of an SNA network from several points of view. It presents network structure in terms of subareas and nodes, in terms of resource control, and in terms of paths and routes. The chapter depicts the SNA functional layers within nodes for several representative configurations.

Chapter 2 describes how resources in an SNA network are activated and deactivated. The chapter explains both cascaded and serial activation and deactivation, explains the implications of resource control, and describes the communication management configuration (CMC) concept.

---

<sup>1</sup> The description of the logical structure, formats, protocols, and operational sequences for transmitting information units through and controlling the configuration and operation of networks.

Chapter 3 describes how SNA networks transmit data between adjacent nodes over a single link or a group of links, and explains how to specify links and transmission groups between nodes.

Chapter 4 describes how an SNA network routes message units from an origin subarea to a destination subarea. This chapter explains how routes and paths are defined in an SNA network and describes explicit routes, virtual routes, multiple explicit routes, class of service, and regulation of data flow along a route.

Chapter 5 explains how SNA sends data between logical units (LUs) via sessions. This chapter describes LU-LU sessions, the types of logical unit, and various session-level protocols.

The Glossary contains definitions of terms and abbreviations related to Systems Network Architecture and products that are designed in accordance with SNA.

**Note:** The descriptions of functions in this publication apply to (1) the functional capabilities defined by the version of SNA current at the time this edition was published; and (2) the Advanced Communications Function levels of SNA products (for example, ACF/TCAM and ACF/VTAM).

Not all of the functional capabilities described are appropriate for all SNA products, and various SNA products may accordingly implement different combinations of these capabilities. Therefore, the reader should not infer from these descriptions that any particular SNA product of interest has all the functional capabilities of the current version of SNA. An IBM marketing representative can supply detailed information about the specific SNA functions provided by a particular SNA hardware or software product or by a particular combination of such products.

## Prerequisite Publication

Systems Network Architecture Concepts and Products, GC30-3072

## Related Publications

Systems Network Architecture—Sessions between Logical Units, GC20-1868

Systems Network Architecture Reference Summary, GA27-3136

Systems Network Architecture-Format and Protocol Reference Manual: Architectural Logic, SC30-3112

IBM Synchronous Data Link Control General Information, GA27-3093

IBM 3270 Data Stream Programmer's Reference, GA23-0059

IBM Cryptographic Subsystem Concepts and Facilities, GC22-9063

## CONTENTS

### **Chapter 1. SNA Network Structure 1-1**

- Major Components of an SNA Network 1-1
- Path and Explicit Route Structure 1-7
- Network Control Structure 1-13

### **Chapter 2. Managing SNA Network Resources 2-1**

- How SNA Resources are Activated and Deactivated 2-1
  - Activation and Deactivation Overview 2-2
  - The Resource Hierarchy and Cascaded Activation 2-9
  - Reestablishing a Configuration after Resources are Deactivated 2-14
  - Automatic Network Shutdown and Subsequent Restart 2-14
- How Control of SNA Resources is Assumed and Shared 2-15
  - Sharing Control of Resources in an SNA Network 2-15
  - The Communication Management Configuration 2-18
- Benefits of SNA Activation, Deactivation and Shared Control Capabilities 2-19
- Specifying Activation, Deactivation, and Control Options 2-21
  - ACF/VTAM Options 2-21
  - ACF/TCAM Options 2-22
- Typical Request Unit Sequences for Activating and Deactivating Resources 2-22

### **Chapter 3. Transmitting Data From Node to Node 3-1**

- Concepts of Data Transmission between Nodes 3-1
  - Links and Link Stations 3-1
  - Types of Links In SNA Networks 3-4
    - SDLC Links 3-4
    - Data Channels 3-4
    - The X.25 Interface Between SNA Nodes and Packet-Switched Data Networks 3-4
  - Basic Link Units 3-5
  - SDLC Link Configurations 3-5
- Path Control Components Involved in Data Transmission between Nodes 3-6
  - Transmitting Data between Adjacent Subarea Nodes 3-6
  - Transmitting Data between a Subarea Node and a Peripheral Node 3-9
- Example of Data Transmission Between Adjacent Link Stations 3-10
- Benefits of the SNA Node-to-Node Transmission Scheme 3-12
- Specifying Links and Associated Resources to SNA Products 3-13
  - Defining SDLC Links, Link Stations, and Transmission Groups 3-13
    - ACF/TCAM Definition 3-13
    - ACF/NCP Definition 3-14
    - ACF/VTAM Definition 3-14
- Typical SDLC Sequences 3-14

### **Chapter 4. Routing Data from Subarea to Subarea 4-1**

- SNA Routing Overview 4-1
  - Path, Explicit Route, and Route Extension 4-1
  - Intermediate Routing Nodes and Boundary-Function Nodes 4-6

Explicit Routes, Virtual Routes, and Transmission Priorities	4-8
Multiple Explicit Routes and Class of Service	4-10
How SNA Networks Route Messages	4-13
Activating and Deactivating Routes	4-16
Regulating Data Flow Along a Route	4-17
Global and Local Flow-Control Algorithms	4-17
Virtual-Route Pacing	4-19
Effect of Severe Congestion	4-21
Effect of Moderate Congestion	4-21
Error Handling for a Route	4-22
Benefits of the SNA Routing Techniques	4-26
Specifying Routes	4-27
Specifying Routes to ACF/NCP	4-28
Specifying Routes to ACF/VTAM	4-28
Specifying Routes to ACF/TCAM	4-29
Typical Request-Unit Sequences for Routing	4-30

## **Chapter 5. Using LU-LU Sessions to Transmit Data between End Users 5-1**

Types of Logical Units	5-1
Benefits of LU Type Classification	5-2
Activating an LU-LU Session	5-2
Negotiable and Nonnegotiable Bind Session Requests	5-5
Half-Sessions	5-5
Half-Session Components	5-5
Managing the Flow of Data	5-6
Request Headers	5-7
Response Headers	5-8
Normal and Expedited Flows	5-8
Request Units	5-9
FMD Request Units	5-9
Data Flow Control and Session Control Request Units	5-10
Grouping Request Units into RU Chains	5-10
Canceling an RU Chain During Transmission	5-11
Response Units	5-11
Specifying Maximum Request Unit Size	5-12
Request and Response Control Modes	5-12
Grouping RU Chains into Brackets	5-13
Normal-Flow Send/Receive Modes	5-15
Half-Duplex Flip-Flop	5-15
Half-Duplex Contention	5-15
Full-Duplex	5-16
Quiescing Data Flow	5-16
Shutting Down Data Flow	5-17
Sequencing Request Units Flowing in a Session	5-17
Reporting Session Status and Signaling the Session Partner	5-18
Data-Handling Protocols	5-18
Using FM Headers to Control LU Activity	5-19
Improving Transmission Efficiency by Compressing and Compacting Data	5-20
Improving Data Security through Cryptography	5-20
Pacing of Data Flow at the Session Level	5-21
Error Recovery at the Session Level	5-24
Session Outage Notification	5-25

Profiles and Usage Fields	5-26
Summary of LU Types and Representative IBM Products	5-27
Selecting and Using a Data Stream	5-28
SNA Character String Controls	5-28
SNA 3270 Data Streams	5-29
String Control Bytes Used for Compressing and Compacting Data	5-29
Typical Request Unit Sequences for Activating Sessions, Transferring Data, and Deactivating Sessions	5-30

**Glossary** X-1

**Index** X-25





## FIGURES

- 1-1. Hardware Configuration of Sample SNA Network 1-2
- 1-2. SNA Components of Sample SNA Network 1-3
- 1-3. SNA Network Divided into Subareas 1-6
- 1-4. SNA Network Structure: Subareas, Nodes, NAUs, and Half-Sessions 1-8
- 1-5. SNA Network Structure: NAUs Communicating via the Path Control Network 1-9
- 1-6. Elements of the Path Control Network 1-11
- 1-7. SNA Network Structure: Paths and Routes 1-12
- 1-8. SNA Network Divided into Domains 1-14
- 2-1. Sample Network for Description of Component Activation 2-4
- 2-2. Adjacent Link Stations, Link Station Control Blocks, and Network Addresses 2-6
- 2-3. SSCP Resource Hierarchy for a Single-Domain Network 2-11
- 2-4. SSCP Resource Hierarchies for a Two-Domain Network 2-12
- 2-5. Resource Sharing in a Multiple-Domain Network 2-17
- 2-6. Communication Management Configuration 2-20
- 2-7. Symbols and Abbreviations Appearing in Sequence Diagrams of Chapter 2 2-23
- 2-8. Activating a Host Node, a Channel-Attached Subarea Node, and the Channel between Them 2-24
- 2-9. Activating Explicit and Virtual Routes between Adjacent Subarea Nodes 2-25
- 2-10. Activating a Channel-Attached Subarea Node and Attached Links 2-26
- 2-11. Activating a Peripheral Node Attached via a Nonswitched SDLC Link 2-27
- 2-12. Activating a Peripheral Node Attached Via a Switched SDLC Link 2-28
- 2-13. Loading a 3705 Communication Controller with an NCP 2-30
- 2-14. Activating an SDLC Link between Subarea Nodes 2-31
- 2-15. Activating Explicit and Virtual Routes between Nonadjacent Subarea Nodes 2-32
- 2-16. Deactivating Virtual Routes, Explicit Routes, and SDLC Links 2-33
- 2-17. Deactivating a Peripheral Node Attached via a Nonswitched SDLC Link 2-35
- 2-18. Deactivating a Peripheral Node Attached via a Switched SDLC Link 2-36
- 2-19. Deactivating a Channel-Attached Subarea Node and Associated Resources 2-37
- 3-1. Components of an SDLC Link 3-3
- 3-2. Point-to-Point SDLC Link Configuration 3-6
- 3-3. Multipoint SDLC Link Configuration 3-7
- 3-4. Loop Configuration 3-8
- 3-5. Nodes, Links, and Link Stations 3-11
- 3-6. Symbols and Abbreviations Appearing in Sequence Diagrams of Chapter 3 3-15

- 3-7. Negative Response to a Poll 3-16
- 3-8. Positive Response to a Poll with Transfer of Data from Secondary Station to Primary Station 3-17
- 3-9. Disconnecting a Secondary Link Station 3-18
- 3-10. A Secondary Link Station Requests Connection and Preparation to Receive Commands 3-19
- 3-11. Primary and Secondary Link Stations Exchange Numbered Frames 3-20
- 3-12. Secondary Link Station Comes Online, Primary Link Station and Secondary Link Stations Exchange Numbered Information Frames 3-21
- 3-13. Busy Secondary Link Station 3-22
- 3-14. Busy Primary Link Station 3-23
- 3-15. Invalid Command 3-24
- 3-16. Numbering Error in Full-Duplex Exchange 3-25
- 3-17. Secondary Link Station Comes Online, Primary Link Station Sends to One Secondary Link Station and Receives from Another 3-26
- 3-18. Interleaved Primary Link Station Transmissions 3-27
- 3-19. Mode Setting and Inquiry Response 3-28
  - 4-1. Path between Two Logical Units 4-2
  - 4-2. Path between Two Host Logical Units 4-4
  - 4-3. Two Paths Connecting Peripheral LUs 4-5
  - 4-4. Explicit Route between Two Subareas 4-7
  - 4-5. Two Explicit Routes 4-9
  - 4-6. Relationship Among an Explicit Route, Virtual Routes, Logical Units, Sessions, and End Users 4-11
  - 4-7. Two Explicit Routes Between the Same Two Subareas 4-12
  - 4-8. Routing Table Segments for Two Explicit Routes 4-15
  - 4-9. Effect of Flow Control on Throughput for a Virtual Route 4-18
- 4-10. Pacing-Group Size Adjustment Algorithm for Path Control at Sending End Node of a Virtual Route 4-23
- 4-11. Actions by an Intermediate Routing Node to Alleviate Congestion 4-24
- 4-12. Elimination of Virtual-Route Pacing Delay by Dynamic Pacing-Group Size Adjustment 4-25
- 4-13. Symbols and Abbreviations Appearing in Sequence Diagrams of Chapter 4 4-31
- 4-14. Fan-out Propagation of Explicit Route Operative (NC-ER-OP) Requests 4-32
- 4-15. Propagation of Routing Information following Activation of Multiple Transmission Groups between the Same Subareas 4-35
- 4-16. Setting Congestion Indicators in FID4 Transmission Headers of PIUs Traversing a Virtual Route 4-37
  - 5-1. Starting an LU-LU Session 5-3
  - 5-2. One-Stage and Two-Stage Pacing 5-23
  - 5-3. Symbols and Abbreviations Appearing in Sequence Diagrams of Chapter 5 5-31
  - 5-4. Activating a Same-Domain LU-LU Session 5-32
  - 5-5. Activating an SSCP-SSCP Session 5-33
  - 5-6. Activating a Cross-Domain LU-LU Session 5-34
  - 5-7. Deactivating a Same-Domain LU-LU Session 5-35
  - 5-8. Deactivating a Cross-Domain LU-LU Session 5-36
  - 5-9. Cross-Domain Takedown Sequence 5-38
  - 5-10. Communication Using Brackets in Half-Duplex Flip-Flop Mode 5-39

- 5-11. Communication Using Half-Duplex Contention Protocols 5-41
- 5-12. LU-LU Communication Using Half-Duplex Flip-Flop Protocols 5-42
- 5-13. Protocols for Quiescing Data Flow 5-43
- 5-14. Protocols for Deactivating LU-LU Session 5-44



## CHAPTER 1. SNA NETWORK STRUCTURE

This chapter presents the structure of an SNA network in terms of its major components, in terms of paths and routes, and in terms of resource control. It then describes the SNA functional layers<sup>1</sup> within SNA nodes for several representative product configurations.

This chapter explains those concepts about SNA network structure that the reader needs in order to understand the other chapters of this book. It reviews the description of network structure given in SNA Concepts and Products, GC30-3072, and enlarges upon some of the concepts presented in that book. A more detailed presentation of SNA network structure is given in SNA Format and Protocol Reference Manual: Architectural Logic, SC30-3112.

### MAJOR COMPONENTS OF AN SNA NETWORK

An SNA network is the part of a user application network<sup>2</sup> 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<sup>3</sup> (NAUs), boundary-function components,<sup>4</sup> and the path control network.<sup>5</sup>

An SNA network contains a set of interrelated logical components that are superimposed on a configuration of physical components in order to allow users of the network to communicate. The logical components are specified by SNA and implemented in IBM software and hardware products.

Figure 1-1 on page 1-2 shows the hardware configuration on which the following description of a sample SNA network is based. In this figure, three IBM 303x host processors are channel attached to 3705 communication controllers. Multiple point-to-point links connect adjacent controllers. Attached to the communication controllers are a variety of SNA terminals.

---

<sup>1</sup> An SNA layer is 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.

<sup>2</sup> User application network is defined in the Preface.

<sup>3</sup> A logical unit, a physical unit, or a system services control point.

<sup>4</sup> Boundary-function components are described in Chapter 3 under "Transmitting Data between a Subarea Node and a Peripheral Node."

<sup>5</sup> The part of the SNA network that includes the data link control and path control layers.

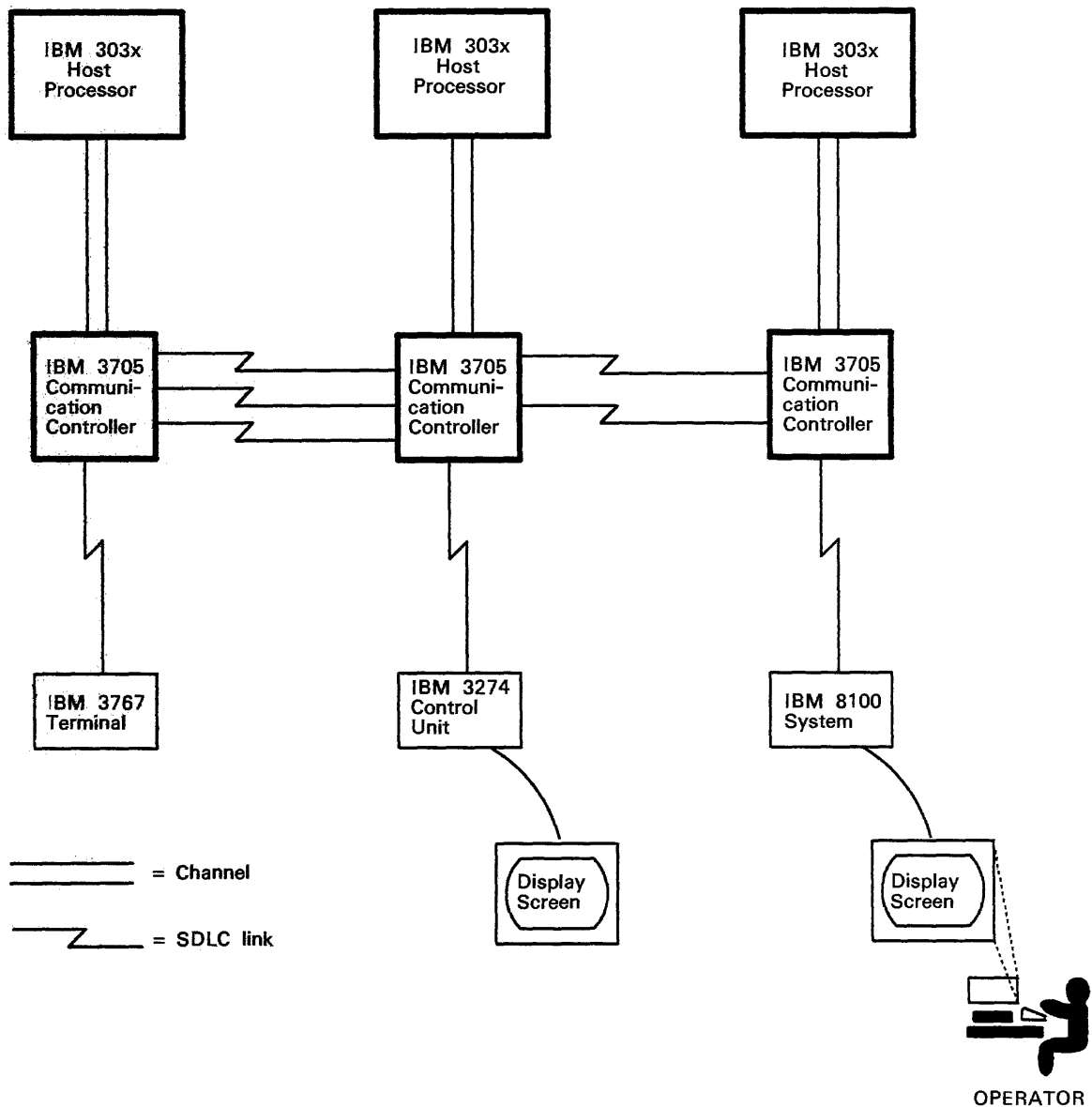


Figure 1-1. Hardware Configuration of Sample SNA Network

Figure 1-2 superimposes on the hardware configuration of Figure 1-1 a set of SNA components. These components, which are implemented in software and in microcode, enable the hardware configuration to function as an SNA network.

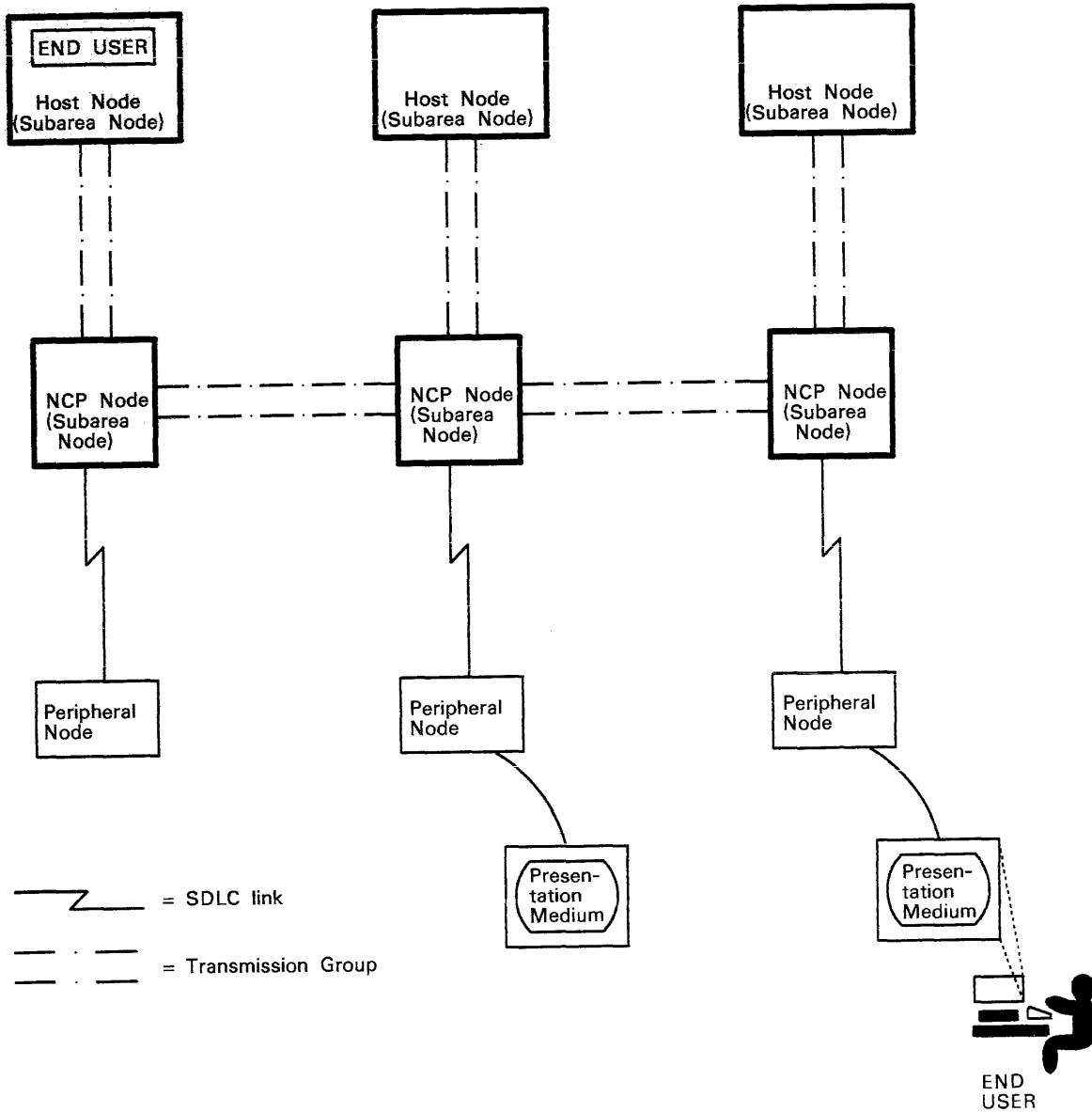


Figure 1-2. SNA Components of Sample SNA Network

When loaded with an SNA access method (ACF/VTAM or ACF/TCAM), each processor becomes a subarea node<sup>6</sup> that contains a system services control point<sup>7</sup> (SSCP). A subarea node that contains an SSCP is called a host node.

Through their SSCPs, SNA access methods control their domains. (A domain is a system services control point (SSCP) and the physical units<sup>8</sup> (PUs), logical units<sup>9</sup> (LUs), links, and associated resources that the SSCP can control by means of activation requests and deactivation requests.) The access methods also contain data link control (DLC) components (called link stations) that help control attached links, and path control (PC) components that help route messages through the network.

In the host processor, the end user of the SNA network is a user-supplied application program that processes messages routed to it. The application program gains access to the network via a logical unit (LU); the logical unit may be located entirely in the access method or partly in the access method and partly in an SNA application subsystem such as CICS/VS, IMS/VS, or JES3. Among other functions, the LU coordinates the sending and receiving of messages and formats data for presentation to the end user.

When loaded with a network control program (ACF/NCP), a 3705 communication controller becomes an NCP node—a kind of subarea node. (In this publication "NCP" refers to ACF/NCP.) These subarea nodes contain data link control (DLC) components (called link stations) that help control channels and Synchronous Data Link Control (SDLC) links. They also contain path control (PC) components that help route messages through the network. The path control components organize into transmission groups<sup>10</sup> the channels between host nodes and NCP nodes and the SDLC links between NCP nodes.

---

<sup>6</sup> A subarea node is a node that uses network addresses for routing and whose routing tables are therefore affected by changes in the configuration of the network. All network addressable units (NAUs), links, and link stations that are addressable within the subarea share a common subarea address and have distinct element addresses.

<sup>7</sup> A focal point within an SNA network for managing the configuration, coordinating network operator and problem determination requests, and providing directory support and other session services for end users of the network.

<sup>8</sup> A physical unit is 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.

<sup>9</sup> A logical unit is 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 SSCPs.

<sup>10</sup> A transmission group is a group of links between adjacent subarea nodes, appearing as a single logical link for routing of messages.



When loaded with IBM-supplied software or microcode, the various SNA terminals shown in Figure 1-1 become SNA peripheral nodes.<sup>11</sup> Peripheral nodes contain path control and data link control components; they also contain logical units (LUs) that serve terminal operators as points of access into the SNA network. Though they vary in complexity, all peripheral nodes contain at least one logical unit.

Figure 1-3 on page 1-6 shows the sample SNA network divided into subareas for routing purposes.<sup>12</sup> A subarea contains one subarea node, and may optionally contain one or more peripheral nodes.

Each subarea node is responsible for special handling of messages to or from peripheral nodes within its subarea. This special handling, which a component called boundary function performs, includes reblocking message units<sup>13</sup> to accommodate differences in buffer size, converting network addresses<sup>14</sup> to local addresses and vice versa, and regulating data flow through session-level pacing.<sup>15</sup> (Session-level pacing is described in Chapter 5 under "Pacing of Data Flow at the Session Level.")

Figure 1-4 on page 1-8 shows:

- The relationship of subareas to the SNA network
- The relationship of nodes and links to subareas
- The relationship of node components to the node

Figure 1-4 shows that an SNA network is composed of subareas connected by transmission groups.

Each subarea contains a single subarea node and may contain one or more peripheral nodes. If the subarea node is a host node, channel attachments connect it to its peripheral nodes. If the subarea node is an NCP node, SDLC links connect it to its peripheral nodes. (NCP nodes are also called communication controller nodes.)

- 
- <sup>11</sup> A peripheral node is a node that uses local addresses for routing and therefore is not affected by changes in network addresses.
- <sup>12</sup> Routing is the function of forwarding 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.
- <sup>13</sup> "Message unit" is a generic term for the unit of data processed by any layer; for example, a basic information unit (BIU), a path information unit (PIU), a request/response unit (RU).
- <sup>14</sup> A network address identifies a link, a link station, or a network addressable unit. Subarea nodes use network addresses; peripheral nodes use local addresses.
- <sup>15</sup> Pacing is a technique by which a receiving component controls the rate of transmission of a sending component to prevent overrun or congestion.

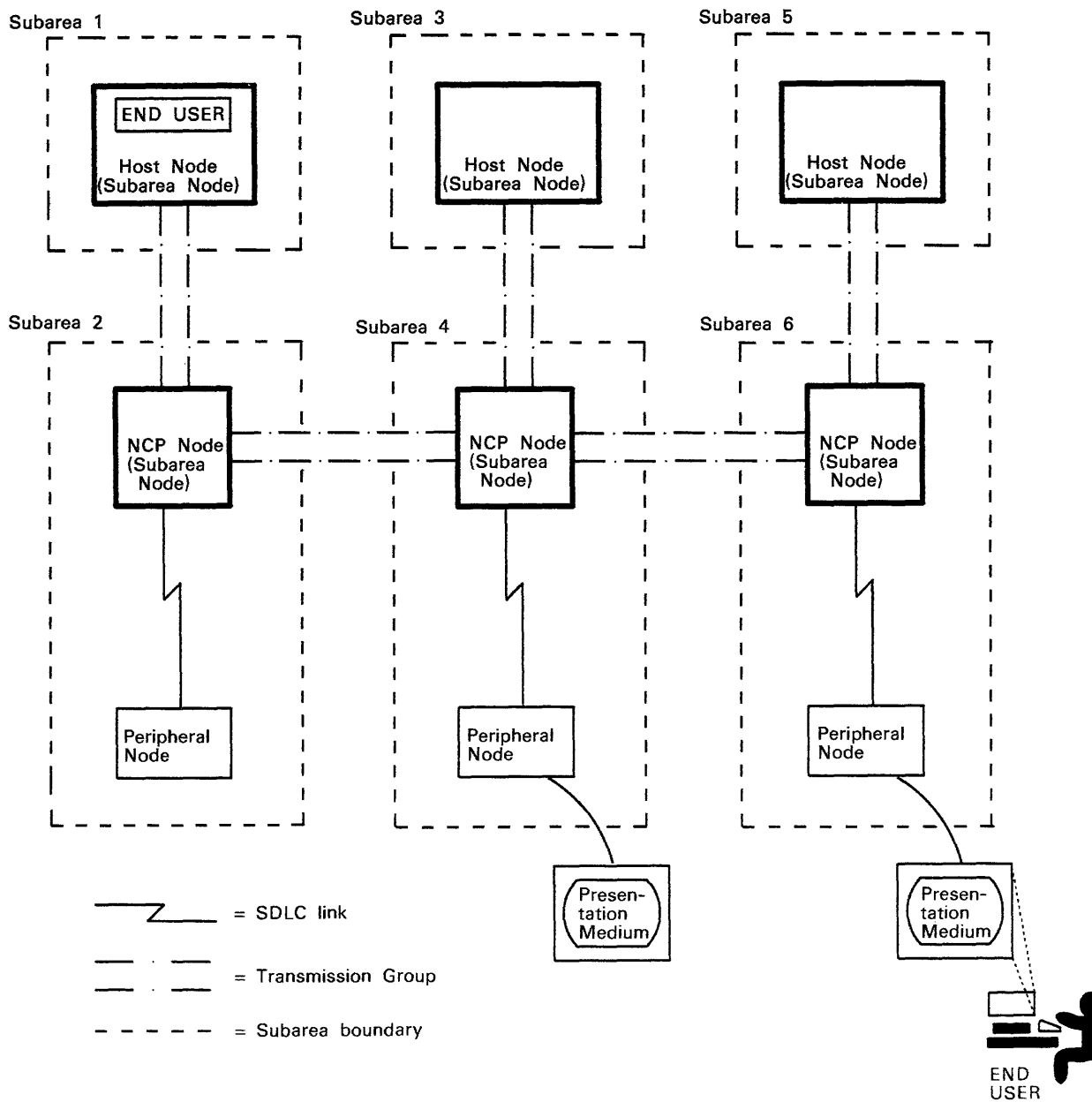


Figure 1-3. SNA Network Divided into Subareas

Each node consists of one or more data link control components called link stations, a single path control element, and one or more network addressable units (NAUs).

Each NAU consists of a NAU services manager and one or more half-sessions. A half-session is a component that provides FMD services,<sup>16</sup> data flow control, and transmission control for one of the sessions of a network addressable unit. Each half-session represents one end of an SNA session<sup>17</sup> involving the NAU.

The NAU services manager provides services for its associated half-sessions. A half-session consists of an FMD services component, a data flow control (DFC) component, and a transmission control (TC) component.

The FMD services component routes requests and responses to particular NAU services manager components and provides session network services or session presentation services, depending on the type of LU involved in the session.

The data flow control (DFC) component (1) controls whether the half-session can send, receive, or concurrently send and receive request units<sup>18</sup> (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.

The transmission control (TC) component keeps track of the status of sessions, synchronizes and paces session-level data traffic, checks session sequence numbers of requests, enciphers and deciphers end-user data, and routes message units received from the path control network to the appropriate points within the NAU.

## PATH AND EXPLICIT ROUTE STRUCTURE

This section describes the structure of an SNA network as it relates to the routing of messages. For a detailed description of message routing, see Chapter 4, "Routing Data from Subarea to Subarea."

Figure 1-5 on page 1-9 shows the communication between NAUs in an SNA network.

---

<sup>16</sup> The abbreviation FMD represents function management data. FMD services is a generic term for session network services and session presentation services, both of which process FMD requests and responses.

<sup>17</sup> An SNA session is a logical connection between two network addressable units (NAUs) that can be activated, tailored to provide various protocols, and deactivated, as requested.

<sup>18</sup> A request unit is a message that contains control information such as a request code or FM header, end-user data, or both.

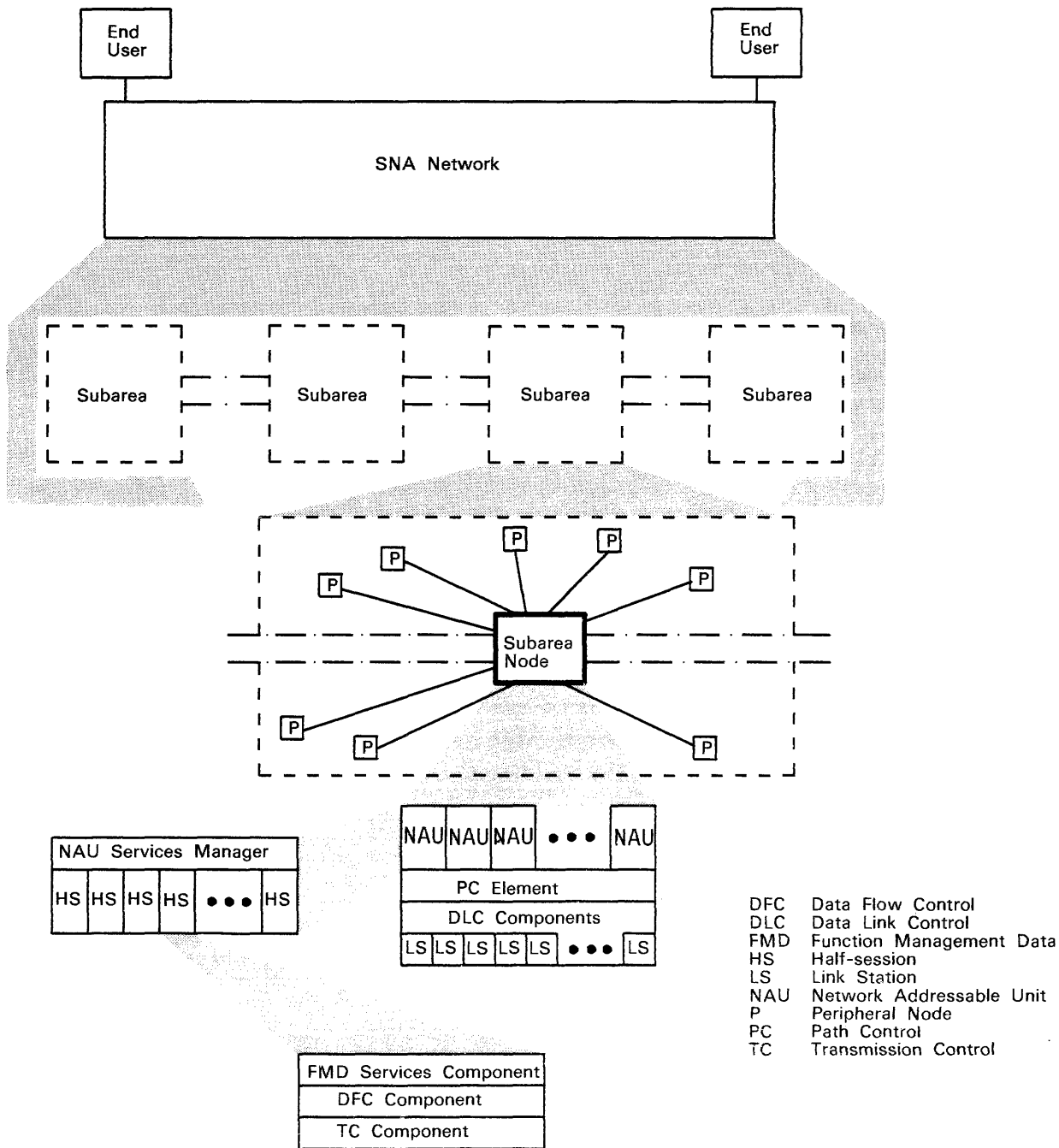


Figure 1-4. SNA Network Structure: Subareas, Nodes, NAUs, and Half-Sessions

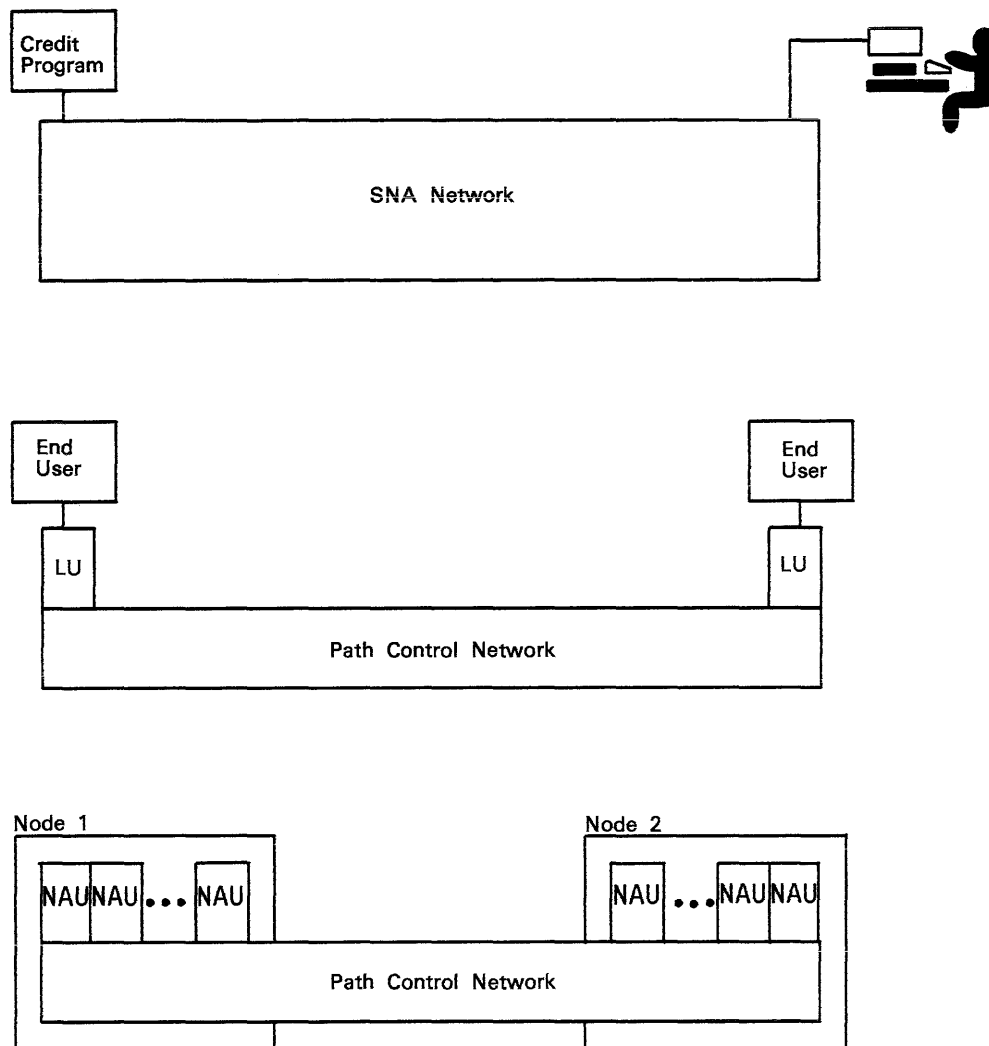


Figure 1-5. SNA Network Structure: NAUs Communicating via the Path Control Network

At the top of the figure, a salesperson and a credit-approval program interact through an SNA network. The salesperson and the program are examples of end users of the network. The part of the network they are using is a pair of LUs, which provide them with ports into the network, and a path control network, which routes the message traffic between them.

As this figure shows, LUs are one type of network addressable unit (NAU). Other types of NAUs are system services control points (SSCPs) and physical units (PUs).

The bottom of Figure 1-5 shows several NAUs in one node using the path control network to communicate with several NAUs that are in another node. The path control network can interconnect NAUs that are in different nodes of the network.

Figure 1-6 on page 1-11 shows how elements of the path control network are dispersed among the nodes in the sample network configuration. The two LUs that allow the salesperson and the credit inquiry program to communicate are connected by path control components in several nodes of the network. These components, together with the transmission groups and links connecting the nodes, make up the path control network for the sample configuration. The path between the two LUs consists of path control components in various nodes and the transmission groups and SDLC links that connect these nodes.

Figure 1-7 on page 1-12 shows in more detail the logical structure of a path between NAUs. The path shown joins a NAU in a subarea node and a NAU in a peripheral node that is not in the subarea node's subarea. The path that connects these two NAUs comprises:

- The path control components of the two subarea nodes and the explicit route<sup>19</sup> that joins them; and
- The boundary function of the subarea node to which the peripheral node is connected, the path control component of the peripheral node, and the SDLC link that joins them. (This link is called the peripheral link.)

The subarea node adjacent to the peripheral node performs boundary function for that peripheral node.

The explicit route consists of transmission groups and the intermediate routing nodes (subarea nodes) that the transmission groups connect. An explicit route contains at least one transmission group, and may contain intermediate routing nodes. Within the intermediate routing nodes, the path control components route messages by selecting the next node on the way to the destination NAU and placing those messages on a transmission group that connects with the selected node. (The node in the middle of Figure 1-6 is an intermediate routing node for the path shown.)

---

<sup>19</sup> An explicit route is the path control network components, including a specific set of one or more transmission groups, 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.

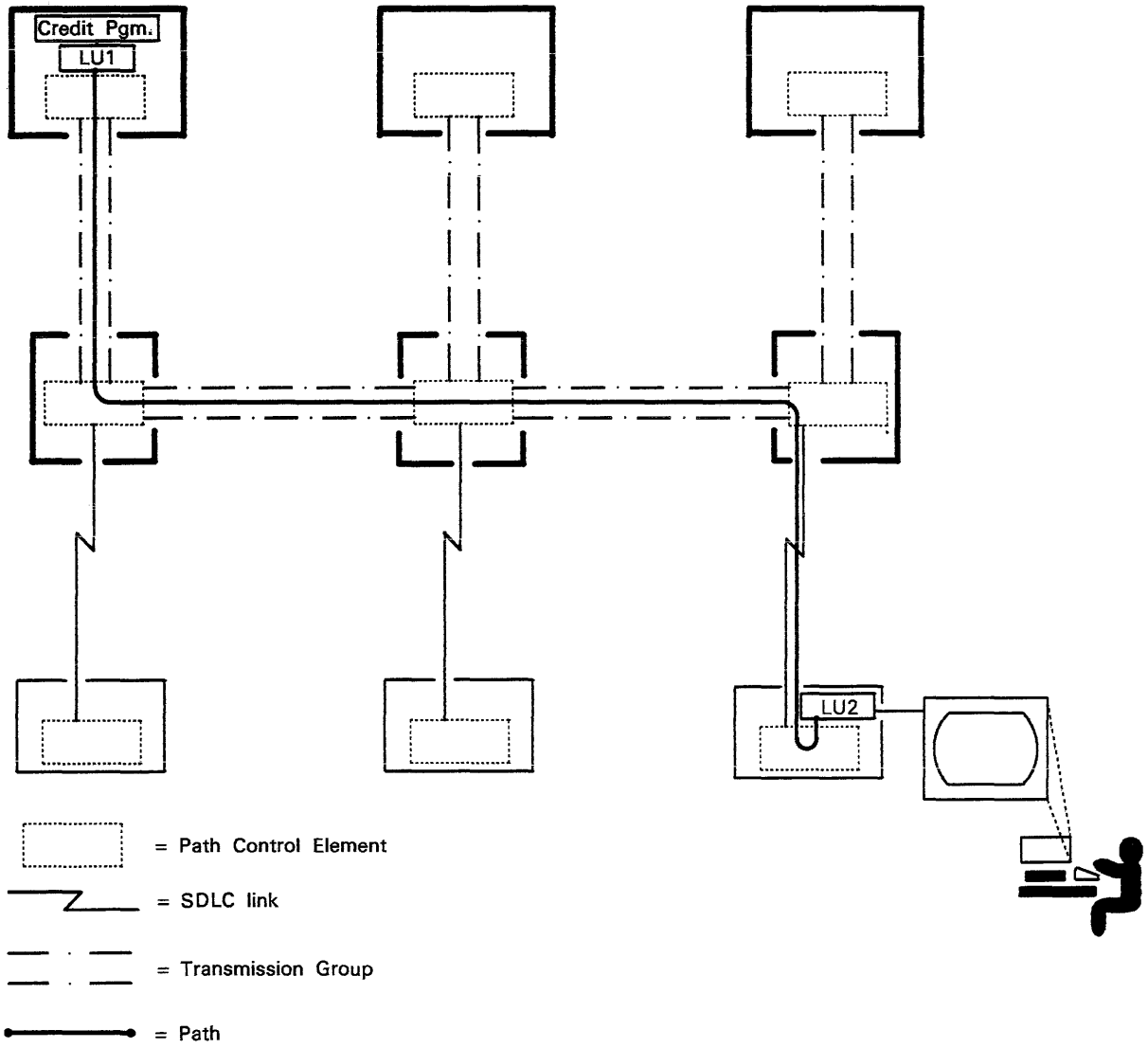
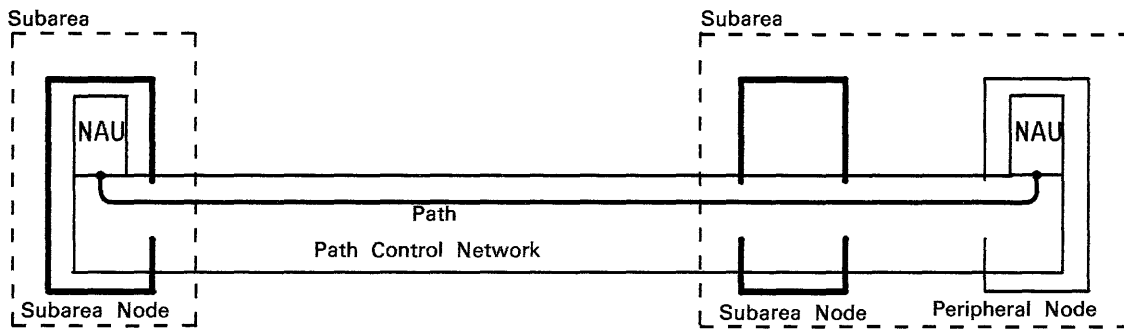
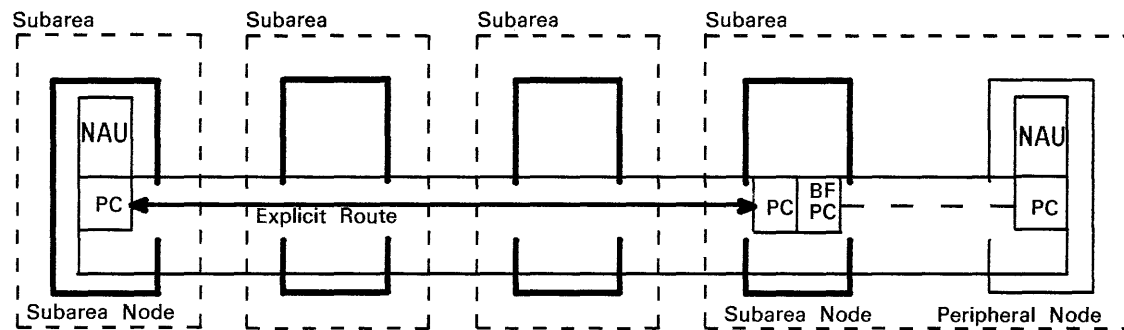


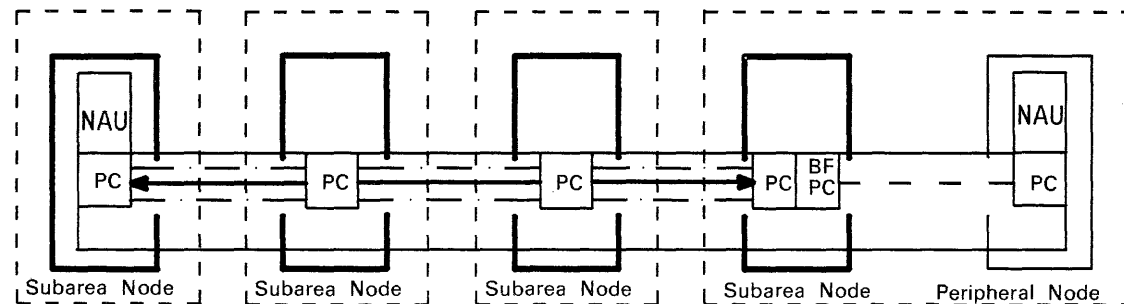
Figure 1-6. Elements of the Path Control Network



(a) Path between two NAUs in Different Subareas



(b) Explicit Route between Path Control Components in Two Subarea Nodes



(c) Explicit Route Formed by Sequence of Transmission Groups and Path Control Components

BF = Boundary Function

NAU = Network Addressable Unit

PC = Path Control

— : — = Transmission Group

Figure 1-7. SNA Network Structure: Paths and Routes



## NETWORK CONTROL STRUCTURE

Figure 1-8 on page 1-14 shows the sample SNA network of Figure 1-2 divided into domains for the purpose of network control. A domain consists of an SSCP and the collection of nodes and associated resources that it can activate. In Figure 1-8, each SSCP controls:

- Two subarea nodes (a host node and an NCP node)
- A channel between the host node and the NCP node
- A peripheral node
- An SDLC link between the NCP node and the peripheral node
- Part of each transmission group linking that SSCP's NCP node to one or two adjacent NCP nodes

Control of SNA network resources is described further in Chapter 2, "Managing SNA Network Resources."

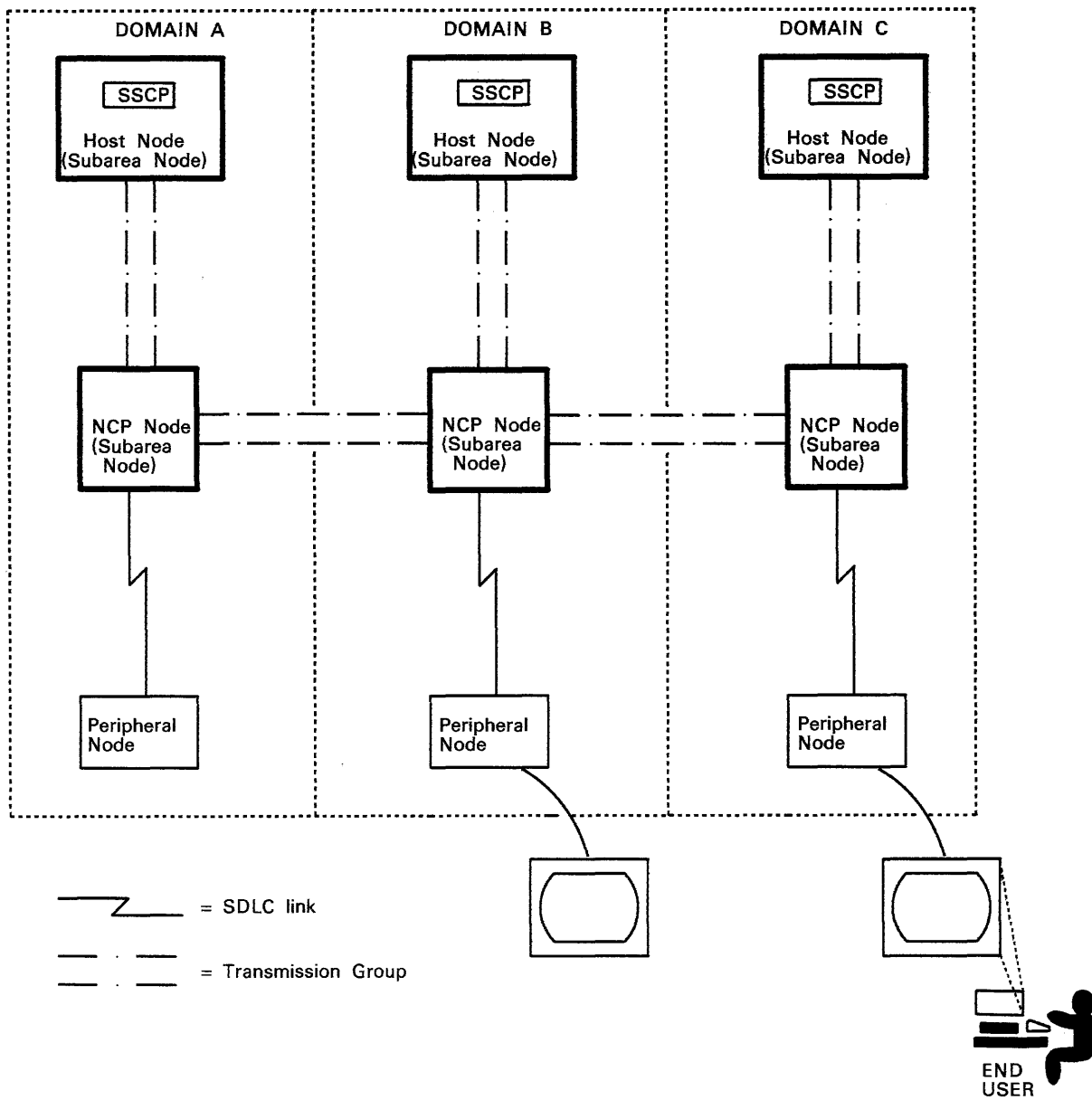


Figure 1-8. SNA Network Divided into Domains

## CHAPTER 2. MANAGING SNA NETWORK RESOURCES

This chapter explains how the resources of an SNA network are activated, deactivated, and controlled.

### HOW SNA RESOURCES ARE ACTIVATED AND DEACTIVATED

An SNA network is a collection of logical resources (such as SSCPs, PUs and LUs), superimposed upon a collection of associated physical resources (such as host processors, 3705 communication controllers, cluster controllers, and terminals), interconnected by links.

Logical resources of the network can be activated after the associated physical resources have had their power turned on. The logical resources are usually deactivated before the power to the associated physical resources is turned off. SSCPs control activation and deactivation of logical resources (other than SSCPs), based upon network design specifications, operator commands, and end-user requests.

This section describes how the logical resources of an SNA network are activated and deactivated.

These are the logical resources that SSCPs activate:

- PUs in subarea nodes and peripheral nodes
- LUs in subarea nodes and peripheral nodes
- Links
- Link stations

Before LU-LU sessions<sup>1</sup> can be activated and two end users can communicate, all logical resources on the path between the end users must be active. (In some cases, a physical unit control point<sup>2</sup> (PUCP) in an NCP node, rather than an SSCP, activates a link between itself and another NCP node. This is explained later in this chapter under "The Communication Management Configuration.")

By activating a resource, an SSCP establishes control over it. The SSCP activates and controls sessions with components of that resource, and is responsible for detecting and correcting error conditions associated

---

<sup>1</sup> An LU-LU session is a session between two logical units in an SNA network. It provides communication between two end users, or between an end user and an LU services component.

<sup>2</sup> A physical unit control point is a component that provides a subset of system services control point (SSCP) functions for activating the physical unit (PU) within its node and its local link resources. Each peripheral node and each subarea node without an SSCP contains a PUCP.

with the resource and its components. For example, the SSCP that controls a peripheral node controls sessions activated with LUs located within that node. Some resources, such as NCP nodes, can be under the control of several SSCPs simultaneously; others, such as peripheral nodes, can be controlled by only one SSCP at a time.

## Activation and Deactivation Overview

This section describes how physical and logical activation and deactivation might proceed for the sample network shown in Figure 2-1 on page 2-4.

The figures in this chapter use certain conventions in denoting SNA network resources. Each SSCP, PU, and LU is assigned a network address and is denoted according to that address. For example, in Figure 2-1 on page 2-4 the SSCP in subarea 1 is assigned network address 1.1 and is denoted SSCP1.1.

SNA network addresses have two fields: (1) a subarea field that identifies the subarea in which the addressed resource is located, and (2) an element field that differentiates the addressed resource from other resources in the same subarea. Network addresses are assigned to SSCPs, PUs, LUs, links, and link stations.

Only subarea nodes know SNA resources in peripheral nodes by their network addresses; within the peripheral nodes, resources are known by local addresses. The boundary function of the subarea node to which a peripheral node is attached makes the required transformations between the network addresses used by subarea nodes and the local addresses used by peripheral nodes. (The use of both network and local addresses allows network address assignments to be changed and access methods and network control programs in subarea nodes to be redefined to reflect changes in network routes without affecting addresses assigned to cluster controllers and terminals.)

In the figures appearing in this chapter, the element address (that is, the element field in the network address) assigned to a PU in a host node is always 0, and the element address assigned to an SSCP is always 1. In a host node element addresses 2 and beyond are assigned to LUs.

---

PLEASE SEE FOLDOUT PAGE AT BACK OF THIS BOOK FOR FIGURE 2-1.

Figure 2-1. Sample Network for Description of Component Activation

---

This page intentionally left blank.

The element address assigned to a PU in an NCP node is always 0. An SDLC link is assigned the first available element address beyond 0 (see PU2.0 and LINK2.1 in Figure 2-1, for example.) After a link is assigned a network address, the link station at the other end of the link from the assigning NCP is assigned the next available element address. (For example, link station LSd in Figure 2-1 is assigned an element address of 2.) If the link extends from the NCP node to a peripheral node, the PU in the peripheral node is assigned the same element address that is assigned to the peripheral node's link station, and any LUs in the peripheral node are assigned sequential element addresses beginning with 1 more than the address of the PU. (See, for example, PU2.2 and LU2.3 in Figure 2-1. PU2.2 has the same element address as LSd, the link station in PNODE2.2.)

Two network addresses are associated with each link between NCP nodes. In Figure 2-1, for example, the top link between NCP2 and NCP3 has a network address of 2.7 in subarea 2 and a network address of 3.1 in subarea 3. PU2.0 associates a network address of 2.7 with this link, while PU3.0 associates a network address of 3.1 with the link.

For links between NCP nodes, the link station at each end of the link is assigned a network address by the NCP in the node at the other end of the link. For example, in Figure 2-1, PU2.0 associates with link station LSh a network address of 2.8 (1 more than the network address of the link for LSh, LINK2.7). PU3.0 associates with link station LSg a network address of 3.2 (1 more than the network address of the link for LSg, LINK3.1).

Though a link station associated with a link between NCP nodes is actually in the node at its end of the link, that link station is logically associated with the node at the other end of the link. For example, in Figure 2-1, link station LSh, which is actually in NCP3's node, has a network address (2.8) that indicates that LSh is in subarea 2. This is the case because PU2.0 is responsible for contacting LSh (via a Contact request) during the link activation procedure. Similarly, PU3.0 is responsible for contacting link station LSg, which is actually in NCP2's node, but has a network address of 3.2.

This relationship is illustrated further in Figure 2-2 on page 2-6. A link station in one node is represented by a control block in the other node to which that link station is connected. In Figure 2-2, for example, link station LSb in node NCP2 is represented by a link station control block in node HOST1. This control block is represented in the figure by a dotted box, and the dotted line shows the relationship between the actual link station in NCP2 and the control block that represents it within HOST1.

From the viewpoint of HOST1, LSb in NCP2 is an adjacent link station. Within HOST1, LSb is represented by a link station control block and is identified by a network address within HOST1's subarea. In this case the address is 1.4.

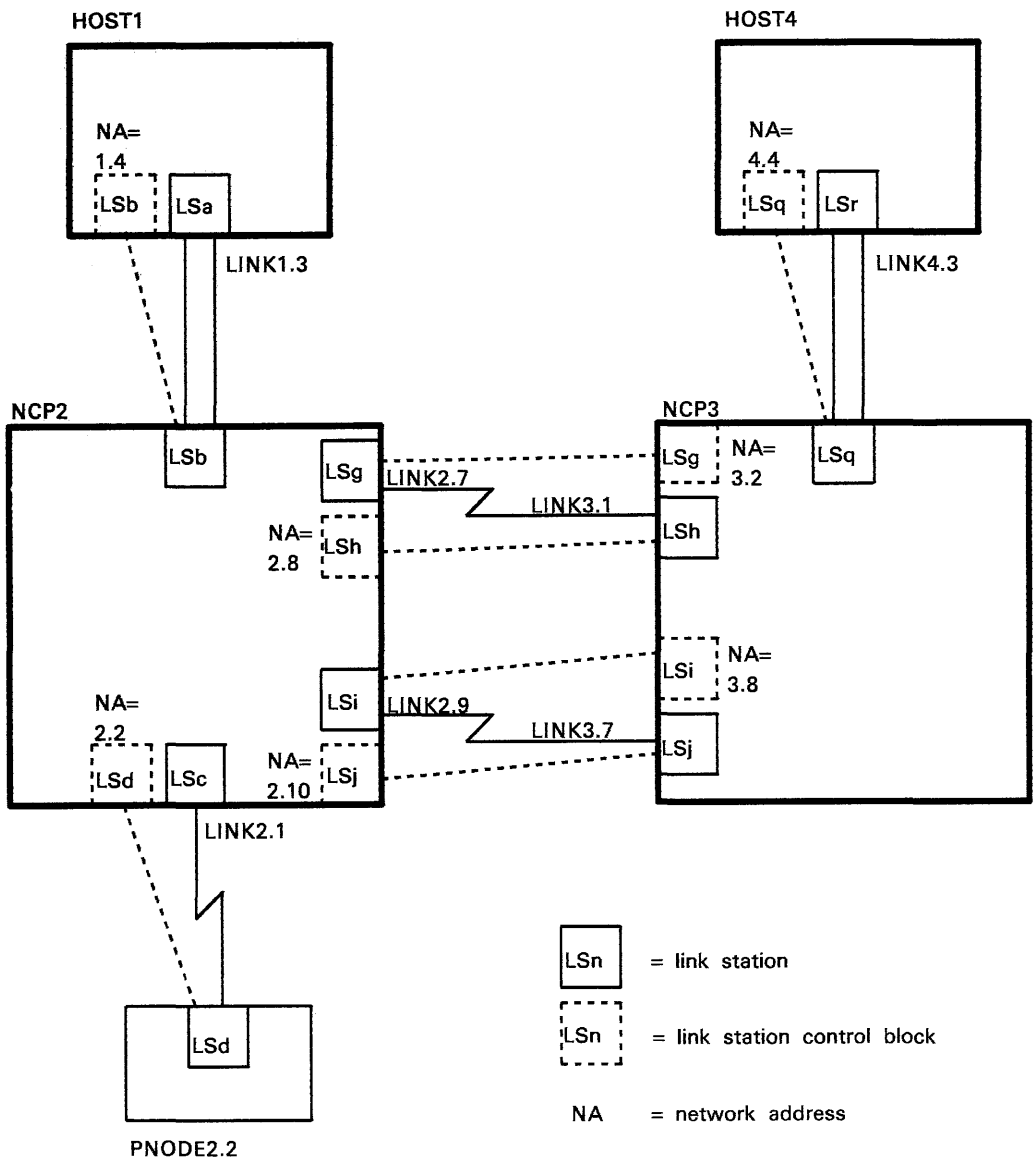


Figure 2-2. Adjacent Link Stations, Link Station Control Blocks, and Network Addresses



Similarly, from the viewpoint of node NCP2, link stations LSd, LSh, and LSj are adjacent link stations. NCP2 contains the link station control block for each of these three adjacent link stations, and NCP2's subarea includes the corresponding network addresses 2.2, 2.8, and 2.10.

Only the node from which a link can be activated contains a link station control block for the link station at the other end of the link. In Figure 2-2, for example, only HOST1 can activate LINK1.3, only HOST4 can activate LINK4.3, and only NCP2 can activate LINK2.1. These nodes therefore contain link station control blocks for the link to NCP2, NCP3, and PNODE2.2, respectively. Because both NCP2 and NCP3 must activate the links between them, each contains the link station control blocks for the adjacent link stations in the other.

Except for resources accessed over a switched SDLC link and those defined with the dynamic reconfiguration facility (described later in this section), resources are assigned network addresses by the SNA access methods and ACF/NCP when these programs are generated. The actual order in which network addresses are assigned is program dependent and depends also on the sequence in which the generation process encounters the resource definition statements. However, the assignment scheme just described reflects the scheme used by ACF/NCP.

ACF/NCP dynamically assigns network addresses to SNA resources in a peripheral node that is accessed over a switched SDLC link after the link connection is completed. ACF/NCP takes these addresses from a pool that it maintains, and returns them to the pool after the connection is broken.

The following sequence of operations will activate all components of the network shown in Figure 2-1. The order shown here is not the only order in which these resources could be activated; the actual order would depend upon the way the system programmer specifies activation parameters to the access method, and the order in which the network operator<sup>3</sup> issues activation requests from the access method.

1. First, operators turn on the power to the physical resources of the network, such as processors, communication controllers, modems, and stations.
2. Next, operators start initial program load (IPL) sequences for processors HOST1 and HOST4. These sequences load operating systems from disk storage into HOST1 and HOST4; the operating systems are then activated.
3. The operators now issue operating system commands to start an SNA access method (ACF/TCAM or ACF/VTAM) in HOST1 and HOST4. The access methods then activate their respective SSCPs (contained within the access-method code): SSCP1.1 and SSCP4.1. The access methods also

---

<sup>3</sup> A person or program responsible for controlling the operation of all or part of a network.

activate their respective physical units and links: SSCP1.1  
activates PU1.0 and LINK1.3 and SSCP4.1 activates PU4.0 and LINK4.3.

4. Under control of the access methods in HOST1 and HOST4, a loader utility program in each host cooperates with a loader program in each channel-attached communication controller to load NCP2 and NCP3 into their respective controllers.
5. After the controllers have been loaded, the loader programs give control to NCP2 and NCP3, which are now able to function as subarea nodes in the network. PU1.0 performs a channel contact operation on LINK1.3. This action causes an explicit route to become operative between HOST1 and NCP2. Following this, SSCP1.1 requests activation of an SSCP-PU session with PU2.0. This request causes the explicit route and the virtual route<sup>4</sup> between HOST1 and NCP2 to be activated and the SSCP-PU session to be activated. (Figure 2-1 does not show the explicit and virtual routes.) PU4.0 and SSCP4.1 in HOST4 perform similar actions that cause NCP3 and the explicit and virtual routes between HOST4 and NCP3 to be activated.
6. SSCP1.1 and SSCP4.1 direct PU2.0 and PU3.0 to activate links in transmission group TG1 between NCP2 and NCP3. (Figure 2-1 shows several transmission groups labeled TG1; these are different transmission groups having the same number.) When at least one link in transmission group TG1 becomes active, PU2.0 and PU3.0 consider the transmission group to be active.
7. When an explicit route between HOST1 and NCP3 is needed for a session, PU1.0 activates such a route. Similarly, PU4.0 activates an explicit route between HOST4 and NCP2 when one is needed for a session. PU1.0 and PU4.0 activate virtual routes using these explicit routes when sessions are assigned to the virtual routes.
8. SSCP4.1 now directs PU3.0 to try to activate a link between the communication controllers for NCP3 and NCP5.
9. After the link has been activated, SSCP4.1 directs PU3.0 to contact NCP5, which may or may not be present in the controller. (NCP5 may be present at this point if, for example, the network has previously been active, NCP3 has failed, and this activation series is an attempt to recover from that failure.) If PU3.0 reports that NCP5 is not present, SSCP4.1 retrieves the load module for NCP5 from a disk data set and sends it to the NCP5 controller via LINK4.3, NCP3, and LINK3.10.

After NCP5 is loaded, PU3.0 and PU5.0 cooperate to activate transmission group TG1 between them. PU4.0 then activates explicit and virtual routes between HOST4 and NCP5. SSCP4.1 then activates a

---

<sup>4</sup> A virtual route is 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 intranode sessions.

session between itself and PU5.0. PU1.0 and PU5.0 now cooperate to establish explicit and virtual routes between HOST1 and NCP5.

10. SSCP1.1 now directs PU2.0 (1) to activate LINK2.1 and LINK2.4 and their associated link stations, and (2) to determine whether peripheral nodes PNODE2.2 and PNODE2.5 are loaded. If not, the access method in HOST1 sends a message to the console, and the operator executes a program that causes the load modules for PNODE2.2 and PNODE2.5 to be loaded. After they are loaded, SSCP1.1 activates SSCP-PU sessions with PU2.2 and PU2.5, thereby activating these physical units. Similar sequences initiated by SSCP4.1 cause PNODE3.4 and PNODE5.2 to be loaded and activated.
11. SSCP1.1 now activates SSCP-LU sessions with LU2.3 and LU2.6, thereby establishing control over them. SSCP4.1 activates SSCP-LU sessions with LU3.5, LU3.6, and LU5.3. Host logical units LU1.2 and LU4.2 are also activated by their SSCPs, although such activation may be implicit rather than explicit, depending upon the access method and LU type.
12. At this point, LU1.2 can activate an LU-LU session with LU2.3 or LU2.6, while LU4.2 can activate an LU-LU session with LU3.5, LU3.6, or LU5.3. Either a host LU or a peripheral LU<sup>5</sup> can ask its controlling SSCP to activate such a session.
13. SSCP1.1 now tries to activate a session with SSCP4.1. As a result, PU1.0 activates an explicit and a virtual route between subarea 1 and subarea 4, and assigns the session to the virtual route.
14. If LU1.2 needs to activate a session with LU5.3, SSCP1.1 must first negotiate session activation with SSCP4.1, which controls access to LU5.3. If SSCP4.1 agrees to activating the session, PU1.0 activates a virtual route on the explicit route between HOST1 and NCP5 and assigns the session to this virtual route.

## The Resource Hierarchy and Cascaded Activation

In the preceding example, resources are activated in a specific order, with superior resources being activated before subordinate resources. For example, SSCP1.1 in Figure 2-1 is activated before NCP2, NCP2 is activated before LINK2.1 and LINK2.4, the links are activated before peripheral nodes PNODE2.2 and PNODE2.5, and the peripheral PUs are activated before the LUs contained within the same nodes. This order, which must be followed in activating an SNA network, is known as the SSCP resource hierarchy.

SNA resources are activated and deactivated either in response to commands from network operators (either human or programmed) or in response to operands (such as ISTATUS and VARY) coded in access-method definition statements. The task of activating a large network would be

---

<sup>5</sup> A logical unit in a peripheral node.

unreasonably difficult if a human operator had to enter a separate command for each resource and remember the network's resource hierarchy when entering these commands. To simplify network activation and deactivation, ACF/VTAM and ACF/TCAM permit the network designer to specify that entire subareas or portions of subareas be activated or deactivated as the result of a single operator command. These procedures are known as cascaded activation and cascaded deactivation.

In the network shown in Figure 2-1, the network designer could specify that SSCP1.1 activate in correct order all subordinate resources whenever NCP2 is activated, and deactivate these resources whenever NCP2 is deactivated.

Cascaded activation can begin or end at any level in the resource hierarchy. For example, the network designer or operator could specify that PNODE2.2 and its subordinate resources be activated whenever NCP2 is activated, but that PNODE2.5 not be activated at that time. In this case, after activating NCP2 the network operator would issue a separate operator command to activate PNODE2.5 and its subordinate resources.

This capability combines the power of cascaded activation with the adaptability required to accommodate such local conditions as differences in work schedules or resource unavailability caused by hardware failures or the need for maintenance.

Figure 2-3 on page 2-11 shows the SSCP resource hierarchy for a simple single-domain network. The SNA access method in HOST1 builds this hierarchy dynamically before subarea node NCP2 is activated. In building the hierarchy, the access method uses NCP network definition statements that the system programmer supplies. After the access method has constructed the hierarchy, SSCP1.1 can activate PU2.0 and the associated resources in an orderly fashion.

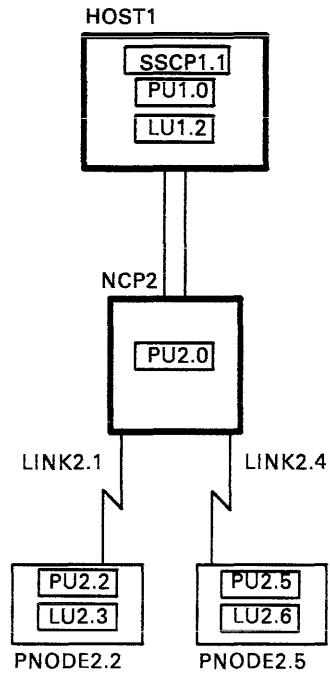
Figure 2-4 on page 2-12 shows resource hierarchies for the more complex two-domain network of Figure 2-1.

As Figure 2-4 illustrates, one resource hierarchy is associated with each active SSCP in the network. Each NCP node and its associated resources constitute a sub-hierarchy. In the figure, each subarea link (that is, a link between subarea nodes) is represented in the hierarchy by (1) a block that represents the local end of the link and (2) a block that represents the link station at the other end of the link.

With respect to the SSCP in a given domain, each SSCP in another domain forms a sub-hierarchy in the given domain's resource hierarchy. Subordinate resources in this sub-hierarchy are the other-domain LUs associated with the other-domain SSCP. The given SSCP uses the hierarchy to determine which SSCP to negotiate with in activating an LU-LU session between an LU in its domain and an other-domain LU.

An SSCP can control all resources in its resource hierarchy (except for other-domain SSCPs and LUs that appear under such SSCPs). In Figure 2-4, PU3.0 and LINK3.3 appear in the resource hierarchies for both SSCP1.1 and SSCP4.1. This is true because SSCP1.1 and SSCP4.1 can

a) Network Configuration



b) Resource Hierarchy for SSCP1.1

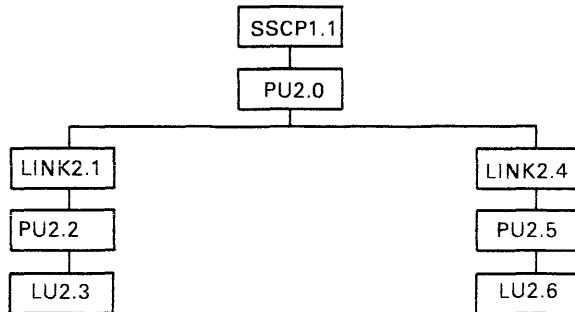
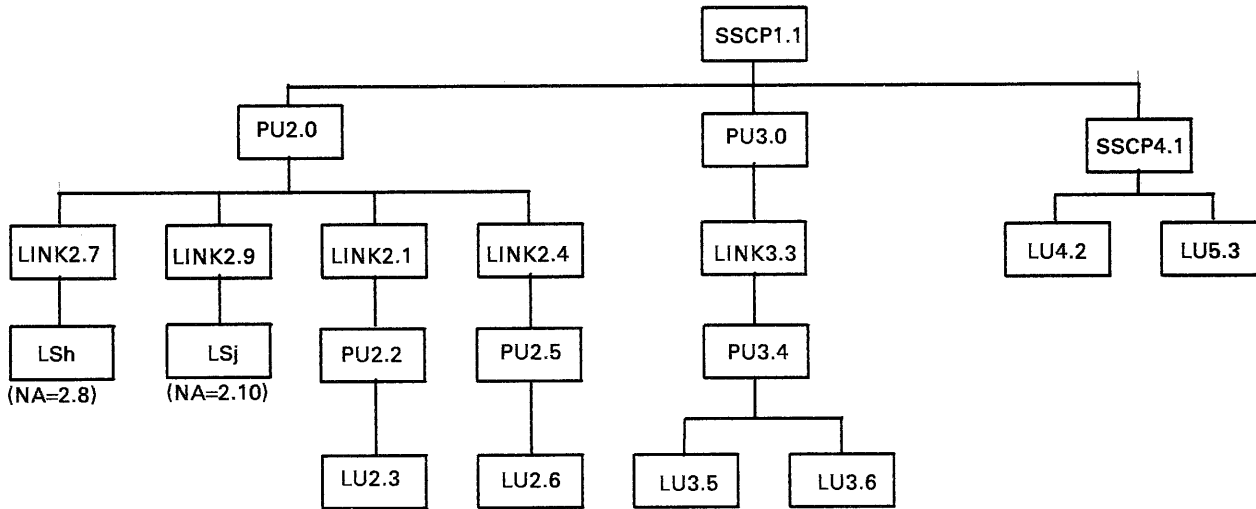


Figure 2-3. SSCP Resource Hierarchy for a Single-Domain Network

a) Resource Hierarchy for SSCP1.1



b) Resource Hierarchy for SSCP4.1

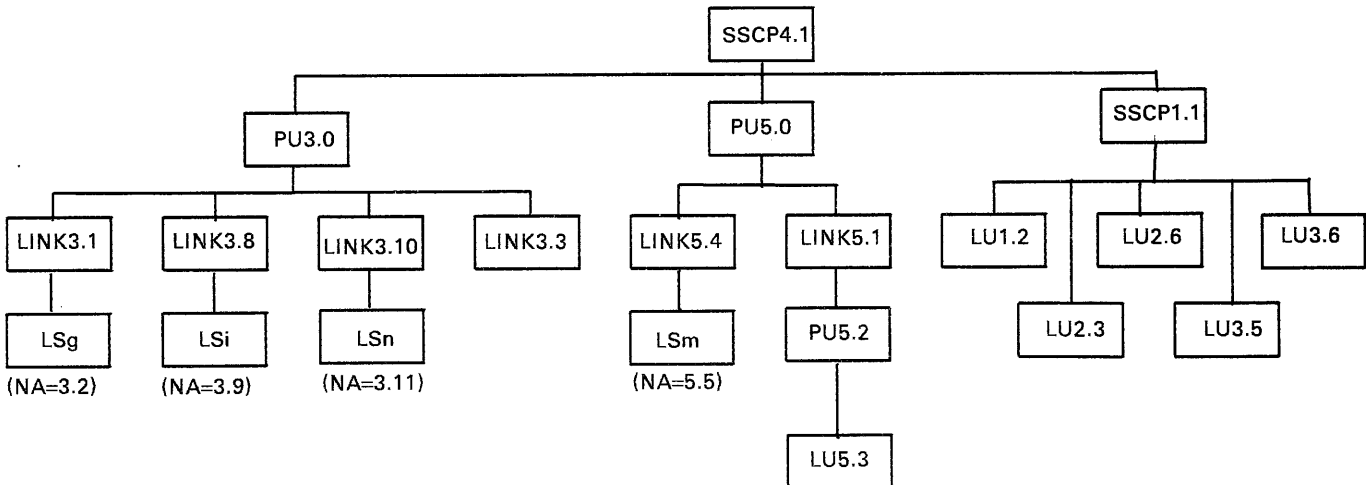


Figure 2-4. SSCP Resource Hierarchies for a Two-Domain Network

share control of NCP3 and LINK3.3 concurrently. Resource control is described further under "How Control of SNA Network Resources is Assumed and Shared" later in this chapter.

The PU for a peripheral node attached directly to a host node appears in the resource hierarchy directly below the controlling SSCP; LUs associated with such a PU appear below the PU.

A switched link appears in the resource hierarchy immediately beneath the PU for the subarea node with which it is associated. When a connection is established between this subarea node and a peripheral node over the switched link, the access method places the PU and LUs associated with the peripheral node into the resource hierarchy below the switched link. When the connection is broken, resources associated with the peripheral node are removed from the resource hierarchy.

The resource hierarchy for a particular SSCP changes over time for a variety of reasons. Some examples follow.

- Sub-hierarchies associated with NCP nodes are added when the SSCP prepares to activate these nodes.
- Resources associated with peripheral nodes on switched links are added to and removed from the resource hierarchy as switched connections with these nodes are made and broken.
- The SNA access methods provide operator commands that allow the network operator to alter the location of resources in a resource hierarchy to reflect changes in control of resources.

For example, an operator at HOST4 in the sample network shown in Figure 2-1 can issue a command that causes SSCP4.1 to assume control of PNODE3.4. This command causes a block for PU3.4 to appear under the block for LINK3.3 in SSCP4.1's hierarchy in Figure 2-4. After PU3.4 is placed in the hierarchy, LU3.5 and LU3.6 are moved from their location subordinate to SSCP1.1 to a location subordinate to PU3.4. SSCP4.1 can now try to activate PNODE3.4 and its associated resources. This attempt will fail if any other SSCP currently controls PNODE3.4 (that is, if any other SSCP is a partner in an SSCP-PU session with PU3.4).

The resource hierarchy is based upon network definitions associated with NCP load modules in each host. These definitions may be changed dynamically during execution via an SNA facility called dynamic reconfiguration.

Dynamic reconfiguration makes use of statements in a host-resident dynamic configuration data set to alter the original configuration specified in NCP resource-definition statements. An operator command associates the changes specified in the data set with the appropriate NCP; after the association is made, an NCP's sub-hierarchy reflects the changes each time the NCP is activated. A network operator can activate another configuration data set to terminate the association and restore the original configuration.

## Reestablishing a Configuration after Resources are Deactivated

After a network has been initially activated, various portions of the network may be deactivated and then reactivated.

When a resource in a resource hierarchy is reactivated, its subordinate resources are generally restored to the activation states they had before the superior resource was deactivated. However, the SNA access methods provide operator commands that can render resources ineligible for cascaded activation. If an operator issues such a command when a resource is inactive, the resource will not be activated when its superior resource is activated; resources subordinate to the ineligible resource will also remain inactive.

After being deactivated because a superior resource was deactivated, an LU-LU session is not reactivated based on its status before it was deactivated. Rather, its reactivation is governed by operands on access-method macro instructions.

An operator command, an error condition, or a failure may cause part of a network to be deactivated. An example of a failure is loss of the ability of an SSCP to communicate with an NCP node whose resources that SSCP controls. Similarly, an operator command or error recovery procedures invoked by network components may cause the deactivated part of the network to be reactivated.

## Automatic Network Shutdown and Subsequent Restart

Upon losing the ability to communicate with the SSCP that activated it, an NCP uses a procedure called automatic network shutdown to deactivate in an orderly way resources associated with the NCP that the SSCP controlled. Using NCP macro operands, the system programmer may specify that the LUs contained within peripheral nodes controlled by the lost SSCP not be deactivated, so that active LU-LU sessions involving LUs within these nodes are not disrupted (unless the LU in the peripheral node has also lost the ability to communicate with its session partner). Links to other subareas also remain active, so that sessions currently assigned to virtual routes that use these paths are not disrupted.

Upon regaining its ability to communicate with an NCP, the SSCP may reactivate the NCP and resources subordinate to that NCP. Certain peripheral nodes permit an SSCP to reactivate an SSCP-PU session with their PUs without disrupting active LU-LU sessions. SSCP-SSCP sessions may also be reactivated without disrupting LU-LU sessions that involve LUs that the two SSCPs controlled.

## HOW CONTROL OF SNA RESOURCES IS ASSUMED AND SHARED

An SNA network is controlled and managed by one or more system services control points (SSCPs). Each SSCP in a network controls a subset of the network's resources. Among the resources controlled by one or more SSCP are:



- Subarea nodes
- Peripheral nodes
- Links and link stations
- Logical units

To gain control of a subarea node or a peripheral node, an SSCP activates an SSCP-PU session with the PU in that node. To gain control of an LU, an SSCP activates an SSCP-LU session with it. To gain control of a link, an SSCP activates the link with an Activate Link (ACTLINK) request. To gain control of a link station, an SSCP sends it a Contact request.

The SSCP controls its domain in a manner specified by the network designer, the network operator, and network-management program products. The SSCP translates network definition statements, operator commands, and network management program commands into SNA requests; these requests activate, deactivate, and change the status of network resources.

## Sharing Control of Resources in an SNA Network

More than one SSCP can share control of some resources. Sharing of resources not only provides flexibility for normal operations but also is important in backup and recovery procedures.

SNA permits some types of resources to be shared concurrently and other types to be shared serially. Concurrent sharing means that a resource may be in an active state for more than one SSCP at the same time. Serial sharing means that a resource may be in an active state for only one SSCP at a time.

These types of resources can be concurrently shared by up to eight SSCPs:

- The PU in a subarea node that contains an NCP (via Activate Physical Unit [ACTPU] requests)
- A nonswitched SDLC link (via Activate Link [ACTLINK] requests)
- An adjacent link station that represents another NCP node (via Contact requests)

These types of resources can be shared serially:

- A switched SDLC link (via Connect Out [CONNOUT] or Activate Connect In [ACTCONNIN] and Contact requests)
- A PU and its associated link station and LUs in a peripheral node (via Activate Physical Unit [ACTPU], Contact, and Activate Logical Unit [ACTLU] requests)
- The peripheral link between a peripheral node and its subarea node (via ACTLINK [if the link is nonswitched] or CONNOUT or ACTCONNIN and CONTACT requests [if the link is switched]).

Figure 2-5 on page 2-17 shows an SNA resource-sharing configuration in which resources attached to a communication controller containing an NCP

are allocated among four domains. The subarea node that contains the NCP is in all four of these domains.

In this figure, subarea nodes HOST3 and HOST4 are channel attached to subarea node NCP1234. Domain C contains HOST3, NCP1234, and peripheral node PNODE3.1, while domain D contains HOST4, NCP1234, and peripheral node PNODE4.1.

Subarea nodes HOST1 and HOST2 are linked to NCP1234 via SDLC links between their own channel-attached NCPs (NCP1 and NCP2, respectively) and NCP1234. Domain A consists of HOST1, NCP1, PNODE1.1, and NCP1234, while domain B contains HOST2, NCP2, PNODE2.1, and NCP1234.

Any of the four host nodes may load NCP1234 into its communication controller. In order to assume control of NCP1234 and its associated resources, the SSCPs in host nodes 1, 2, 3, and 4 must have available to them network definitions for the NCP and its resources.

Each SSCP that shares control of NCP1234 may activate the NCP by sending it an Activate Physical Unit (ACTPU) request. This activates an SSCP-PU session between the SSCP and the NCP. After activating this session and each link to peripheral nodes controlled by the NCP, the SSCP can activate the peripheral nodes by sending the NCP an ACTPU request for each one. This results in an active SSCP-PU session between the SSCP and each peripheral node. The SSCP then controls that peripheral node and can send Activate Logical Unit (ACTLU) requests to activate SSCP-LU sessions with LUs within the node. Logical units can then activate same-domain LU-LU sessions via Bind Session (BIND) requests.

When an NCP receives an Activate Physical Unit (ACTPU) request for a peripheral node, it rejects the request if the node already has an active session with another SSCP. Hence, these resources and the LUs associated with them are available to SSCPs sharing the NCP on a first-come, first-served basis.

An SDLC link connecting several PUs may be activated by each of the SSCPs that share control of the NCP's resources; the SSCPs can issue Contact requests to different adjacent link stations that share the link. As a result, each of several SSCPs may simultaneously have active sessions with different peripheral nodes on the same link.

In Figure 2-5, if peripheral node PNODE1.1 is active in domain A, the SSCP in domain A must deactivate it before the SSCP in domain B can activate it. If the SSCP in domain B activates PNODE1, then the SSCPs in domains A, C, and D must be informed of this fact so that they will negotiate with the SSCP in domain B to start LU-LU sessions between LUs in domains A, C, and D and LUs associated with PNODE1.1. SNA access methods provide operator commands for informing SSCPs of a change in resource control.

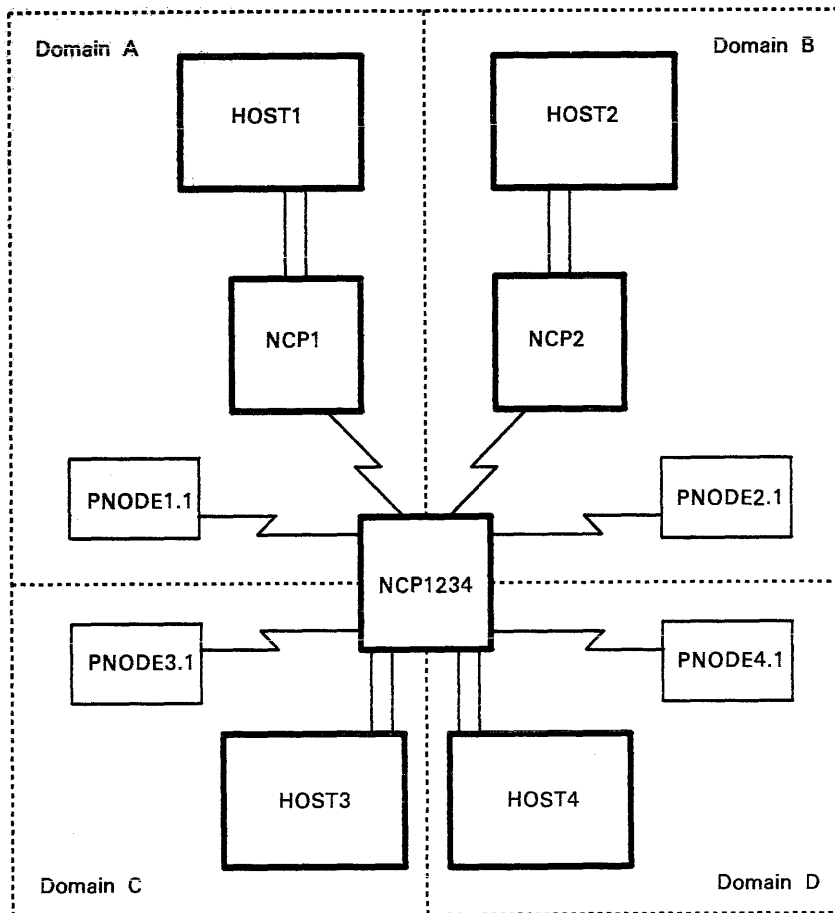


Figure 2-5. Resource Sharing in a Multiple-Domain Network

### The Communication Management Configuration

As the controller of resources, an SSCP must initiate and respond to many SNA requests. For example, an SSCP must process requests to contact channel-attached NCPs; activate (and deactivate) resources; and activate same-domain LU-LU sessions, SSCP-SSCP sessions, and cross-domain LU-LU sessions. Also, an SSCP participates in error recovery and maintenance for the resources it controls. When an SSCP controls and manages resources, its host node has less time for application processing.

SNA access methods support a communication management configuration (CMC) in which the SSCP in one host node (called a CMC host) controls and manages most of the resources in the network, leaving other host nodes (called application hosts) more available for application processing. A communication management configuration also simplifies operator control of network resources.

The SSCP in the CMC host can control the NCPs, SDLC links, and peripheral nodes in the network; The SSCP in each application host controls its host LUs and any locally attached resources. An application host uses the network without controlling many of the network's resources.

In a communication management configuration, application hosts need to use their channel-attached NCPs as ports into the network. To do so, application hosts must be able to establish a logical connection with those NCPs. This connection is ordinarily established during the activation process; the SSCP in the application host activates an SSCP-PU session with the PU in an NCP before it routes data through that NCP into the network. However, ACF/TCAM provides an operator command to establish the logical connection between the application host and the channel-attached NCP without first activating the SSCP-PU session.

ACF/NCP's SDLC monitor mode function (SMMF) is useful in establishing a communication management configuration. SMMF allows two NCPs to activate links between them even if the PU in one or both NCPs does not currently have an active session with an SSCP. When SMMF is used, the physical unit control point (PUCP) in the NCP issues the SNA commands otherwise issued by an SSCP to activate one end of a link between NCPs. Figure 2-14 on page 2-31 shows the sequence of commands for a link that is being activated with the aid of a PUCP.

The communication management configuration has several benefits. Because the SSCP in the CMC host is responsible for activating, deactivating, and maintaining most of the resources in the network, control can be centrally located at the CMC host. Also, because the CMC host manages the network resources, application hosts can concentrate on application processing.

A communication management configuration may have some disadvantages, however. For example, all network-managing capability is lost if the CMC host fails, unless one or more of the application hosts are used as backup CMC hosts capable of assuming control of at least some of the network resources. Also, in a large network, the CMC host processes many requests to manage resources; a bottleneck that slows down the network might develop if the CMC host has insufficient processing power to handle peak demands upon it.

Figure 2-6 on page 2-20 shows a possible communication management configuration. The dotted lines indicate the domains in this network. In the network shown in Figure 2-6, HOST1 is the CMC host. HOST1 can use both NCP1 and NCP2 for explicit routes to other resources it controls. If either NCP1 or NCP2 fails, HOST1 can still have operative explicit routes to other resources in its domain (except those

peripheral resources attached to the NCP that fails); another host node need not assume control of the failed NCP.

Also, if NCP2 is used as the port for the primary explicit routes to NCP3 and NCP4, and NCP1 is used as the port for the primary explicit routes to NCP5 and NCP6, it is less likely that congestion that slows down the network will develop at either NCP1 or NCP2. However, congestion could still develop at HOST1 if it has insufficient processing power to manage a network of this size.

## **BENEFITS OF SNA ACTIVATION, DEACTIVATION AND SHARED CONTROL CAPABILITIES**

The SNA activation and deactivation capabilities allow the system programmer considerable flexibility in specifying how the network is to be activated and deactivated. Some benefits of these capabilities are as follows.

The cascaded activation capability allows the programmer to expand or contract the scope of operator commands used for resource activation. Using parameters on network definition statements, the system programmer can cause most of the network to be activated automatically via cascaded activation, while requiring selected resources to be activated individually via individual operator commands.

For example, the system programmer can specify that when an operator activates an NCP node, the access method is to automatically activate all associated link stations, peripheral nodes, and LUs except for one particular peripheral node and its LUs, which the operator must individually activate.

SNA also provides a cascaded deactivation capability that allows a network operator to deactivate more than one network resource with a single operator command. For example, if the network operator issues a command to deactivate an NCP node, the access method automatically deactivates the peripheral nodes, link stations, and LUs associated with that NCP node.

When reactivating network resources after part of the network failed or was deactivated, the operator may either restore the inactive portion of the network to the activation status it had just before it failed or was deactivated, or restore only a part of the inactive portion to its previous activation status.

In summary, the SNA network activation and deactivation capabilities allow the network designer great latitude in customizing the order and method of activation to meet the needs of the particular network.

Some benefits of the SNA shared control capability are as follows.

By permitting SSCPs to share control of network resources, SNA allows the network management task to be distributed among network nodes according to their capabilities or according to the time of day.

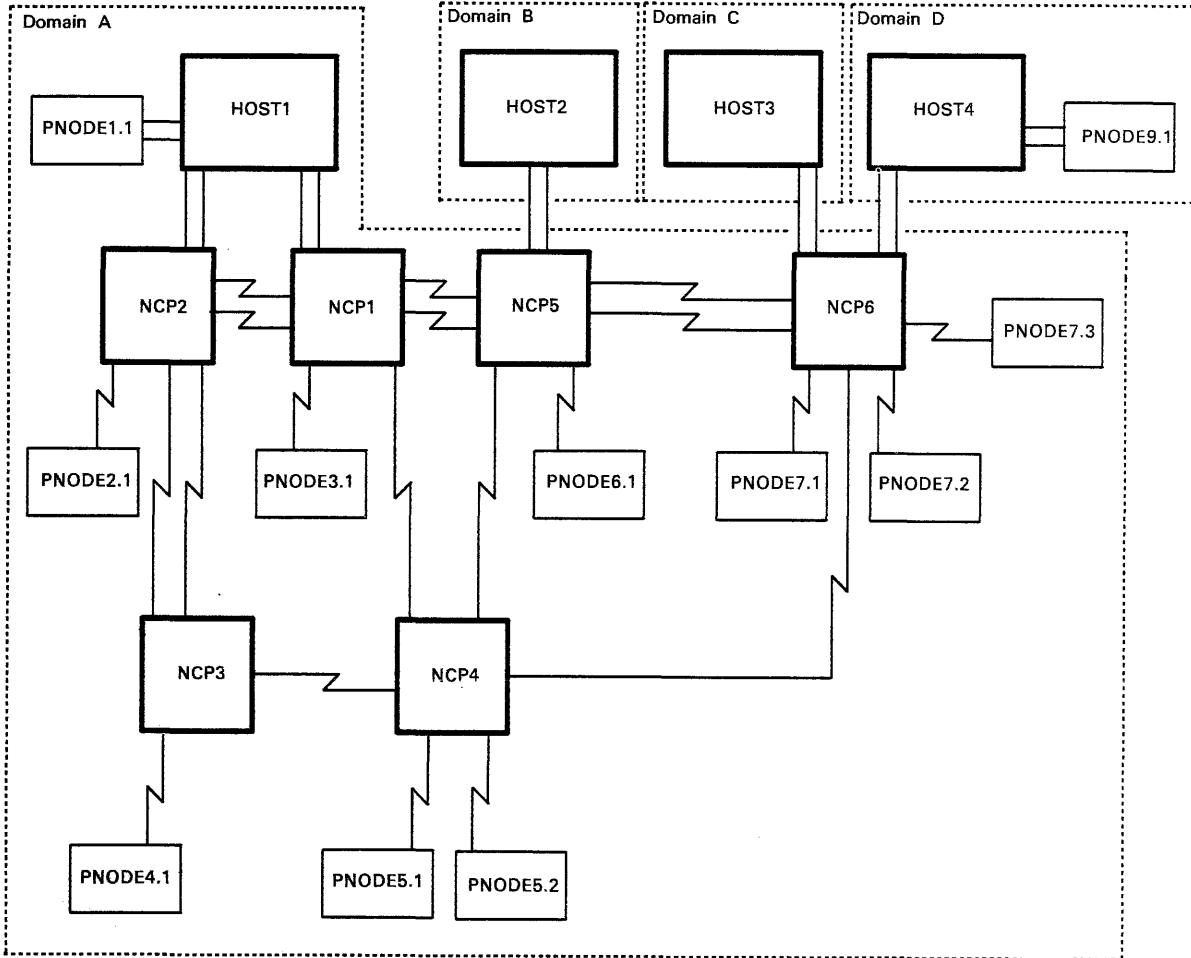


Figure 2-6. Communication Management Configuration

Network management overhead can be allocated among SSCPs based upon the anticipated available processing power of the nodes in which the SSCPs reside.

When a node that contains an SSCP fails, other SSCPs can assume control of the failed SSCP's resources, thereby allowing these resources to continue to participate in the network.

The physical unit control point (PUCP) in nodes that do not contain an SSCP can share control of some such node resources as its PU, links to other nodes, and adjacent link stations. This capability allows such resources to be activated from within the node.

SNA's shared control scheme allows the network designer to implement the communication management configuration, in which an SSCP in one host node assumes control of most of the network resources, thereby allowing the other host nodes in the network to concentrate on application processing.

## SPECIFYING ACTIVATION, DEACTIVATION, AND CONTROL OPTIONS

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

### 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 have available to it 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 upon 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 may use the VARY operator command to alter the activation status of an SNA resource.

## ACF/TCAM Options

In order for ACF/TCAM to assume control of an SNA resource, the resource must be defined to it. ACF/TCAM TERMINAL macros define NCPs, links, physical units, and logical units to ACF/TCAM. For NCP-controlled resources, ACF/TCAM must also have available to it 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 via the ACTIV operand of the TERMINAL macro that defines the resource.

The network operator may use the Activate SNA Resource and Deactivate SNA Resource basic operator commands to alter the activation status of an SNA resource.

## TYPICAL REQUEST UNIT SEQUENCES FOR ACTIVATING AND DEACTIVATING RESOURCES

Figure 2-8 through Figure 2-19 present typical request unit sequences for activating and deactivating various portions of the network shown in Figure 2-1. (Figure 2-7 on page 2-23 gives the meaning of each of the symbols and abbreviations appearing in these request unit sequences.) Though they do not cover all possible activation and deactivation scenarios, these sequences should give a general idea of the way in which activation and deactivation occur in an SNA network.

Typical request unit sequences for initiating and terminating LU-LU sessions are given in Chapter 5, "Using LU-LU Sessions to Transmit Data between End Users."



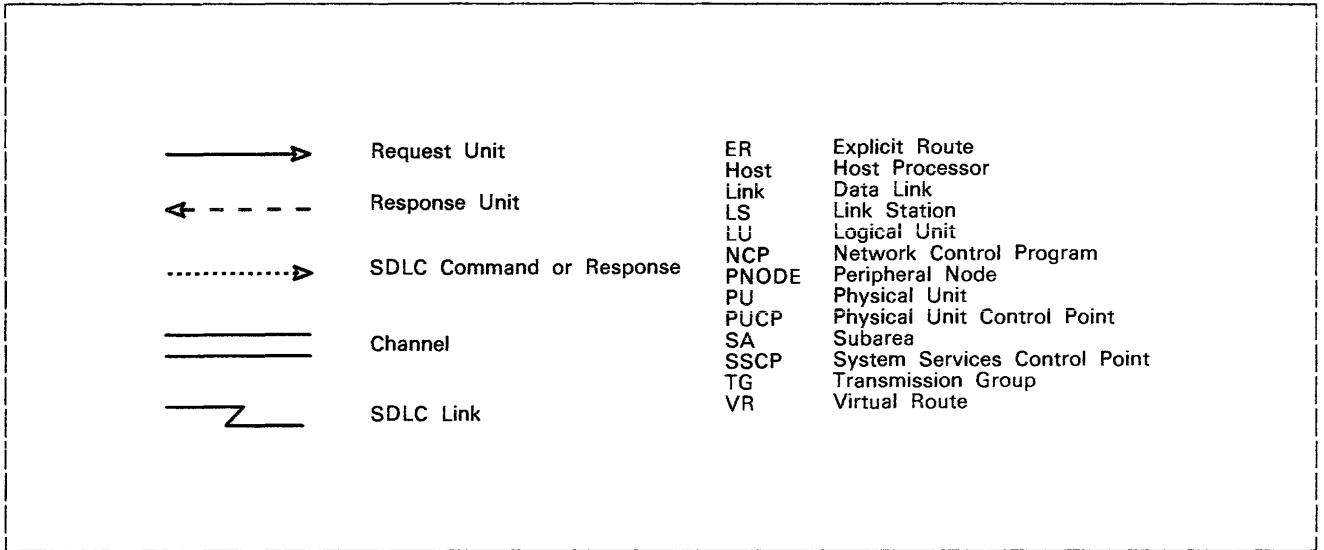
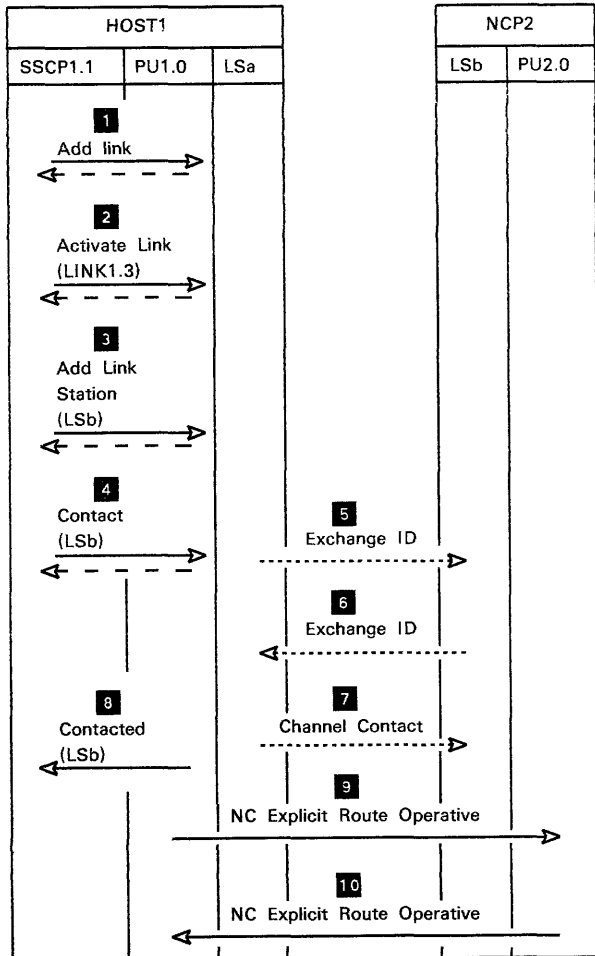
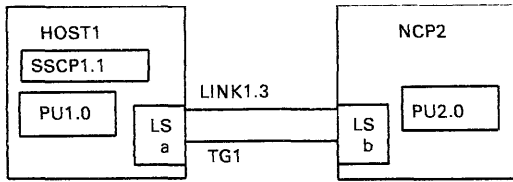


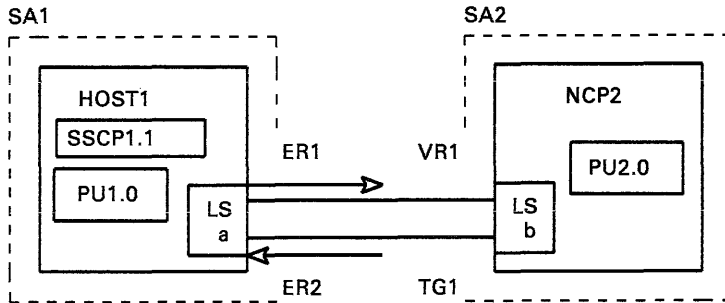
Figure 2-7. Symbols and Abbreviations Appearing in Sequence Diagrams of Chapter 2



1. SSCP1.1 requests PU1.0 to furnish SSCP1.1 with a network address for the designated link. PU1.0 does so.
2. SSCP1.1 tells PU1.0 to activate LINK1.3 and to prepare to issue and receive data link control commands and responses for the link.
3. SSCP1.1 requests PU1.0 to furnish SSCP1.1 with a network address for the designated link station. PU1.0 does so.
4. SSCP1.1 tells PU1.0 to contact the adjacent link station LSb. The representation of LSb in HOST1 has a network address of 1.4.
5. PU1.0 sends PU2.0 information about HOST1, including the maximum number of bytes that HOST1 will accept across the channel at one time.
6. PU2.0 informs PU1.0 that the parameters sent by PU1.0 are acceptable, and sends PU1.0 information about PU2.0.
7. PU1.0 completes the activation of TG1 by accepting the parameters sent by PU2.0.
8. PU1.0 informs SSCP1.1 that message units can now be sent to PU2.0 through link station LSa.
9. PU1.0 tells PU2.0 which subareas can be reached from HOST1, and which explicit routes are used to reach subareas.
10. PU2.0 tells PU1.0 which subareas can be reached from NCP2, and which explicit routes are used to reach these subareas.

(Figure 2-7 gives the meanings of the symbols and abbreviations that appear in this figure.)

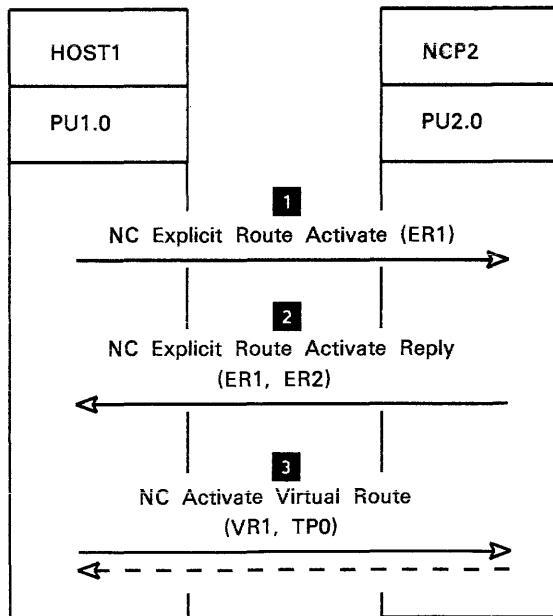
Figure 2-8. Activating a Host Node, a Channel-Attached Subarea Node, and the Channel between Them



1. PU1.0 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.

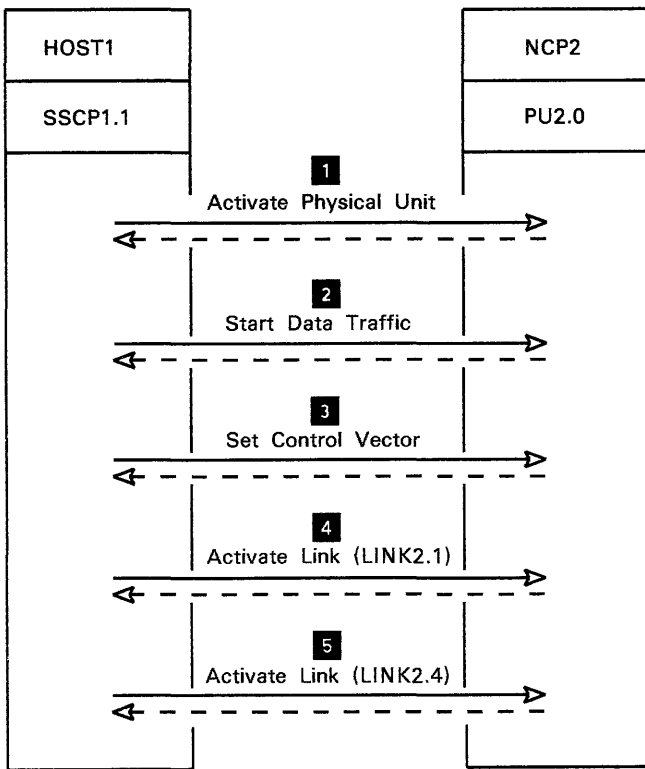
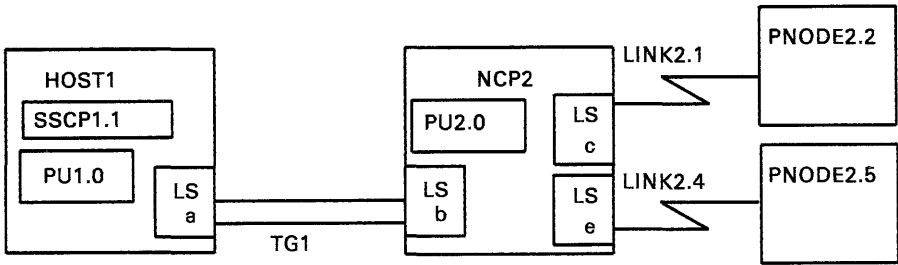
2. PU2.0 completes activation of the explicit route between subarea 1 and subarea 2, replying that the reverse explicit route number for this explicit route is 2. (ER1 and ER2 in this case refer to the same explicit route; an explicit route is known by two explicit route numbers--one for each direction.) PU2.0 indicates to PU1.0 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. PU1.0 activates a virtual route between subareas 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 host-to-NCP direction) and 2 (in the reverse direction). An inactive virtual route is activated when it is needed to satisfy a session activation request.



(Figure 2-7 gives the meanings of the symbols and abbreviations that appear in this figure.)

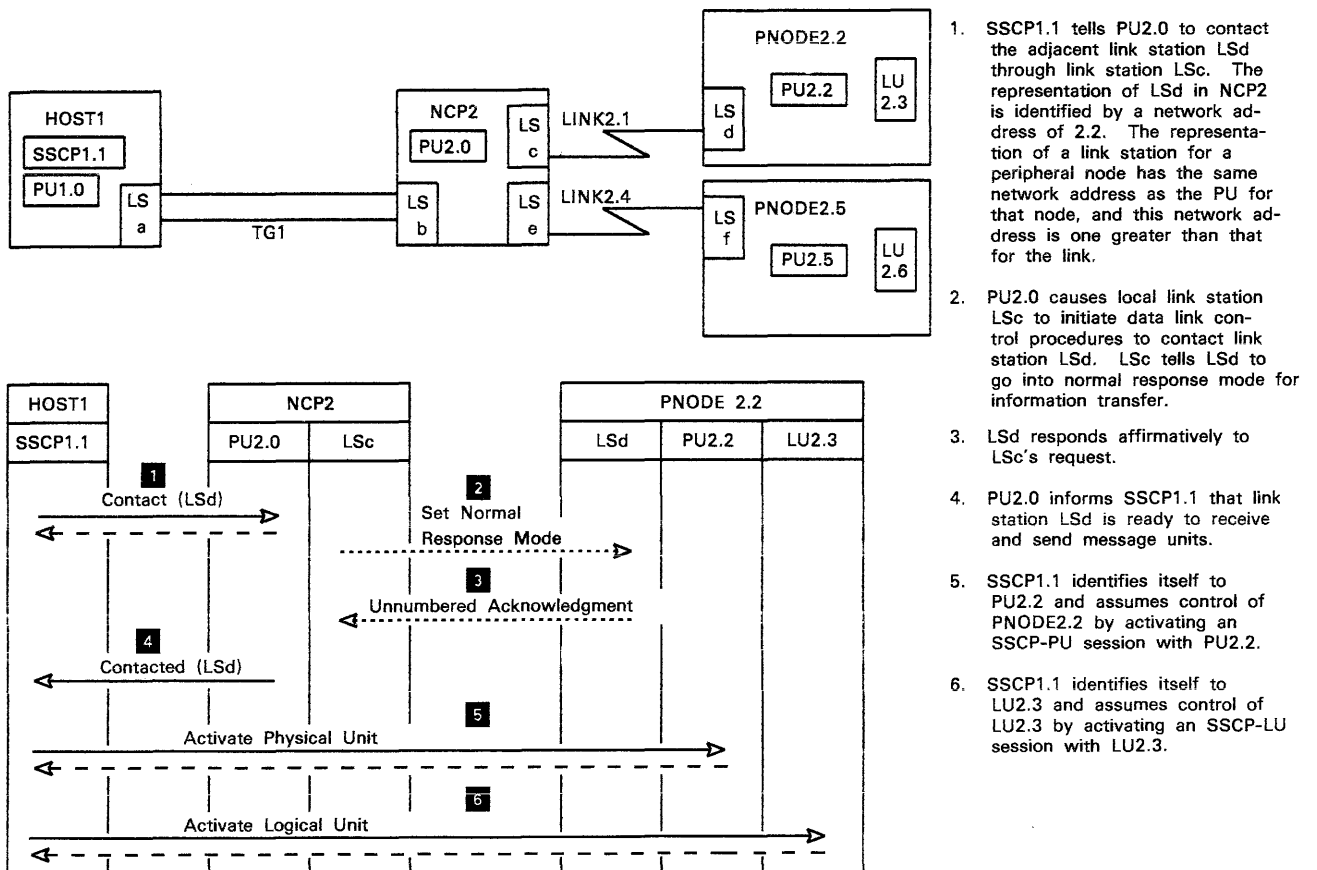
Figure 2-9. Activating Explicit and Virtual Routes between Adjacent Subarea Nodes



1. SSCP1.1 identifies itself to PU2.0 and assumes control of NCP2 by activating an SSCP-PU session with PU2.0.
2. SSCP1.1 enables the flow of FMD message units over the SSCP-PU session with PU2.0.
3. SSCP1.1 sends PU2.0 the current date and time.
4. SSCP1.1 tells PU2.0 to activate LINK2.1 and to prepare to issue data link control commands for the link.
5. SSCP1.1 tells PU2.0 to activate LINK2.4 and to prepare to issue data link control commands for the link.

(Figure 2-7 gives the meanings of the symbols and abbreviations that appear in this figure.)

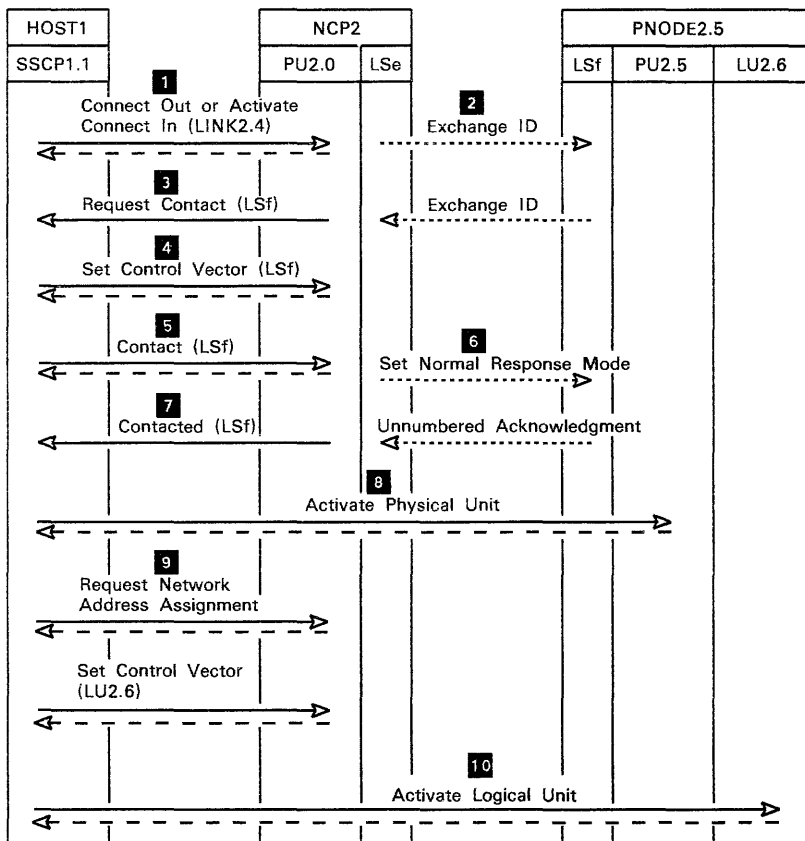
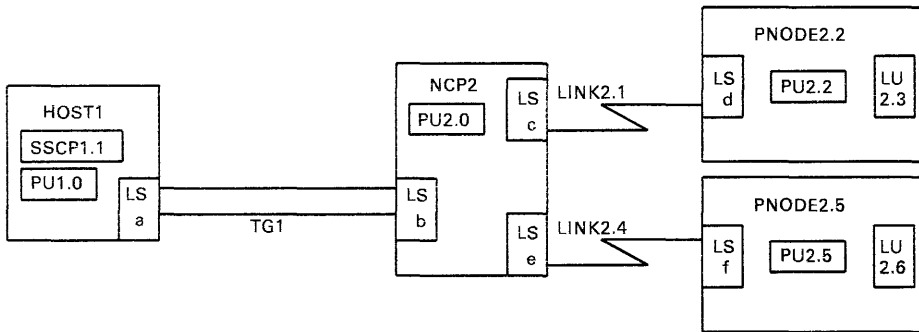
Figure 2-10. Activating a Channel-Attached Subarea Node and Attached Links



1. SSCP1.1 tells PU2.0 to contact the adjacent link station LSd through link station LSc. The representation of LSd in NCP2 is identified by a network address of 2.2. 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.
2. PU2.0 causes local link station LSc to initiate data link control procedures to contact link station LSd. LSc tells LSd to go into normal response mode for information transfer.
3. LSd responds affirmatively to LSc's request.
4. PU2.0 informs SSCP1.1 that link station LSd is ready to receive and send message units.
5. SSCP1.1 identifies itself to PU2.2 and assumes control of PNODE2.2 by activating an SSCP-PU session with PU2.2.
6. SSCP1.1 identifies itself to LU2.3 and assumes control of LU2.3 by activating an SSCP-LU session with LU2.3.

(Figure 2-7 gives the meanings of the symbols and abbreviations that appear in this figure.)

Figure 2-11. Activating a Peripheral Node Attached via a Nonswitched SDLC Link



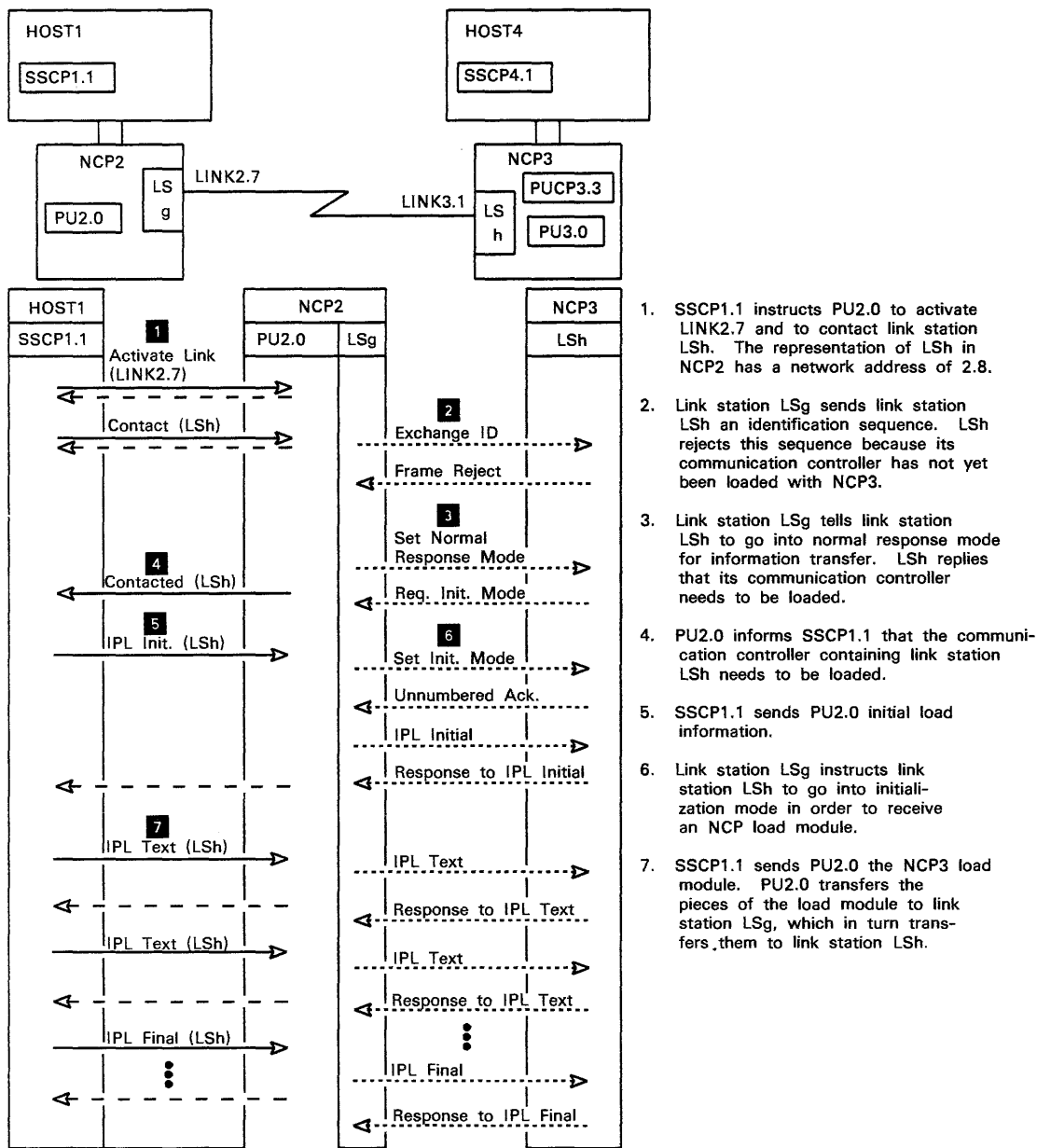
(Figure 2-7 gives the meanings of the symbols and abbreviations that appear in this figure.)

Figure 2-12 (Part 1 of 2). Activating a Peripheral Node Attached Via a Switched SDLC Link

- 
1. SSCP1.1 either tells PU2.0 to establish a connection over switched link LINK2.4 or tells PU2.0 to enable switched link LINK2.4 to accept incoming calls.
  2. Once a connection over a switched link has been established, PU2.5 informs PU2.0 about its physical unit and node including its identity, node type and characteristics, and the maximum acceptable message unit size for PNODE2.5.
  3. PU2.0 informs SSCP1.1 that a connection has been established to link station LSf. The representation of LSf in NCP2 has a network address of 2.5.
  4. SSCP1.1 sends PU2.0 information related to link station LSf, including the local SDLC address of LSf.
  5. SSCP1.1 tells PU2.0 to contact link station LSf.
  6. PU2.0 causes local link station LSe to initiate data link control procedures to contact link station LSf. LSe tells LSf to go into normal response mode for information transfer. LSf responds affirmatively to this request.
  7. PU2.0 informs SSCP1.1 that link station LSf is ready to receive and send message units.
  8. SSCP1.1 identifies itself to PU2.5 and assumes control of PNODE2.5 by activating an SSCP-PU session with PU2.5.
  9. SSCP1.1 sends PU2.0 the appropriate local addresses for LUs associated with node PNODE2.5. PU2.0 returns network addresses for the LUs to SSCP1.1. SSCP1.1 then sends PU2.0 control information about each LU associated with PNODE2.5.
  10. SSCP1.1 identifies itself to LU2.6 and assumes control of LU2.6 by activating an SSCP-LU session.

Figure 2-12 (Part 2 of 2). Activating a Peripheral Node Attached Via a Switched SDLC Link

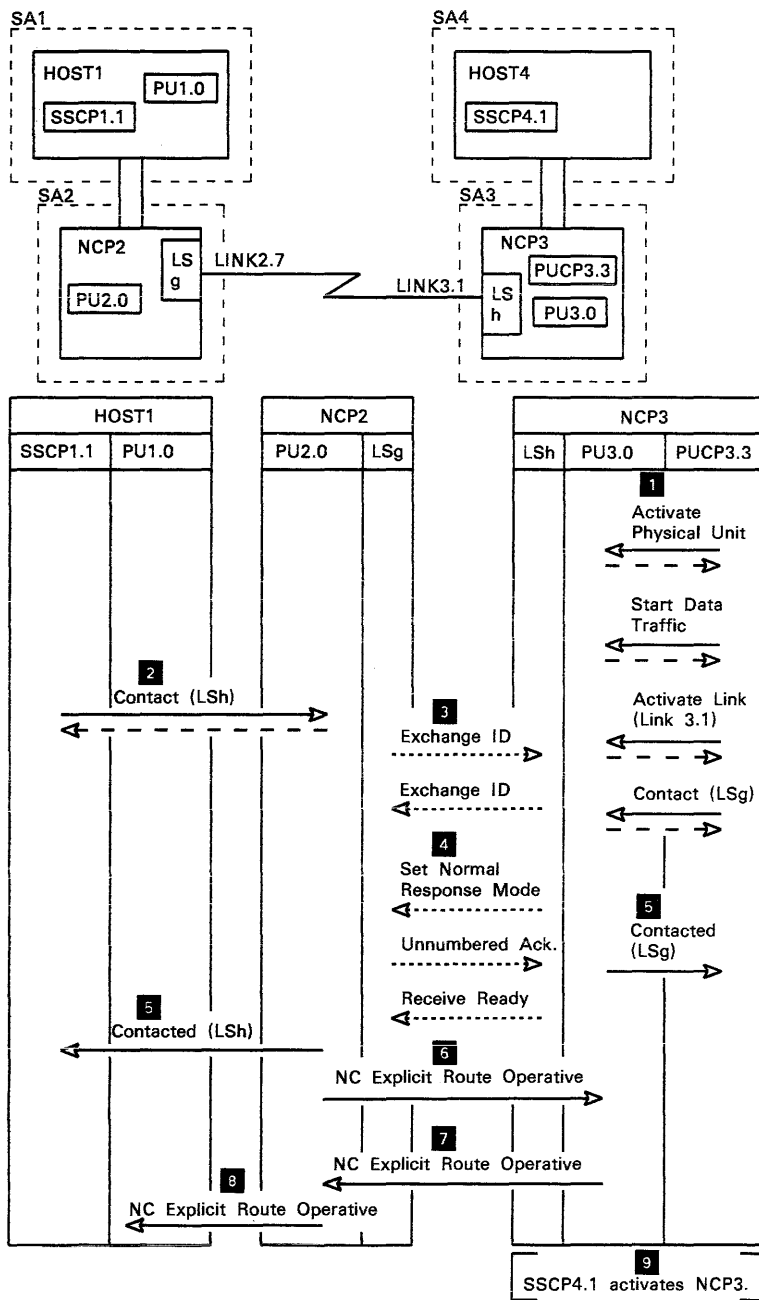
---



(Figure 2-7 gives the meanings of the symbols and abbreviations that appear in this figure.)

Figure 2-13. Loading a 3705 Communication Controller with an NCP

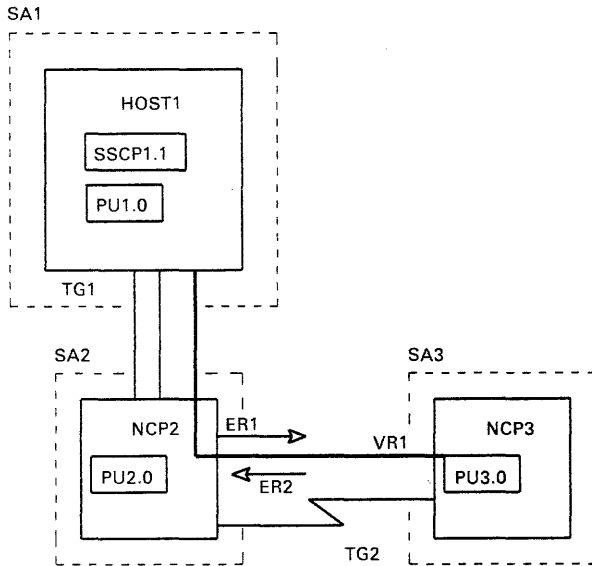




1. When NCP3 has been loaded, PUCP3.3 is given control. PUCP3.3 sends PU3.0 commands required to activate the NCP and LINK3.1, and instructs PU3.0 to contact link station LSg. The representation of LSg in NCP3 has a network address of 3.2.
2. SSCP1.1 instructs PU2.0 to contact link station LSh. The representation of LSh in NCP2 has a network address of 2.8.
3. Link stations LSg and LSh exchange identification sequences and prepare the link for data exchange.
4. LSh assumes the role of the primary link station because its PU has a higher subarea address than does the PU associated with LSg.
5. Based upon receipt of the Receive Ready data link control command, PU2.0 informs SSCP1.1 that a transmission group between NCP2 and NCP3 has been activated. PU3.0 informs PUCP3.3 of this fact.
6. PU2.0 tells PU3.0 which subareas can be reached from NCP2, and which explicit routes are used to reach these subareas. (PU2.0 may also send this information, as modified by information in PU2.0's routing tables, to PU1.0.)
7. PU3.0 tells PU2.0 which subareas can be reached from NCP3, and which explicit routes to use to reach these subareas. (PU3.0 may also send this information, as modified by information in PU3.0's routing tables, to the PU associated with HOST4, PU4.0).
8. PU2.0 propagates to PU1.0 the routing information it has received from PU3.0.
9. When HOST4 assumes control of NCP3 and its resources, PUCP3.3 relinquishes such control.

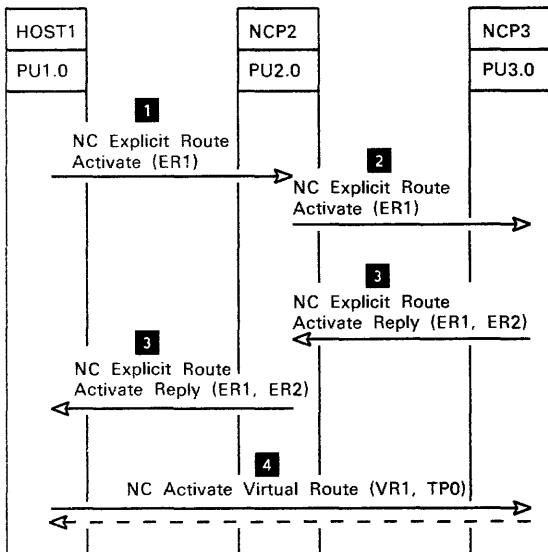
(Figure 2-7 gives the meanings of the symbols and abbreviations that appear in this figure.)

Figure 2-14. Activating an SDLC Link between Subarea Nodes



1. PU1.0 initiates the activation of an explicit route between subarea 1 and subarea 3 by sending the appropriate information to PU2.0. This route has an explicit route number of 1.
2. PU2.0 passes this request to PU3.0.
3. PU3.0 returns to PU2.0 the information that the reverse explicit route number for this explicit route is 2, and that the explicit route has a length of 2 transmission groups. (ER1 and ER2 refer in this case to the same explicit route; an explicit route is known by two explicit route numbers--one for each direction.) PU2.0 passes this information to PU1.0.
4. PU1.0 activates a virtual route between subareas 1 and 3. 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 host-to-NCP direction) and 2 (in the reverse direction).

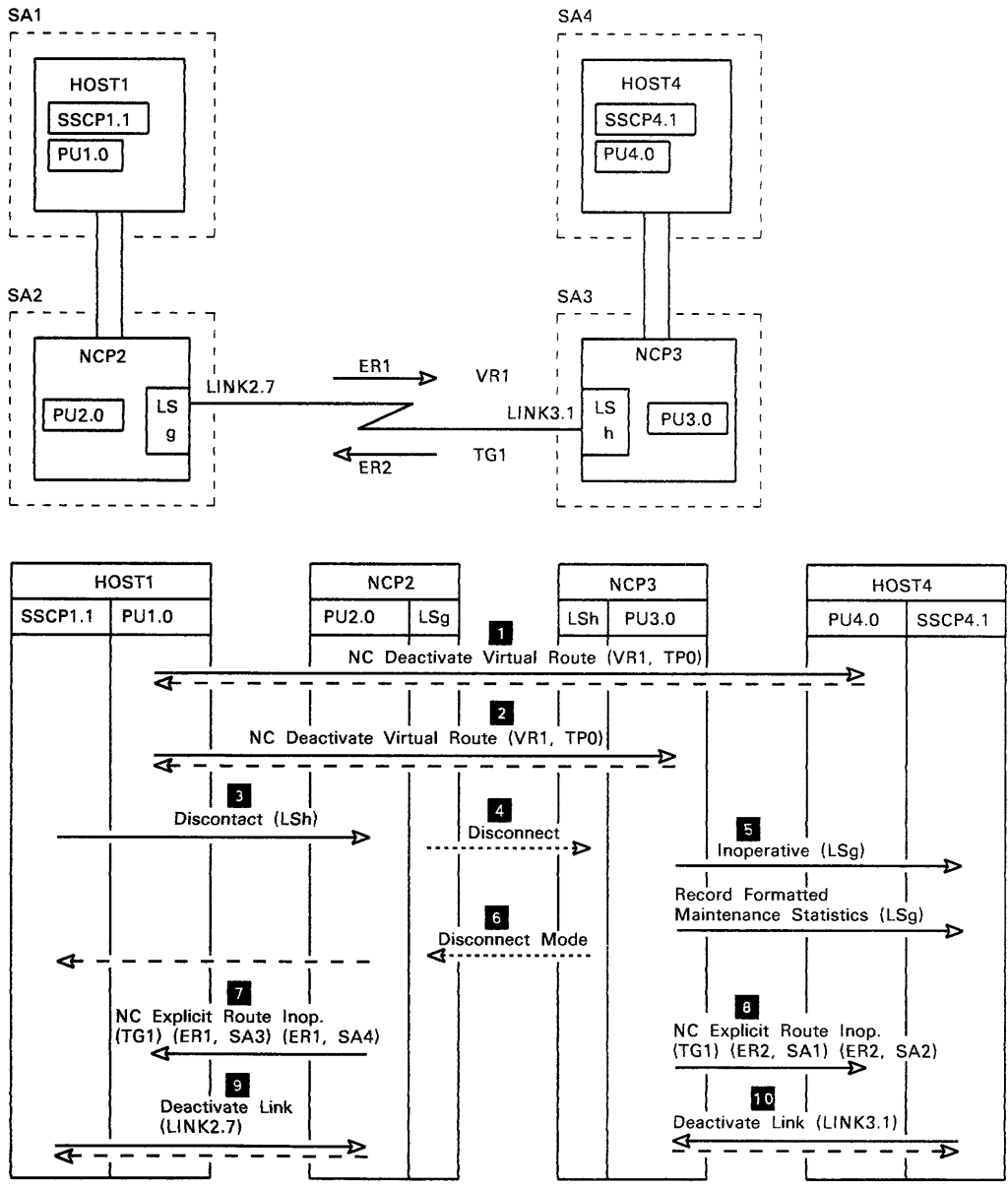
Note: PU1.0 and PU4.0 activate explicit and virtual routes between subarea 1 and subarea 4 in a manner similar to that shown here.



(Figure 2-7 gives the meanings of the symbols and abbreviations that appear in this figure.)

Figure 2-15. Activating Explicit and Virtual Routes between Nonadjacent Subarea Nodes

This page intentionally left blank.



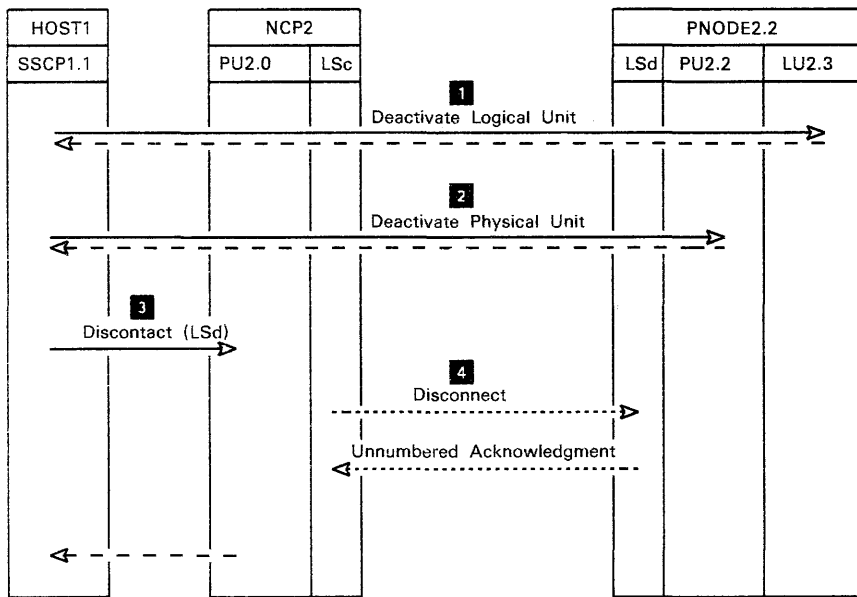
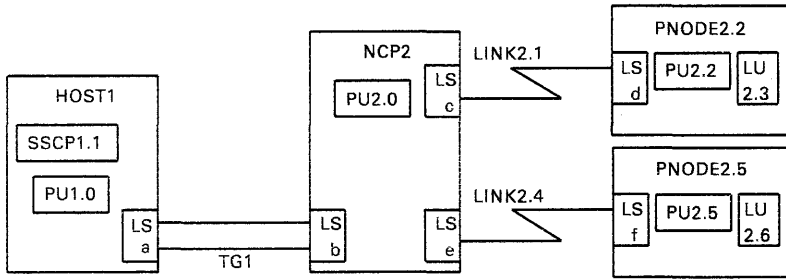
(Figure 2-7 gives the meanings of the symbols and abbreviations that appear in this figure.)

Figure 2-16 (Part 1 of 2). Deactivating Virtual Routes, Explicit Routes, and SDLC Links

- 
1. PU1.0 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. PU1.0 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. SSCP1.1 tells PU2.0 to break contact with link station LSh. The representation of LSh in NCP2 has a network address of 2.8.
  4. PU2.0 causes local link station LSg to initiate data link control procedures to break contact with link station LSh. LSg tells LSh to go into disconnect mode.
  5. PU3.0 informs SSCP4.1 that link station LSg is inoperative, and sends SSCP4.1 maintenance statistics relating to LSg. The representation of LSg in NCP3 has a network address of 3.2.
  6. LSh informs LSg that LSh has gone into disconnect mode.
  7. PU2.0 informs PU1.0 that TG1 has had a routing interruption that rendered inoperative ER1 to subarea 3 and ER1 to subarea 4.
  8. PU3.0 informs PU4.0 that TG1 has had a routing interruption that rendered inoperative ER2 to subarea 1 and ER2 to subarea 2.
  9. SSCP1.1 tells PU2.0 to deactivate NCP2's end of LINK2.7 between NCP2 and NCP3.
  10. SSCP4.1 tells PU3.0 to deactivate NCP3's end of LINK3.1 between NCP2 and NCP3.

Figure 2-16 (Part 2 of 2). Deactivating Virtual Routes, Explicit Routes, and SDLC Links

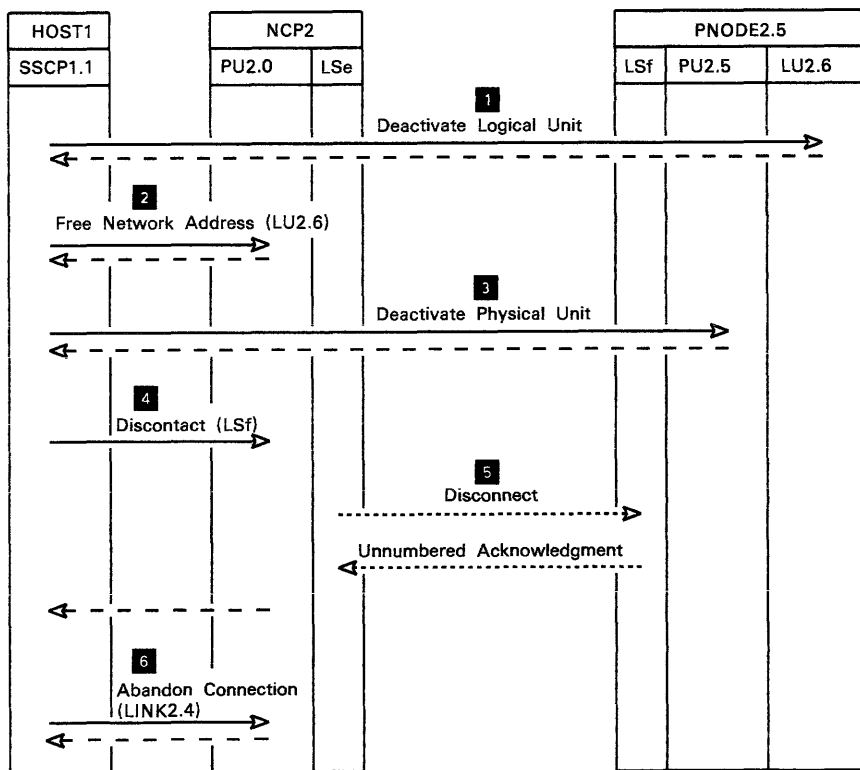
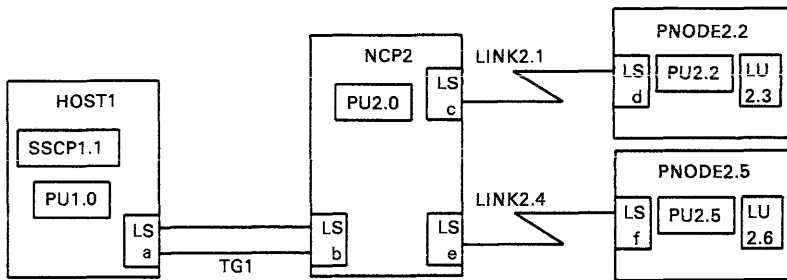
---



1. SSCP1.1 relinquishes control of LU2.3 by deactivating the SSCP-LU session between itself and LU2.3.
2. SSCP1.1 relinquishes control of PNODE2.2 by deactivating the SSCP-PU session between itself and PU2.2.
3. SSCP1.1 tells PU2.0 to break contact with link station LSc.
4. PU2.0 causes local link station LSc to initiate data link control procedures to break contact with link station LSd. LSc tells LSd to go into disconnect mode, and LSd responds affirmatively.

(Figure 2-7 gives the meanings of the symbols and abbreviations that appear in this figure.)

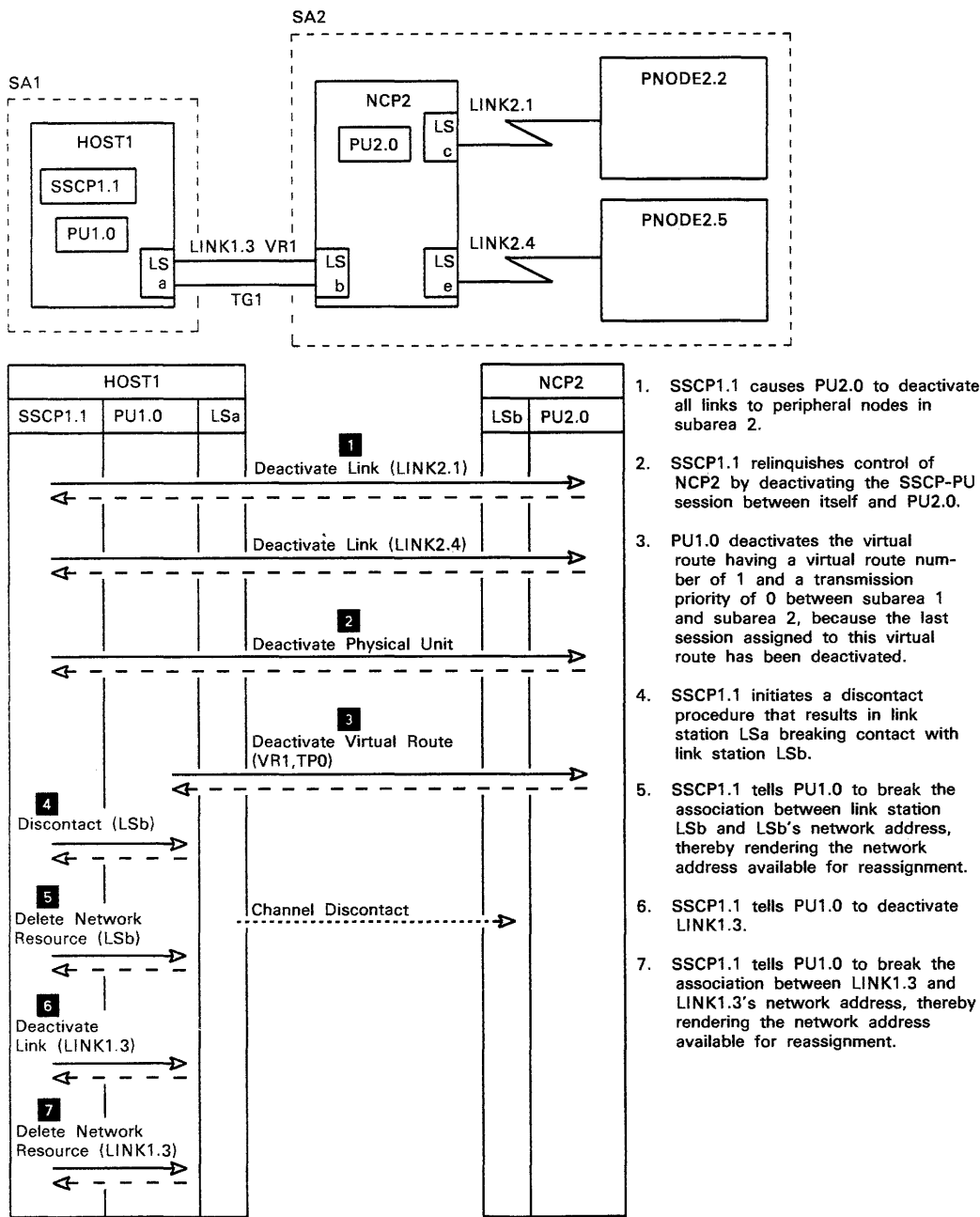
Figure 2-17. Deactivating a Peripheral Node Attached via a Nonswitched SDLC Link



1. SSCP1.1 relinquishes control of LU2.6 by deactivating the SSCP-LU session between itself and LU2.6.
2. SSCP1.1 tells PU2.0 to disassociate LU2.6's network address from LU2.6. The freed network address is returned to a pool in NCP2, from which it may be reassigned.
3. SSCP1.1 relinquishes control of PNODE2.5 by deactivating the SSCP-PU session between itself and PU2.5.
4. SSCP1.1 tells PU2.0 to break contact with link station LSf. The representation of LSf in NCP2 has a network address of 2.5.
5. PU2.0 causes local link station LSe to initiate data link control procedures to break contact with link station LSf. LSe tells LSf to go into disconnect mode, and LSf responds affirmatively.
6. SSCP1.1 tells PU2.0 to break the switched connection between NCP2 and PNODE2.5 (LINK2.4). PU2.0 does so.

(Figure 2-7 gives the meanings of the symbols and abbreviations that appear in this figure.)

Figure 2-18. Deactivating a Peripheral Node Attached via a Switched SDLC Link



(Figure 2-7 gives the meanings of the symbols and abbreviations that appear in this figure.)

Figure 2-19. Deactivating a Channel-Attached Subarea Node and Associated Resources



## CHAPTER 3. TRANSMITTING DATA FROM NODE TO NODE

This chapter presents some general Synchronous Data Link Control (SDLC) concepts, explains how the SNA data link control layer transfers data over links between adjacent SNA nodes, describes how the SNA path control components regulate the transfer, and provides some typical sequences of SDLC commands. More detailed information on Synchronous Data Link Control is available in IBM Synchronous Data Link Control General Information, GA27-3093.

The machines that contain SNA nodes are physically connected by such means as cable, twisted-pair wire, and satellite and microwave link connections. The SNA data link control layer in each node serves as the interface between the SNA node and the physical link connections to adjacent nodes in the network. Within each node, data link control elements called link stations coordinate the exchange of information across link connections without imposing on the higher layers of the SNA network the characteristics of those connections. (Examples of such characteristics are propagation delay, initialization of the connection, multiplexing of traffic involving more than one adjacent link stations on the link, and the error-management protocols that are used.)

Selecting the appropriate link for transmission of a message unit is a routing function that SNA networks perform with path control elements in each node. Path control also sequences path information units (PIUs)<sup>1</sup> during data transfer between nodes and determines the rate at which nodes exchange message units. Finally, path control transfers message units between two nodes in the network that are end points of routes that may pass through several intermediate routing nodes.

Links and nodes together form paths between network addressable units (NAUs). Chapter 4, "Routing Data from Subarea to Subarea," describes how SNA path control routes data between NAUs.

### CONCEPTS OF DATA TRANSMISSION BETWEEN NODES

It is important to be familiar with the following concepts before reading the remainder of this chapter. The Glossary at the back of this publication gives formal definitions for these concepts.

#### Links and Link Stations

An SNA network is made up of nodes (which are implemented in machines such as host processors, communication controllers, and terminals) and

---

<sup>1</sup> A PIU is a message unit consisting of a transmission header (TH) alone, or of a TH followed by a basic information unit (BIU).

links that connect the nodes. Data that is to travel from one node to an adjacent node must be transmitted over a link connecting the two nodes. Figure 3-1 on page 3-3 shows the components of an SDLC link.

In SNA networks, a link consists of a link connection, which physically connects two machines containing SNA nodes, and two or more link stations. Each link station connects the link to an SNA node. A link connection consists of an SDLC link or a data channel (such as a System/370 channel) and (in the case of an SDLC link) two modems.

A channel connects a host processor to a communication controller or other input/output (I/O) unit. A channel transmits bits in parallel; it may transmit one or more bytes in a single operation.

An SDLC link connects a communication controller to another such controller or to one or more terminals or cluster controllers. Whereas a channel transmits bits in parallel, one or more bytes at a time, an SDLC link transmits data serially by bit, one bit at a time. An SDLC link can use various physical transmission media, such as metallic conductors, or a combination of these with microwave or satellite connections.

Modems convert binary signals to make them compatible with voice-grade transmission equipment. A modem is required at each connection point on the SDLC link. (Because no signal conversion or byte assembly and disassembly is done across a channel, modems are not required for channel connections.)

Whereas the link connection consists of the hardware used to transmit data from one machine to another, the link station consists of hardware and software that allows an SNA node within a machine to communicate with a link connection. The link station controls the link and the link connection and serializes and deserializes data transmitted over the link connection.

Control of the link connection consists in physically "activating" the link connection at both ends so that data can be transmitted over it. Link control consists in executing protocols that transmit data over the link. Serialization and deserialization consist in breaking up bytes into bits at the sending end of the link connection and assembling bits into bytes at the receiving end.

The link stations for a channel are implemented in the channel programs at the host and in channel-adapter I/O supervisors within the network control program (NCP). Among other functions, these programs group data to ensure efficient use of the channel, transfer data over the channel, and ensure that each path information unit (PIU) sent to the host begins in a new host I/O buffer.

The SDLC protocols that control data transmission over the link connection are somewhat more complex than the channel protocols. One factor contributing to this complexity is the greater vulnerability to transmission errors of SDLC links as compared to channels. Another factor is the ability of SDLC to operate over multipoint lines, which

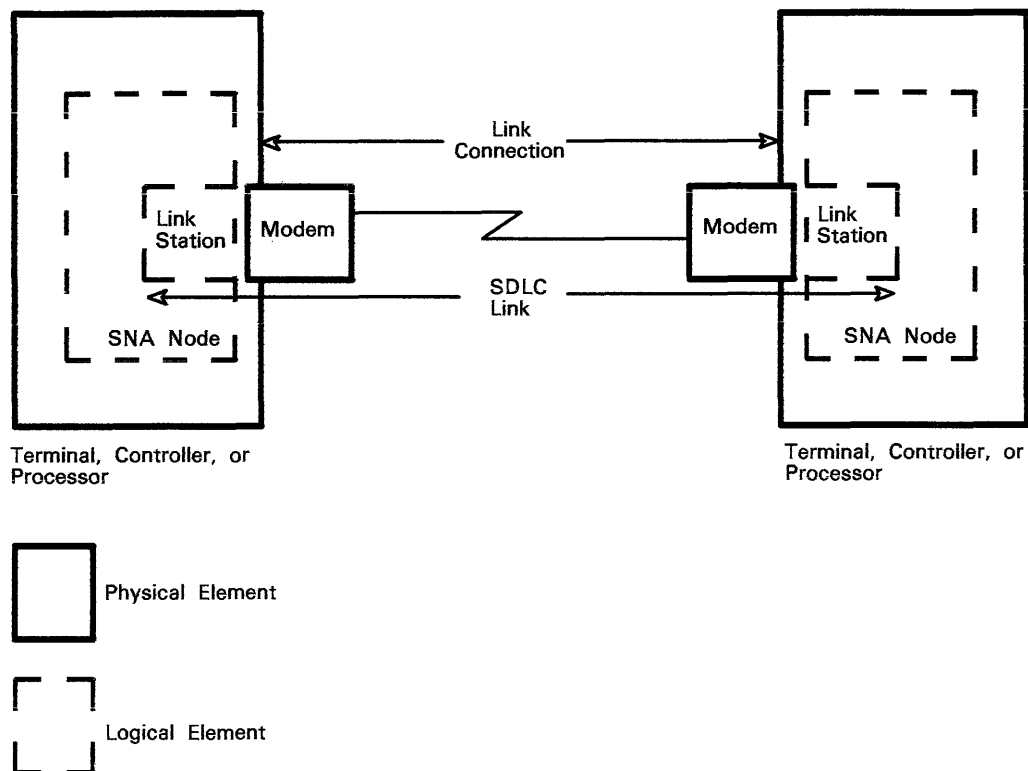


Figure 3-1. Components of an SDLC Link

require simultaneous management of many terminals; in contrast, channels are always point-to-point.

Each link has a link station at each point where a node connects to the link.

There are two types of link stations: primary link stations and secondary link stations. A primary link station controls the link; it issues link-level commands that control the secondary link stations. Secondary link stations receive these commands and respond to them. The primary link station notifies each secondary link station on the link when the secondary may transmit data and when it may expect to receive data. A link can have only one primary link station at a time. All communication on the link is between the primary link station and the secondary link stations; secondary link stations may not communicate with each other.

## Types of Links In SNA Networks

SNA networks can have two types of links: data channels and SDLC links. In addition, SNA networks support X.25 virtual-call and permanent virtual circuits as SDLC links via the X.25 NCP Packet Switching Interface program product (program number 5668-981).

### SDLC Links

SDLC is a serial-by-bit discipline for transmitting data from one point to another. It checks for and corrects transmission errors to help ensure that transmissions are correctly received.

SDLC is used where direct-wired connections are impractical. It may be used on telephone lines, microwave links, satellite channels, or any other voice-quality means of communicating.

### Data Channels

A data channel (referred to hereafter as a channel) is a device that connects a processor and main storage with input/output (I/O) control units. A channel transmits one or more bytes of data at a time between the main storage and the I/O devices attached to the channels.

Channels are used to free the host processor from local I/O processing. They transmit data over cables that connect input/output devices directly to a host processor. In SNA networks, channels are used to transmit data over cables between host processors and communication controllers, cluster controllers, and terminals.

Data link control for data channels is described in the ACF/TCAM and ACF/VTAM publications.

### The X.25 Interface Between SNA Nodes and Packet-Switched Data Networks

Certain IBM products provide an interface between the SNA nodes they contain and packet-switched data networks. This interface conforms to recommendation X.25 of the International Telegraph and Telephone Consultative Committee (CCITT).

The X.25 interface that such products support causes X.25 permanent and switched virtual circuits to appear to SNA nodes as SDLC links. Two implementations of this interface are the X.25 NCP Packet Switching Interface, program number 5668-981 (a program product that operates in the 3705 communication controller) and the IBM 5973-LO2 Network Interface Adapter (a device used to connect SNA peripheral nodes to a packet-switched data network).

## Basic Link Units

Data flows over a link in the form of basic link units<sup>2</sup> (BLUs). BLUs sent across a channel consist of one or more PIUs, blocked so as to increase channel usage and accommodate differences in buffer size between the host computer and the receiving communication controller or terminal. BLUs sent over an SDLC link contain link control information at their beginning and end in fields called the link header and link trailer, respectively. A BLU sent over an SDLC link is also called a frame.

The link header includes an address field that identifies the secondary link station that is to receive or send the frame over the link, and a control field that indicates whether the frame contains data or control information.

The link trailer contains a frame check sequence field that the receiving link station uses to check the frame for errors that may have occurred during transmission over the link.

The sending link station inserts a special flag sequence of bits at the beginning and end of each frame. This sequence delimits the frame to the receiving link station.

## SDLC Link Configurations

An SDLC link and its associated resources may be in one of three configurations: point-to-point, multipoint, or loop.

In a point-to-point configuration, a single link connection joins two link stations. This is the simplest data link configuration possible. (See Figure 3-2 on page 3-6.)

In a multipoint configuration, a single link connection joins more than two link stations. (See Figure 3-3 on page 3-7.)

A loop configuration is a special form of multipoint configuration in which the stations are connected serially by a link connection called a loop. Transmissions pass from one station to another and ultimately complete the loop. (See Figure 3-4 on page 3-8.)

In a loop configuration, all transmissions travel in the same direction on the link, whereas in a non-loop multipoint configuration, transmissions usually travel in both directions on the link.

A point-to-point half-duplex configuration may be switched or nonswitched. A switched configuration is one in which the physical connection is temporary, as in a telephone call. In a nonswitched configuration, the physical connection is permanent.

---

<sup>2</sup> A basic link unit is the unit of data and control information transmitted over a link by data link control.

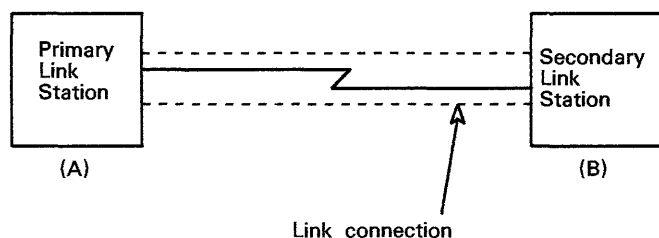


Figure 3-2. Point-to-Point SDLC Link Configuration

## PATH CONTROL COMPONENTS INVOLVED IN DATA TRANSMISSION BETWEEN NODES

Within an SNA network, data is transferred from one subarea node to another, or between a subarea node and a peripheral node. A different set of path control components is used in each case.

### Transmitting Data between Adjacent Subarea Nodes

Subarea nodes have more path control capabilities than do peripheral nodes. Subarea nodes can perform intermediate routing—passing message units received from one node to another node—and control the flow of traffic within the subarea-routing part of the network in response to changing traffic loads.

Subarea-routing path control routes message units within subarea nodes and between one subarea node and another. The parts of subarea-routing path control are:

- Transmission-group control (TGC)
- Explicit-route control (ERC)
- Virtual-route control (VRC)

Explicit-route control and virtual-route control primarily concern routing of message units between end subareas and are therefore described in Chapter 4, "Routing Data from Subarea to Subarea."

SNA permits multiple SDLC links to be defined between NCP nodes. Multiple SDLC links that are operating concurrently between such nodes are referred to as parallel links. Parallel links can be grouped into

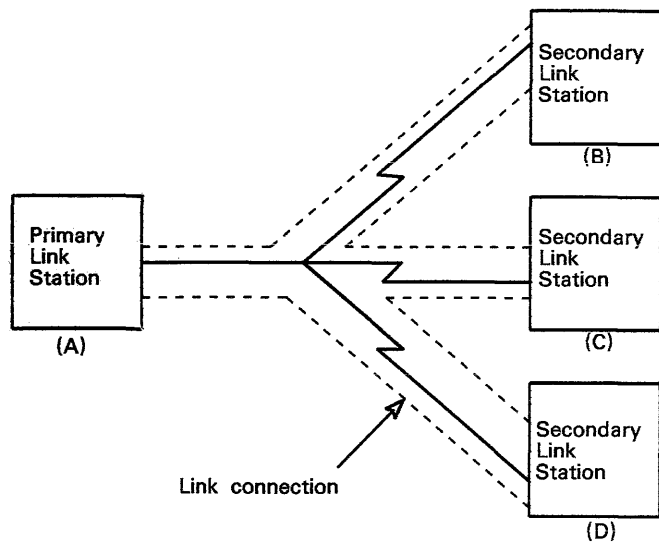


Figure 3-3. Multipoint SDLC Link Configuration

transmission groups; each transmission group represents a logical connection between nodes. The network designer places links with similar characteristics into the same transmission group. Message traffic scheduled for transmission over a multiple-link transmission group is divided among the links in order to use the composite capacity of the links most effectively.

Because each SDLC link in a transmission group is controlled by its own protocol, individual links can be added or deleted from the group as needed, error statistics can be gathered separately for each link, and scheduling of message traffic among the links can take account of degraded performance of particular links in the group.

SNA session protocols require that all requests and responses arrive at their destination in the same sequence as they were transmitted from their origin. However, requests and responses may flow over a transmission group in a sequence different from that in which they were transmitted by their origin. This can happen because different links in the group may operate at different rates, requests transmitted over the various links may vary in length, and delays may result from retransmission over some links but not others. Transmission-group control at the receiving end of a transmission group therefore resequences requests and responses as necessary to restore them to their original order.

The transmission-group scheduling and error-recovery protocols assume that all links in a transmission group operate at nearly the same speed and experience nearly the same transmission delay. Transmission groups

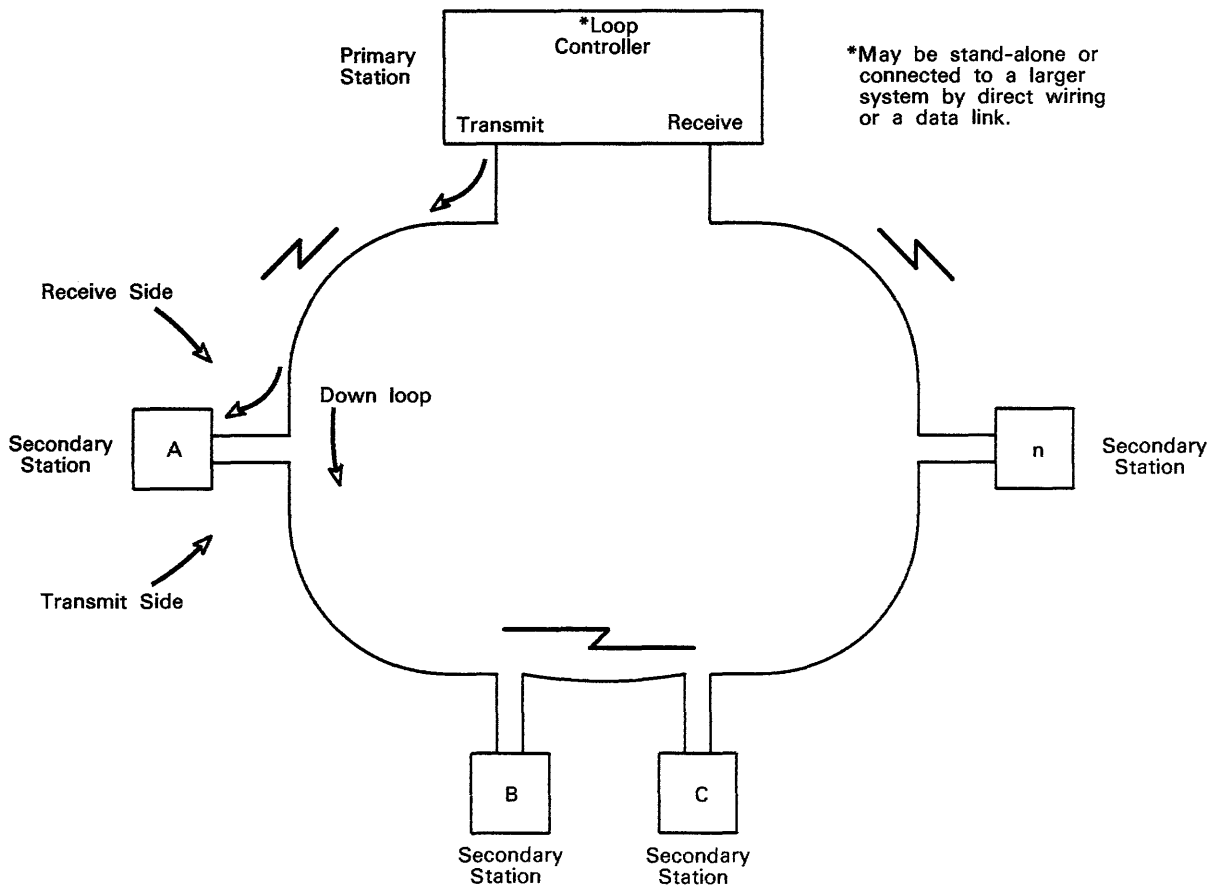


Figure 3-4. Loop Configuration

for which this is not true will create relatively more out-of-sequence requests. This will increase the storage needed to resequence out-of-order requests and increase the average delay over the transmission group toward the delay imposed by the slowest link in the group.

Transmission-group control:



- Blocks<sup>3</sup> and deblocks basic transmission units (BTUs) over channels
- Retransmits BTUs when transmission errors occur
- Converts excessively long PIUs into exception requests<sup>4</sup> and indicates the appropriate sense code
- Places all PIUs to be transmitted over a transmission group into a transmission priority list for that transmission group according to their priority and transmits them according to priority
- Sequences PIUs at sending nodes and resequences them at receiving nodes
- Ensures that PIUs sent on more than one link in a transmission group are not duplicated at the receiving end of the transmission group
- Indicates when a congested condition exists
- Checks the validity of each received BTU before accepting it

### Transmitting Data between a Subarea Node and a Peripheral Node

A peripheral node requires the services of the boundary-function (BF) component in the subarea node to which it is connected (called the adjacent subarea node) in order to communicate with NAUs in that or some other subarea node. The boundary-function component consists of a boundary-function path control component and a boundary-function node for each peripheral node attached to the subarea node.

A boundary-function node consists of a BF physical unit, which represents the physical unit in the peripheral node, and a BF logical unit for each logical unit in the peripheral node.

Each boundary-function node:

- Decides whether session-activation parameters it receives are acceptable to the boundary function
- Determines whether the boundary function has sufficient resources to support sessions with the peripheral PU and LUs
- Manages session-level sequence numbering for low-function peripheral nodes
- Controls session-level pacing for the attached peripheral node

Boundary-function path control:

- Translates network addresses, which subarea nodes use, to local addresses, which peripheral nodes use, and vice versa

---

<sup>3</sup> Blocking is an optional function of path control that combines multiple path information units (PIUs) into a single basic transmission unit (BTU).

<sup>4</sup> An exception request is a message unit that replaces another message unit in which an error has been detected. The exception request contains a 4-byte sense field that identifies the error in the original message unit and, except for some path errors, is sent to the destination of the original message unit; if possible, the sense data is returned in a negative response to the originator of the replaced message unit.

- Transforms transmission headers<sup>5</sup> (THs) from the format that subarea nodes use, to a shorter format that peripheral nodes use, and vice versa
- Routes data over peripheral links<sup>6</sup> to adjacent link stations in attached peripheral nodes
- Segments message units destined for peripheral nodes into message unit segments and reassembles segments coming from peripheral nodes into message units

The boundary-function path control component in a subarea node cooperates with the path control component in an adjacent peripheral node to transfer message units over the link between the two nodes.

## EXAMPLE OF DATA TRANSMISSION BETWEEN ADJACENT LINK STATIONS

This section describes for a sample network how adjacent link stations cooperate to transmit a message over several links on a path from one end user to another. For information on how the path control components of nodes on a path determine which links to use in routing a message from one end to another see Chapter 4, "Routing Data from Subarea to Subarea."

The sample network in Figure 3-5 on page 3-11 consists of a single host node (HOST1) to which an NCP node (NCP2) and a peripheral node (TERM4) are attached. NCP2 and TERM4 are both channel attached to HOST1. SDLC links connect NCP2 to another NCP node (NCP3) and a peripheral node (TERM5). Three links connect NCP2 to NCP3.

An SDLC link connects NCP3 to three peripheral nodes: TERM6, TERM7, and TERM8.

Whereas the SDLC links that connect NCP2 to TERM5 and NCP3 are point-to-point, the link that connects NCP3 to TERM6, TERM7, and TERM8 is a multipoint link shared by the three peripheral nodes.

The set of link stations within a node constitutes that node's data link control layer. For example, link stations LSb, LSe, LSg, LSi, and LSk together constitute NCP2's data link control layer.

Consider a message sent from an LU in HOST1 to an LU in TERM7. Path control in HOST1 determines, from the destination network address in the TH of each PIU in the message, that it should route each PIU over the channel between HOST1 and NCP2. Path control in HOST1 therefore passes each PIU to link station LSa.

---

<sup>5</sup> A transmission header is 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.

<sup>6</sup> A peripheral link connects a peripheral node to a subarea node.

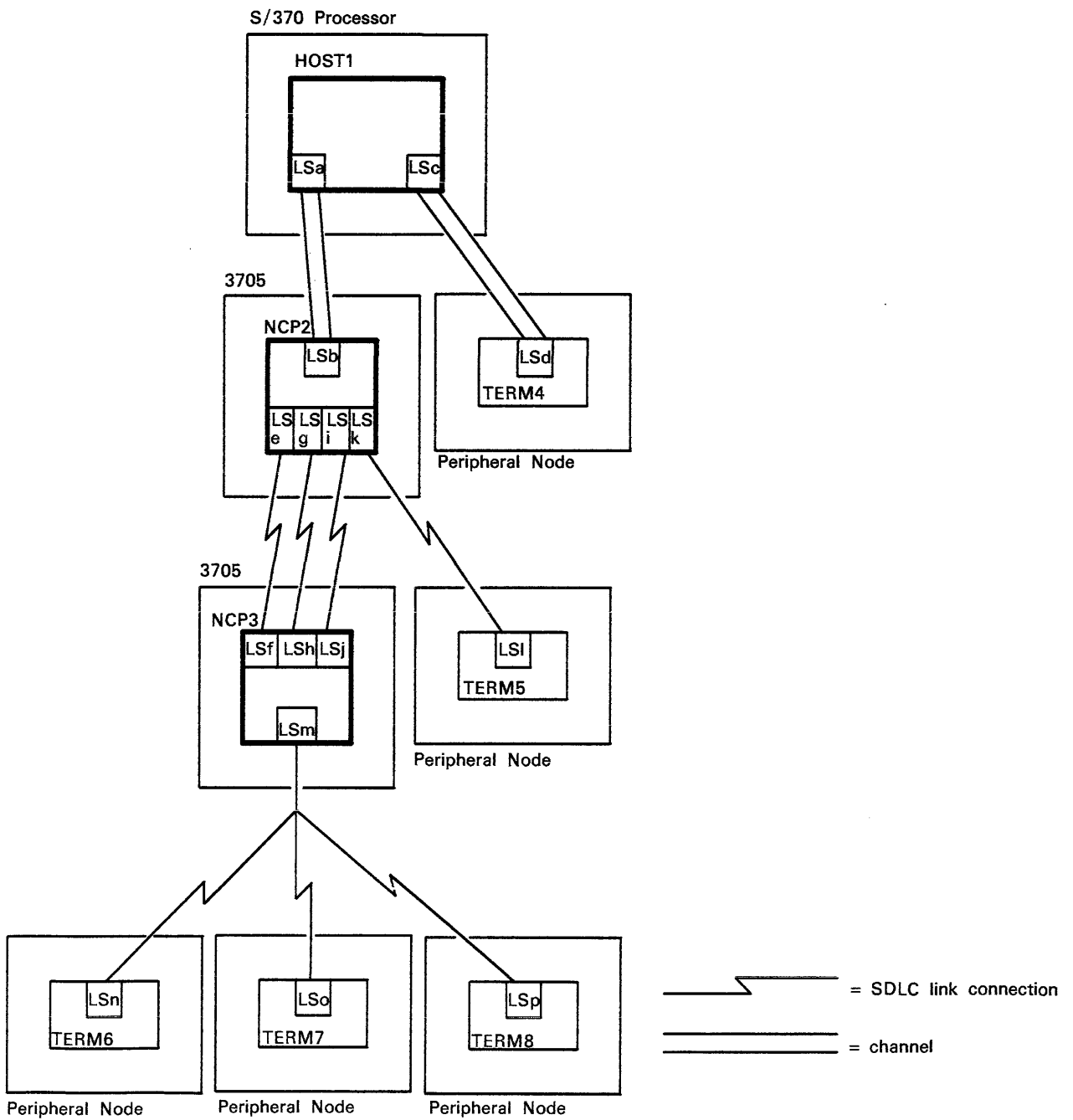


Figure 3-5. Nodes, Links, and Link Stations

Link station LSa creates a basic link unit (BLU) from PIUs that HOST1 is sending to NCP2 and executes a channel program to transfer the BLU over the channel to NCP2.

From a routing table and information in the TH of each PIU it receives, NCP2 knows that it should transmit those PIUs to NCP3 over a specific transmission group between NCP2 and NCP3. If all three links between NCP2 and NCP3 are in this transmission group, path control in NCP2 may assign each PIU in the message to any of the link stations for those links: that is, LSe, LSg, and LSi.

Because different PIUs in the same message may be transmitted over different links in the transmission group, path control in NCP2 assigns each PIU a sequence number and puts the number in the TH. These sequence numbers allow path control in NCP3 to resequence any out-of-order PIUs in the message before it routes the message to the next node on the path. Duplicate PIUs are discarded. Path control assigns each PIU to a link station for transmission to NCP3.

Whenever path control in NCP2 assigns a PIU to link station LSg, this link station builds a link header and trailer to transform the PIU into an SDLC frame. If it is the primary station on the link, link station LSg places the address of link station LSh in the link header and sends the frame over the link that joins them.

Upon receiving the frame, link station LSh checks its length, checks the frame check sequence bits in the link trailer to verify that no transmission errors have occurred, deletes the link header and link trailer, and passes the resulting PIU to path control in NCP3. Path control resequences PIUs into their original order, then assigns the PIUs to LSm, the link station associated with terminal TERM7.

Link station LSm creates a frame for each PIU and places the address of link station LSo in the header of each frame destined for TERM7.

Adjacent link stations are those link stations that can communicate with each other over a link connection that joins them. In Figure 3-5, link stations LSc and LSd are adjacent link stations, as are LSg and LSh, and LSm and LSo. Link stations LSg and LSj, in contrast, are not adjacent because no link connection joins them. Link stations LSo and LSp, although each is adjacent to LSm, are not adjacent to each other because both are secondary link stations and therefore cannot communicate.

## BENEFITS OF THE SNA NODE-TO-NODE TRANSMISSION SCHEME

By isolating link considerations to the SNA data link control layer, SNA shields upper layers of the network from differences among transmission media, such as the differences between a data channel and an SDLC link, or those between wire connections and microwave connections.

Allowing multiple links (called parallel links) between subarea nodes permits increased node availability and higher traffic rates between nodes.

By allowing parallel links between subarea nodes to be divided into transmission groups, SNA lets network designers place links with similar characteristics (for example, satellite or terrestrial, high or low speed, good or mediocre quality) into the same group. Traffic is distributed among the links in a transmission group in order to take advantage of the composite transmission capacity of the links.

The SDLC protocol accommodates a wide range of data transfer rates and it has better error checking abilities than the older Binary Synchronous Communication (BSC) protocol. SDLC error retry parameters can be dynamically adjusted; links can therefore be kept active despite intermittent bursts of transmission errors.

## **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**

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

#### ACF/TCAM Definition

To control an SDLC link, (1) ACF/TCAM in a host node must have a **TERMINAL** macro that defines the link, and (2) the host node must control the NCP node to which the link is attached.

The programmer defines a link for ACF/TCAM in the message control program (MCP) of the host node that controls the link. To control a link, a host node must also control the NCP node attached to the link.

To define a link, a programmer codes a **TERMINAL** macro that specifies **TERM=LINK**.

To define link stations for ACF/TCAM, a programmer can:

- Specify in the **LKSTA** operand of the **TERMINAL** macro the names of any link stations that may be used to load or dump the NCP that the **TERMINAL** macro defines; or
- Specify the **LKSTA** operand on either the "IPL a 3705 NCP" or the "Dump 3705 NCP Storage" basic operator commands.

To define a transmission group for ACF/TCAM, a programmer specifies its TG number in the NCP PU macro. For each subarea link in the transmission group, the programmer must specify the same TG number for both ends of the link.

#### 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 TG 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.

### TYPICAL SDLC SEQUENCES

Some examples of typical exchanges of SDLC commands and responses by link stations appear in Figure 3-7 through Figure 3-19. (Figure 3-6 on page 3-15 gives the meaning of each of the abbreviations appearing in these SDLC sequences. The SDLC commands, modes, and the poll/final bit that appear in this figure and in the SDLC sequence diagrams are explained in IBM Synchronous Data Link Control General Information, GA27-3093.) These sequences are not limited to the configurations shown, nor should they be viewed as the only possible sequences of events. Rather, they are examples showing the logic and protocols of SDLC.




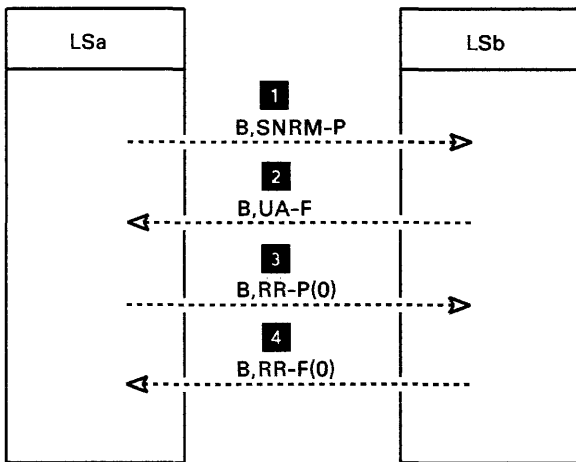
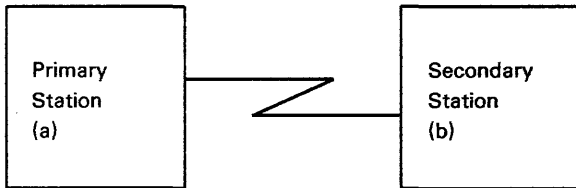
	'Not' symbol	I	Information frame
	SDLC Command or Response	LS	Link Station
	SDLC Link	P	Poll (setting of Poll/Final bit)
B	Address of link station B	REJ	Reject command
C	Address of link station C	RIM	Request Initialization Mode command
CRC	Cyclic Redundancy Check	RNR	Receive Not Ready command
DISC	Disconnect command	RR	Receive Ready command
F	Final (setting of Poll/Final bit)	SIM	Set Initialization Mode command
FRMR	Frame Reject command	SNRM	Set Normal Response Mode command
		UA	Unnumbered Acknowledgment command
		X	Broadcast address (all stations addressed)
		XID	Exchange identification command
		XXX	Undefined control field (invalid SDLC command)

Figure 3-6. Symbols and Abbreviations Appearing in Sequence Diagrams of Chapter 3

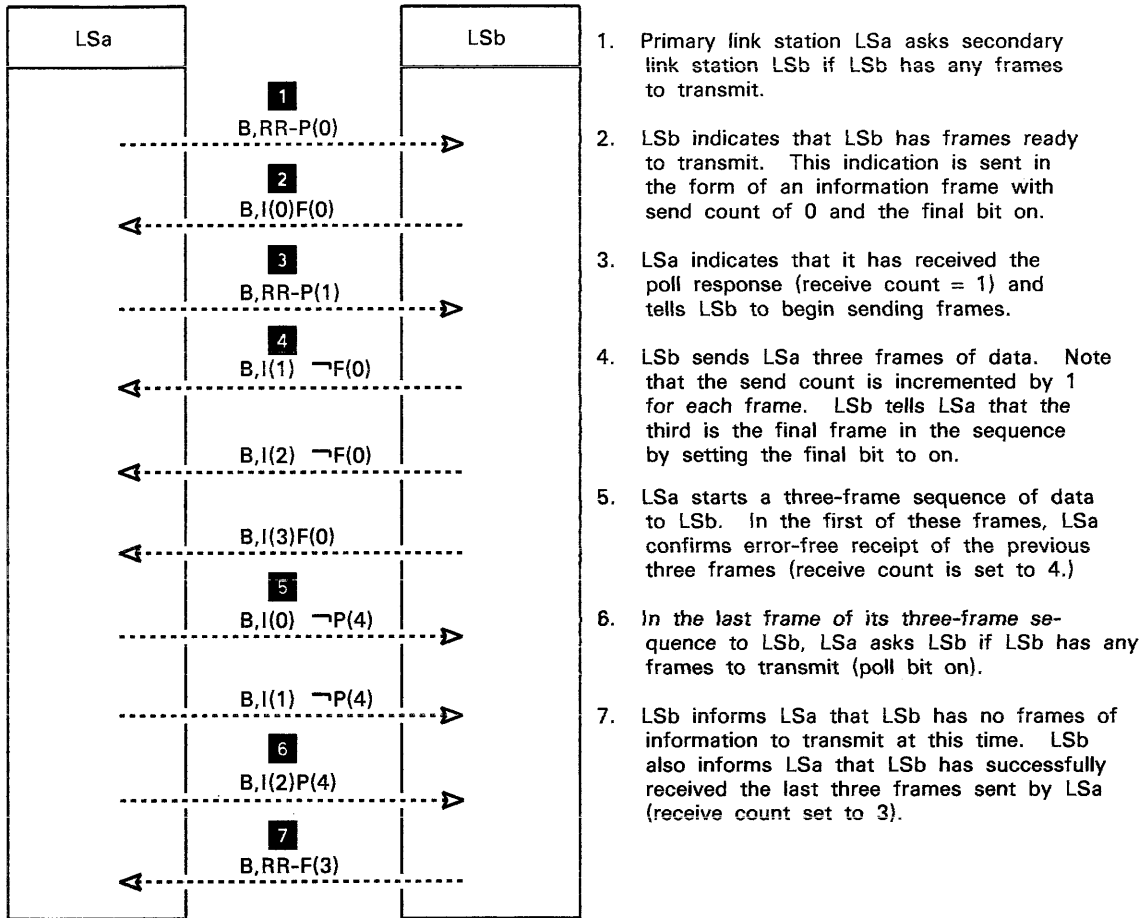


1. Primary link station LSa places secondary link station LSb in normal response mode for data transfer.
2. LSb tells LSa that LSb is now capable of sending or receiving data.
3. LSa asks LSb if LSb has any frames containing message units to transmit.
4. LSb informs LSa that LSb has no frames to transfer at this time.

(Figure 3-6 gives the meanings of the symbols and abbreviations that appear in this figure.)

Figure 3-7. Negative Response to a Poll

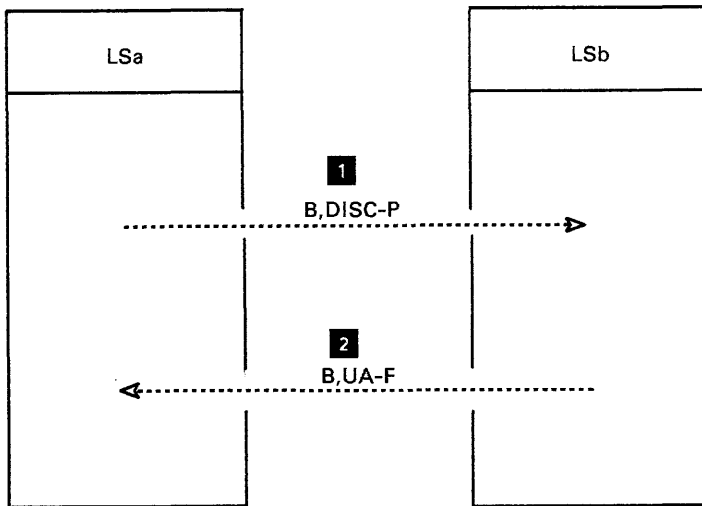




1. Primary link station LSa asks secondary link station LSb if LSb has any frames to transmit.
2. LSb indicates that LSb has frames ready to transmit. This indication is sent in the form of an information frame with send count of 0 and the final bit on.
3. LSa indicates that it has received the poll response (receive count = 1) and tells LSb to begin sending frames.
4. LSb sends LSa three frames of data. Note that the send count is incremented by 1 for each frame. LSb tells LSa that the third is the final frame in the sequence by setting the final bit to on.
5. LSa starts a three-frame sequence of data to LSb. In the first of these frames, LSa confirms error-free receipt of the previous three frames (receive count is set to 4.)
6. In the last frame of its three-frame sequence to LSb, LSa asks LSb if LSb has any frames to transmit (poll bit on).
7. LSb informs LSa that LSb has no frames of information to transmit at this time. LSb also informs LSa that LSb has successfully received the last three frames sent by LSa (receive count set to 3).

(Figure 3-6 gives the meanings of the symbols and abbreviations that appear in this figure.)

Figure 3-8. Positive Response to a Poll with Transfer of Data from Secondary Station to Primary Station

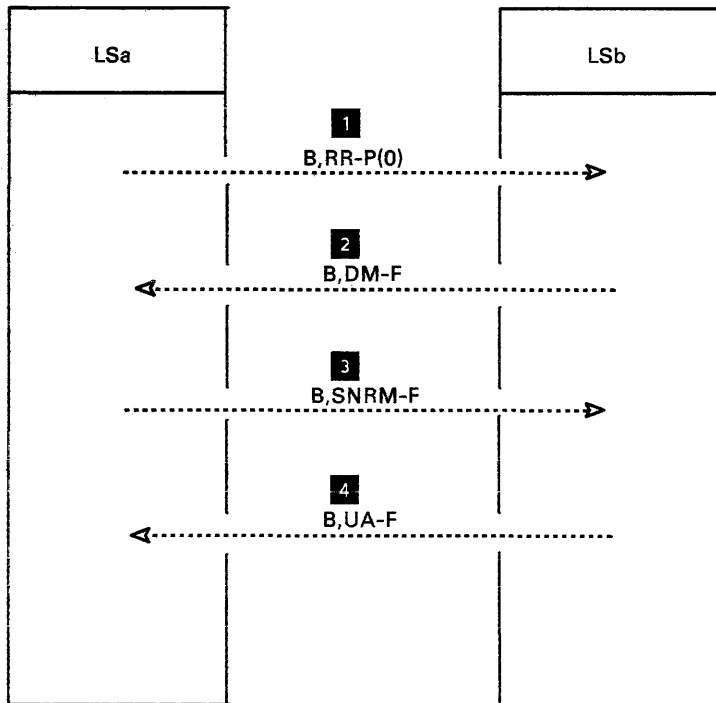


1. Primary link station LSa commands secondary link station LSb to disconnect.
2. LSb confirms the command and disconnects itself.

(Figure 3-6 gives the meanings of the symbols and abbreviations that appear in this figure.)

Figure 3-9. Disconnecting a Secondary Link Station

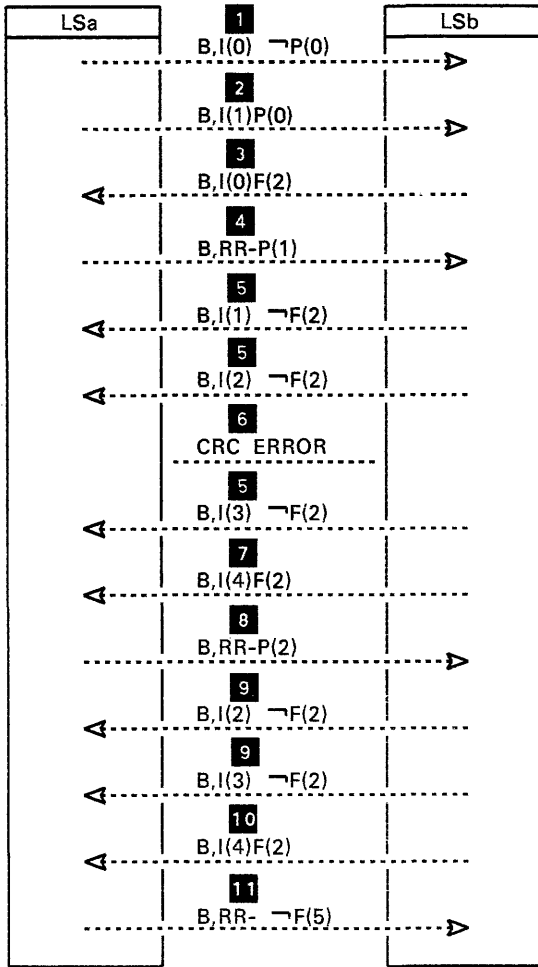
---



1. Primary link station LSa polls secondary link station LSb to see if LSb has any frames to transmit.
2. LSb reports that its status is Disconnected Mode and that it therefore cannot transmit frames.
3. LSa sets LSb's mode to Normal Response Mode. LSb may now participate in transmission of frames.
4. LSb responds that its mode is set and that it is prepared to transmit and receive frames.

(Figure 3-6 gives the meanings of the symbols and abbreviations that appear in this figure.)

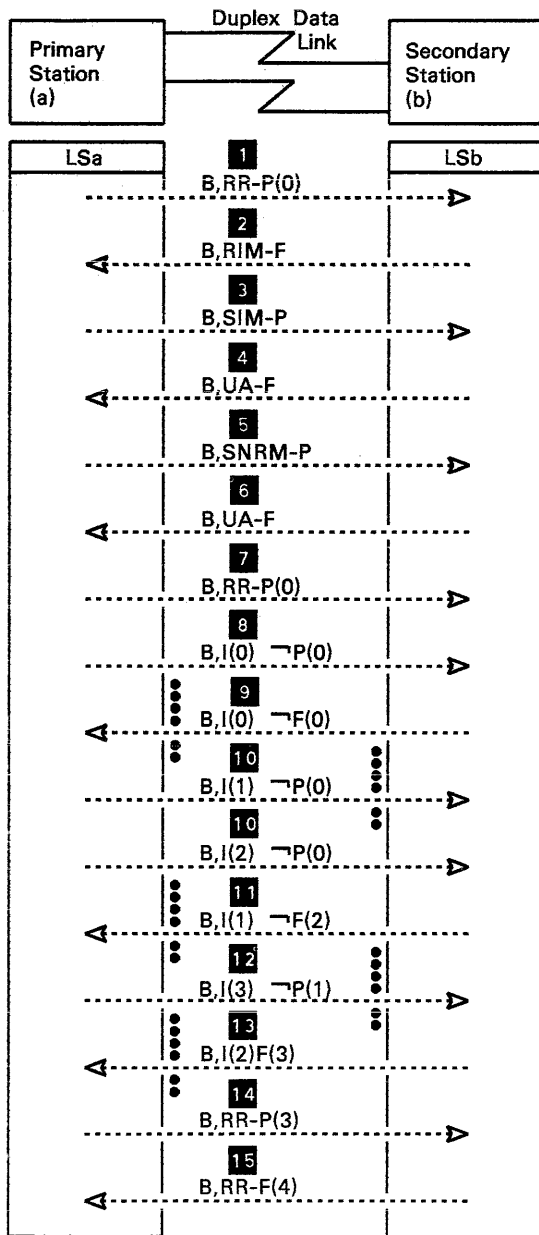
Figure 3-10. A Secondary Link Station Requests Connection and Preparation to Receive Commands



1. Link station LSa sends numbered information frame 0.
2. LSa sends numbered information frame 1 and polls link station LSb for confirmation.
3. LSb responds with its own numbered information 1 and LSa's frames 0-1 and indicates to LSa that this is LSb's final frame.
4. LSa confirms LSb's frame 1 and indicates LSa is ready to receive more frames.
5. LSb sends numbered information frames 1-3.
6. LSa receives frame 1 without error, but discovers a CRC error in frame 2. LSa therefore discards frames from 2 on, including frame 2.
7. LSb sends numbered information frame 4 and indicates to LSa that this is LSb's final frame.
8. LSa confirms LSb's frame 1. The other frames are not confirmed because they were discarded in step 6. LSa polls LSb for more frames.
9. LSb re-sends frames 2 and 3.
10. LSb re-sends frame 4 and indicates to LSa that this is LSb's final frame.
11. LSa confirms LSb's frames 2-4.

(Figure 3-6 gives the meanings of the symbols and abbreviations that appear in this figure.)

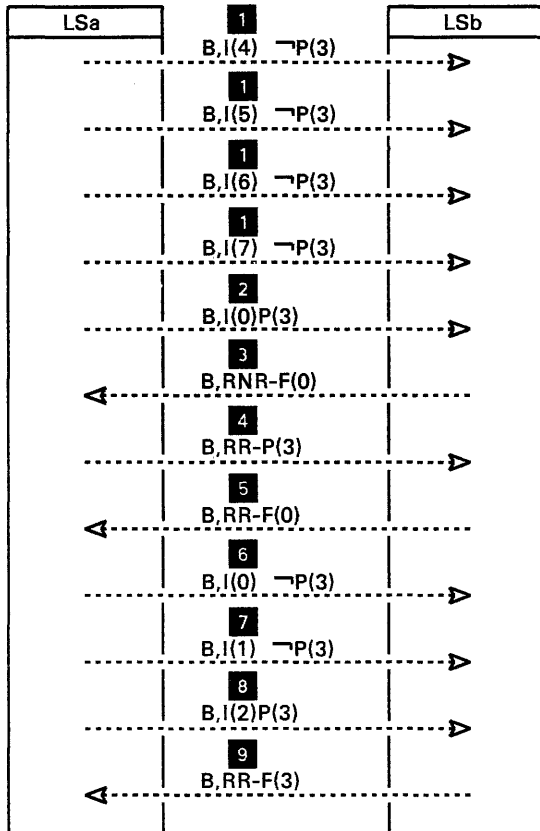
Figure 3-11. Primary and Secondary Link Stations Exchange Numbered Frames



1. Link station LSa informs link station LSb that LSa is ready to receive frames.
2. LSb responds that LSb must be initialized before LSb can transmit numbered information frames.
3. LSa complies and sets LSb's mode to initialization mode in preparation for frame exchange.
4. LSb sends a unnumbered acknowledgment that its mode has been set to initialization mode.
5. LSa sends a command to set normal response mode at LSb so that LSb may respond to numbered information frames.
6. LSb sends an unnumbered acknowledgment that its mode is now response mode and it can now respond to numbered information frames from LSa.
7. LSa informs LSb that LSa is ready to receive numbered information frames and LSa polls LSb for frames.
8. LSa transmits numbered information frame 0 and indicates that there are more frames coming.
9. LSb transmits its numbered information frame 0 and indicates that there are more frames coming.
10. LSa transmits its numbered information frames 1 and 2. LSa indicates that there are more frames coming.
11. LSb transmits its numbered information frame 1, indicates that there are more frames coming, and confirms LSa's frames 0 and 1.
12. LSa transmits its numbered information frame 3, indicates that there are more frames coming, and confirms LSb's frame 0.
13. LSb transmits its numbered information frame 2, indicates that it is LSb's final frame, and confirms LSa's frame 2.
14. LSa indicates it is ready to receive, confirms LSb's frame 2, and polls LSb for more frames.
15. LSb indicates it is ready to receive, confirms LSa's frame 3, and indicates that it is LSb's final frame.

(Figure 3-6 gives the meanings of the symbols and abbreviations that appear in this figure.)

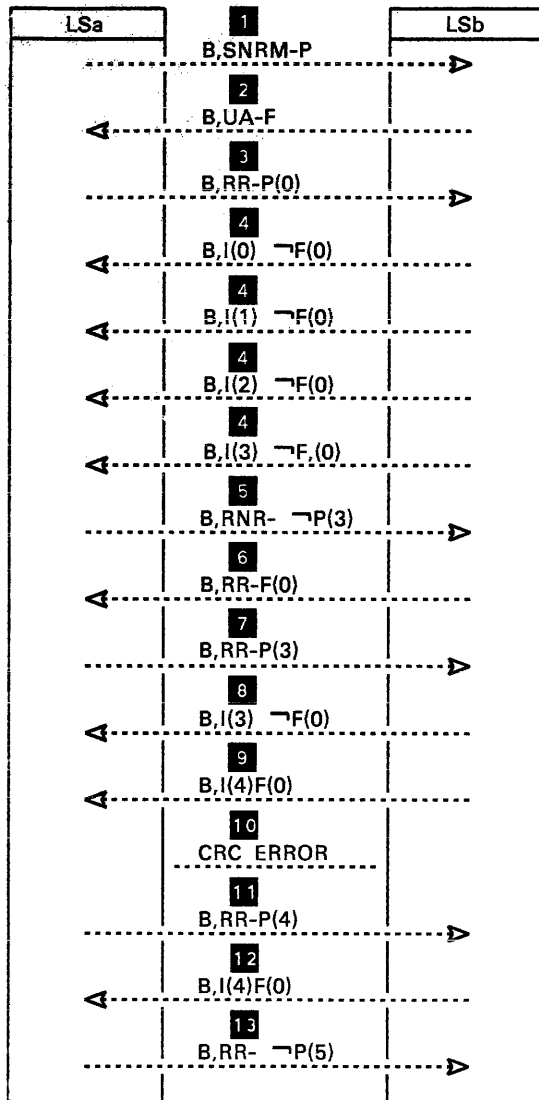
Figure 3-12. Secondary Link Station Comes Online, Primary Link Station and Secondary Link Stations Exchange Numbered Information Frames



1. Link station LSa sends its numbered information frames 4-7, confirms link station LSb's numbered information frame 2, and indicates that more frames follow.
2. LSa sends its frame 0 and polls LSb for confirmation.
3. LSb confirms frames 4-7, and indicates LSb is not ready to receive more frames at this time.
4. LSa polls LSb to see if LSb is ready to receive more frames yet.
5. LSb indicates LSb is now ready to receive and that LSb has no further frames to send.
6. LSa re-sends frame 0.
7. LSa sends frame 1.
8. LSa sends frame 2 and polls LSb for confirmation.
9. LSb confirms LSa's frames 0-2, indicates that LSb is ready to receive more, and indicates that LSb has no further frames to send.

(Figure 3-6 gives the meanings of the symbols and abbreviations that appear in this figure.)

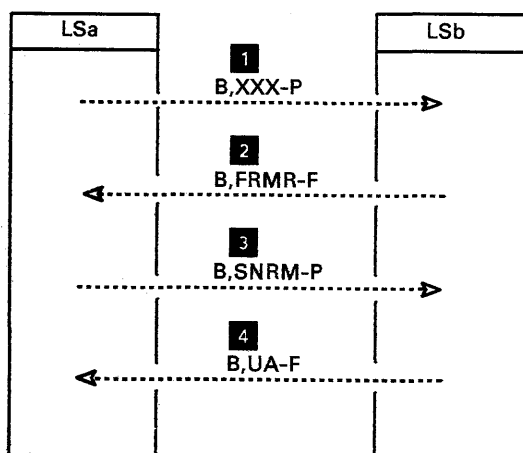
Figure 3-13. Busy Secondary Link Station



1. Link station LSa sets link station LSb's mode to normal response mode, preparing LSb to send and receive numbered information frames.
2. LSb confirms that LSb received the mode-setting command.
3. LSa indicates that LSa is ready to receive frames from LSb and polls LSb for frames.
4. LSb sends numbered information frames 0-3 and indicates that LSb has more frames to send.
5. LSa becomes busy, transmits to LSb the confirmation of LSb's frames 0-2, and indicates to LSb that LSa is not ready to receive.
6. LSb indicates to LSa that LSb is ready to receive when LSa's busy condition clears.
7. LSa's busy condition clears; LSa informs LSb that LSa is now ready to receive and polls LSb for frames.
8. LSb re-transmits numbered information frame 3.
9. LSb sends frame 4 and indicates that frame 4 is LSb's final frame.
10. LSa discovers a CRC error upon receiving frame 4.
11. LSa indicates to LSb that LSa is ready to receive, and that LSa is expecting LSb's frame 4.
12. LSb re-sends frame 4 and indicates that frame 4 is LSb's final frame.
13. LSa confirms LSb's frame 4 and indicates that LSa is ready to receive.

(Figure 3-6 gives the meanings of the symbols and abbreviations that appear in this figure.)

Figure 3-14. Busy Primary Link Station

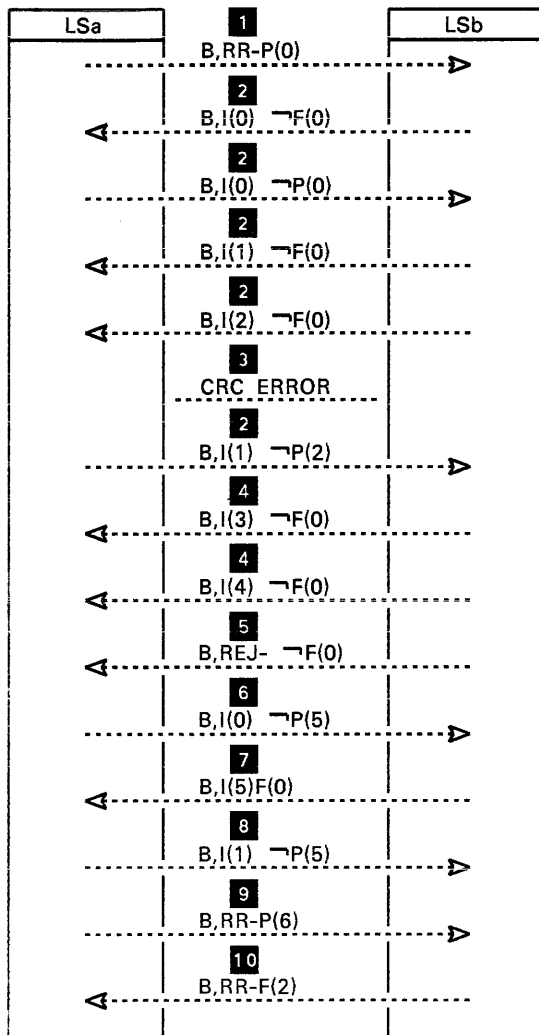


1. Link station LSa sends link station LSb a frame with an undefined control field.
2. LSb rejects LSa's undefined frame.
3. LSa resets LSb's error condition by resetting LSb's response mode to normal response mode.
4. LSb confirms that LSb received the mode-setting command.

(Figure 3-6 gives the meanings of the symbols and abbreviations abbreviations that appear in this figure.)

Figure 3-15. Invalid Command

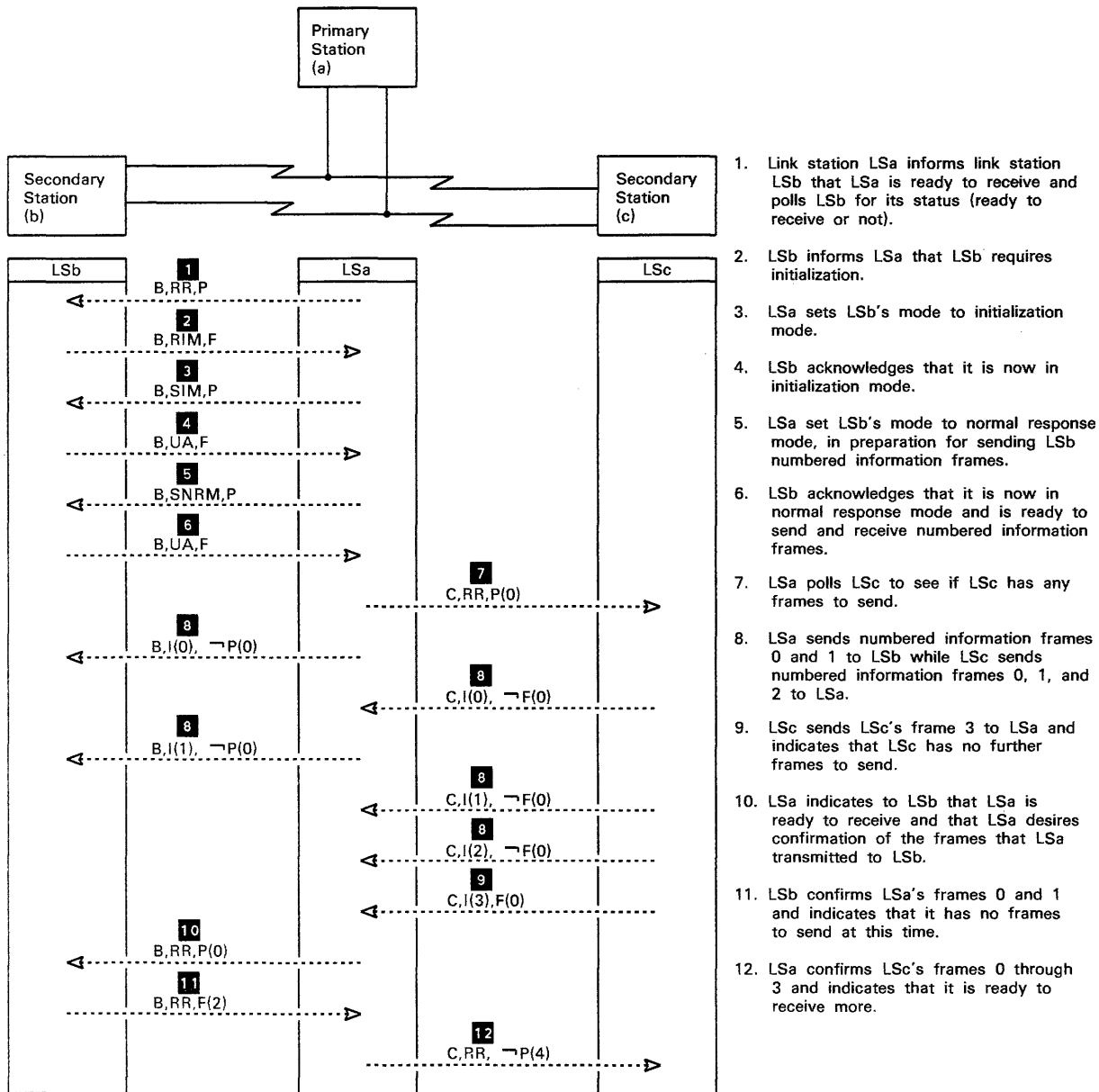




1. Link station LSa informs link station LSb that LSa is ready to receive numbered information frames from LSb and polls LSb for frames.
2. LSa and LSb exchange numbered information frames (note that LSa's frames are longer than LSb's).
3. LSb receives LSa's frame 0 with a CRC error.
4. LSb continues sending numbered frames, LSb does not confirm LSa's frame 0.
5. LSb receives LSa's frame 1 and since LSb has not successfully received LSa's frame 0, LSa's frame 1 is out-of-sequence. LSb therefore rejects LSa's frame 0.
6. LSa re-sends LSa's frame 0 and confirms LSb's frames 0-4.
7. LSb sends LSb's frame 5 and indicates that LSb has no further frames to send.
8. LSa re-sends LSa's frame 1.
9. LSa confirms LSb's frame 5, indicates it is ready to receive more frames from LSb, and polls LSb for more frames.
10. LSb confirms LSa's frame 1, indicates it is ready to receive more frames from LSa, and indicates that LSb has no further frames to send.

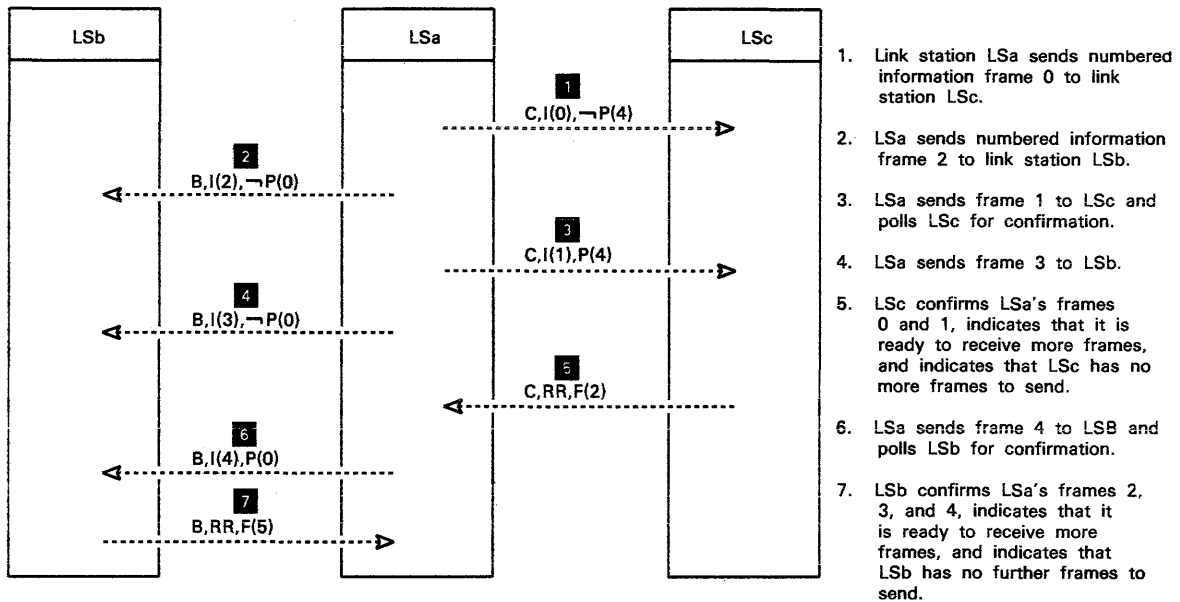
(Figure 3-6 gives the meanings of the symbols and abbreviations that appear in this figure.)

Figure 3-16. Numbering Error in Full-Duplex Exchange



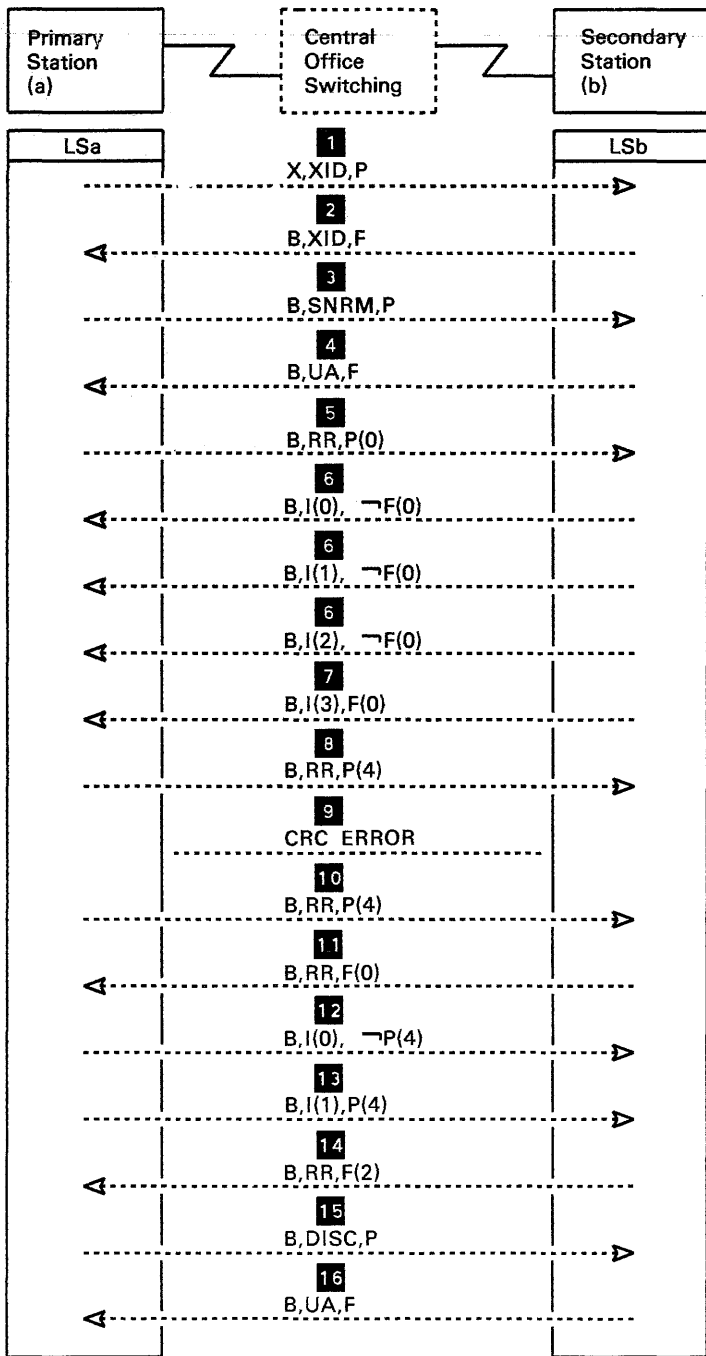
(Figure 3-6 gives the meanings of the symbols and abbreviations that appear in this figure.)

Figure 3-17. Secondary Link Station Comes Online, Primary Link Station Sends to One Secondary Link Station and Receives from Another



(Figure 3-6 gives the meanings of the symbols and abbreviations that appear in this figure.)

Figure 3-18. Interleaved Primary Link Station Transmissions



1. Link station LSA sends a frame with a general (broadcast) address to link station LSB. The frame requests LSB's address and identity.
2. LSB returns the frame, but puts LSB's address in the place of the broadcast address.
3. LSA sets LSB's response mode to normal response mode, preparing LSB to send and receive numbered information frames.
4. LSB acknowledges that its mode is now normal response mode.
5. LSA indicates to LSB that LSA is ready to receive numbered information frames and polls LSB for frames.
6. LSB sends frames 0 through 2 and indicates that it has more frames to send.
7. LSB sends frame 3 and indicates that it has no more frames to send.
8. LSA confirms LSB's frames 0 through 3, indicates that LSA is ready to receive more frames, and polls LSB for frames.
9. A CRC error occurs and LSB rejects LSA's frame. LSB does not confirm LSA's RR.
10. LSA resends its confirmation, receive ready, poll frame.
11. LSB confirms LSA's frame and indicates that LSB is ready to receive frames.
12. LSA sends frame 0.
13. LSA sends frame 1 and polls LSB for confirmation.
14. LSB confirms LSA's frames 0 and 1.
15. LSA commands LSB to disconnect.
16. LSB confirms that it is entering disconnected mode.

(Figure 3-6 gives the meanings of the symbols and abbreviations that appear in this figure.)

Figure 3-19. Mode Setting and Inquiry Response

## CHAPTER 4. ROUTING DATA FROM SUBAREA TO SUBAREA

The designer of an SNA network specifies one or more routes between each pair of subareas containing network addressable units that may communicate. Depending upon the applications a network is to serve, the network designer will wish to achieve some combination of the following objectives for each route in the network:

- Minimize the time that each message spends on a route
- Maximize the number of messages per unit of time that two subareas may exchange
- Minimize the cost of the physical components of the routes
- Maximize route availability
- Minimize data loss or retransmission caused by failures of physical components in the route
- Employ the most secure route for a given session
- Keep transmission queues and traffic congestion at intermediate routing nodes under control

These objectives involve compromises. For example, a network designer might minimize the time that each message spends on a route by assigning the route relatively few messages per unit of time, thereby achieving the first objective above at the expense of the second.

This chapter describes how SNA helps network designers achieve these objectives.

### SNA ROUTING OVERVIEW

This section explains what paths and routes are in an SNA network and describes how SNA networks route data traffic.

#### Path, Explicit Route, and Route Extension

In SNA, a path consists of a series of nodes, links, and path control and data link control components that are traversed by PIUs exchanged between two network addressable units. In an SNA network, one or more paths are defined for each pair of network addressable units that can communicate.

Figure 4-1 on page 4-2 shows a path between network addressable units in an SNA network. In this figure, two end users (an application program in HOST1 and an operator at a display terminal attached to PNODE4.2) communicate via a session between LU1.2 and LU4.3.

HOST1 contains an SNA host node, within which is an SNA logical unit denoted LU1.2. If the access method in HOST1 is ACF/TCAM, then either (1) LU1.2 is a device message handler (DMH) and associated code, or (2)

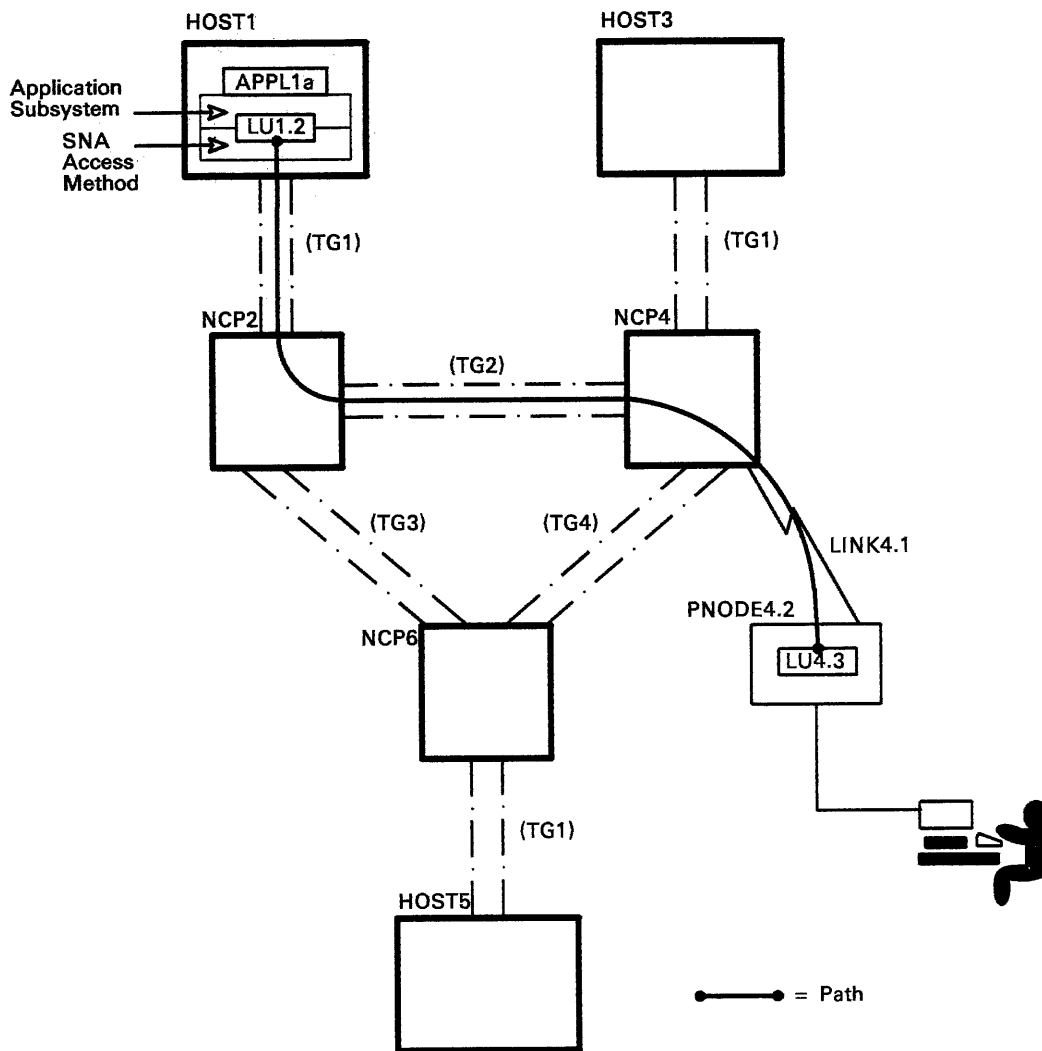


Figure 4-1. Path between Two Logical Units

LU1.2 is located partly in the access method and partly in application program APPL1a. If the access method in HOST1 is ACF/VTAM, then LU1.2 is located partly in the access method and partly in an application subsystem, such as CICS/VS or IMS/VS, through which the application program denoted APPL1a gains access to the network.

The path shown in Figure 4-1 proceeds from LU1.2 in HOST1 through transmission group TG1, NCP2, TG2, NCP4, and LINK4.1 to LU4.3 in PNODE4.2.

TG1 is a System/370 channel that connects HOST1 to the communication controller that contains NCP2. NCP2 and NCP4 are subarea nodes. TG2 consists of one or more SDLC links that connect NCP2 and NCP4.

LINK4.1 is a single SDLC link that connects NCP4 and PNODE4.2.

Peripheral node PNODE4.2 contains the other LU involved in the session, LU4.3. PNODE4.2 might, for example, be an IBM 3274 control unit, or an IBM 8130 processor under the control of DPPX.<sup>1</sup> Attached to PNODE4.2 is a display terminal. The terminal and its operator constitute the end user communicating with APPL1a. This end user gains access to the SNA network via the LU denoted LU4.3.

Figure 4-1 shows a path between a host LU and a peripheral LU. In SNA, paths may also exist between two host LUs. Figure 4-2 on page 4-4 shows a path between LU1.2 in HOST1 and LU3.2 in HOST2.

Paths can exist between a host LU and LUs in all types of SNA peripheral nodes, but only a few types of such nodes (for example, the IBM 6670 Information Distributor) can have a path between their LUs and other peripheral LUs. Figure 4-3 on page 4-5 shows how peripheral LUs may communicate with each other even when their nodes do not support a direct path between them.

In Figure 4-3, PATH1 connects LU1.2 in HOST1 with LU2.3 in peripheral node PNODE2.2. PATH2 connects LU1.2 with LU4.3 in PNODE4.2. To communicate with LU4.3, LU2.3 sends its messages to LU1.2 over PATH1. LU1.2 sends the messages to LU4.3 over PATH2. To communicate with LU2.3, LU4.3 sends its messages to LU1.2, which forwards them to LU2.3. ACF/TCAM is particularly suitable for this kind of communication, because with ACF/TCAM in HOST1, LU1.2 is a device message handler that can readily be designed to forward messages to their appropriate destinations.

An SNA path physically consists of one or both of the following components:

- An explicit route between two subarea nodes
- A peripheral link between a subarea node and a peripheral node

---

<sup>1</sup> Distributed Processing Programming Executive for the IBM 8100 Information System.

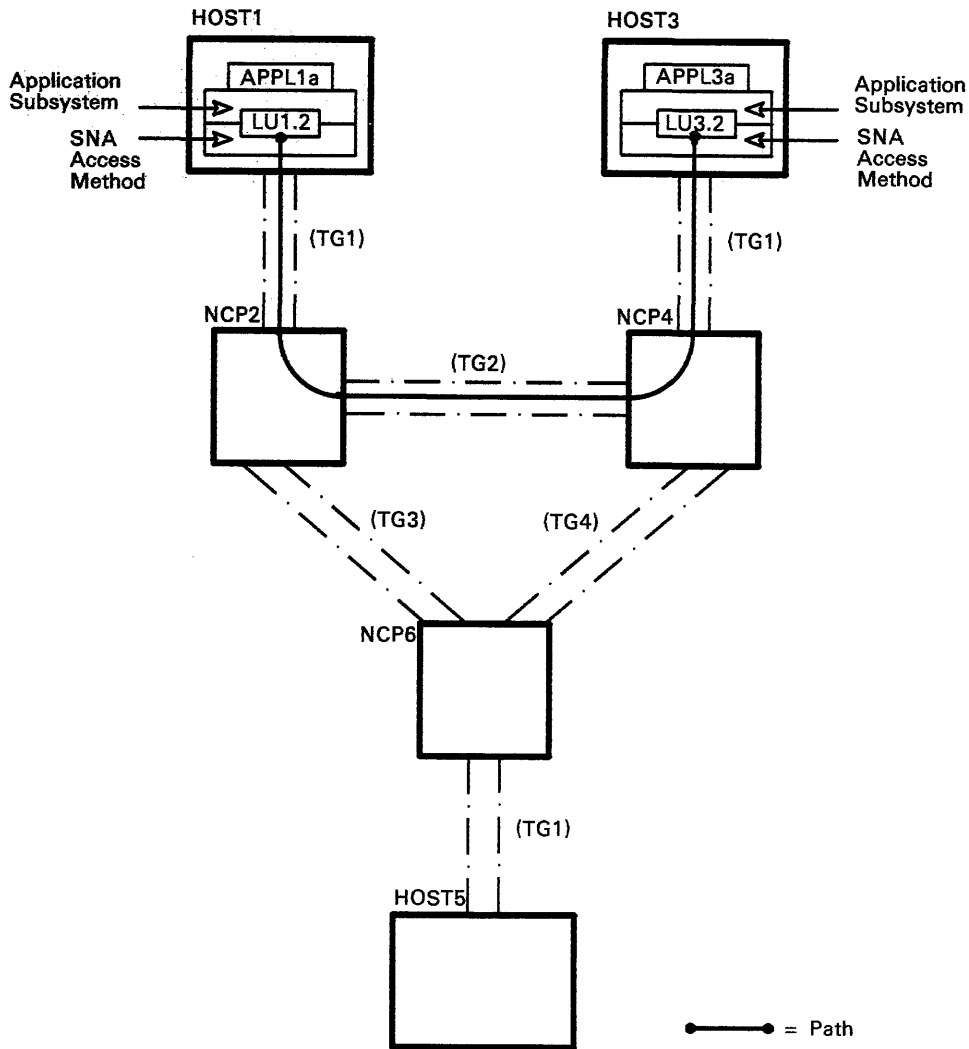


Figure 4-2. Path between Two Host Logical Units



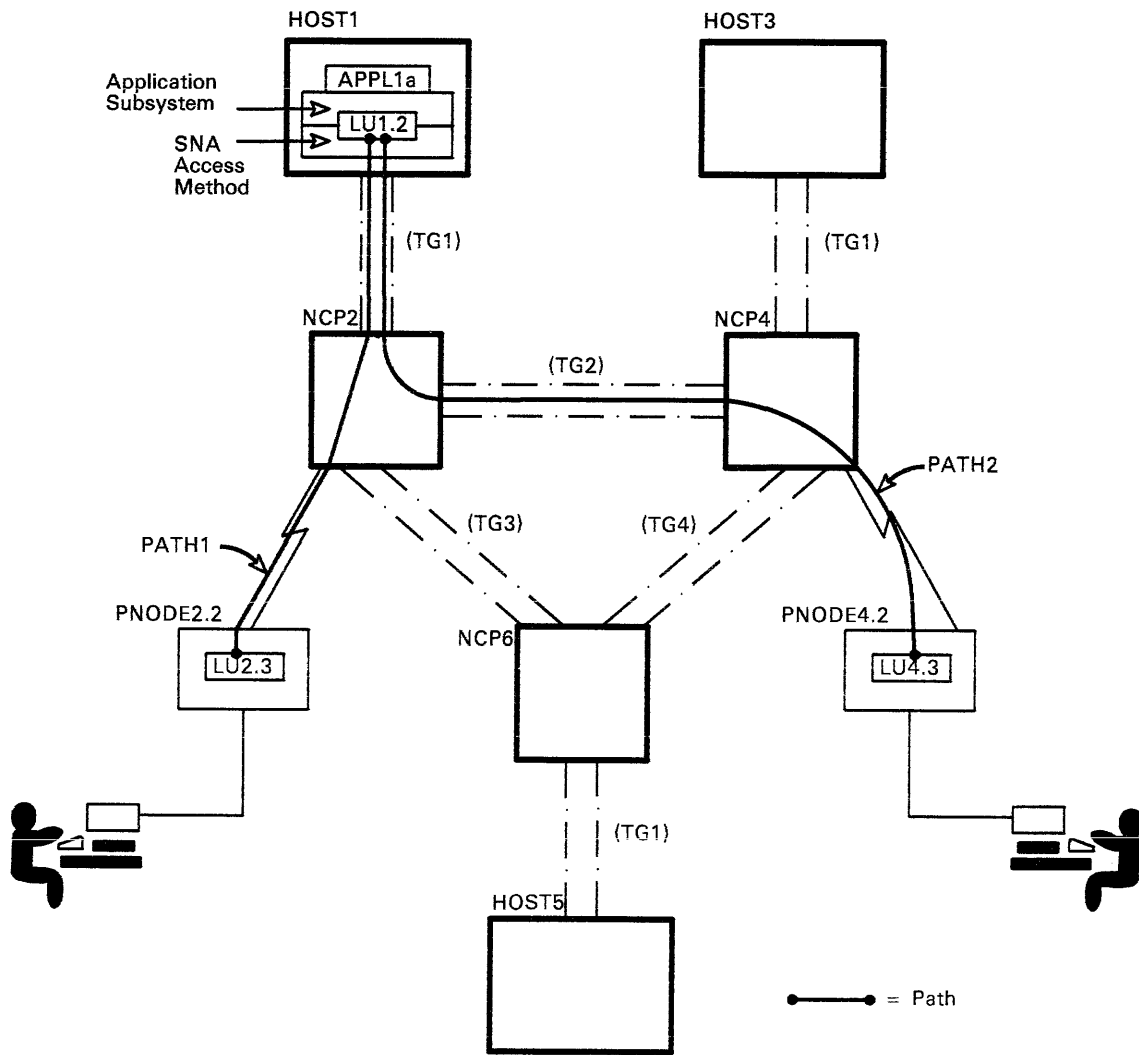


Figure 4-3. Two Paths Connecting Peripheral LUs

An SNA explicit route is the part of the path that lies between the two end subareas on the path. Figure 4-4 on page 4-7 shows the explicit route that is part of the path shown in Figure 4-1. This explicit route lies between the two subarea nodes HOST1 and NCP4. LINK4.1 is the peripheral link that completes the path.

An SNA virtual route imposes a priority usage on an explicit route to distinguish service levels among sessions. The combination of priority and explicit-route selection can be used to provide different classes of service for sessions. The concepts of virtual routes and classes of service are described below under "Explicit Routes, Virtual Routes, and Transmission Priorities."

A peripheral link is the part of a path that lies between a subarea node and an adjacent peripheral node. For the path in Figure 4-4, the peripheral link (LINK4.1) extends from NCP4 to PNODE4.2.

An SNA path can connect:

- A NAU in a subarea node and a NAU in a peripheral node attached to the subarea node by an SDLC link
- NAUs in two subarea nodes
- A NAU in a host node and a NAU in a peripheral node attached to that host node's channel
- A NAU in a host node and a NAU in a peripheral node attached to another host node's channel

These combinations of explicit routes and peripheral links can make up a physical path between NAUs:

- A path between a NAU in a subarea node and a NAU in a peripheral node attached to a different subarea node by an SDLC link (as shown in Figure 4-1) consists of an explicit route and a peripheral link.
- A path between NAUs in two subarea nodes (as shown in Figure 4-2) consists of an explicit route but no peripheral link.
- A path between a NAU in a host node and a NAU in a peripheral node attached to that host node by a channel consists only of a peripheral link (the channel).
- A path between a NAU in a host node and a NAU in a peripheral node that is channel attached to another host node consists of an explicit route and a peripheral link (the channel).

## Intermediate Routing Nodes and Boundary-Function Nodes

For the explicit route shown in Figure 4-4, subarea node NCP2 is on the route but is not one of the two end subarea nodes for the route. Such a subarea node is called an intermediate routing node; it participates in an explicit route but its subarea does not contain either of the end points of the route. An intermediate routing node examines the destination address associated with each PIU flowing on an explicit route and places each PIU on the transmission group that leads to the next subarea node on the route to that destination.

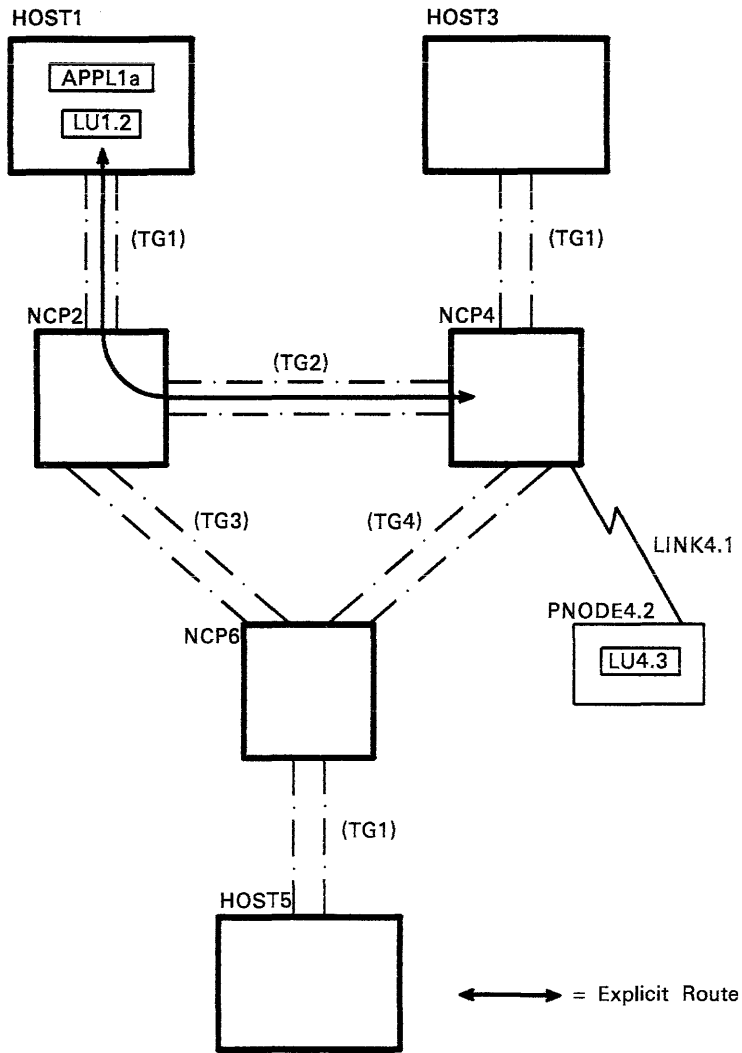


Figure 4-4. Explicit Route between Two Subareas

Subarea node NCP4, on the other hand, is an end subarea node on the route and is also attached to a peripheral node that is an end node on the path associated with the route. Subarea node NCP4 performs certain boundary functions required for transmission of PIUs to and from a peripheral node.

A subarea node may simultaneously provide boundary functions for some explicit routes and serve as an intermediate routing node for other explicit routes. Consider, for example, the two explicit routes shown in Figure 4-5 on page 4-9—one between HOST3 and NCP6, and the other (already described) between HOST1 and NCP4. NCP4 serves as an intermediate routing node on the explicit route between HOST3 and NCP6 while providing boundary functions on the explicit route between HOST1 and NCP4.

### Explicit Routes, Virtual Routes, and Transmission Priorities

Routing may be considered at two levels: a physical level and a logical level. At the physical level, an explicit route is a sequence of nodes and transmission groups by which two subareas communicate. Data can flow over an explicit route in either direction, using the same set of nodes and links in both directions. In a given direction, up to 16 explicit route numbers can be used to identify explicit routes between two subarea nodes.

At the logical level, a virtual route is a logical connection between a pair of subareas. Each virtual route in the network is assigned to an explicit route and takes on the physical characteristics of that route. The explicit route is referred to as the underlying explicit route for that virtual route. In addition, each virtual route has certain flow-control characteristics. Up to 48 virtual routes may be defined between two subarea nodes, thereby dividing the message traffic on the explicit routes into subsets that differ in their flow-control characteristics.

Two flow-control characteristics are associated with a virtual route: transmission priority and virtual-route pacing.

Virtual-route pacing is a congestion-control mechanism that is described later in this chapter under "Regulating Data Flow Along a Route."

Each virtual route in a network is assigned one of three transmission priorities by the network designer. If many virtual routes share the same explicit route, messages flowing on virtual routes that have a higher transmission priority are transmitted ahead of messages flowing on virtual routes that have a lower transmission priority.

When activated, each SNA session is assigned to a virtual route, and all messages flowing on the session use that virtual route.

Sessions using a particular explicit route may vary in their data transmission requirements. For example, sessions associated with data-base inquiry applications usually require faster data transmission

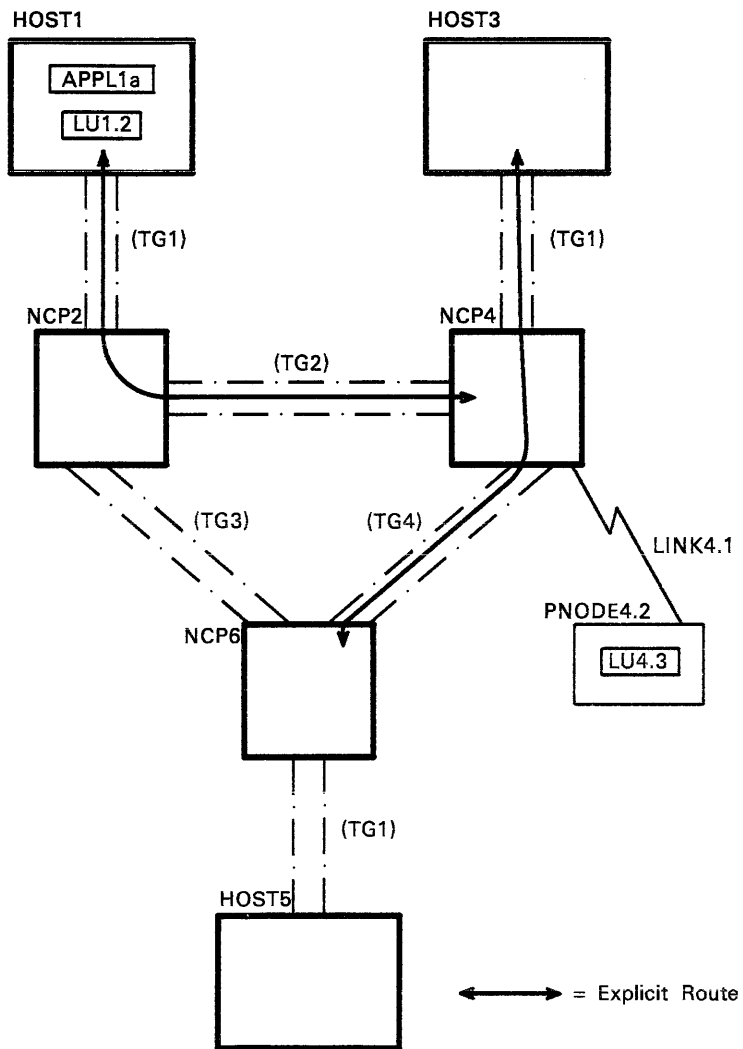


Figure 4-5. Two Explicit Routes

and more predictable response times than sessions associated with data collection applications. By assigning sessions using the same explicit route to different virtual routes having different transmission priorities, the network designer can cause messages associated with high-priority applications to flow ahead of messages associated with low-priority applications.

Figure 4-6 on page 4-11 shows how virtual and explicit routes are related to each other and to LUs, sessions, and end users. Associated with explicit route ER1 are two virtual routes. These are identified as virtual route (VR1, TP0) and virtual route (VR1, TP1). TP0 and TP1 are transmission priorities.

One LU-LU session between LU1.2 and LU2.4 uses virtual route (VR1, TP1). Application program APPLa1 uses this session to communicate with APPLc1.

Two LU-LU sessions are assigned to virtual route (VR1, TP0): one between LU1.2 and LU2.4 and one between LU1.3 and LU2.3. Application program APPLa2 uses the first of these sessions to communicate with application program APPLc2, and application program APPLb1 uses the second session to communicate with application program APPLd1.

Message units flowing between APPLa1 and APPLc1 are favored for transmission over message units flowing between other end users because APPLa1 and APPLc1 use the session assigned to the virtual route that has the higher priority: (VR1, TP1).

In this figure, explicit route ER1 may be made up of many transmission groups and intermediate routing nodes. If one of the LUs in HOST1 and HOST2 (which are subarea nodes) is instead in a peripheral node, then ER1 requires a peripheral link to reach those LUs.

## Multiple Explicit Routes and Class of Service

The explanation thus far has considered single explicit routes between subareas. An SNA capability called multiple routing allows different explicit routes to be established between a pair of subareas. This capability can be used to increase the probability that a route will be available when needed for a session.

Each SNA session between a pair of subareas is associated with a specific explicit route, and several sessions may be associated with the same route. A session is assigned to a virtual route at session activation, and all data for a given session flows over the same route. One or more virtual routes are assigned to each explicit route.

Figure 4-7 on page 4-12 illustrates two explicit routes between subarea 1 and subarea 4. One of these is "shorter" than the other in that it involves one less node and one less transmission group. The network designer might assign six virtual routes to the two explicit routes between subarea 1 and subarea 4. Note that both explicit routes share transmission group TG1 and network control program NCP2.

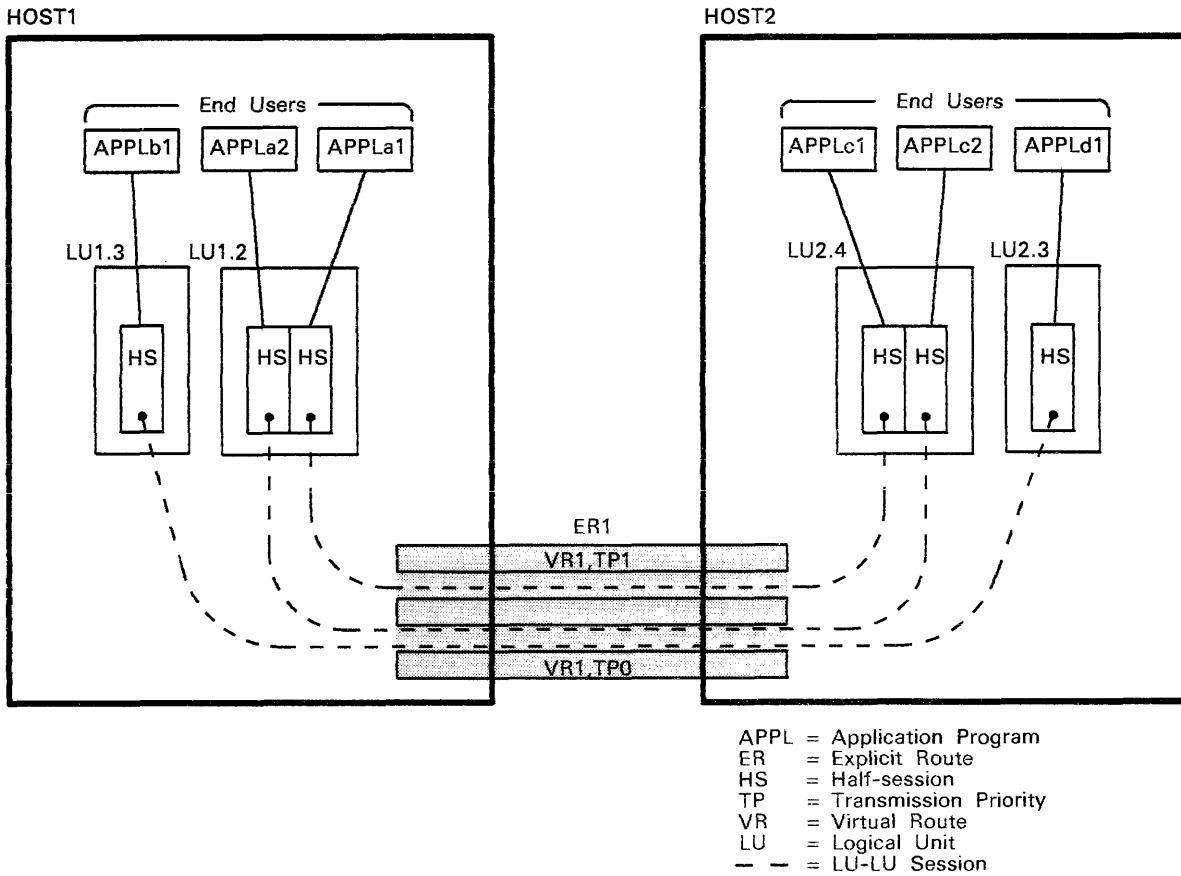


Figure 4-6. Relationship Among an Explicit Route, Virtual Routes, Logical Units, Sessions, and End Users

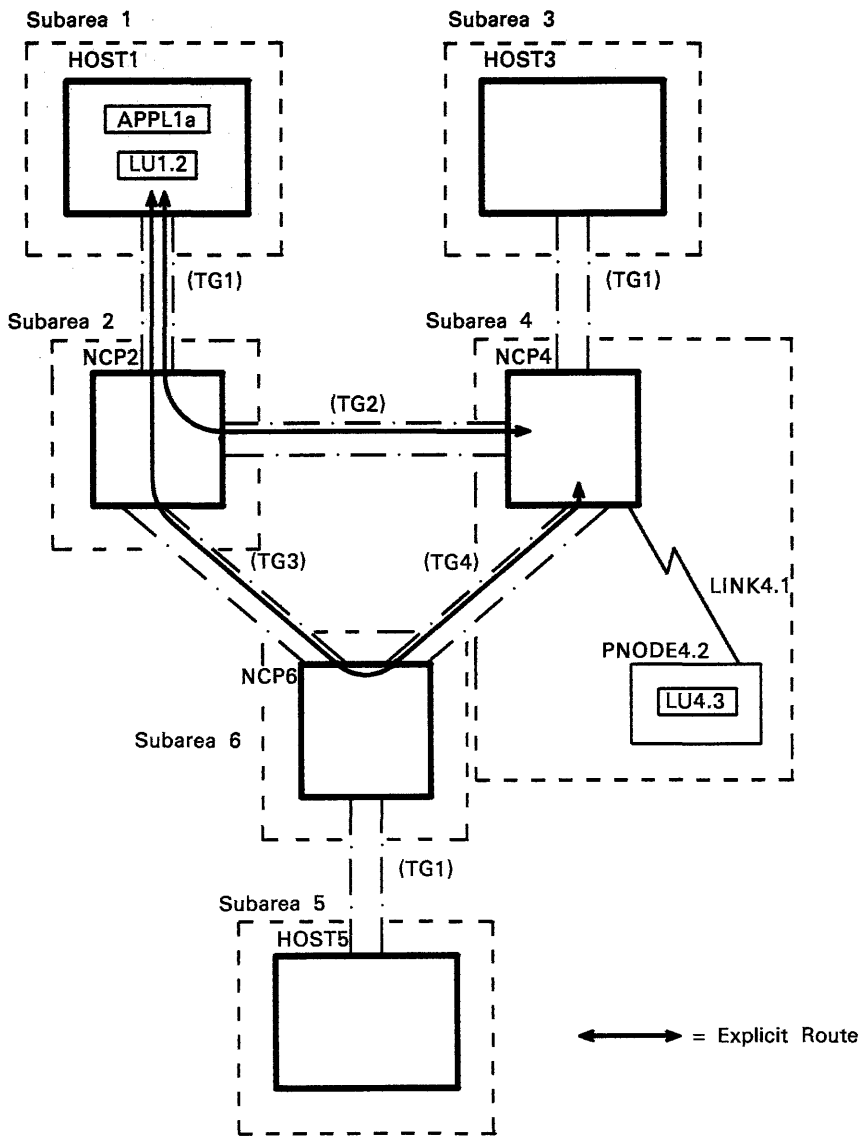


Figure 4-7. Two Explicit Routes Between the Same Two Subareas



Some explicit routes, and therefore the virtual routes that use them, may be inherently better than others for certain kinds of applications. Sessions associated with inquiry/response applications may require the quickest available route to enhance response time; sessions associated with other kinds of applications may require the least expensive route or the more secure route.

Sometimes several virtual routes may have approximately the same characteristics and, therefore, be equally suitable for transferring data for a particular kind of session. SNA allows the network designer to specify a list of several virtual routes that can be assigned to sessions between LUs in two subareas. A specific class of service is furnished to any sessions that use one of the listed routes. Because sessions are assigned to the first available route in the list, routes should appear in the list in descending order of desirability.

The class of service for a session is selected when the session is initiated and remains in effect until the session ends. The SSCP resolves the class-of-service name to a list of virtual routes, from which one is selected when the session is activated. If none of the virtual routes in the list of virtual routes is available, an attempt is made to activate an explicit route for one of the virtual routes in the list. If no explicit route can be activated, a request to activate a session is rejected. In this case, the requester must try to activate the session later. ACF/TCAM and ACF/VTAM provide a monitoring function that allows a host LU to be notified automatically when a virtual route providing the requested class of service becomes available.

SNA support for multiple explicit routes between subareas lets sessions be assigned to different virtual routes when an operational virtual route is lost. A virtual route may be lost because one or more of the physical elements in the explicit route to which it was assigned failed or was deactivated. In this event, all active sessions that use the virtual route are terminated and appropriate session partners are notified. If a different virtual route between the same two subareas is active, or can be made active, SSCPs and LUs can reinitiate the interrupted sessions over the different route.

For example, in Figure 4-7, if the explicit route that links subareas 1 and 4 via TG1, NCP2, and TG2 became unavailable because TG2 failed, disrupted sessions could be reassigned to virtual routes on the other explicit route shown in the figure, because it also links HOST1 with NCP4.

## How SNA Networks Route Messages

An SNA network routes messages on a decentralized, node-by-node basis. The network routes messages between subareas in accordance with the subarea portion of the network address within each PIU in the message. Upon reaching its destination subarea, each PIU is delivered to the NAU specified by the element portion of its network address.

The system programmer defines for each node the information it needs in order to properly process PIUs flowing on routes that pass through the node. Stored in routing tables in each node, this information is used by path control to properly route PIUs.

Message units are assigned to a virtual route on a session-by-session basis; that is, all message units that flow on the same session use the same virtual route.

For LU-LU sessions, the Bind Session (BIND) request that initiates the session specifies a class of service for that session. (The class of service may be specified directly, as a class-of-service name [COS name], or indirectly as a mode name. COS names and mode names are described further in Chapter 5 under "Activating an LU-LU Session.") The SSCP provides a list of virtual routes associated with each class of service. The session is assigned to the first available virtual route in the list; all message units that flow on the session use the assigned route.

Associated with each virtual route is an underlying explicit route. When building the FID4 transmission header (TH)<sup>2</sup> for a message unit, the path control component of the originating node specifies in the header the virtual route and explicit route over which the message unit will flow. The path control component of each node along the explicit route uses the explicit-route number and the subarea portion of the destination network address in the TH to determine the next transmission group the message unit is to be sent over on its way to its destination subarea.

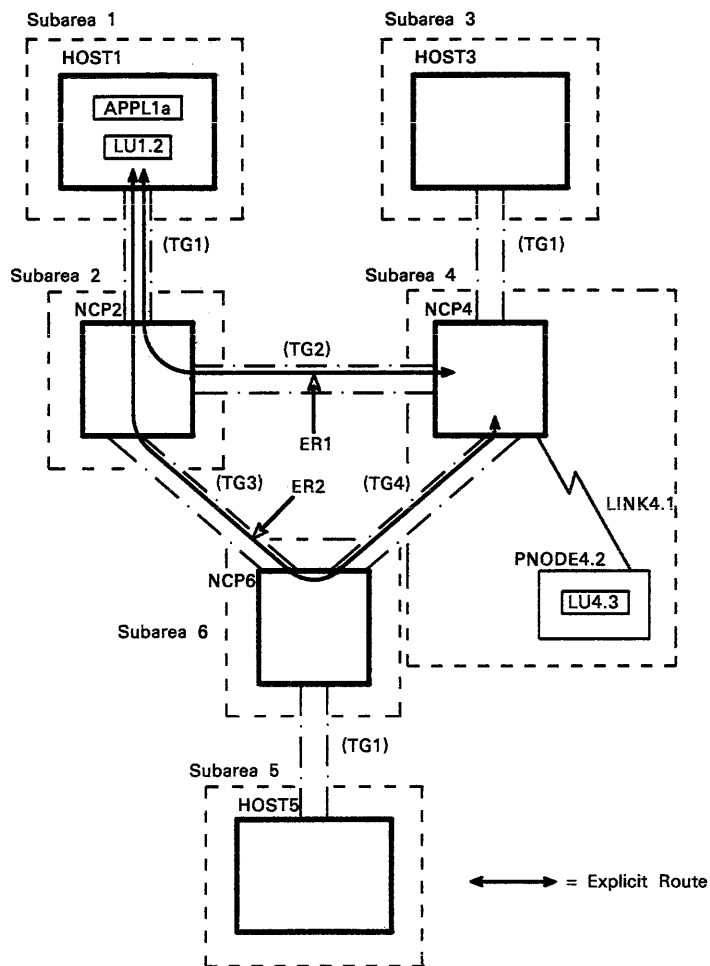
Figure 4-8 on page 4-15 shows two explicit routes between subareas 1 and 4. Associated with each explicit route is a single virtual route. Beneath the route diagram is a series of simplified routing-table segments used by each node on the two routes.

Consider a PIU that flows from subarea 1 to subarea 4 over explicit route ER1. From the destination network address specified in the transmission header of the PIU, path control in HOST1 determines that the PIU's destination is in subarea 4. The HOST1 routing table indicates that TG1 is the transmission group over which the PIU must pass to the next node, NCP2, on its way to subarea 4. Path control then schedules the PIU for transmission on TG1.

Upon receiving the PIU, path control in NCP2 looks at the NCP2 routing table to determine the next transmission group that the PIU must pass over on its way to subarea 4. The table shows that TG2 is the required transmission group for a PIU whose TH specifies subarea 4 and explicit

---

<sup>2</sup> A FID4 is the format of transmission header used for sending traffic between adjacent subarea nodes when both nodes support explicit-route and virtual-route protocols. The various formats of transmission headers are listed in the Glossary under the entry "format identification (FID) field."



Routing Table Name	Destination Subarea	Explicit Route Number	Next Subarea	Next Node	Transmission Group to Next Node
HOST1	4	1	2	NCP2	TG1
	4	2	2	NCP2	TG1
NCP2	4	1	4	NCP4	TG2
	4	2	6	NCP6	TG3
	1	1	1	HOST1	TG1
	1	2	1	HOST1	TG1
NCP4	1	1	2	NCP2	TG2
	1	2	6	NCP6	TG4
NCP6	4	2	4	NCP4	TG4
	1	2	2	NCP2	TG3

Figure 4-8. Routing Table Segments for Two Explicit Routes

route 1. Path control therefore schedules the PIU for transmission on TG2.

Upon reaching NCP4, the PIU has traversed its route. From the element portion of the destination network address in the TH, path control in NCP4 determines which NAU in subarea 4 is to receive the PIU. From a table similar to the routing table, path control in NCP4 determines which route extension<sup>3</sup> to use in sending the PIU to its destination NAU in subarea 4.

Note that in Figure 4-8 no single node knows all of the components involved in ER1 or ER2. All that any particular node knows about each explicit route is the next node on that route and the transmission group used to reach that node.

## Activating and Deactivating Routes

SNA access methods and network control programs activate and deactivate explicit and virtual routes as needed. Before a route can be activated, a network operator (human or programmed) must enter the commands needed to activate the physical network resources (nodes and links) through which the route passes. When the nodes and links have been activated, the transmission groups in the explicit route are operational; the explicit route itself is therefore operational, although not yet active.

Once operational, an explicit route is eligible to be activated. Before a virtual route can use an operational explicit route, the subarea node in which the explicit route originates sends an activation request to the subarea node at the other end of the route. This request is called a Network Control Explicit Route Activate (NC-ER-ACT) request. As it passes through the transmission groups that make up the explicit route, this request verifies that the route is usable, is complete throughout its length, and does not "loop"—that is, pass through any node more than once. The request also determines the length of the explicit route in number of transmission groups, or "hops," it traverses.

Eligible explicit routes and associated virtual routes are automatically activated as required to allow session traffic to flow. The network operator never explicitly activates or deactivates either explicit routes or virtual routes.

A virtual route is automatically deactivated when all sessions assigned to it have ended. An explicit route is automatically deactivated when a physical component of the route (such as a transmission group) is deactivated or becomes inoperative.

---

<sup>3</sup> A route extension is 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.

## Regulating Data Flow Along a Route

The flow of data through a network varies over time, with the degree of variation depending on the applications for which the network is used, among other factors.

In many networks, peak traffic conditions during the conduct of business can be anticipated and planned for. But traffic peaks can also result from such conditions as the failure of an explicit route and the consequent diversion of its traffic to different explicit routes. The different route may become overburdened with the extra traffic it is required to carry. Unexpected demands for data transmission by certain end users (terminal operators or applications) also may impose peak traffic conditions on parts of the network. Thus, not only does the total traffic in a network vary, but the amount of traffic that flows in various parts of the network may fluctuate as well.

Whenever the rate at which data is presented to a network exceeds the capacity of the network routes over which the data must flow, data congestion may result. Response times may lengthen and throughput may be impaired. Severe or prolonged congestion in one part of a network may be propagated to other parts, causing overall network efficiency to suffer.

SNA provides three capabilities that help alleviate congestion:

- Capability to operate multiple links between adjacent nodes
- Capability to operate multiple explicit routes between subareas
- Capability to regulate data flow through global and local flow-control algorithms

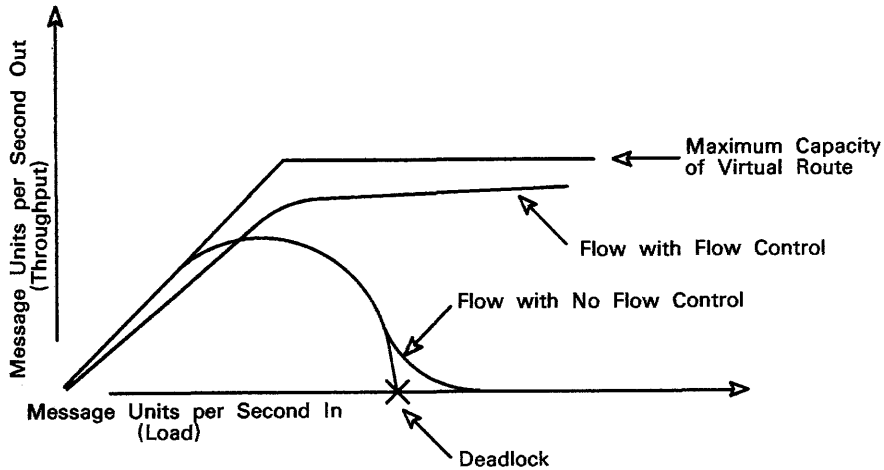
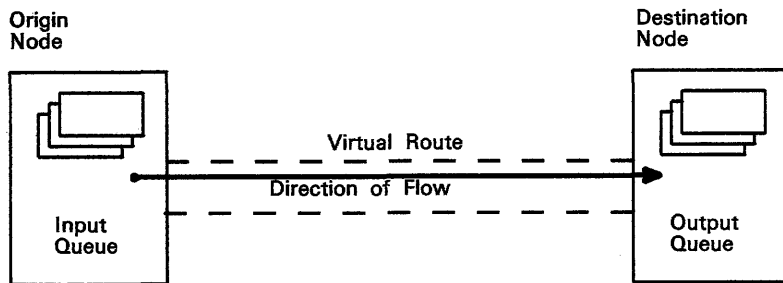
The first capability is described in Chapter 3 under "Transmitting Data between Adjacent Subarea Nodes;" the second is described earlier in this chapter under "Multiple Explicit Routes and Class of Service."

### Global and Local Flow-Control Algorithms

Flow-control algorithms can operate on either a global or a local level. A global flow-control algorithm provides a coordinated, network-wide mechanism to prevent congestion, both at the endpoints of a path and at intermediate points. Local flow-control algorithms operate within individual nodes; they manage the network traffic through each node separately, based on information contained within the node. Local flow control algorithms can mediate congestion at endpoints of a path or provide feedback control to influence global flow-control algorithms.

Figure 4-9 on page 4-18 illustrates the effect that global and local flow-control capabilities may have on the throughput, expressed as message units per second, that a virtual route can achieve.

Two principal causes of congestion along a virtual route are:



The problem: To maximize throughput on a virtual route.

The solution:

- Regulate flow at the ends of the virtual route (global flow control)
- Regulate arrivals at individual nodes along the virtual route (local flow control)
- Selectively assign resources to arriving message units at each node along the virtual route, based on:
  - Transmission priority
  - Virtual route status
  - Resource availability

Figure 4-9. Effect of Flow Control on Throughput for a Virtual Route

- Relatively slow processing at the destination node of a virtual route
- Depletion of buffers in the intermediate routing nodes

If the origin node in Figure 4-9 puts message units on the virtual route faster than the destination node can accept and process them, congestion can occur at the destination node. To prevent this occurrence, each destination node provides a pool of buffers to receive message units being sent over each virtual route terminating in that node. When this pool becomes depleted, the destination node uses a global flow-control mechanism to adjust the flow to the capacity of the receiver or to halt the flow altogether until buffers are available.

When an NCP is serving as an intermediate routing node for many busy virtual routes, NCP buffers may become filled with message units waiting to be placed on the next transmission group on the way to their destinations. SNA provides both a local and a global flow-control mechanism to alleviate intermediate-node congestion.

The purpose of global flow control is to synchronize the rate of traffic entry into a network with the rate of traffic delivery, thereby protecting network resources from traffic overload.

The two SNA flow-control algorithms are session-level pacing and virtual-route pacing. Session-level pacing is a data-flow coordination mechanism that is used to prevent one of the partners in an SNA session from being overloaded with PIUs from the other session partner. Session-level pacing is described in Chapter 5 under "Pacing of Data Flow at the Session Level."

### Virtual-Route Pacing

Virtual-route pacing synchronizes the rate of data flow between the two end subareas of a virtual route. VR pacing combines the traffic flowing on all LU-LU sessions assigned to the same virtual route into one subarea-to-subarea flow. Virtual-route pacing manages traffic volume by controlling data flow through the network and by permitting a metered flow at each of the routing boundary-function nodes. Network feedback includes the ability to withhold a VR-pacing response in the FID4 transmission header of a PIU when sufficient resources are not available. Virtual-route pacing is bidirectional and the protocol is the same in both directions.

SNA networks implement virtual-route pacing as follows:

- Each node along a virtual route can detect congestion and set the appropriate congestion indicator bits in transmission headers
- Both end nodes of a virtual route can:
  - Pace the data flow to the other end node
  - Test bits in transmission headers to determine if congestion is occurring at any node along the virtual route
  - Withhold message units from the virtual route when congestion is detected and release them to the route when congestion ends

In virtual-route pacing, PIUs to be transmitted are placed in sequences of PIUs called "pacing groups." A subarea node sending PIUs must receive authorization from the subarea node receiving PIUs before sending a new pacing group. The nodes dynamically adjust the number of PIUs in each pacing group between a minimum and a maximum based on congestion detected on a virtual route between two subareas.

Bits within the FID4 TH identify a PIU as a virtual-route (VR) pacing request or a VR-pacing response. The first PIU in each pacing group is a VR-pacing request. A VR-pacing request indicates to the receiving node that the sending node is sending a pacing group. Once that pacing group is sent, the sending node will send no more PIUs until it receives a VR-pacing response from the receiving node. A VR-pacing response authorizes the subarea node sending PIUs to transmit another pacing group.

When the virtual route is activated, a minimum number of PIUs that can make up a pacing group (that is, a minimum pacing-group size) and a maximum number of PIUs that can make up a pacing group (that is, a maximum pacing-group size) are calculated for that virtual route. Unless the system programmer has specified otherwise when defining the network, the minimum pacing-group size is equal to the number of transmission groups that make up the explicit route that underlies the virtual route. The maximum pacing-group size is equal to three times the minimum pacing-group size.

These maximum and minimum pacing-group sizes are set at both ends of the virtual route. Because this choice of range is arbitrary, and because network configurations vary considerably, ACF/TCAM and ACF/VTAM permit the system programmer to specify appropriate minimum and maximum pacing-group sizes for each virtual route as that route is activated.

After a virtual route is activated, a subarea node at one end of the route can transmit a pacing group equal to the minimum pacing-group size. This becomes the current pacing-group size. A VR-pacing count is initialized with the minimum pacing-group size. The first PIU of the pacing group contains a VR-pacing request. Except for the current pacing group, no more pacing groups can be transmitted until the subarea node that is receiving the PIUs sends a VR-pacing response.

As a subarea node transmits PIUs, the VR-pacing count is decreased by 1 for each PIU transmitted; the subarea node can continue to send PIUs until the VR-pacing count reaches 0. At that point, the sending node can transmit no more PIUs until it receives a VR-pacing response.

Upon receiving a VR-pacing response, a subarea node increases the VR-pacing count by the value calculated to be the pacing-group size for the next pacing group to be sent. The subarea node then sends any unsent PIUs in the current pacing group. Finally, it sets a bit for the first PIU of the new pacing group, indicating that it is a VR-pacing request, and proceeds to transmit the new pacing group.

Path control in a sending subarea node dynamically adjusts a virtual route's current pacing-group size based upon the level of congestion



present on the explicit route that underlies the virtual route. Either an intermediate routing node or the receiving end node may become congested.

### Effect of Severe Congestion

Upon detecting severe congestion along one direction of an explicit route, the path control component of an intermediate routing node sets the Reset Window Indicator (RWI) bit in the FID4 transmission header of the next PIU flowing in the opposite direction along each virtual route that uses the explicit route. Upon receiving a PIU with RWI set over a virtual route, path control at the end subarea for the virtual route tries to alleviate the congestion by immediately (1) truncating the current pacing group and (2) adjusting the pacing-group size for the route to its minimum value.

If the path control component of an end node on an explicit route becomes severely congested because it is receiving too many PIUs over the explicit route, this component can take two actions. (1) Path control can use the RWI bit to cause the sending nodes of congested virtual routes to reduce their pacing-group sizes to the minimum. (2) Path control can withhold VR-pacing responses for the congested virtual routes, thus causing no more PIUs to be sent along these routes after the current pacing groups have been sent.

If it does not receive VR-pacing responses, the sending subarea node must avoid severely depleting its buffers by accepting too many PIUs that need to be sent over the congested virtual routes. The SNA access methods and ACF/NCP allow the system programmer to specify parameters that help limit buffer depletion.

### Effect of Moderate Congestion

If the path control component of an intermediate routing node detects moderate congestion along an explicit route, it sets the Change Window Indicator (CWI) bit in the FID4 transmission header of one PIU flowing in the direction in which the virtual route is congested. Upon detecting that the CWI bit in the TH is set, the path control component in the receiving end node of the virtual route sets the Change Window Reply Indicator (CWRI) bit in the TH of the next VR-pacing response. Upon receiving this response, path control decrements its pacing-group size by 1.

If path control in a sending subarea node receives a VR-pacing response whose TH indicates no congestion along the underlying explicit route, and if all PIUs in the current pacing group have been sent, then path control increases by 1 the pacing-group size for the next pacing group to be sent. If all PIUs in the current pacing group have not yet been sent, then path control does not change the pacing-group size.

If all PIUs in the current pacing group have been sent, then the virtual route is not being fully utilized; path control therefore tries to improve the utilization by increasing the pacing-group size by 1.

If all PIUs in the current pacing group have not been sent, then the current pacing-group size is sufficiently large to allow PIU traffic to be sent on the virtual route without pacing delays, so the pacing-group size is not changed. ("Pacing delay" refers to the pacing-imposed interval between the moment when sending of one pacing group is completed and the moment when sending of the next pacing group begins, as illustrated in Figure 4-12 on page 4-25. Ideally, this pacing delay should be zero.)

The path control component that adjusts the pacing-group size for virtual-route pacing never raises it above the maximum or lowers it below the minimum pacing-group size specified for the virtual route.

Figure 4-10 on page 4-23 is a flowchart that shows the pacing-group size algorithm used by the path control component of the sending end node of a virtual route.

Figure 4-11 on page 4-24 shows how moderate and severe congestion cause an intermediate routing node on a virtual route to set indicators that cause the end nodes of the route to take corrective action.

Figure 4-12 on page 4-25 shows how dynamic adjustment of the pacing-group size results in eliminating a VR-pacing delay, thereby increasing the utilization of the explicit route that underlies the virtual route that is experiencing the delay.

## Error Handling for a Route

Data flowing along an explicit route in a network passes through a series of network elements, all of which must be operational. Explicit routes are subject to disruption when any of their elements fail or are deactivated. When either occurs, all sessions that use virtual routes associated with the affected explicit route are disrupted, and the data flows associated with those sessions stop. Examples of network elements of a path that can fail and thereby cause session disruption are: a host access method, a host LU, an NCP, a transmission group, a link connection to a link station, or a link station.

When a transmission group fails, the network notifies network operators at hosts on both sides of the failure. The notification identifies which transmission group has failed and which subareas, explicit routes, and virtual routes have been lost to the host that receives the notification. The notification thus identifies the location of the failure so that corrective action can be taken.

Many elements of an SNA network, such as SSCPs in host access methods and NCPs in NCP nodes, contain information about the current status of active sessions. When a failure in the network disrupts a session, it is important that this changed session status be communicated to the

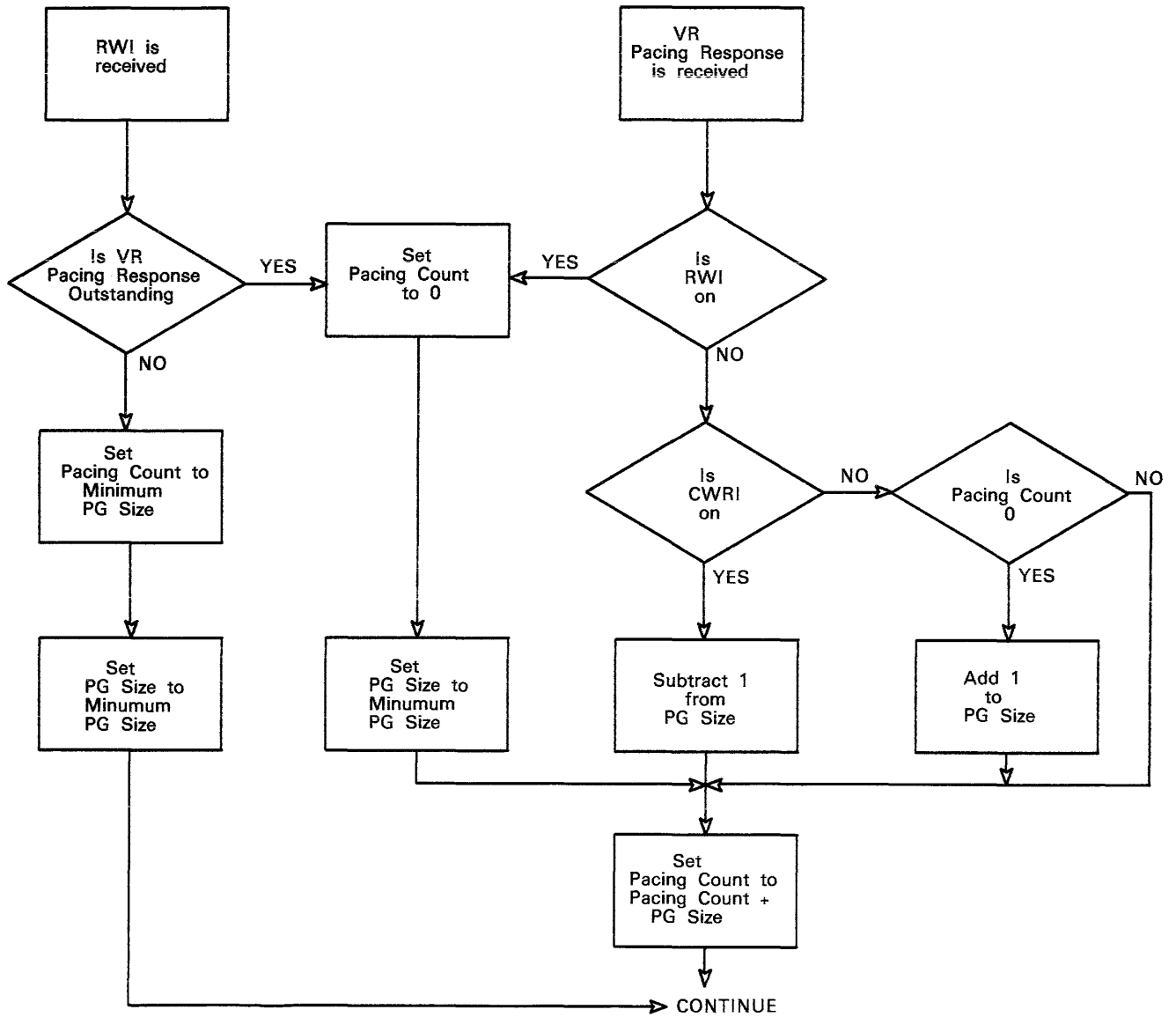
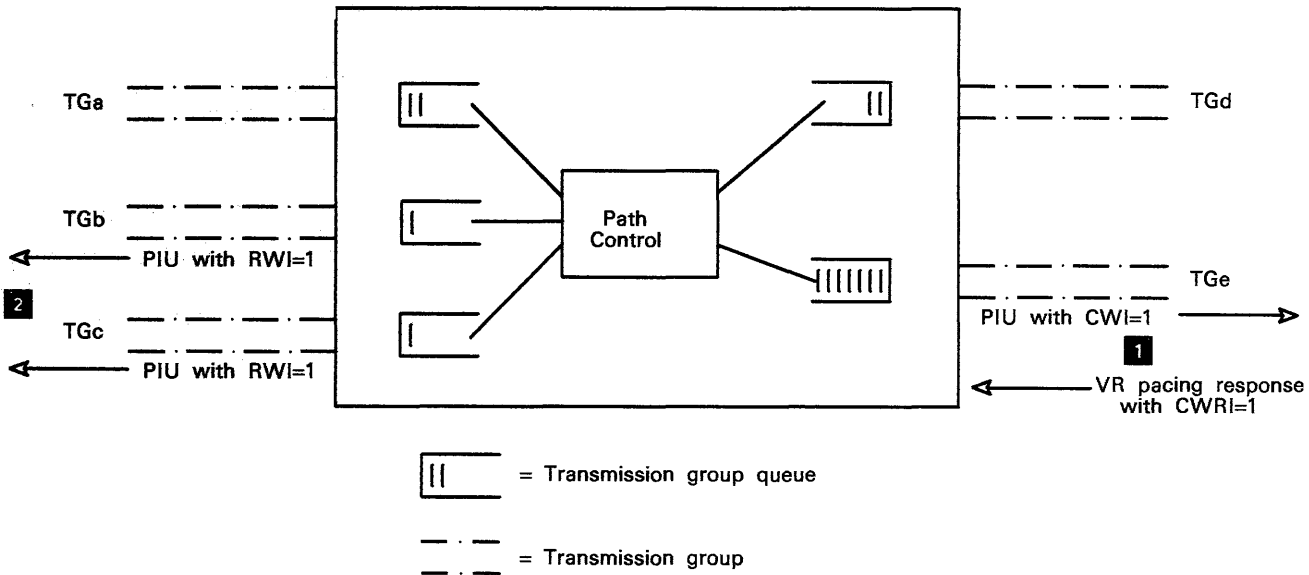


Figure 4-10. Pacing-Group Size Adjustment Algorithm for Path Control at Sending End Node of a Virtual Route

session partners and (in the case of LU-LU sessions) to the SSCPs of the session partners.



Indicators:

- CWI - Ask VR partner to decrement sending pacing group size by 1.
- CWRI - Receiver of CWRI should decrement the size of the pacing group it sends by 1.
- RWI - Receiver of RWI should reset its sending pacing group sizes and pacing group counts to the minimum.

Explicit Routes:

- ER1 = TGa + TGd
- ER2 = TGb + TGe
- ER3 = TGc + TGe

1. If a PIU encounters moderate congestion at TGe, path control turns on the Change Window Indicator (CWI) in the header of that PIU before forwarding it. The next VR pacing response returned on the virtual route of that PIU indicates, by setting the Change Window Reply Indicator (CWRI), that the sender of the original PIU should decrement the pacing group size by 1.
2. If severe congestion exists at TGe, path control turns on the Reset Window Indicator (RWI) in any PIUs flowing in the reverse direction over explicit routes that are sources of PIUs for TGe. In the example above, virtual routes using ER2 and ER3 would be notified to truncate their current pacing groups and reset their pacing group sizes and pacing counts to the minimum values in the direction of TGe.

Figure 4-11. Actions by an Intermediate Routing Node to Alleviate Congestion

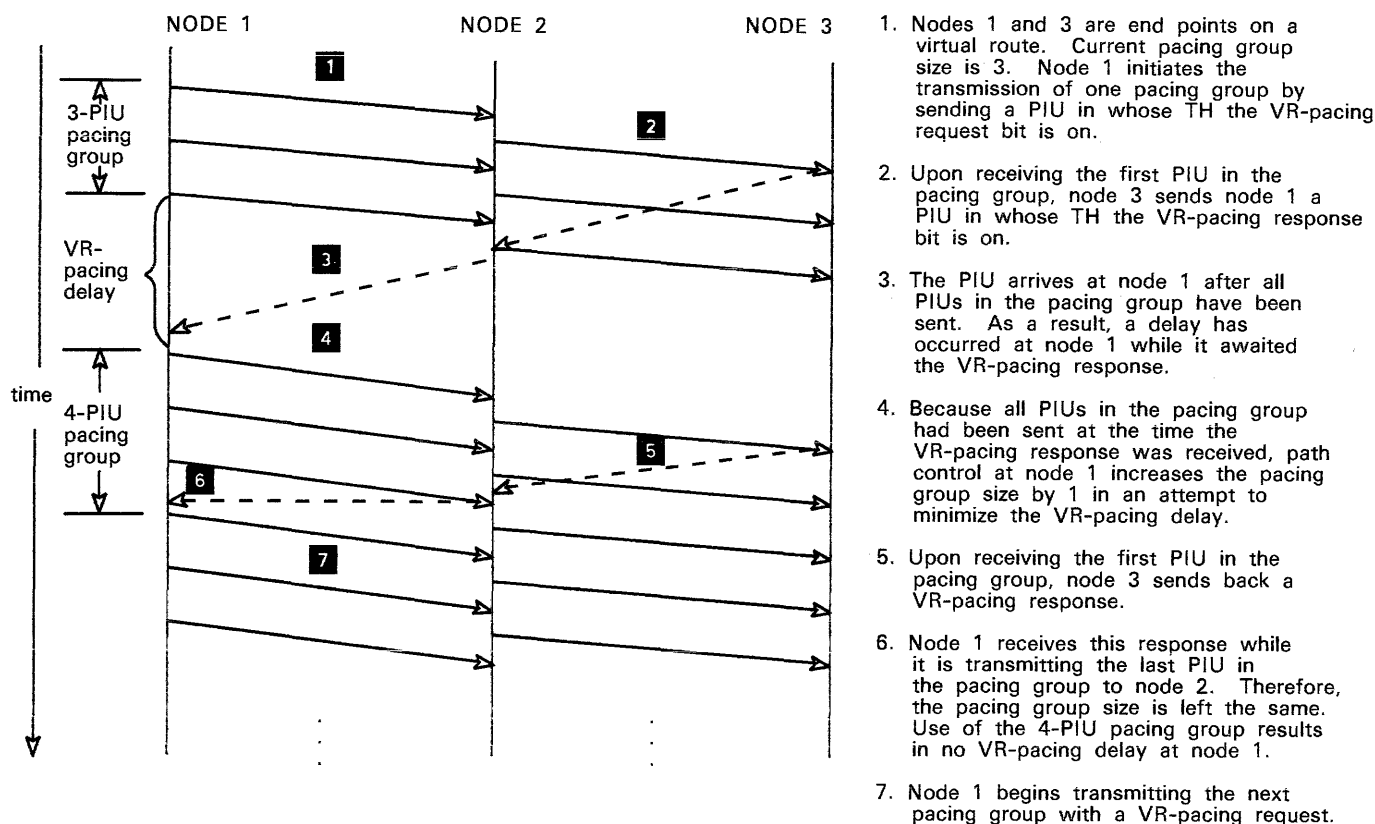


Figure 4-12. Elimination of Virtual-Route Pacing Delay by Dynamic Pacing-Group Size Adjustment

When a failure or deactivation of a virtual route disrupts any SNA session, NCPs and access methods on either side of the disruption notify the session partners and all other affected SNA functions that the disruption occurred. This notification lets the sessions be reinitiated over a different route, if one is available, or over the original route after that route is repaired and reactivated.

Once notified that their sessions are disrupted because the virtual route failed, the affected SSCPs and LUs can try to reinitiate those sessions. ACF/TCAM provides a facility for automatically trying to reinitiate such sessions. The network designer can specify that ACF/TCAM try to automatically reinitiate disrupted sessions over a

different virtual route, if one is available, having the same class of service. Affected LUs in peripheral nodes can also try to reinitiate disrupted LU-LU sessions over an available virtual route having the same or a different class of service.

If no other virtual route is available having the same or a different class of service, disrupted sessions cannot be reinitiated until a route becomes available. ACF/TCAM provides a virtual-route-availability monitoring function that allows a host NAU to be notified automatically when a virtual route contained in the requested class of service becomes available. When the NAU is notified, it automatically retries the session-initiation request.

"Session reinitiation" means that a new session is established between the same pair of LUs; it does not mean that the disrupted session is reactivated at the point of failure. ACF/TCAM does not automatically resynchronize sessions as part of its automatic reinitiation process. Because failure of an explicit route may cause some user data to be lost, the LUs participating in each disrupted LU-LU session must perform resynchronization processing to determine whether data loss occurred and to react accordingly.

One kind of failure in an explicit route—an SDLC link failure—need not cause sessions to be disrupted. Because of the way data is queued for a transmission group, the failure or deactivation of a link in a multiple-link transmission group causes the data for all sessions using virtual routes associated with the explicit route to be sent over the remaining active links in the transmission group. Thus, unless the failing link was the only operational link in the transmission group, no sessions are interrupted and no data is lost.

## BENEFITS OF THE SNA ROUTING TECHNIQUES

This section describes the benefits of the SNA techniques for routing data and relates SNA routing facilities to the routing objectives described above.

SNA does not require that an entire path be defined in each node along a path. Rather, SNA distributes path definition over all of the nodes so that each node contains only a part of the path definition. This technique saves storage space in the individual nodes.

By assigning sessions to a virtual route during session activation, SNA networks avoid the inflexibility that would occur if a session always had to use the same route. They also avoid the processing overhead associated with routing schemes that use load-adaptive techniques during an active session to determine the route for a message as it traverses the network.

SNA provides several means to expedite the flow of message traffic over a path. (1) By allowing a network designer to associate transmission priorities with virtual routes, SNA lets messages flowing in certain sessions be expedited. Messages on sessions assigned to higher-priority

virtual routes can flow faster than messages assigned to lower-priority virtual routes. (2) The ability of SNA networks to have multiple SDLC links in a transmission group and multiple explicit routes between subareas of the network can help minimize the time messages spend on network paths. (3) The ability of SNA networks to detect and control congestion along virtual routes is another way to help minimize the time required to transmit messages.

The SNA data link control procedures and the virtual-route congestion control techniques are two means to promote efficient utilization of path components and thereby help minimize the cost of these components.

The ability of SNA networks to have multiple SDLC links in a transmission group and multiple explicit routes between subareas can help increase the data-handling capacity and the availability of network paths. Multiple links increase the data-carrying capacity between adjacent nodes; multiple explicit routes allow one explicit route to be substituted for another explicit route that fails. SNA networks can also notify session partners when route failures disrupt their sessions and let them try to reinitiate disrupted sessions over different routes.

The SNA data link control techniques for detecting and, through retransmission, correcting errors as they occur helps minimize the loss of data caused by physical errors occurring in path components. SNA also provides network management techniques to identify error-prone components of a path by collecting and displaying error statistics for such components.

Finally, the ability of SNA networks to notify network operators when explicit routes fail, to notify session partners when route failure disrupts their sessions, and to let session partners reinitiate disrupted sessions over different explicit routes, can help minimize the time needed to detect and reactivate an inoperable path.

## SPECIFYING ROUTES

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

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

These are defined to ACF/NCP, ACF/VTAM, and ACF/TCAM as follows.

Links between adjacent subarea nodes are defined as explained in Chapter 3, "Transmitting Data from Node to Node."

Transmission groups are defined to ACF/NCP.

Explicit routes are defined to ACF/NCP and ACF/TCAM. ACF/TCAM and ACF/VTAM provide statements that allow the system programmer to resolve potential routing conflicts.

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

To define explicit routes among subareas using the facilities described below can be tedious for large networks. IBM provides a field-developed program (FDP) 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. Information about this program can be found in Routing Table Generator Program Description/Operations Manual, SB21-2806.

### 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 provided by PATH macros, each NCP builds an explicit-routing table similar to that shown in Figure 4-8 on page 4-15.

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, which are similar in format to ACF/NCP PATH macros. Besides 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 may 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 by 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.

## Specifying Routes to ACF/TCAM

ACF/TCAM builds its explicit-routing table based upon information received from channel-attached NCPs via Explicit Route Operative (ER-OP) requests sent when explicit routes are activated (as shown in Figure 2-13). However, ACF/TCAM also provides PATH macros that are used to resolve routing conflicts that can occur when an ACF/TCAM host node receives notification (via Explicit Route Operative requests) that more than one of its channel-attached NCPs are part of the same explicit route to the same destination subarea. The ACF/TCAM PATH macros, which are similar in format to ACF/NCP PATH macros, are assembled into a module that is loaded when ACF/TCAM is initialized and is used to resolve routing conflicts.

ACF/TCAM provides a VRN operand on its PATH macro to map virtual routes to explicit routes. If this operand is not coded, ACF/TCAM maps a virtual route to an explicit route on an identity basis; that is, VR1 maps to ER1, VR2 maps to ER2, and so on.

ACF/TCAM can calculate minimum and maximum pacing-group sizes for virtual-route pacing, based upon the number of transmission groups in an

explicit route. The system programmer can provide a different module for this purpose and point to it via the WSZEXIT operand of the INTRO macro.

ACF/TCAM 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/TCAM provides a default class-of-service table.

ACF/TCAM lets a system programmer select a class of service for an SNA session in either of two ways. The programmer can:

- Code the COS operand of the IEDBENT macro that defines a Bind image table entry; or
- Initialize an option field created by an OPTION macro that specifies USE=COSNAME. An Unbind user-exit associated with the device message handler (DMH) of a primary LU may be used to specify the automatic session reinitiation of LU-LU sessions.

A Bind user-exit associated with the device message handler (DMH) of a primary LU may be used to modify the virtual route entries in a class of service just before a virtual route is selected.

If no virtual route in the selected class of service is available, a session cannot be activated. ACF/TCAM provides a route-availability monitoring option (RAMO) that notifies the primary session partner immediately when a virtual route in the selected class of service becomes available. The system programmer uses a Bind user-exit associated with the DMH of a primary LU to specify that RAMO is effective for an LU-LU session.

## TYPICAL REQUEST-UNIT SEQUENCES FOR ROUTING

Figure 4-14 through Figure 4-16 are diagrams of typical request-unit sequences used in routing traffic through an SNA network. (Figure 4-13 on page 4-31 gives the meaning of each of the symbols and abbreviations appearing in these sequence diagrams.) Figure 4-14 on page 4-32 and Figure 4-15 on page 4-34 show how Network Control Explicit Route Operative (NC-ER-OP) requests are propagated through the network. Figure 4-14 shows how these requests flow from pairs of subarea nodes after a transmission group is activated between the nodes.

Propagated from subarea to subarea, Network Control Explicit Route Operative (NC-ER-OP) requests identify which subareas can be reached when the transmission group is activated. Upon receiving this information, a subarea node determines which explicit routes it may activate to reach other subareas in the network.

Figure 4-15 is similar to Figure 4-14, but shows how routing information is propagated when different explicit routes between the same two subareas are assigned to different transmission groups.

The propagation scheme shown in Figure 4-14 and Figure 4-15 is also used, after a transmission group is deactivated, to propagate information about loss of logical connections between subareas. Network Control Explicit Route Inoperative (NC-ER-INOP) requests are used to propagate this information.

Figure 4-16 shows how congestion indicators are used to alleviate both moderate and severe congestion at nodes along virtual routes.

In addition to the sequences presented here, certain sequences in Chapter 2 also illustrate facets of routing. Figure 2-8, Figure 2-9, Figure 2-14, and Figure 2-15 show how explicit and virtual routes are activated; Figure 2-16 and Figure 2-19 show how virtual routes are deactivated.

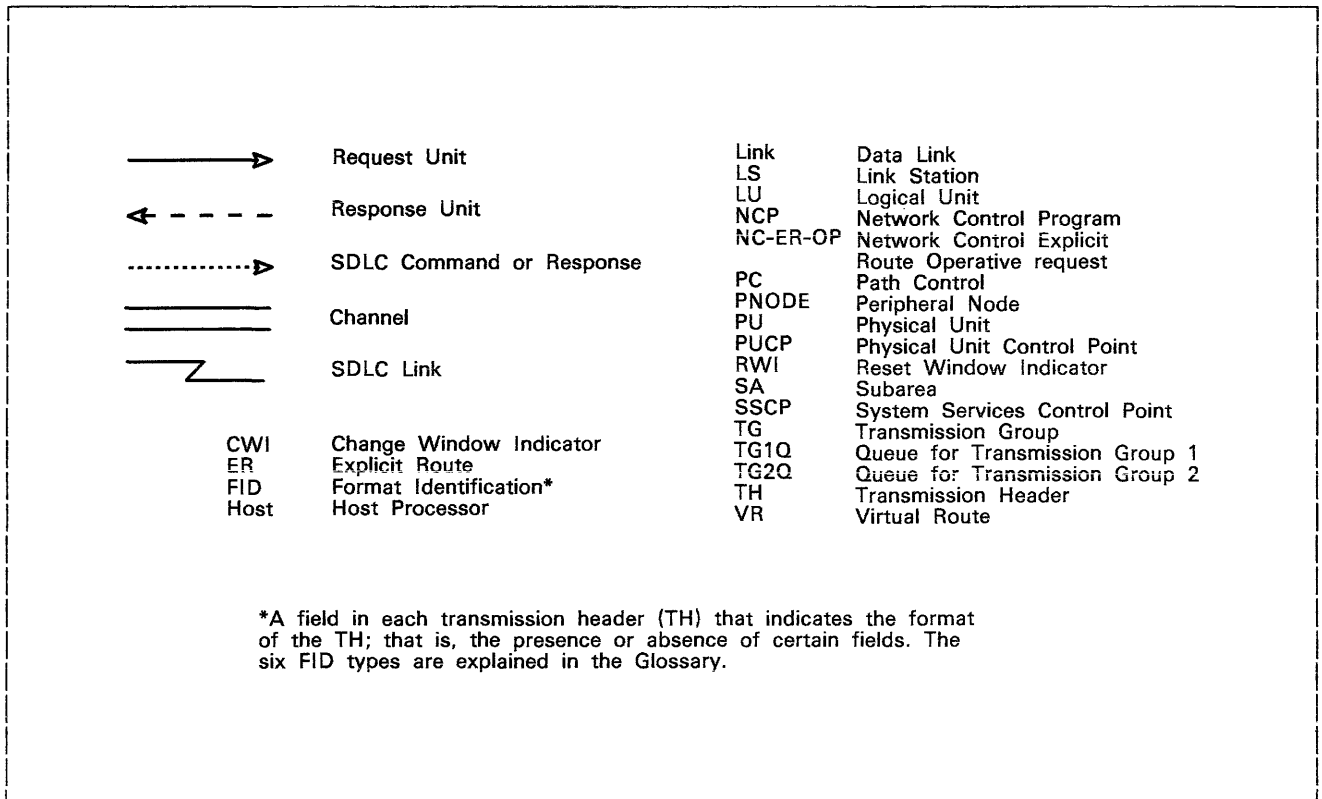
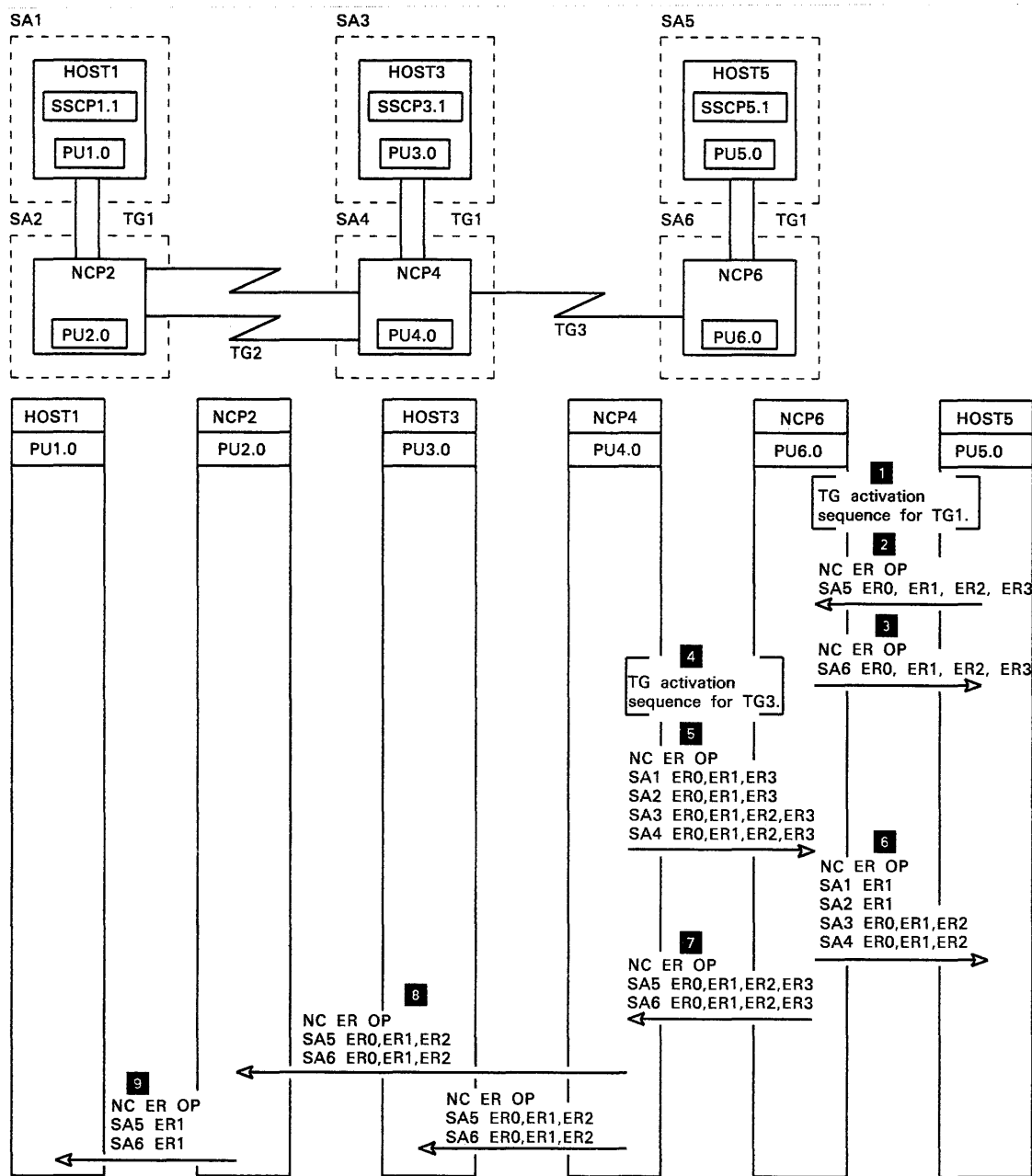


Figure 4-13. Symbols and Abbreviations Appearing in Sequence Diagrams of Chapter 4



(Figure 4-13 gives the meanings of the symbols and abbreviations that appear in this figure.)

Figure 4-14 (Part 1 of 2). Fan-out Propagation of Explicit Route Operative (NC-ER-OP) Requests

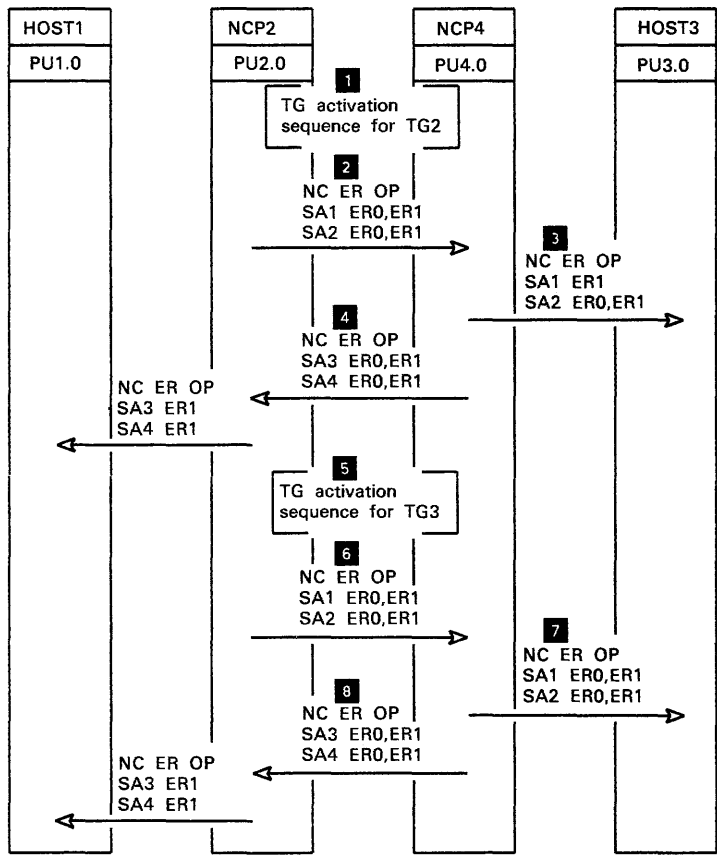
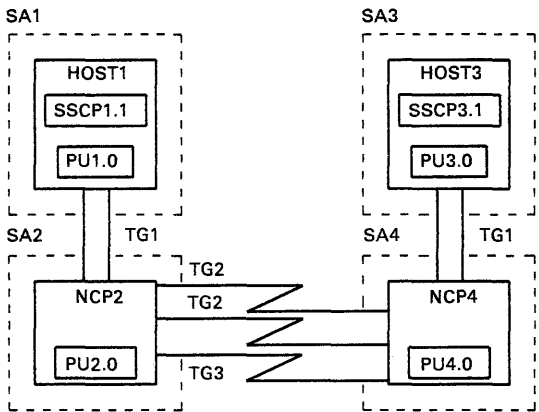
---

This sequence of request units assumes that all transmission groups are active except for TG1 between HOST5 and NCP6, and for TG3, which are inactive.

1. Transmission group TG1 is activated. See Chapter 2.
2. PU5.0 tells PU6.0 that PU5.0 can handle data routed to subarea 5 over explicit routes 0, 1, 2, and 3, once these routes are activated. PU6.0 uses this information in deciding which explicit routes to activate.
3. PU6.0 tells PU5.0 that PU6.0 can handle data routed to subarea 6 over explicit routes 0, 1, 2, and 3.
4. Transmission group TG3 is activated. See Chapter 2.
5. PU4.0 tells PU6.0 that PU4.0 can handle data routed to the following subareas over the following explicit routes once these routes are activated:
  - o Subareas 1 and 2 via explicit routes 0, 1, and 3
  - o Subareas 3 and 4 via explicit routes 0, 1, 2, and 3
6. PU6.0 tells PU5.0 that PU6.0 can handle data routed to the following subareas over the following explicit routes:
  - o Subareas 1 and 2 via explicit route 1
  - o Subareas 3 and 4 via explicit routes 0, 1, and 2
7. PU6.0 tells PU4.0 that PU6.0 can handle data routed to subareas 5 and 6 over explicit routes 0, 1, 2, and 3
8. After modifying the information from PU6.0 to reflect the information in its own routing table, PU4.0 propagates this information to the PUs in the other subarea nodes connected to it--that is, PU2.0 in NCP2 and PU3.0 in HOST3.
9. PU2.0 modifies the information it received from PU4.0 and propagates it to PU1.0.

Figure 4-14 (Part 2 of 2). Fan-out Propagation of Explicit Route Operative (NC-ER-OP) Requests

---

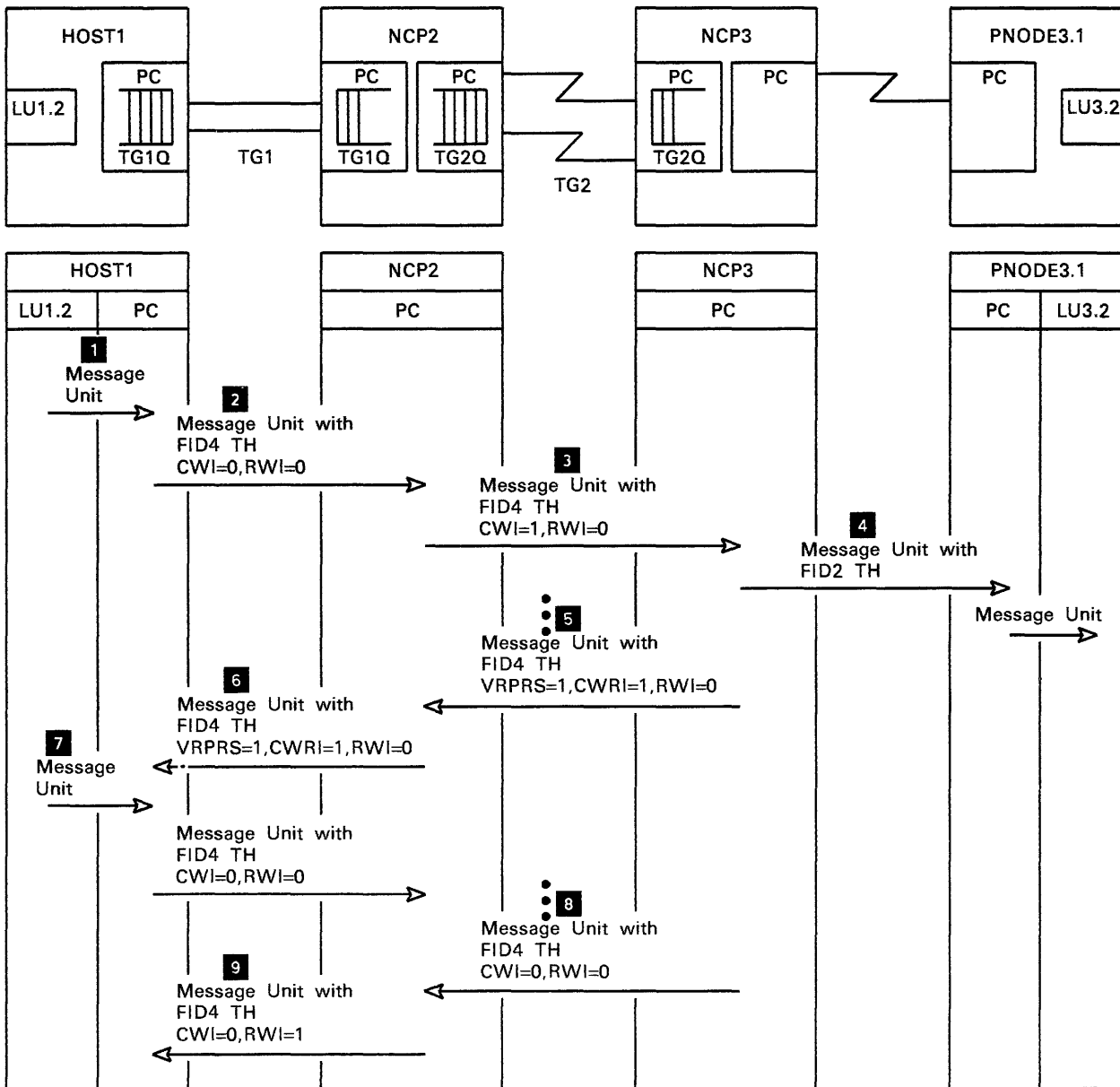


- This sequence of request units assumes that transmission groups TG2 and TG3 are active and that both TG1s are inactive.
1. TG2 is activated.
  2. PU2.0 tells PU4.0 that PU2.0 can reach subareas 1 and 2 over explicit routes 0 and 1.
  3. PU4.0 tells PU3.0 that PU4.0 can reach subarea 1 over explicit route 1 and subarea 2 over explicit routes 0 and 1. PU4.0 uses TG3 to reach subarea 1 over explicit route 0; since TG3 has not yet been activated, explicit route 0 is not yet available.
  4. PU4.0 sends the appropriate information on connectivity to PU2.0, which modifies the information to reflect the contents of its own routing tables and then propagates the information to PU1.0.
  5. TG3 is activated.
  6. PU2.0 sends PU4.0 the same information that it sent in Step 2.
  7. PU4.0 in turn propagates this information to PU3.0. Because TG3 is now active, PU4.0 tells PU3.0 that PU4.0 can now reach subarea 1 via explicit route 0, as well as via explicit route 1.
  8. PU4.0 sends PU2.0 the same information that it sent in Step 4. PU2.0 propagates this information to PU1.0.

(Figure 4-13 gives the meanings of the symbols and abbreviations that appear in this figure.)

Figure 4-15. Propagation of Routing Information following Activation of Multiple Transmission Groups between the Same Subareas

This page intentionally left blank.



(Figure 4-13 gives the meanings of the symbols and abbreviations that appear in this figure.)

Figure 4-16 (Part 1 of 2). Setting Congestion Indicators in FID4 Transmission Headers of PIUs Traversing a Virtual Route



---

These are the steps in the request unit sequences that set congestion indicators in transmission headers of PIUs traversing a virtual route.

1. LU1.2 in HOST1 generates a message unit destined for LU3.2 in PNODE3.1 and passes it to the path control component of HOST1 for processing.
2. Path control in HOST1 adds a FID4 TH to the message unit and places it on the transmission queue for transmission to NCP2 via TG1.
3. Path control in NCP2 places the message unit on the transmission queue for transmission to NCP3 via TG2.
4. Path control in NCP3 converts the TH to a FID2 TH and sends the message unit to path control in PNODE3.1, which in turn routes it to its destination, LU3.2.
5. When it prepares the next VR-pacing response (VRPRS) for the congested virtual route between HOST1 and NCP3, path control in NCP3 turns on the change window reply indicator (CWRI) bit in the FID4 TH and sends the pacing response to NCP2.
6. Path control in NCP2 routes a message unit containing the VR-pacing response (VRPRS) to path control in HOST1. Upon detecting that the CWRI bit is on, path control in HOST1 decrements by 1 the pacing group size for the next pacing group of message units to be sent over the virtual route to NCP3, in an attempt to lessen congestion along this route.
7. At some later time, LU1.2 generates another message unit for LU3.2, and HOST1 path control routes this message unit to NCP2 over transmission group TG1. When it enqueues the message unit for transmission to NCP3 over TG2, path control in NCP2 notices that TG2 has become severely congested.
8. After this congestion has been detected, a message unit flows from NCP3 to NCP2 on the virtual route between NCP3 and HOST1.
9. Path control in NCP2 turns on the reset window indicator (RWI) bit in the TH of the message unit and routes the message unit to HOST1. The RWI bit indicates that severe congestion has been detected along the virtual route in the direction of HOST1. Upon receiving the message unit with the RWI bit on in its TH, path control in HOST1 immediately truncates the current pacing group of message units flowing on the virtual route to NCP3, and resets the pacing group size to the minimum. (Message units that were in the old pacing group but have not yet been sent are sent in the new pacing group.)

Figure 4-16 (Part 2 of 2). Setting Congestion Indicators in FID4 Transmission Headers of PIUs Traversing a Virtual Route

---



## CHAPTER 5. USING LU-LU SESSIONS TO TRANSMIT DATA BETWEEN END USERS

Before an end user of an SNA network can communicate with any other end user, their respective logical units must be connected in a mutual relationship called a session. Because the session joins two logical units, it is called an LU-LU session. The term session partners is often applied to logical units engaged in an active session.

An LU-LU session synchronizes the state of the interaction between the end users. The session is a temporary relationship, or connection, between logical units that lets the end users exchange data. Activating a session between logical units makes available the appropriate resources, such as buffer storage and processor capacity, for as long as the session is active.

A single LU can represent one or more end users and can participate in several LU-LU sessions at the same time with one or more other LUs. The component within an LU that provides the set of resources used by a particular LU-LU session is called a half-session.

The exchange of data by end users is subject to a number of procedural rules, or protocols, that the logical units specify before beginning the exchange. These protocols represent an agreement between the end users about how the session is to be conducted, where alternatives exist. The protocols specify such things as the format of the data, the amount of data to be sent by one end user before the other one replies, and the action to be taken if errors occur. While the session is being activated, the session partners negotiate and agree upon the protocols that they will follow in conducting the session.

This chapter introduces some of the protocols that LU-LU sessions use and names the SNA request units associated with each protocol.

### TYPES OF LOGICAL UNITS

IBM classifies SNA products according to LU type.<sup>1</sup> An LU type constitutes a particular set of SNA functions provided for communication between end users that a product can perform. The product can perform those functions if it supports that type of LU. Some of these SNA functions are mandatory; others are optional.

In general, products that support a particular type of LU are used for a particular kind of communication. One type of LU is typically used for communication between an application program and an IBM 6670 Information Distributor. Communication between an application program and a display terminal, on the other hand, typically uses a different type of LU at

---

<sup>1</sup> Formerly called LU-LU session type.

the ends of the session. And communication between two application programs typically uses still another type of LU.

The LU type designation is a convenient means of classifying SNA hardware and software products according to the subsets of SNA functions that their logical units can perform.

In order for two SNA products to communicate over an LU-LU session, both products must support the same LU type. Some SNA products support one LU type, others support more than one. A list of SNA products and their corresponding LU types appears in the SNA Reference Summary, GA27-3136.

## Benefits of LU Type Classification

Knowing the type (or types) of LU that a particular SNA product supports can reveal much about the kinds of applications for which that product can be used and the kinds of SNA functions the product can perform. Knowing the LU type is therefore useful to those who need to select the appropriate SNA products for a given application.

Knowing the LU type is also of value to those who must diagnose problems in the network; because the LU type identifies the specific functions that an LU may perform, one can concentrate on problems associated with those functions.

## ACTIVATING AN LU-LU SESSION

The process of activating an LU-LU session is diagrammed in Figure 5-1 on page 5-3. First, an LU sends a session-initiation request to its SSCP. This request identifies the two LUs between which a session is to be activated. The requesting LU sends an Initiate Other (INIT-OTHER) or Initiate Self (INIT-SELF) request to the SSCP. The INIT-SELF and INIT-OTHER requests are called session-initiation requests.

The SSCP performs several functions, in a category called session services, that help in the session-initiation process. Among other functions, the SSCP:

- Verifies the authority of the initiating LU to initiate a session between the specified LUs
- Resolves network names<sup>2</sup> to network addresses
- Queues and dequeues session-initiation requests (these requests can be queued in the event that the required LU-LU session cannot be activated immediately)
- Selects appropriate session parameters to be used by the LUs when their session is active
- Synchronizes the initiation process

---

<sup>2</sup> A network name is the symbolic identifier by which end users refer to a network addressable unit, a link station, or a link.

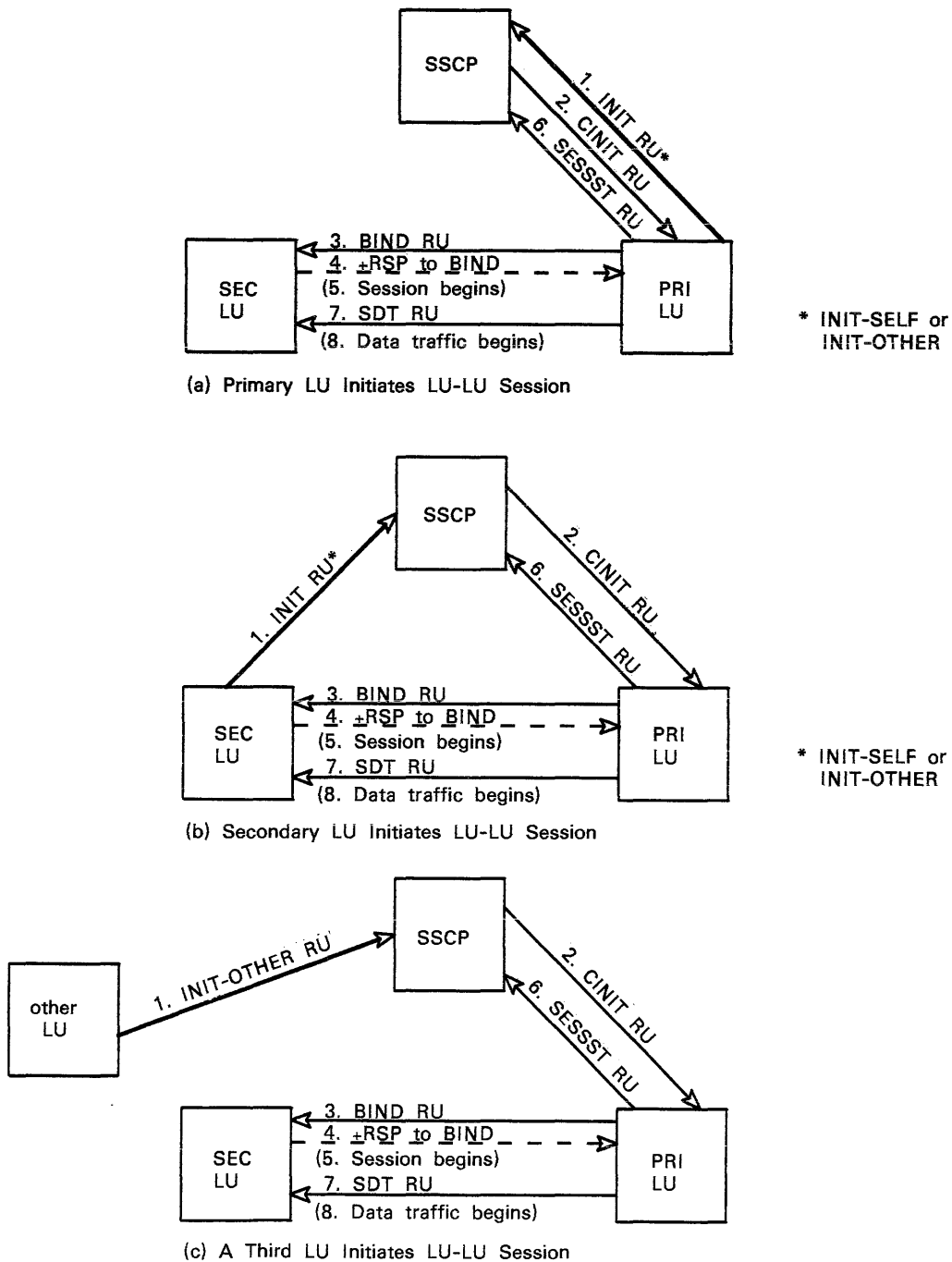


Figure 5-1. Starting an LU-LU Session

Session services in the SSCP are complemented by session services in each LU. The session-initiation requests (INIT-SELF and INIT-OTHER) are carried on SSCP-LU sessions.

In the session-initiation request, the end user can specify a particular class of service to be used for the LU-LU session. The class of service may be specified directly, by a class-of-service name (COS name), or indirectly, by a mode name. In the latter case, the SSCP for the secondary LU determines from the mode name a default COS name.

The SSCP of the primary LU resolves the COS name to a specific list called a virtual route identifier list. During session activation, the session is assigned to the first available virtual route in the list. The SSCP uses the specified mode name, along with optional installation-defined parameters, to select the set of rules and protocols to be used for the session. The SSCP selects this set, called a Bind image, from a group of such sets within a Bind image table. The selected Bind image is the content of the Bind Session (BIND) request that will be used to activate the LU-LU session. The SSCP then transmits the Bind image within a Control Initiate (CINIT) request to the LU that is to be the primary LU of the session.

The session activation process then follows, beginning when the primary LU sends the Bind Session (BIND) request to the LU that is to be the secondary LU in the session. The Bind Session request specifies, as parameters, the protocols and rules that the primary and secondary LUs are to observe in conducting their session. These protocols are grouped in several categories that correspond to the SNA layers responsible for enforcing them. Some of the principal protocols are described in the remainder of this chapter.

Upon receiving the Bind Session request, the secondary LU may return a positive response to the primary LU. Then the session begins. (The secondary LU may instead return a negative response, if it is unable to participate in a session with the primary LU.) The primary LU in turn notifies the SSCP, via a Session Started (SESSST) request, that the LU-LU session has been successfully activated. (If the activation process is unsuccessful, the primary LU sends the SSCP a Bind Failure (BINDF) request instead. This request indicates the reason for the failure.)

The preceding description of session initiation and activation applies to a same-domain LU-LU session—that is, a session in which both LUs are in the same domain. The process is similar for cross-domain sessions, except that the Control Initiate (CINIT) request may be sent by an SSCP different from the one that received the Initiate Self (INIT-SELF) or Initiate Other (INIT-OTHER) request. The latter SSCP sends a Cross-Domain Initiate (CDINIT) request to the SSCP that issues the CINIT request. The LUs involved in the session to be activated are not aware of whether their session is a same-domain or a cross-domain session.

The description above is just an overview and omits a number of steps and procedural options involved in initiating and activating LU-LU sessions. Complete details of the process, including the activation of

the prerequisite SSCP-PU and SSCP-LU sessions (and SSCP-SSCP sessions, if the LUs are in different domains), appear in the SNA Format and Protocol Reference Manual: Architectural Logic, SC30-3112.

## Negotiable and Nonnegotiable Bind Session Requests

The LU that sends the Bind Session (BIND) request is called the Bind Session sender, or primary LU (PLU), and the other LU is called the Bind Session receiver, or secondary LU (SLU). The Bind Session receiver examines the proposed protocols that the Bind Session sender specified. If the protocols are acceptable to the receiver—that is, the receiver has the ability to participate in those protocols—it accepts the Bind Session request by returning a positive response to the Bind Session sender. The LU-LU session is then activated.

The action that the Bind Session receiver takes if it cannot participate in the specified protocols depends on whether the Bind Session (BIND) request is of the negotiable or nonnegotiable kind. If the request is nonnegotiable, the receiver rejects the request by returning a negative response. For example, if the specified protocols include the exchange of compressed data, but the receiver is unable to handle such data, the receiver rejects the Bind Session request. If the Bind Session request is negotiable, on the other hand, the receiver can return a response that requests different protocols. The Bind Session sender is then responsible for determining whether it can participate using the requested alternate protocols.

## HALF-SESSIONS

Each LU-LU session consists of two ends, or half-sessions. A half-session is a component within the LU that provides the resources needed by the session. An LU can activate as many half-sessions as it has the resources to support. When the session partner receives the request to activate the session, it too allocates the resources necessary for it to participate in the session. If it cannot support the session requested, it rejects the activation request and the session does not begin. Each LU has a session limit (enforced by its SSCP) that specifies how many LU-LU sessions it can participate in at the same time. If the LU is already engaged in its maximum number of sessions, it cannot engage in any additional ones.

### Half-Session Components

Each half-session consists of the following components:

- Transmission control
- Data flow control
- Function management data (FMD) services

Among its functions, transmission control:

- Activates and deactivates the data traffic that flows between the half-sessions
- Checks sequence numbers within request and response units<sup>3</sup>.
- Paces data traffic that flows between the half-sessions
- Enciphers and deciphers, when necessary, the data flow between the half-sessions
- Enforces the maximum request unit size for traffic that passes between the half-session and the path control network

Among its functions, data flow control:

- Controls the concurrency of data flows between the half-sessions—that is, one way at a time or in both directions concurrently
- Chains request units together, at the end user's option
- Delimits chains of request units into larger groupings called brackets
- Controls the interlocking of requests and responses in accordance with the request and response control modes selected during session activation
- Assigns sequence numbers to normal-flow requests
- Correlates requests and their responses
- Interrupts the flow of data between half-sessions in either direction without affecting other control protocols of the session

Function management data (FMD) services protocols are divided into two categories.

The services provided for half-sessions involved in LU-LU sessions are called session presentation services. These services encode and compress data, format the data to appear on display screens, and perform other functions related to the presentation of user data.

The services provided for half-sessions involved in active sessions between an SSCP and an LU, a PU, or another SSCP are called session network services. These services allow the SSCP to monitor and control the processing and communication resources of the SNA network.

## MANAGING THE FLOW OF DATA

This section describes how the flow of data between logical units is managed through the use of headers, request units, response units, control modes, parameters, and indicators.

---

<sup>3</sup> A response unit is a message unit that acknowledges a request unit; it may contain prefix information received in a request unit. If positive, the request unit may contain additional information (such as session parameters in response to Bind Session), or if negative, contains sense data that defines the exception condition.



## Request Headers

Each request that flows between half-sessions begins with a request header (RH). This 3-byte field contains a number of control indicators for managing the flow of request units. The RH contains indicators for FMD services, data flow control, and transmission control components of the half-sessions. Examples of the indicators are:

- RU category: Indicates which category of function the request unit corresponds to: session control (SC), data flow control (DFC), or function management data (FMD). (Another category, network control (NC), is not used by LUs.)
- Sense data indicator (SDI): Indicates whether the request unit associated with the RH contains a 4-byte sense data field. Only an exception request (EXR) contains this field, which indicates the kind of condition that caused the exception condition.
- Chaining control indicators (BCI and ECI): Indicate that a sequence of contiguously transmitted requests is being grouped in a chain of RUs. Two indicators, BCI (begin chain indicator) and ECI (end chain indicator) together denote the relative position of the associated RU within a chain.

If the BCI is on (=1) and the ECI is off (=0), the RU is the first RU of the chain. If both indicators are off, the RU is the middle RU of the chain. If the BCI is off and the ECI is on, the RU is the last RU of the chain.

If both indicators are on, the RU is both the first and the last—and therefore the only—RU of the chain.

- Form-of-response requested indicators (DR1I, DR2I, ERI): Indicate one of three forms of response that a half-session, upon receiving a request, is to return to the sending half-session. These forms of response requested, represented by various combinations of three indicators (definite response 1 indicator [DR1I], definite response 2 indicator [DR2I], and exception response indicator [ERI]), are described later in this chapter.
- Pacing indicator (PI): Indicates that a request is the first request in a pacing group and that the sending half-session can accept a pacing response.
- Bracket control indicators (BBI, EBI): Indicate the beginning or end of a group of RU chains (brackets) exchanged between a pair of half-sessions, as described later in this chapter. BBI is the begin-bracket indicator; EBI is the end-bracket indicator.
- Change direction indicator (CDI): indicates that the sending half-session is to become the receiving half-session, and vice versa, thus reversing the direction of data flow between them. A half-session that is communicating with another half-session in

half-duplex mode can set this indicator on only in the RH for the last (or only) RU in a chain of RUs.

- Enciphered data indicator (EDI): Indicates that data in the associated RU is enciphered under session-level cryptography (described later in this chapter under "Improving Data Security through Cryptography").

## Response Headers

Each response that flows between half-sessions begins with a response header (RH). Like the request header, the response header is a 3-byte field that contains a number of control indicators for FMD services, data flow control, and transmission control. (A request header is distinguished from a response header by the setting of a single bit, called the request/response indicator, in the header.) Examples of the indicators are:

- RU category: Indicates which category of function the response corresponds to; the RU category specified by a response header is the same as that specified by the corresponding request header.
- Sense data indicator (SDI): Indicates whether the response unit includes a 4-byte sense data field. A negative response always includes sense data, which indicates the kind of condition causing the negative response.
- Chaining control indicators (BCI, ECI): Always indicate that the response unit is the only RU in the chain. (The BCI and ECI indicators—explained above under "Request Headers"—are both on.)
- Definite response indicators (DR1I, DR2I): Match the setting of the DR1I and DR2I indicators in the corresponding request header.
- Response type indicator (RTI): Indicates whether the response is positive or negative.
- Pacing indicator (PI): Set in a response header that a receiving half-session sends to signal the sending half-session that it may transmit to the receiving half-session another group of request units on the normal flow. (Pacing is described later in this chapter under "Pacing of Data Flow at the Session Level.")

## Normal and Expedited Flows

The exchange of request units during a session is divided into normal and expedited flows in each direction between the half-sessions. Certain request units are designated as normal-flow RUs, and others are designated as expedited-flow RUs. Those designated as normal-flow RUs never flow on the expedited flow, and vice versa. (Isolated pacing responses<sup>4</sup> which are FMD-RUs, can be sent on either the normal flow or the expedited flow.)

In each direction of flow between a pair of half-sessions, the normal-flow RUs and the expedited-flow RUs are independently sequence numbered or identified, and the normal and expedited flows are controlled under separate protocols. Although the two flows are separate, the control of these flows is coupled such that requests carried on the expedited flow can change the state of the normal flow—as, for example, by resetting sequence numbers on the normal flow or quiescing the normal-flow data traffic.

Within half-sessions and within the boundary function, expedited-flow RUs bypass the transmission-control queues occupied by normal-flow RUs; the expedited-flow RUs may thus pass normal-flow requests in the queues.

The provision for normal and expedited flows within the SNA network allows various useful session-level flow-control protocols to be imposed on end-user data traffic (which is carried on normal flows) without blocking the passage of crucial control traffic (which is carried on expedited flows).

Use of the normal and expedited flows within half-sessions varies by RU category (described below), as follows:

- Function management data RUs (FMD-RUs) are sent only on the normal flow.
- Data flow control RUs (DFC-RUs) are sent on either the normal flow or the expedited flow, depending on the particular RU.
- Session control RUs (SC-RUs) are sent only on the expedited flow.

A fourth RU category, network control RUs (NC-RUs), applies only to PU-PU flows. These are direct flows between physical units in the network, and do not involve sessions. PU-PU flows, which transfer only network control RUs, use only the expedited flow.

## Request Units

All request units are in one of three categories, as specified in the RU category field of the request header associated with the request unit: FMD (function management data), data flow control, or session control.

### FMD Request Units

FMD request units, abbreviated FMD-RUs, are divided into two major categories: those that carry user data and those that carry data related to network services. FMD-RUs that carry user data may contain function management (FM) headers. These headers are described later in this chapter under "Using FM Headers to Control LU Activity." FMD-RUs for

---

<sup>4</sup> Described later in this chapter under "Pacing of Data Flow at the Session Level."

network services contain network services (NS) headers if the format indicator of the associated RH specifies that the FMD-RU is a field-formatted request. (Such a request is encoded into fields, each having a specified format such as binary codes, binary counts, bit-significant flags, and symbolic names; a format indicator in the request/response header (RH) for the request is set to 1.)

### Data Flow Control and Session Control Request Units

With one exception, all data flow control RUs, abbreviated DFC-RUs, flow on LU-LU sessions. The one exception is the Logical Unit Status (LUSTAT) RU, which can flow either on LU-LU sessions or SSCP-LU sessions; in the latter case, it flows from the LU to the SSCP.

Session control RUs, abbreviated SC-RUs, are all expedited-flow RUs. They flow on SSCP-SSCP, SSCP-PU, SSCP-LU, and LU-LU sessions.

### **Grouping Request Units into RU Chains**

Individual request units may be grouped into a sequence of request units called an RU chain. Chaining of RUs provides the means for sending and receiving a sequence of requests as one entity for error-recovery purposes. The receiver returns to the sender one response for the entire chain, or no response at all. When an error is detected in some request within the chain, that request and any following ones in the chain are purged.

A chain consists of one or more request units or a single response unit. Each chain has the following properties:

- All the requests are transferred over the normal flow, or all are transferred over the expedited flow.
- All requests flow in the same direction.
- The first request is marked (in its request header [RH]) as "first in chain" (FIC).
- The last request is marked as "last in chain" (LIC).
- All requests that are between the first and last requests are marked as "middle in chain" (MIC).
- If a chain contains a single request, that request is marked as "only in chain" (OIC).

The Begin Chain (BC) and End Chain (EC) indicators are used to mark the requests as the first, last, middle, or only request in the chain.

Each RU chain is of one of these types:

- Definite-response chain: The last request in the chain is marked definite-response; all other requests in the chain are marked exception-response.
- Exception-response chain: Every request in the chain is marked exception-response.
- No-response chain: Every request in the chain is marked no-response.

The receiver of a definite-response chain of RUs always responds by returning to the sender either a positive response or a negative response. The receiver returns a positive response only at the end of the chain, that is, following the last RU of the chain. (Positive responses are never sent for any other RU in the chain.) The receiver returns a negative response after receiving the RU in error. Thus, a negative response could be returned after the first RU, an intermediate RU, or the last RU in the chain. The RH of the last RU in the chain specifies definite response; all other RHs specify exception response.)

The receiver of an exception-response chain does not return a positive response. The receiver returns only negative responses (indicating exception conditions). As in the case of a definite-response chain, a negative response could be returned after the first RU, an intermediate RU, or the last RU in a chain.

The receiver of a no-response chain returns neither a positive nor a negative response.

Only normal-flow requests can be grouped into chains of multiple RUs; expedited-flow requests and all responses flow as single-RU chains.

### Canceling an RU Chain During Transmission

A sender of a multiple-RU chain may need to cancel the chain while transmitting it. The sender can do so by sending a Cancel RU to the receiver of the chain. The Cancel RU notifies the receiver that the chain has ended and that the receiver should discard any RUs in the chain it has already received.

### **Response Units**

For each request that flows in the network there may or may not be a corresponding response. The sender of a request unit specifies, in the request header, whether and under what conditions the receiver of the request unit is to return a response. A response unit indicates to the sender whether the response is positive or negative.

A positive response acknowledges that the request was successfully received and processed. Positive responses for some kinds of RUs also provide certain information that the sender of the RU needs. For example, the positive response to an Activate Logical Unit (ACTLU) RU returns to the sender parameters that indicate the session capabilities of the logical unit that is being activated.

A negative response acknowledges that the corresponding request was received, and includes information that identifies how the request was in error. Each negative response includes 4 bytes of sense data that indicate in detail the kind of error, plus up to 3 bytes of the original request. The sense data includes a category value that defines one of several categories of error, a modifier value that indicates a specific error, and user-defined data or sense-code-specific fields. The

categories of errors and examples of specific errors within each category are as follows:

- Request Reject Error: resource not available, intervention required, missing password, end user not authorized, link inactive, request not executable, invalid session parameters, LU busy, explicit route not in a valid state.
- Request Error: RU data error, RU length error, function not supported, invalid FM header.
- State Error: sequence number error, data traffic quiesced, session control protocol violation, response owed before sending request.
- RH Usage Error: Incomplete RH, pacing not supported, RU category incorrectly specified.
- Path Error: Link failure, unrecognized destination address, LU not active, explicit route inoperative or undefined, incomplete transmission header.

## Specifying Maximum Request Unit Size

The maximum size of each request unit that half-sessions exchange is specified by the RU-size parameter of the Bind Session (BIND) request that activates the session. The reason for limiting the size is that some logical units are in SNA nodes that have limited storage available for buffers; a restriction on buffer size is also a restriction on RU length.

To send a message (user data) that is longer than the maximum RU size specified in the Bind Session request, an LU may divide the information into several RUs. These RUs, which contain related data, can be grouped into RU chains.

## Request and Response Control Modes

Request and response control modes determine the relationship between the sending of a request and the returning of a response, as follows.

In immediate-request mode, a half-session can send no RU chain on a given flow (normal or expedited) as long as a previously sent RU chain that requires a definite response is outstanding on that flow (that is, the receiver of that chain has not yet returned the response). In delayed-request mode, the restriction imposed by immediate-request mode does not apply: a number of RU chains may be sent before a required response is received for any of them.

Immediate-request mode is used generally on the expedited flow in each direction between two half-sessions.

In immediate-response mode, the receiver of RU chains responds to them in the order it received them. That is, RU chains are received and the corresponding responses are returned on a first-in, first-out basis. In delayed-response mode, the receiver of RU chains may return its responses to their sender in an order different from the order in which it received them. (An exception is the Chase RU: Upon receiving a Chase RU, the receiver must return all required responses for previously received RU chains before returning its response to the Chase RU.)

The request and response control modes to be used on the normal flows in any session are determined by session-activation parameters on the Bind Session (BIND) RU that activates the session. The modes to be used in one direction of flow may be chosen independently of, and do not affect, the modes to be used in the other direction.

## Grouping RU Chains into Brackets

A sequence of RU chains transmitted on the normal flow may be grouped in an entity called a bracket. This can be done to associate together the RUs pertaining to a particular transaction (unit of work) and prevent other, unrelated requests and responses transmitted during the session from appearing within that transaction.

A bracket is delimited by the begin-bracket indicator (BBI) in the RH of the first request of the first chain in the bracket and the end-bracket indicator (EBI) in the RH of the first request of the last chain in the bracket.

A set of bracket protocols allows half-sessions to contend with each other in activating a bracket, and assist the half-sessions in resolving the "race" condition that can result from that contention. Parameters in the Bind Session (BIND) request that activates a session specify whether a bracket protocol is to be used within the session.

A set of rules for use of brackets determines how brackets are initiated and terminated.

If brackets are to be used within a session, the Bind Session parameters specify one of the half-sessions as first speaker and the other as bidder. The first speaker has the freedom to begin a bracket without requesting permission from the other half-session to do so. The bidder, in contrast, must request and receive permission from the first speaker to begin a bracket. The bidder may use a Bid request to ask this permission.

A positive response to Bid notifies the bidder that the first speaker will not begin a bracket, but instead will wait for the bidder to begin one. A negative response to Bid notifies the bidder that the first speaker has denied permission for the bidder to begin a bracket.

The first speaker may later send a Ready to Receive (RTR) request to grant permission for the bidder to begin a bracket. In its negative response to a Bid request, the first speaker indicates whether it will

later send an RTR request to the bidder. If it does indicate a forthcoming RTR request, the bidder can either wait for that RTR request or send the Bid request again. If the first speaker does not indicate a forthcoming RTR request, the bidder must wait for a begin-bracket indicator from the secondary or send the Bid request again if it still wishes to begin a bracket.

Instead of sending a Bid request, a half-session designated as bidder may try to begin a bracket simply by sending an RU chain in which the begin-bracket indicator is set in the RH of the first request. In this case the first speaker either grants the attempt (by returning a positive response) or rejects it (by returning a negative response). As in the case of a response to a Bid request, the first speaker indicates whether or not it will send an RTR request later.

The first speaker need not send an RTR request only in reply to a Bid request. It may send an RTR request to the bidder, without having first received a Bid request, to determine whether the bidder wishes to begin a bracket. The bidder replies with either a positive response, which indicates that it will initiate the next bracket, or a negative response, which indicates that the RTR request is not required—that is, that the bidder does not intend to begin the next bracket.

Unlike the bidder, the first speaker does not need permission to begin a bracket and therefore never sends a Bid request. Instead, any request it sends in which the begin-bracket indicator is on begins the bracket.

A number of rules govern the use of the bracket indicators (BBI and EBI); these rules are given in SNA Format and Protocol Reference Manual: Architectural Logic, SC30-3112.

Bracket termination is governed by one of two rules. Bracket termination rule 1 is conditional termination, controlled by the form of response requested (definite response, exception response, or no response) for the last RU chain in the bracket. If this chain requests a definite response, the bracket is not terminated until the sending half-session receives and processes a positive response. If it receives and processes a negative response, the bracket is continued. A negative response for any but the last request in the chain lets the sender specify whether the chain is to be terminated or continued.

To terminate the bracket, the sender sends a Cancel request in which the EBI (end-bracket indicator) is on. Or the sender can end the chain (and thus the bracket) by sending a request that specifies exception response or no response.

To continue the bracket, the sender sends a Cancel request in which the EB indicator is off, or ends the chain (while continuing the bracket) by sending a request that specifies definite response.

Bracket termination rule 2 is unconditional termination: A bracket is always terminated after processing of the last request of the chain in which the EBI indicator is on. (The indicator is in the first request of the chain.)



## Normal-Flow Send/Receive Modes

Requests pass through the network on the normal flow in one of three send/receive modes: half-duplex flip-flop, half-duplex contention, or full-duplex. The mode used is selected by session parameters in the Bind Session (BIND) request that activates the session. Application requirements determine which mode should be used; these modes are independent of, and are unaffected by, the similarly named half-duplex and full-duplex modes used on data links in the network.

### Half-Duplex Flip-Flop

In half-duplex flip-flop mode, two half-sessions alternate sending RU chains. When the half-session that is currently sending needs to permit its session partner to begin sending, it sets a change-direction indicator (CDI) in the RH of the last chain it sends. Upon detecting that the CDI in the request indicates "change direction," the receiving half-session can begin to send RU chains.

The protocol varies depending on whether the bracket protocol is also used during the session. If brackets are not used, one half-session is designated first sender and the other first receiver. The sender sends normal-flow requests and the receiver replies with responses. When the sender is finished sending requests, it sets the change-direction indicator in the last request it sends. If brackets are used, one half-session is designated bidder and the other is designated as first speaker, as explained above under "Grouping RU Chains into Brackets."

When bracket protocol is specified for a session, but neither half-session is transmitting a bracket (a condition known as "between-brackets" state), the half-sessions are both in contention state. In this state, either half-session may begin sending. If one half-session receives requests from its session partner while it is sending, a condition called "contention" has occurred. The first speaker always wins the contention condition; it can reject (with a negative response) the requests it receives. Its session partner is the contention loser; it continues to receive the requests the contention winner is sending.

When the session is not in the between-brackets state, that is, one half-session or the other is sending brackets, the half-sessions are subject to the protocol described above for half-duplex flip-flop mode when the bracket protocol is not in effect.

### Half-Duplex Contention

In half-duplex contention mode, either half-session can begin to send an RU chain. Transmission of chains continues until the half-session has no more to send. At this point the two half-sessions are once again in contention, and either can begin to send a new RU chain.

If contention occurs (one half-session begins to send requests while it is receiving from its session partner), the contention is won by the half-session that was designated contention winner in the Bind Session parameters. The contention loser queues any requests it receives while it is sending. The contention winner can either reject (with a negative response) or queue any requests it receives while it is sending. The contention winner or loser reverts to contention state after it sends or receives the last request of a chain.

### Full-Duplex

In full-duplex mode, both half-sessions can send requests at the same time; the data flows in the two directions are entirely independent. Any correlation of the two flows must occur at a level higher than the data flow control level of the half-sessions.

### **Quiescing Data Flow**

A half-session may need to interrupt its partner half-session while the partner is sending requests on the normal flow. The quiesce protocol provides a means for doing so. Only the normal flow is affected; the session partner can continue to send expedited-flow requests.

This protocol may be used for various reasons. For example, a half-session that is currently receiving from its session partner may need to end the session (via an Unbind Session [UNBIND] request) after it finishes receiving the rest of the current RU chain. Or a half-session may need to stop receiving temporarily because it has run low on some resource, such as a buffer pool, that it needs in order to continue receiving.

Either half-session in a session may interrupt its session partner. The half-session that needs to interrupt its partner sends a Quiesce at End of Chain (QEC) request. Upon receiving this request, the session partner can send no more requests except a Quiesce Complete (QC) request. The QC request notifies the other half-session that the sending half-session has stopped sending on the normal flow.

While quiesced, a half-session accepts all normal-flow requests and responds appropriately. Any normal-flow requests it needs to send in reply to requests it has received must be sent later. (Alternatively, the quiesced half-session can respond to a request requiring a reply by returning a negative response whose sense code indicates "reply not allowed.")

The half-session that caused its session partner to quiesce can later release it from that condition by sending it a Release Quiesce (RELQ) request. Upon receiving this request, the quiesced half-session can resume sending to its session partner.

## Shutting Down Data Flow

Another way to stop a half-session from sending is with the shutdown protocol. Like the quiesce protocol, only the normal flow is affected. Unlike the quiesce protocol, which either half-session can invoke, only a primary half-session can invoke the shutdown protocol. This protocol may be used when the primary half-session needs to end the session in an orderly manner.

To stop its partner from sending, the primary half-session sends it a Shutdown (SHUTD) request. Upon receiving this request, and after reaching a point in its sending where it is convenient, the secondary half-session stops sending. The secondary half-session determines at what point it is convenient for it to stop. That point might, for example, be after it has completed sending a complete bracket.

Upon reaching that point, the secondary half-session sends a Shutdown Complete (SHUTC) request. After receiving a positive response to this request, the secondary has shut down its normal-flow transmission.

As in the case of the quiesce protocol, its session partner (the primary half-session) can send a Release Quiesce (RELQ) request; upon receiving this, the secondary can resume transmission on the normal flow. Also as in the case of the quiesce protocol, a shutdown half-session continues to receive and respond to normal-flow requests, or responds to such requests with a negative response that specifies "reply not allowed."

## Sequencing Request Units Flowing in a Session

For LU-LU sessions, most requests sent on the normal flow are assigned sequence numbers. These numbers are used for orderly, sequential communication between half-sessions.

Each half-session maintains a pair of sequence numbers. One is called the send sequence number; the half-session passes a send sequence number along with each request it gives to path control for transmission by the path control network. Path control places the number in the transmission header it appends to the request. The other is called the receive sequence number; path control obtains it from the transmission header for the request and passes it to the half-session.

For each request it sends or receives, the half-session updates the send or receive sequence number accordingly.

Each response on the normal flow also has a sequence number; the number for each response is the same as the number for the corresponding request. The half-session correlates a response with its request by matching their sequence numbers.

The first request a half-session sends after it is activated is numbered 1; each request thereafter has a number 1 greater than the preceding request. The highest possible sequence number is 65 535. The next

sequence number assigned is 0, the next is 1, and so on. (Only in this "wraparound" situation is a sequence number of 0 used.)

The progression of sequence numbers can be altered by sending a Clear or a Set and Test Sequence Numbers (STSN) request.

Some requests sent on the normal flow, and all requests sent on the expedited flow, use identifiers instead of sequence numbers. The identifier is unique for each outstanding request sent within a layer.

The data flow control layer assigns identifiers to DFC-RUs sent on the expedited flow; the session control component of the transmission control layer assigns identifiers to the SC-RUs that it processes.

If the transmission header that accompanies a request contains a sequence number field, normal-flow requests sent on LU-LU sessions always use sequence numbers. Normal-flow requests sent on SSCP-SSCP and SSCP-LU sessions always use identifiers. Normal-flow requests sent on SSCP-PU sessions may use sequence numbers or identifiers (in either case, only if the accompanying transmission header has a sequence number field). The transmission services (TS) profile specified determines whether sequence numbers or identifiers are used.

Half-sessions in some peripheral nodes do not use a sequence number. The transmission headers used between half-sessions in these nodes and the boundary function in the subarea nodes to which they are connected do not have a sequence number field. The transmission headers contain a 1-byte identifier that specifies whether the request is flowing on the SSCP-PU session, the SSCP-LU session, or a particular LU-LU session. Only one request at a time can be outstanding between the half-session and the boundary function, so sequence numbers are not needed.

## Reporting Session Status and Signaling the Session Partner

A half-session may need to inform its session partner about local conditions that affect the session. An LU half-session can do this by sending a data RU or it can use the LU Status (LUSTAT) request. This normal-flow request sends 4 bytes of status information to its session partner; the status may be information about the LU or any end-user information desired. LUSTAT is typically used to report failures and error recovery conditions for a local device used by an LU.

Sometimes the LU cannot use (or does not need to use) the normal flow to send status information to its session partner. The LUSTAT request cannot be sent, for example, while the half-session is receiving normal-flow data. The LU can instead use an expedited-flow request, Signal (SIG), to send the desired information.

## Data-Handling Protocols

SNA provides data-handling protocols to control devices, manage data, and compress and compact data. These protocols change the way

information is presented so as to match the needs of individual end users. For example, these protocols can transform data that is packaged within brackets, RU chains, and RUs into lines of data that are printed on a printer.

Upon receiving data requests from path control, an LU half-session determines the destination of the data and then translates, transforms, or combines the data as appropriate before passing it to the application or device. These data-handling services, called session presentation services, provide common formats and protocols to insulate each end user from unnecessary details of another end user's operation.

These protocols may be expressed in LU-LU sessions by including appropriate control information in the data stream. All LU types allow controls in the data stream, but this is the only way control information is passed in LU types 2 and 3. Other LU types (1, 4, and 6) can also send control information in FM headers and, except for LU type 6, in string control bytes (SCBs). Both FM headers and SCBs are part of a data RU; although they need contain neither, data RUs often contain one or more FM headers and one or more SCBs.

### Using FM Headers to Control LU Activity

A single programmed logical unit in a subarea node or a peripheral node may be a point of access for many end users. Function management (FM) headers are a means of directing the destination LU in a session to customize its processing of data RUs to meet the needs of individual end users represented by that LU, and to direct the end-user data to particular physical components, such as displays and printers. FM headers are "transparent" to the data flow control and transmission control elements of an LU; only the presentation services component within the LU is concerned with FM headers.

Three types of FM headers and their uses are as follows:

Type 1 FM Header: A type 1 header selects destinations associated with an LU. A destination may be represented by a device (such as a display or printer), a data set residing on a device, or merely a data stream. The LU also supplies control information relative to the initiation, interruption, resumption, and conclusion of destination selection, and indicates whether the data is compressed or compacted.

Type 2 FM Header: The type 2 header provides further information about previously selected destinations. This header, which is always used with a type 1 header, defines destination attributes and requests specific data management tasks for that destination. Type 2 headers may be sent with or without data.

Type 3 FM Header: The type 3 header requests specific data management tasks that apply to all destinations in the LU. (In contrast, a type 2 header applies only to a specific destination.): The preceding FM header

types may be used to help host-resident application subsystems communicate with logical units operating in batch mode in peripheral nodes.

FM header types 4 through 8 are also available. They are used for process selection and process-to-process communication in intersystem communication environments. Their use is described in SNA—Sessions Between Logical Units, GC20-1868.

### Improving Transmission Efficiency by Compressing and Compacting Data

Two means to improve the efficiency of data transmission through an SNA network are available: data compression and data compaction.

Data compression involves recognizing repeated characters in data to be sent to a destination. Instead of repeating a string of identical characters, a 1- or 2-byte code is substituted. Data compaction involves restructuring parts of a data stream so that some bytes represent more than one character of data. The end user selects the characters to be compacted.

For certain kinds of traffic, data compression or compaction can reduce significantly the number of characters transmitted in a data RU. This increases the amount of data that a route that handles such traffic can accommodate. These techniques involve the use of computing time to save transmission time. They may improve throughput when the network's paths are overloaded or are relatively slow speed paths. On the other hand, compaction may not be efficient when the data traffic has few occurrences of character strings that can be compressed or compacted and the traffic uses high-speed or broadband communication paths.

If the FM header indicates that an SCB (string control byte) follows, the data immediately following the header is an SCB. The SCB identifies the compression or compaction characteristics of the data.

Once compression or compaction is selected, one of the session partners must send an FM header to establish the compression or compaction criteria. The LU that originates the data must compress or compact it and build the SCBs that identify the compression or compaction characteristics. The LU that receives the compressed or compacted data must analyze the SCBs and decompress or decompact the data accordingly, thus restoring it to its original form.

### Improving Data Security through Cryptography

Cryptography is the technique of converting clear, understandable data into data that is incomprehensible to all except those who have the means to convert it back into clear data. In data processing, cryptography is implemented by an algorithm, agreed upon in advance, and a specific cryptographic key that is changeable and known only to the communicators.

A cryptographic algorithm can be represented as an extremely large number of possible mathematical transformations. Each transformation defines how sequences of intelligible data are converted (enciphered) into sequences of apparently random noise that are unintelligible to humans or machines. For each of these transformations, an inverse operation is required to change (decipher) the data back into its original form. The cryptographic key is a secretly held sequence of numbers or characters that identifies the specific transformation to be used.

Assuming that data encryption and decryption are performed using the same cryptographic key, a cryptographic algorithm provides data security between two nodes of a network if both nodes have the algorithm installed (in hardware or software) and if both know the key. Only the key must be kept secret; the details of the algorithm are assumed to be known to everyone.

The SNA access methods (ACF/TCAM and ACF/VTAM) supply an interface to IBM's Programmed Cryptographic Facility program product, which enables SNA users to secure the data flowing in LU-LU sessions. All stations that participate in cryptographic LU-LU sessions must be equipped with a cryptographic feature that is compatible.

The Programmed Cryptographic Facility uses the Data Encryption Standard (DES) as its cryptographic algorithm. (The DES is published by the National Bureau of Standards.) It also provides a key generator utility program, which is used to create, maintain, and delete cryptographic keys; and a cryptographic key resource manager, which assigns and manages the cryptographic keys used by an installation. Key management provides for the generation, transformation, and control of operational keys in such a way that the network user is not aware of their manipulation, thus increasing overall data security.

SNA provides two kinds of data encryption: mandatory and selective. If encryption is mandatory, all FMD request units flowing in an LU-LU session are enciphered and deciphered, with the participating logical units having no option. If encryption is selective, FMD request units flowing in an LU-LU session are enciphered and deciphered at the user's request on an RU-by-RU basis or on an RU-chain-by-RU-chain basis. The request header (RH) contains a bit that indicates whether the RU is enciphered.

For further information about cryptography, see Data Security through Cryptography, GC22-9062.

## **Pacing of Data Flow at the Session Level**

Session-level pacing is a data-flow coordination mechanism for preventing one of the partners in an LU-LU session from being flooded with FM data requests from the other partner. Session-level pacing permits the receiving LU to control the rate at which it receives FM data requests on the normal flow. (Expedited-flow requests are not

paced.) Pacing is generally used when one LU in an LU-LU session is capable of sending requests faster than the other LU can process them.

When pacing is in effect, the request sender is allowed to send a limited number of normal-flow requests; it can send no more requests until the receiver indicates, by returning a pacing response, its readiness to receive more. The sender can then send more requests, up to the stated limit. If the receiver sends the pacing response early, that is, before having received the maximum number of requests, the sender can send its next group of requests immediately following the first group.

Normal-flow responses may or may not be marked as queued responses. If a request is held up from transmission by a pacing delay, responses marked as queued responses that are queued behind the request are also held up. In contrast, responses that are not marked as queued responses are not held up by a pacing delay, but may pass queued requests at queuing points in transmission control and boundary-function transmission control.

A session-level pacing response may be an ordinary FM data response with the pacing indicator turned on in the response header. In some sessions, however, an FM data response does not ordinarily flow when a pacing response is required (for example, if the request sender specifies that it is to receive only negative responses). In such sessions, the receiving LU formats and sends as needed an isolated pacing response (IPR). This is an FM data response consisting of an RH with the session-level pacing response indicator on, and no RU. Isolated pacing responses can be sent on either the normal flow or the expedited flow.

The pacing of requests flowing toward a given LU is called receive pacing with respect to that LU. The pacing of requests flowing away from a given LU is called send pacing with respect to that LU.

Pacing is done in one or two stages, as shown in Figure 5-2 on page 5-23. One-stage pacing directs the flow of requests between the sending LU and the receiving LU. (See Figure 5-2(a).) Two-stage pacing directs the flow of requests (1) from the sending LU to the boundary function on the path between the two LUs and then (2) from the boundary function to the receiving LU, as shown in Figure 5-2(b).

An LU-LU session may be paced in both directions. That is, both requests flowing from the primary to the secondary LU and requests flowing from the secondary to the primary LU may be paced.

Four pacing parameters are defined for each LU-LU session:

- Secondary send pacing-group size
- Secondary receive pacing-group size
- Primary send pacing-group size
- Primary receive pacing-group size



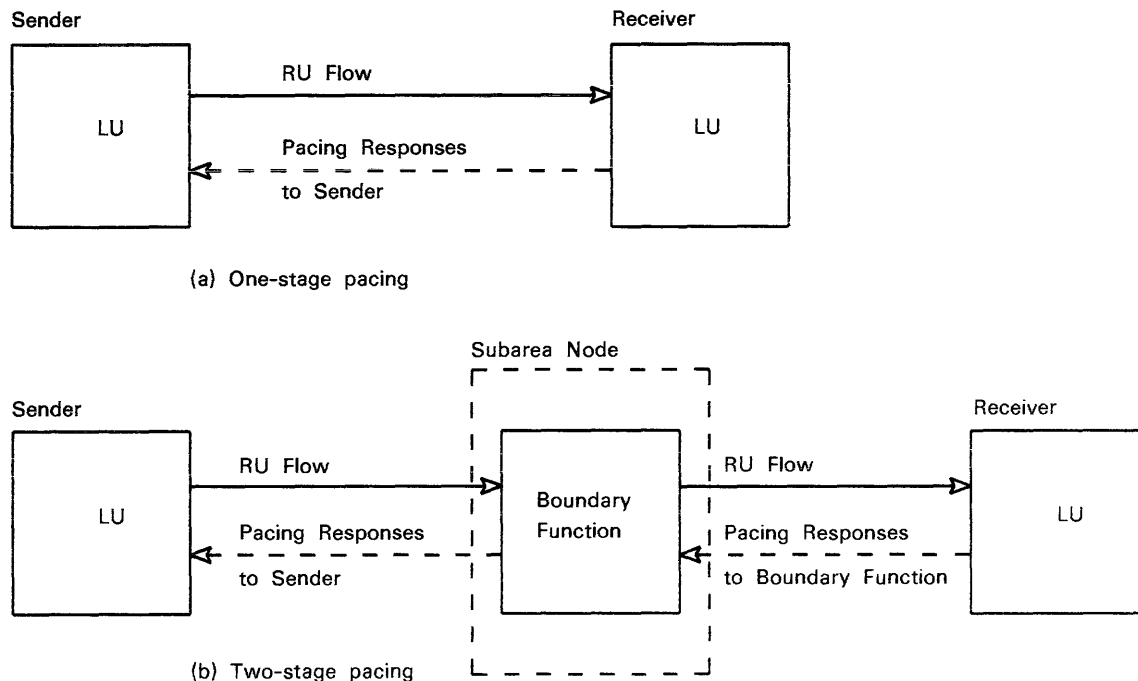


Figure 5-2. One-Stage and Two-Stage Pacing

The primary send pacing-group size and the secondary receive pacing-group size together specify how pacing is to be done for FM data requests flowing from the primary LU to the secondary LU. The primary send pacing-group size specifies how many requests may flow from the primary LU to the boundary function before the primary LU receives a pacing response. The secondary receive pacing-group size specifies how many requests may flow between the boundary function and the secondary LU before the boundary function receives a pacing response. A similar scheme applies to requests that flow in the other direction, that is, from the secondary LU to the primary LU.

A pacing-group size of 0 means that no pacing occurs: the number of RUs sent is not limited. A pacing-group size of 0 may reduce total network traffic by obviating the need for pacing responses, but caution should be exercised in specifying a 0 count lest the receiver of FM data requests flowing in an LU-LU session be unable to accept the rate of input to it.

The Bind Session (BIND) RU specifies, for each direction of flow, whether one-stage, two-stage, or no pacing is to be used, and specifies the counts to be applied.

## Error Recovery at the Session Level

Errors are a naturally occurring phenomenon in the complex environment for which SNA protocols are defined. They have a number of causes, ranging from random electrical noise to mismatched definitions for session partners. They can vary in persistency from transient to fully disruptive.

As defined by SNA, an error is a violation of:

- A general architectural rule, such as using an undefined RU request code
- A session rule, such as using an FM protocol not defined within the FM profile or usage field of the Bind Session (BIND) RU
- A rule established by an FM header within a session
- A state-dependent rule, such as a sequence number received on the normal flow that was not one greater than the previous sequence number

Either session partner can be responsible for recovering from errors and unsuccessful transmissions. Parameters in the Bind Session (BIND) RU indicate which partner has the responsibility; in some cases this responsibility can be shared.

The sending LU has the responsibility to send an error-free data stream, and thus must check and return to the end user any errors that are detected. Thus, many of the errors are temporary. They are corrected either by the path control network or by an operator (for example, a printer being out of forms) and are not seen by the LU. For the temporary errors found by the LU's half-session, the half-session frequently can recover from the error by retransmitting the RU chain that failed.

One of the main recovery problems with LU-LU sessions is having to resynchronize the session after an unrecoverable transmission error occurs. If a half-session has little recovery capability, it may send to its session partner SNA requests that cause the transmission in error to be deleted or the session to be terminated. A half-session that can restart the transmission may use sequence numbers in its session recovery actions. The Set and Test Sequence Numbers (STSN) RU is useful in this effort.

There may be times when the session cannot be resumed, but the end user's job is not complete. In this event, the session partners can conclude the session and activate a new session where the previous one ended. Sequence numbers of the new session may be reset to the nonzero value of the previous session and the new session then activated.

The recovery approach a half-session uses depends on the kind of error and the half-session's error recovery capability. That capability can range from terminating a session to performing sophisticated error correcting actions.

Some program products, such as IMS/VS and CICS/VS, contain facilities for backing work in progress out of data sets when session-level errors occur. Based upon the use of synchronization points, these facilities help preserve the integrity of data sets that are being updated online.

Both IMS/VS and ACF/TCAM have extensive checkpoint/restart capabilities that allow new sessions to be activated because of hardware failure; the first RU chain to flow on the new session is the chain that was next to be sent when the failure occurred.

ACF/TCAM, ACF/VTAM, and ACF/NCP allow sessions to continue when links fail in a multiple-link transmission group, as long as at least one link in the group remains active. ACF/VTAM and ACF/TCAM allow new sessions to be automatically reinitiated to replace sessions that were interrupted because a virtual route failed; the new sessions may be activated over a different virtual route.

#### Session Outage Notification

For a number of reasons, an active session between two LUs (or other network addressable units) can fail. In these cases, SNA provides the means for notifying the affected half-sessions so that they can try restart processing. This notification is called session outage notification.

A virtual route used by active sessions may be disrupted as a result of the failure of the last remaining link in a transmission group used by the explicit route that underlies the virtual route. Or the virtual route may be forcibly deactivated. In either case, both ends of the virtual route inform NAU services managers in each affected node. The NAU services managers then generate and send session deactivation requests (Unbind Session [UNBIND], Deactivate Logical Unit [DACTLU], Deactivate Physical Unit [DACTPU], or Deactivate Cross-Domain Resource Management [DACTCDRM]) to the affected half-sessions in their subareas, notifying them of the failure. Restart processing can involve assigning a different virtual route in order to bypass the failure and reactivate the sessions. Session reinitiation by a terminal operator or network operator initiates the restart processing.

A route extension used by active sessions may fail. Should this occur, the PU in the node at each end of the route extension sends session deactivation requests to all affected half-sessions on its side of the failure. Within an SSCP, any affected half-sessions involved in SSCP-PU and SSCP-LU sessions are reset as a result of receiving an Inoperative (Inop) RU from the PU in the subarea node to which the route extension is attached.

## Profiles and Usage Fields

Some of the session protocols (such as those for request and response control modes, brackets, and pacing) are selectable at session activation. Specific combinations of these selectable protocol options are known as profiles; they are specified in the profile field of the RU that activates the session.

Profiles that refer to presentation of end-user data are called presentation services (PS) profiles; they apply only to LU-LU sessions. Profiles that refer to transmission control options are called transmission services (TS) profiles; profiles that refer to data flow control (DFC) and function management (FM) data options are called FM profiles; and profiles that refer to SSCP options for cross-domain SSCP sessions are called cross-domain resource management (CDRM) profiles.

The PS, TS, and FM profiles to be used in any session are specified when the session is activated. These profiles are specified as parameters in session activation requests and their responses: the requests are Bind Session (BIND), Activate Physical Unit (ACTPU), Activate Logical Unit (ACTLU), and Activate Cross-Domain Resource Management (ACTCDRM), respectively, for LU-LU, SSCP-PU, SSCP-LU, and SSCP-SSCP sessions. The CDRM profile also is specified during SSCP-SSCP session activation, via a control vector in the ACTCDRM request and its response.

Some FM and TS profiles require supplemental information about the protocols they specify beyond what the profile field itself contains. Such supplemental information is carried in usage fields within the session activation request.

The FM usage field has three subfields. One subfield contains protocol rules that the primary and secondary half-sessions must jointly enforce (for example, whether the normal-flow requests are to flow in one direction at a time [half-duplex flow] or in both directions at once [full-duplex flow]). Another subfield specifies the rules that the secondary LU is to follow; for example, whether the secondary half-session may end a bracket. The final subfield specifies the rules that the primary half-session is to follow.

The TS usage field specifies pacing parameters and the maximum RU sizes allowed on the normal flow.

The PS usage field supplements the information specified by the TS and FM profile and usage fields by identifying additional FM options that primary and secondary LU half-sessions use. Similarly to the FM usage field, a PS usage field has three subfields: one for rules jointly followed by both primary and secondary half-sessions, one for rules followed by the primary half-session, and one for rules followed by the secondary half-session.

Detailed lists of the FM, TS, and PS profiles and usage fields can be found in the SNA Reference Summary, GA27-3136.

## 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 that type of LU.

- 0 A type of LU that uses SNA-defined protocols for transmission control and data flow control, but uses end-user or product-defined protocols to augment or replace FMD services protocols. For example, an LU for an application program using IMS/VS and an IBM 3600 Finance Communication System in which the operator of the 3600 terminal is updating the passbook balance for a customer's savings account.
- 1 A type of LU for an application program that communicates with single- or multiple-device data processing terminals 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 3767 Communication Terminal in which the terminal operator is correcting a data base that the application program maintains. The data stream is the SNA character string (SCS).
- 2 A type of LU for an application program that communicates with a single display terminal in an interactive environment, using the SNA 3270 data stream. For example, an LU for an application program that uses IMS/VS and an IBM 3277 Display Station, in which the 3277 operator is creating and sending data to the application program.
- 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 3284 Printer attached to an IBM 3791 Controller.
- 4 A type of LU for: (1) an application program that communicates with a single-or multiple-device data processing or word processing terminal 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 (SCS) for data processing environments and Office Information Interchange Level 2 for word processing environments.
- 6 A type of LU for an application subsystem that is to communicate with another application subsystem in a distributed processing environment. For example, an LU for an application program that uses CICS/VS to communicate with an application program that uses IMS/VS.

Both session partners in an LU-LU session must be of the same LU type. SNA does not permit, for example, one half-session to be a component of

a type 1 LU and the other to be a component of a type 2 LU. However, a single LU can have more than one LU type. For example, an LU can contain the resources to participate in LU-LU sessions as both a type 1 and a type 4 LU simultaneously, provided that the session partner for the type 1 LU is another type 1 LU and the session partner for the type 4 LU is another type 4 LU.

A complete list of the LU-LU session characteristics for each LU type appears in SNA Reference Summary, GA27-3136. These characteristics include the TS and FM profiles and PS characteristics permitted and the sense codes that are applicable to the LU type.

## Selecting and Using a Data Stream

### SNA Character String Controls

SNA character string (SCS) 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. They also set modes of device operation, define data to be used in a unique fashion, or are 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 SCS data stream consists of a sequential string of SCS control codes and data characters. Control codes may be intermixed with graphic data characters. SCS 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 SCS control codes. One-byte parameters that specify functions or binary values are permitted with some codes.

SCS 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 (SCBs) for functions such as selecting destinations, managing data, and compressing or compacting data.

SCS 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 SCS 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 SCS controls are Backspace, Carriage Return, Form Feed, Horizontal Tab, Indent Tab, Presentation Position, Select Left Platen, Set Horizontal Format, Superscript, and Word Underscore. A complete

list of SCS controls and a description of how each is used appears in SNA—Sessions Between Logical Units, GC20-1868.

SCS controls can be used by LU types 0, 1, 4, and 6.

### SNA 3270 Data Streams

The SNA 3270 data stream consists of user-provided data and commands that are transmitted between the primary logical unit and the secondary logical unit of an LU-LU session. Control information, which governs the way data is handled and formatted, is also transmitted.

The SNA 3270 data stream is the only data stream used by LU types 2 and 3. It is an optional data stream for LU types 0 and 6. The data stream supports both display and printer applications.

An application program communicates with a display operator using one of two methods. In one method, the display surface is left unformatted by the application program 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 it is the only partition in which the operator can enter data or requests.

Specific information on the 3270 functions that can be specified in a SNA 3270 data stream appears in IBM 3270 Data Stream Programmer's Reference, GA23-0059. Specific information on the sets of functions that can be specified in an SNA 3270 data stream appears in SNA—Sessions between Logical Units, GC20-1868.

### String Control Bytes Used for Compressing and Compacting Data

Data compression and compaction are data-handling techniques used by LU types 1 and 4 to shorten the transmission time of data during the session. Each half-session compresses or compacts the data before sending it to its session partner; the session partner in turn decompresses or decompacts the data before passing it to the end user.

In data compression, a user-selected character called the prime compression character is used to replace repetitive sequences of that character within a data stream with a string control byte (SCB). One SCB can replace up to 63 characters (bytes), but the characters must be a string of the same prime character. An SCB can also define strings of repeated characters that are not the same as the prime compression character. In this case the SCB is followed by a byte containing the nonprime character so that the receiver knows which is the repeated character.

Data can also be compacted to shorten the length of transmissions. In the compaction technique, two bytes of data are replaced by one byte. Up to 16 characters from the total character set used by the session can be compacted; the end user optionally provides a compaction table to indicate which characters are to be compacted. An SCB is used in the data stream to define the beginning and end of each block of compacted data.

When a half-session sends compressed or compacted data, it sets on the compression indicator (CMI) or compaction indicator (CPI) in the FM header that precedes the data. When either or both of these flags are on, the half-session builds and inserts one or more SCBs into the data stream. An SCB either begins an RU chain or follows any FM headers that may be present in the chain.

Further information on the use of data compression, data compaction, and SCBs appears in SNA—Sessions between Logical Units, GC20-1868.

## TYPICAL REQUEST UNIT SEQUENCES FOR ACTIVATING SESSIONS, TRANSFERRING DATA, AND DEACTIVATING SESSIONS

Figure 5-4 through Figure 5-6 present typical request unit sequences used to activate a same-domain LU-LU session, an SSCP-SSCP session, and a cross-domain LU-LU session.

Figure 5-7 through Figure 5-9 present typical request unit sequences used to deactivate same-domain and cross-domain LU-LU sessions and SSCP-SSCP sessions.

Figure 5-10 through Figure 5-14 present some typical request unit sequences used within LU-LU sessions. They illustrate the use of bracket protocols, half-duplex contention and half-duplex flip-flop protocols, protocols for quiescing data flow, and protocols for deactivating the LU-LU session.

(Figure 5-3 on page 5-31 gives the meanings of each of the symbols and abbreviations appearing in these request unit sequences.)







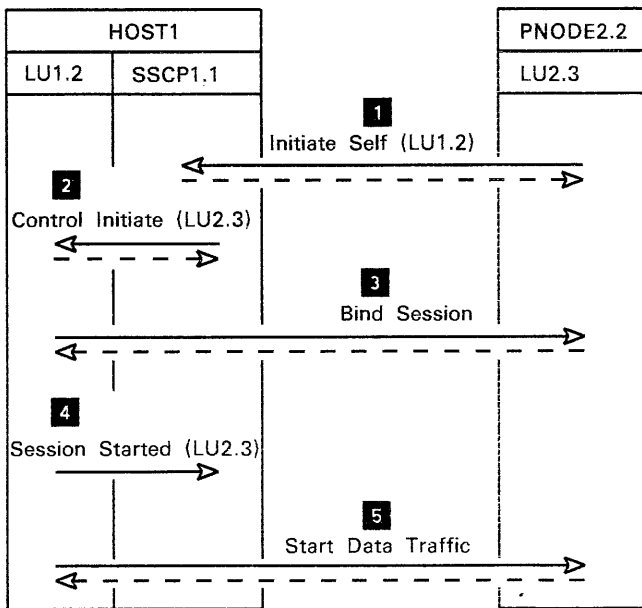
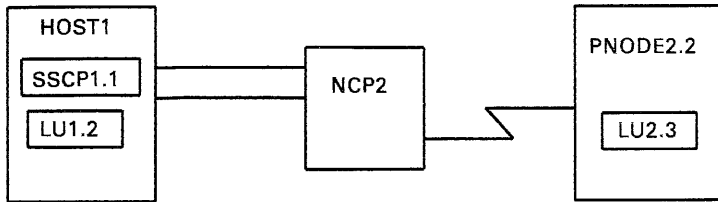
	Request Unit	Host	Host Processor
	Response Unit	Link	Data Link
	Channel	LS	Link Station
	SDLC Link	LU	Logical Unit
BB	Begin Bracket (indicator)	NCP	Network Control Program
BC	Begin Chain (indicator)	PNODE	Peripheral Node
Bid	Bid request	PU	Physical Unit
CD	Change Direction (indicator)	PUCP	Physical Unit Control Point
DR*	Definite Response 1 (DR1) or Definite Response 2 (DR2) or both (indicators)	QEC	Quiesce at End of Chain
ER	Exception Response	RELQ	Release Quiesce request
		+RSP	Positive Response
		-RSP	Negative Response
		RTR	Ready to Receive Request
		SA	Subarea
		SSCP	System Services Control Point
		TG	Transmission Group
		VR	Virtual Route

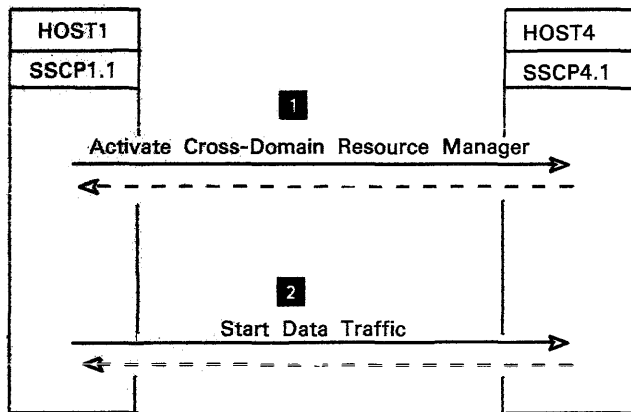
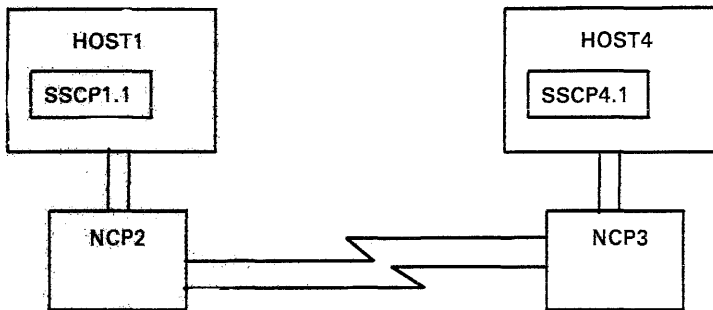
Figure 5-3. Symbols and Abbreviations Appearing in Sequence Diagrams of Chapter 5



1. LU2.3 requests that SSCP1.1 set up a session between LU2.3 and LU1.2. LU1.2 is to be the primary LU.
2. SSCP1.1 tells LU1.2 to activate a session with LU2.3, and informs LU1.2 of the attributes of LU2.3.
3. LU1.2 activates a session with LU2.3 and passes LU2.3 rules to be observed during this session.
4. LU1.2 informs SSCP1.1 that LU1.2 has activated a session with LU2.3.
5. LU1.2 enables the flow of FMD and DFC message units over its LU-LU session with LU2.3.

(Figure 5-3 gives the meanings of the symbols and abbreviations that appear in this figure.)

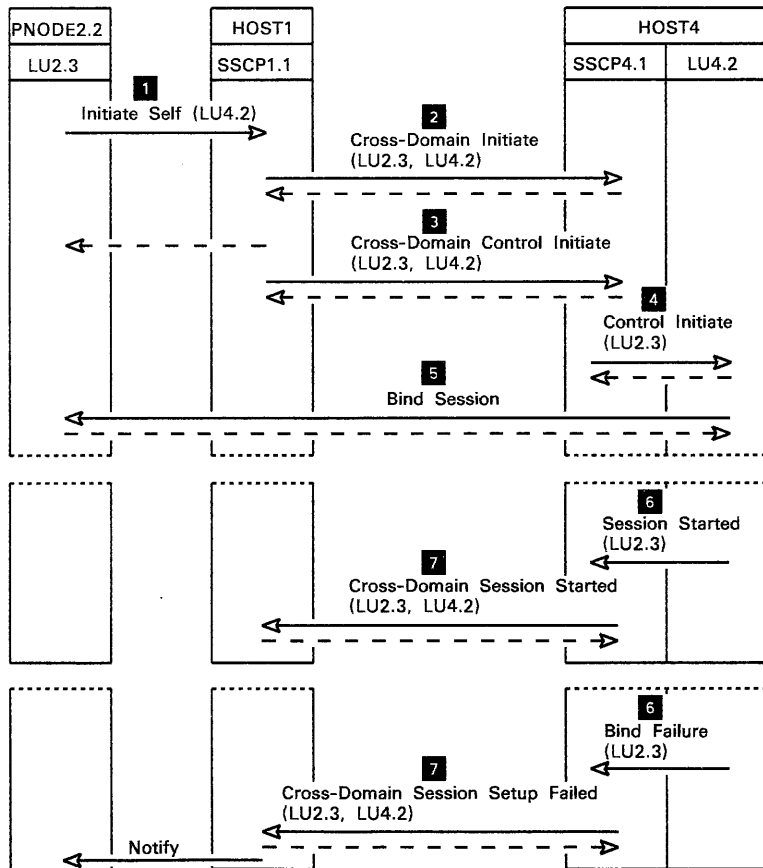
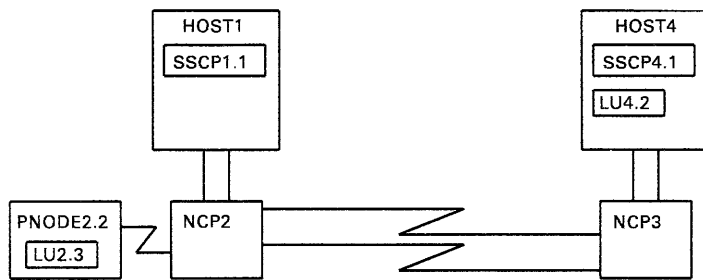
Figure 5-4. Activating a Same-Domain LU-LU Session



1. SSCP1.1 activates a session with SSCP4.1.
2. SSCP1.1 enables the flow of FMD message units over its SSCP-SSCP session with SSCP4.1.

(Figure 5-3 gives the meanings of the symbols and abbreviations that appear in this figure.)

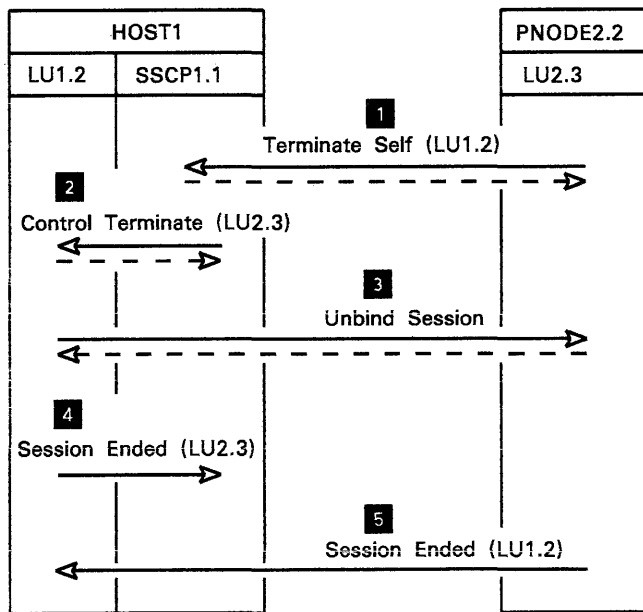
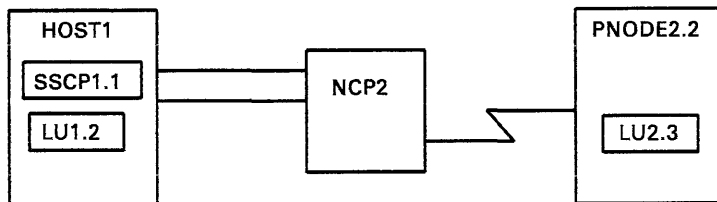
Figure 5-5. Activating an SSCP-SSCP Session



1. LU2.3 requests that SSCP1.1 help set up a session between LU2.3 and LU4.2. LU4.2 is to be the primary LU.
  2. SSCP1.1 tells SSCP4.1 of LU2.3's request. After receiving a response from SSCP4.1, SSCP1.1 returns to LU2.3 a response to LU2.3's request.
  3. SSCP1.1 sends SSCP4.1 the session rules that must be in effect for the session between LU2.3 and LU4.2.
  4. SSCP4.1 tells LU4.2 to activate a session with LU2.3, using the session rules passed by SSCP1.1 in step 3.
  5. LU4.2 activates a session with LU2.3 using the session rules given by SSCP4.1 in step 4.
- If a session is activated successfully:
6. LU4.2 informs SSCP4.1 that LU4.2 has activated a session with LU2.3.
  7. SSCP4.1 informs SSCP1.1 of this fact.
- If the session is not activated successfully:
6. LU4.2 informs SSCP4.1 that LU4.2 has failed in its attempt to activate a session with LU2.3.
  7. SSCP4.1 informs SSCP1.1 of this fact.

(Figure 5-3 gives the meanings of the symbols and abbreviations that appear in this figure.)

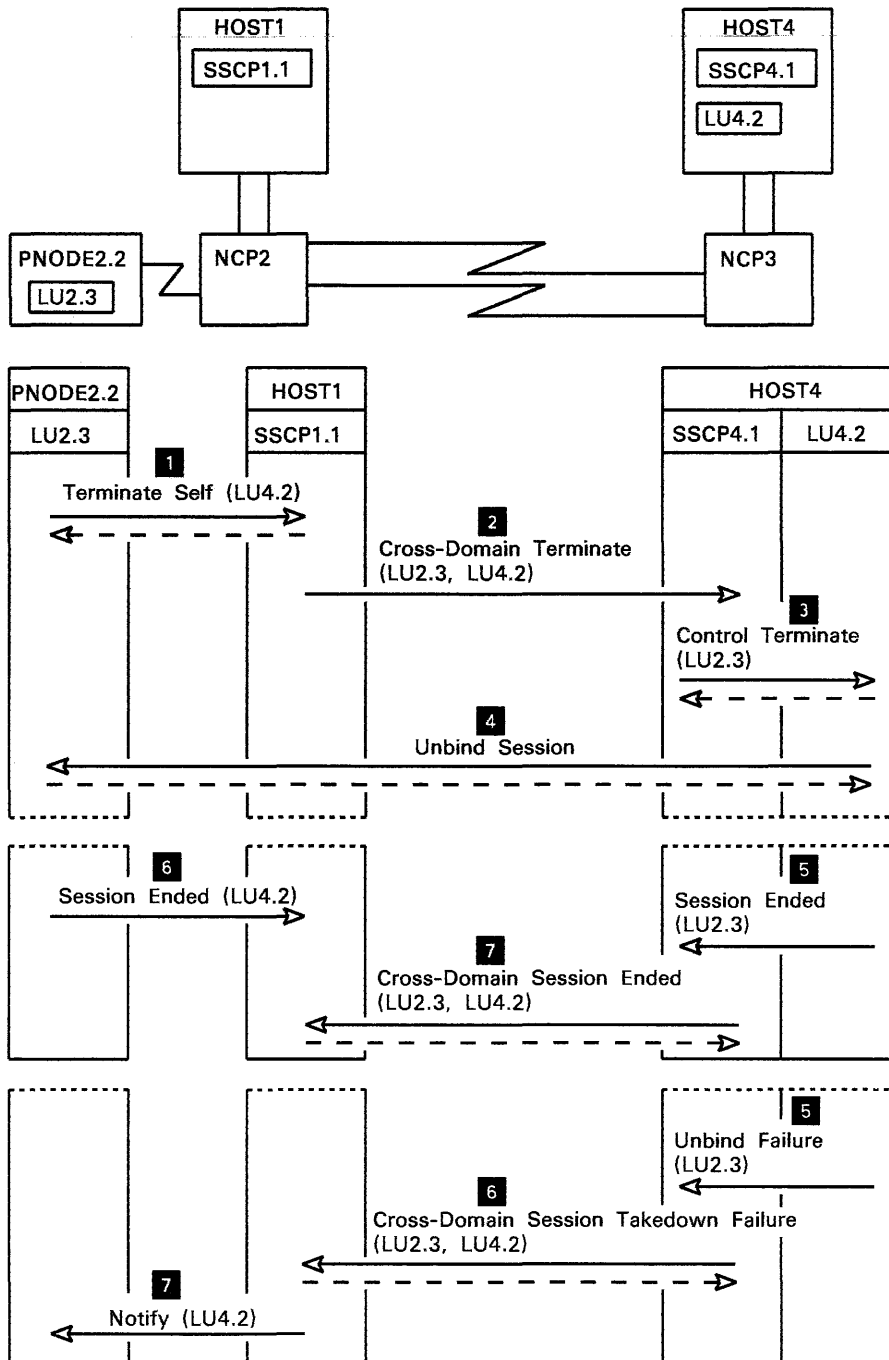
Figure 5-6. Activating a Cross-Domain LU-LU Session



1. LU2.3 requests that SSCP1.1 help terminate the session between LU1.2 and LU2.3. LU1.2 is the primary LU.
2. SSCP1.1 tells LU1.2 to deactivate its session with LU2.3.
3. LU1.2 deactivates its session with LU2.3.
4. LU1.2 informs SSCP1.1 that LU1.2 has deactivated its session with LU2.3.
5. The boundary function of LU2.3 informs SSCP1.1 that the session between LU2.3 and LU1.2 has been deactivated. (Although the RU bearing this information has the network address of LU2.3 in the origin field of its TH, this RU actually comes from the LU2.3 boundary function in NCP2.)

(Figure 5-3 gives the meanings of the symbols and abbreviations that appear in this figure.)

Figure 5-7. Deactivating a Same-Domain LU-LU Session



(Figure 5-3 gives the meanings of the symbols and abbreviations that appear in this figure.)

Figure 5-8 (Part 1 of 2). Deactivating a Cross-Domain LU-LU Session

- 
1. LU2.3 requests that SSCP1.1 help deactivate the session between LU2.3 and LU4.2. LU4.2 is the primary LU.
  2. SSCP1.1 tells SSCP4.1 of LU2.3's request.
  3. SSCP4.1 tells LU4.2 to deactivate its session with LU2.3.
  4. LU4.2 deactivates its session with LU2.3.

If the session is deactivated successfully:

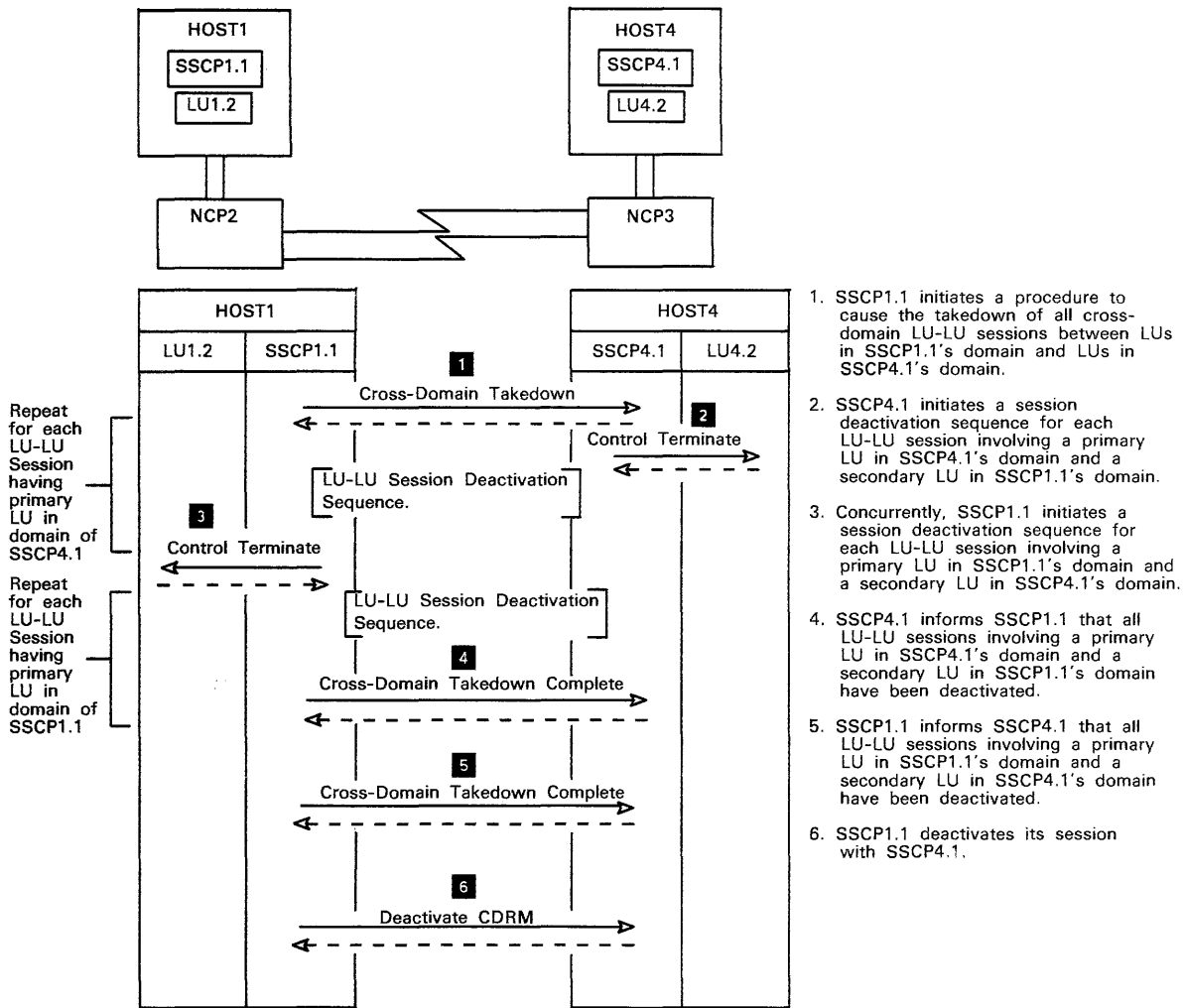
5. LU4.2 informs SSCP4.1 that LU4.2 has deactivated its session with LU2.3.
6. The boundary function of LU2.3 informs SSCP1.1 that the session between LU2.3 and LU4.2 has been deactivated. (Although the RU bearing this information has the network address of LU2.3 in the origin field of its TH, this RU actually comes from the LU2.3 boundary function in NCP2.)
7. SSCP4.1 informs SSCP1.1 that the cross-domain session has been deactivated.

If the session is not deactivated successfully:

5. LU4.2 informs SSCP4.1 that LU4.2 has failed in its attempt to deactivate a session with LU2.3.
6. SSCP4.1 informs SSCP1.1 that the attempt to deactivate the session between LU2.3 and LU4.2 has failed.
7. SSCP1.1 informs LU2.3 that LU4.2 believes that LU4.2's attempt to deactivate the session has failed.

Figure 5-8 (Part 2 of 2). Deactivating a Cross-Domain LU-LU Session

---

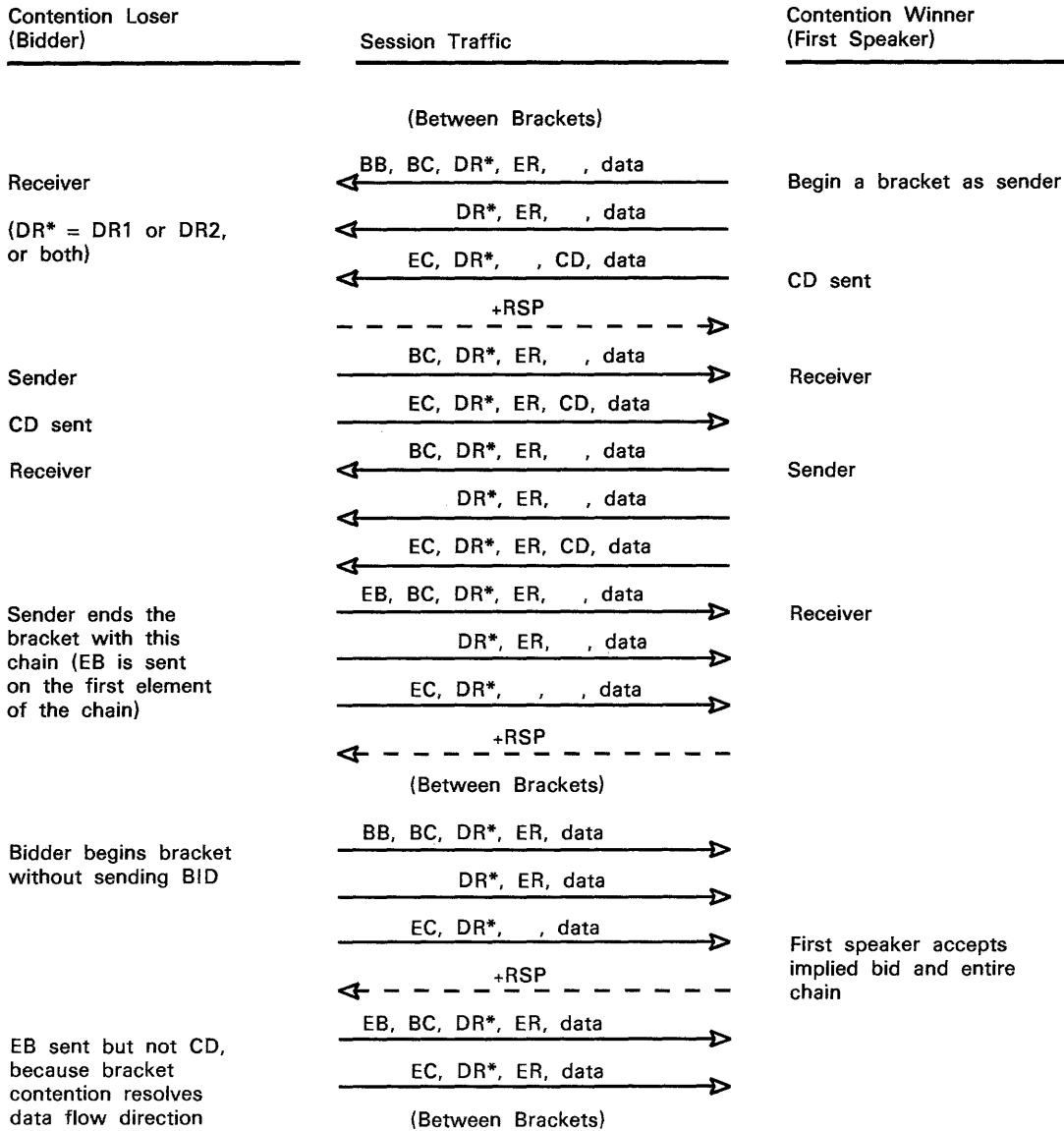


(Figure 5-3 gives the meanings of the symbols and abbreviations that appear in this figure.)

Figure 5-9. Cross-Domain Takedown Sequence



This page intentionally left blank.



(Figure 5-3 gives the meanings of the symbols and abbreviations that appear in this figure.)

Figure 5-10 (Part 1 of 2). Communication Using Brackets in Half-Duplex Flip-Flop Mode

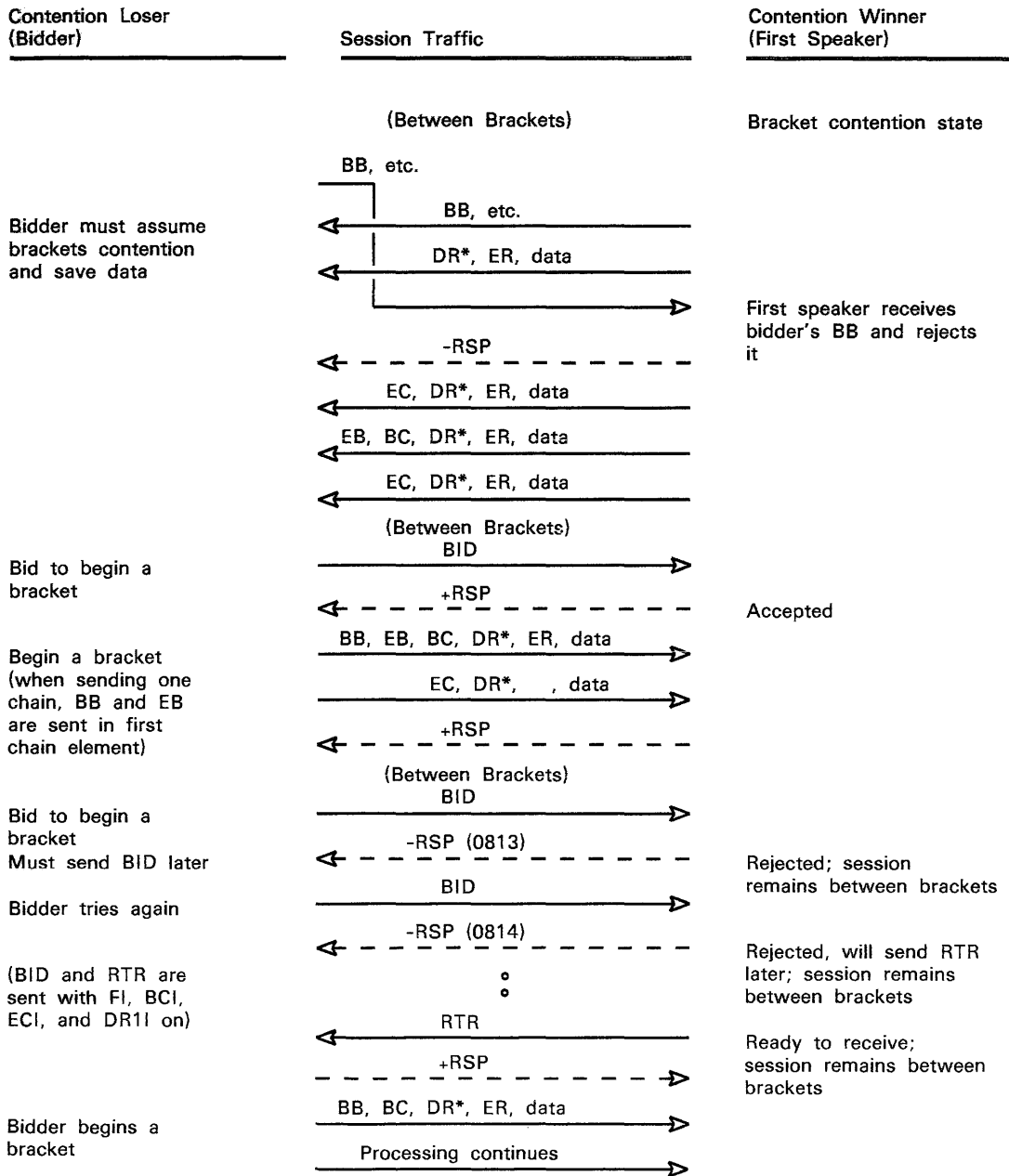
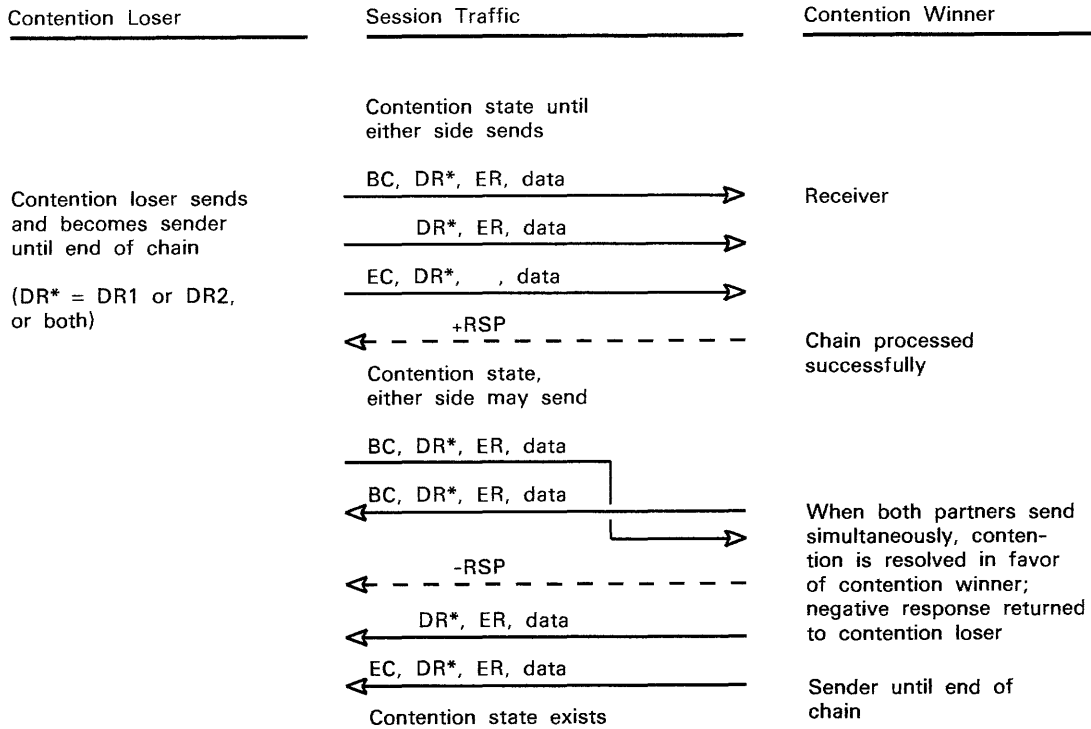
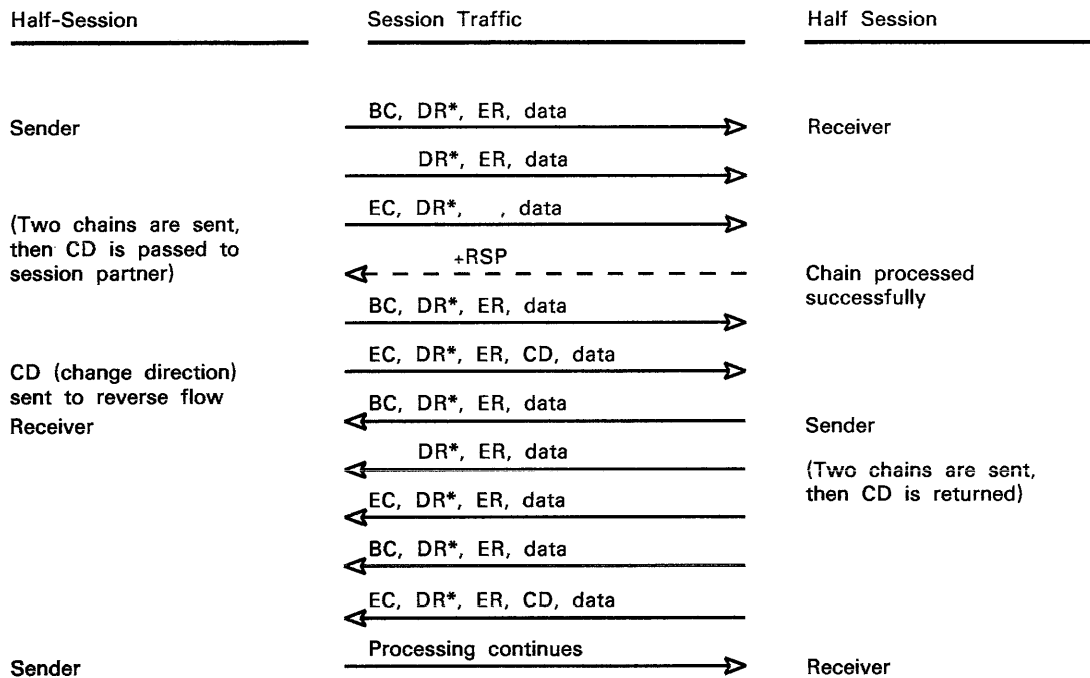


Figure 5-10 (Part 2 of 2). Communication Using Brackets in Half-Duplex Flip-Flop Mode



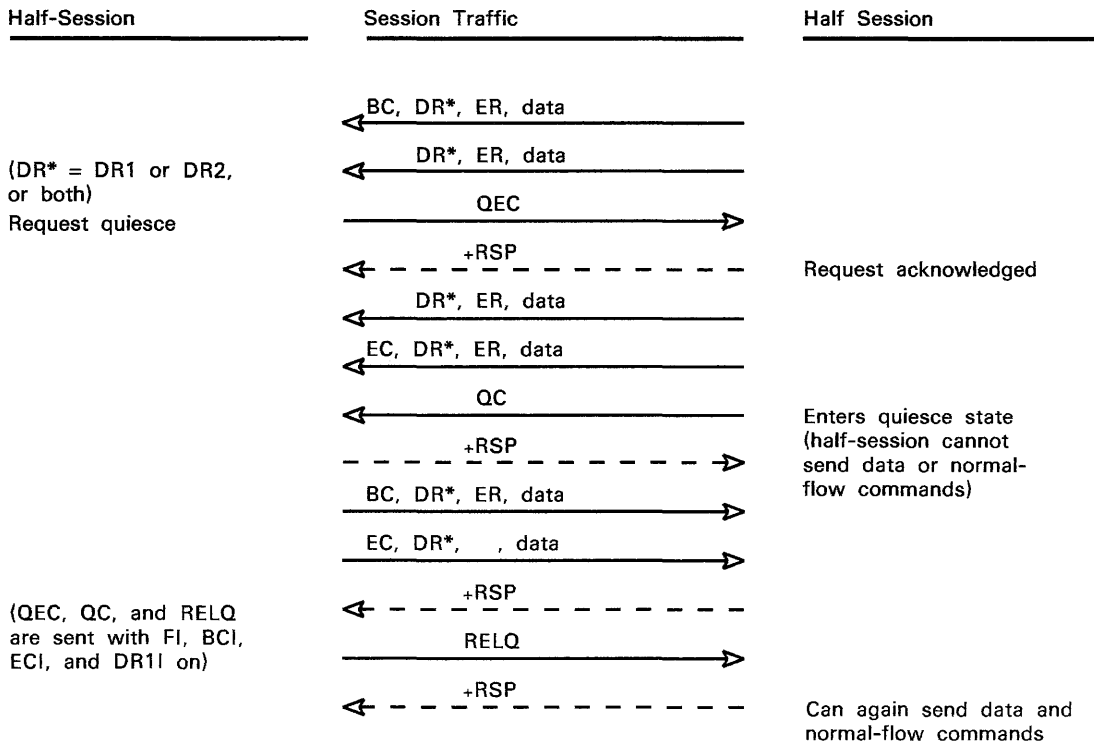
(Figure 5-3 gives the meanings of the symbols and abbreviations that appear in this figure.)

Figure 5-11. Communication Using Half-Duplex Contention Protocols



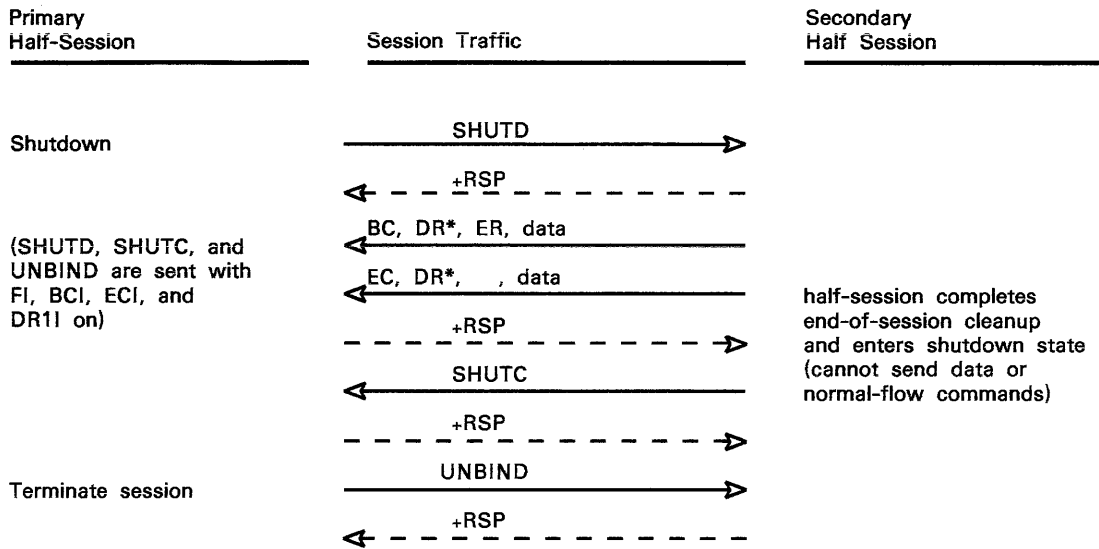
(Figure 5-3 gives the meanings of the symbols and abbreviations that appear in this figure.)

Figure 5-12. LU-LU Communication Using Half-Duplex Flip-Flop Protocols

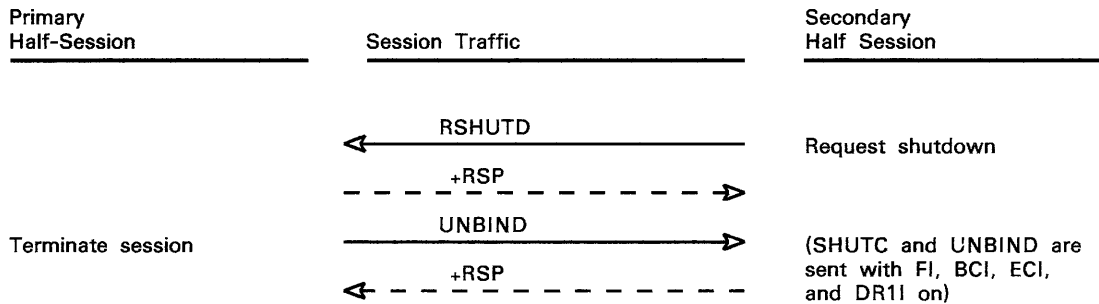


(Figure 5-3 gives the meanings of the symbols and abbreviations that appear in this figure.)

Figure 5-13. Protocols for Quiescing Data Flow



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



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

(Figure 5-3 gives the meanings of the symbols and abbreviations that appear in this figure.)

Figure 5-14. Protocols for Deactivating LU-LU Session





## GLOSSARY

This Glossary contains definitions of terms and abbreviations related to Systems Network Architecture and to products designed in accordance with Systems Network Architecture. These terms, abbreviations, and definitions are taken from the IBM Vocabulary for Data Processing, Telecommunications, and Office Systems, GC20-1699. Terms defined by Systems Network Architecture are preceded by "In SNA,"; terms not defined by SNA are not so identified.

**ACF.** Advanced Communications Function.

**active state.** In SNA, the state in which a component of an SNA node is able to perform the functions for which it was designed.

**adjacent domains.** In ACF/VTAM, domains sharing a common subarea node (for example, a communication controller) or two domains connected by a link. See also domain.

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

**Note:** Several secondary link stations that share a link connection do not exchange data with each other and therefore are not adjacent to each other.

**adjacent nodes.** In SNA, two nodes that are connected by one or more links with no intervening nodes.

**adjacent subareas.** In ACF/TCAM, two subareas connected by one or more links, with no intervening nodes.

**Advanced Communications Function (ACF).** A group of IBM program

products (principally ACF/TCAM, ACF/VTAM, ACF/VTAME, and ACF/NCP/VS) that uses the concepts of Systems Network Architecture (SNA), including distribution of function and resource sharing.

**Note:** ACF/NCP/VS, ACF/VTAME, and the Multisystem Networking Facility of ACF/TCAM and ACF/VTAM allow the interconnection of two or more domains into one multiple-domain network.

**Advanced Communications Function for the Network Control Program (ACF/NCP/VS).** A program product that provides communication controller support for single-domain and multiple-domain data communication.

**Advanced Communications Function for the Telecommunications Access METHOD (ACF/TCAM).** A program product that provides single-domain data communication capability, and, optionally, multiple-domain capability.

**Advanced Communications Function for the Virtual Telecommunications Access Method (ACF/VTAM).** A program product that provides single-domain data communication capability and, optionally, multiple-domain capability.

**Advanced Communications Function for VTAM Entry (ACF/VTAME).** A program product that provides single-domain and multiple-domain data communication capability for an IBM 4331 that may include communication adapters.

**alternate code.** A code, frequently ASCII, selected at session activation, to be used to encode end-user data in the request unit (RU) instead of using extended binary-coded decimal interchange code (EBCDIC).

**API.** Application program interface.

**application program interface (API).**

The formally defined programming language interface between an IBM system control program or program product and its user.

**basic information unit (BIU).** In SNA, 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).** In SNA, the unit of data and control information transmitted over a link by data link control.

**basic transmission unit (BTU).** In SNA, 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 of PIUs.

**begin bracket.** In SNA, 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 end bracket. See also bracket.

**bidder.** In SNA, the LU-LU half-session defined at session activation as having to request and receive permission from the other LU-LU half-session to begin a bracket. Contrast with first speaker. See also bracket protocol.

**BIND image.** In SNA, the session parameters passed in a Control Initiate (GINIT) request by the system services control point (SSCP) to the primary logical unit (PLU); the parameters specify the proposed protocol options for an LU-LU session.

**BIU.** Basic information unit.

**BIU segment.** In SNA, the portion of ~~a basic information unit (BIU) that is~~ contained within a path information unit (PIU). It 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.** In SNA, an optional function of path control that combines multiple path information units (PIUs) into a single basic transmission unit (BTU).

**Note:** When blocking is not done, a BTU consists of one PIU.

**BLU.** Basic link unit.

**boundary function.** In SNA, (1) a capability of a subarea node to provide protocol support for adjacent peripheral nodes, such as: (a) transforming network addresses to local addresses, and vice versa; (b) performing session sequence numbering for low-function peripheral nodes; and (c) providing session-level pacing support. (2) The component that provides these capabilities. (3) See also path control network, network addressable unit.

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

**Note:** A subarea node may be a boundary node, an intermediate routing node, both, or neither, depending on how it is used in the network.

**bracket.** In SNA, 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. A bracket must be completed before another bracket can be started. Examples of brackets are data base inquiries/replies, update transactions, and remote job entry output sequences to work stations. See begin bracket, end bracket, RU chain.

**bracket protocol.** In SNA, a data flow control protocol in which exchanges

between the two LU-LU half-sessions are achieved through the use of brackets, with one LU 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, first speaker.

**BTU.** Basic transmission unit.

**CDRM.** Cross-domain resource manager.

**chain.** See RU chain.

**channel.** See data channel.

**change-direction protocol.** In SNA, a data flow control protocol in which the sending logical unit (LU) stops sending normal-flow requests, signals this fact to the receiving LU using the change-direction indicator (in the request header of the last request of the last chain), and prepares to receive requests.

**character-coded request.** In SNA, a request that is encoded and transmitted as a character string: the format indicator in the request header (RH) for the request is set to zero. Synonymous with unformatted request. Contrast with field-formatted request.

**CICS/VS.** Customer Information Control System/Virtual Storage.

**class of service.** In SNA, a designation of the path control network characteristics, such as path security, transmission priority, and bandwidth, that apply to a particular session. The end user designates class of service at session initiation by using a symbolic name that is mapped into a list of virtual routes, any one of which can be selected for the session to provide the requested level of service.

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

**command.** In SNA, (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 adapter.** An optional hardware feature, available on certain processors, that permits communication lines to be attached to the processors.

**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 3705 Communications Controller.

**communication controller node.** A term used to refer to a subarea node containing no system services control point (SSCP).

**compaction.** In SNA, the transformation of data by packing two characters in a byte so as to take advantage of the fact that only a subset of the allowable 256 characters is used; the most frequently sent characters are compacted.

**compaction table.** In SNA, a table used by a sending LU-LU half-session to transform data so that fewer bytes are sent to the receiving half-session. The receiving LU-LU half-session uses the same table to reverse the process, thereby restoring the data to its original form. See also compaction.

**compression.** In SNA, the replacement of a string of up to 64 repeated

characters by an encoded control byte to reduce the length of the data stream sent to the LU-LU session partner. The encoded control byte is followed by the character that was repeated (unless that character is the prime compression character, typically the space character). See also string control byte.

**concurrent control count.** In SNA, the number of control points concurrently controlling a network resource. See also share limit.

**configuration services.** In SNA, 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 maintenance services, management services, network services, physical unit control point, session services, SSCP

**connection point manager.** In SNA, a component of the transmission control layer that: (1) performs session-level pacing of normal-flow requests, (2) checks sequence numbers of received request units, (3) verifies that request units do not exceed maximum permissible size, (4) routes incoming request units to their destinations in the half-session, and (5) enciphers and decipheres FMD request units when cryptography is selected. The connection point manager coordinates the normal and expedited flows for one half-session.

**Note:** The sending connection point manager in a half-session builds the request/response header (RH) for outgoing request/response units, and the receiving connection point manager interprets the request/response headers

that precede incoming request/response units.

**control point.** In SNA, a physical unit control point (PUCP) or a system services control point (SSCP).

**cross-domain.** In SNA, pertaining to control or resources involving more than one domain.

**cross-domain keys.** In SNA, a pair of cryptographic keys used by a system services control point (SSCP) to encipher the session cryptography key that is sent to another SSCP and to decipher the session cryptography key that is received from the other SSCP during initiation of cross-domain LU-LU sessions that use session-level cryptography.

**cross-domain LU-LU session.** In SNA, a session between logical units (LUs) in different domains.

**cross-domain resource manager (CDRM).** In ACF/TCAM and ACF/VTAM, the functions of the system services control point (SSCP) that control initiation and termination of cross-domain sessions.

**Note:** CDRM functions span domains; "cross-domain" is therefore valid here, but is invalid with respect to LUs and resources, which cannot span domains. Contrast with same-domain session.

**cross-subarea.** In SNA, pertaining to control or resources involving more than one subarea node.

**cryptographic algorithm.** A set of rules that specify the mathematical steps required to encipher and decipher data.

**cryptographic key.** In systems using the Data Encryption Standard (DES) algorithm, a 64-bit value (containing 56 independent bits and 8 parity bits) provided as input to the algorithm in

determining the output of the algorithm. See also cross-domain keys, session cryptography key.

**cryptographic session.** An LU-LU session in which a function management data (FMD) request may be enciphered before it is transmitted and deciphered after it is received. See also mandatory cryptographic session, selective cryptographic session.

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

**DAF.** Destination address field.

**DAF'.** Destination address field prime.

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

**data circuit.** Synonym for link connection.

**data count field (DCF).** In SNA, a binary count of the number of bytes in the basic information unit (BIU) or BIU segment associated with the transmission header (TH).

**data encrypting key.** In SNA, a key used to encipher and decipher data transmitted in a session that uses cryptography. See also session cryptography key.

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

**data flow control (DFC).** In SNA, a request/response unit (RU) category used for requests and responses exchanged between the data flow control layer in one half-session and the data flow control layer in the session partner.

**data flow control (DFC) layer.** In SNA, 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.** In SNA, 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.** In SNA, a continuous stream of data elements being transmitted, or intended for transmission, in character or binary-digit form, using a defined format.

**data stream format.** In SNA, the format of the data elements (end-user data) in the request unit (RU). See also SNA character string (SCS), SNA 3270 data stream.

**DCF.** Data count field.

**DEF.** Destination element field.

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

**delayed-request mode.** In SNA, an operational mode in which the sender may

continue sending request units on the normal flow after sending a definite-response request chain on that flow, without waiting to receive the response to that chain. Contrast with immediate-request mode.

**delayed-response mode.** In SNA, an operational mode in which the receiver of normal-flow 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.

**Note:** An exception is the response to the DFC request CHASE: all responses to normal-flow request units received before CHASE must be sent before the response to CHASE is sent.

**destination address field (DAF).** In SNA, a field in a FIDO or FID1 transmission header that contains the network address of the destination. See also destination address field prime (DAF'), destination element field (DEF), destination subarea field (DSAF), format identification (FID) field, local session identification (LSID). Contrast with origin address field (OAF).

**destination address field prime (DAF').** In SNA, a field in a FID2 transmission header that contains the local address of the destination network addressable unit (NAU). See also destination address field (DAF), format identification (FID) field. Contrast with origin address field prime (OAF').

**destination element field (DEF).** In SNA, a field in a FID4 transmission header that contains an element address which, combined with the subarea address in the destination subarea field (DSAF), gives the complete network address of the destination network addressable unit (NAU). See also format identification (FID) field. Contrast with origin element field (OEF).

**destination subarea field (DSAF).** In SNA, a field in a FID4 transmission

header that contains a subarea address which, combined with the element address in the destination element field (DEF), gives the complete network address of the destination network addressable unit (NAU). See also format identification (FID) field. Contrast with origin subarea field (OSAF).

**DFC.** Data flow control.

**distributed function.** In SNA, functions, such as network management, processing, and error recovery operations, that are situated in different places, as contrasted with functions that are concentrated at a central location.

**DLC.** Data link control.

**domain.** In SNA, 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.

**domain operator.** In a multiple-domain network, the person or program that controls the operation of the resources controlled by one system services control point (SSCP). Contrast with network operator.

**DSAF.** Destination subarea field.

**element address.** In SNA, a value in the element address field of the network address identifying a particular resource within a subarea. See also subarea address.

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

**end user.** In SNA, the ultimate source or destination of application data flowing through an SNA network. An end user may be an application program or a terminal operator.

**end-user to SSCP echo check.** Synonym for LU connection test.

**ER.** Explicit route.

**exception request (EXR).** In SNA, a request that replaces another message unit in which an error has been detected.

**Note:** The exception request contains a 4-byte sense field that identifies the error in the original message unit and, except for some path errors, is sent to the destination of the original message unit; if possible, the sense data is returned in a negative response to the originator of the replaced message unit.

**exception response.** In SNA, a value in the form-of-response-requested field of a request header: the receiver is requested 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. See also negative response.

**expedited flow.** In SNA, a data flow designated in the transmission header (TH) that is used to carry network control, session control, and various data flow control request/response units (RUs); the expedited flow is separate from the normal flow (which carries primarily end-user data) and can be used for commands that affect the normal flow. Contrast with normal flow. See also isolated pacing response.

**Note:** The normal and expedited flows move in both the primary-to-secondary and secondary-to-primary directions. Requests and responses on a given flow (normal or expedited) usually are processed sequentially within the path,

but the expedited flow traffic may be moved ahead of the normal-flow traffic within the path at queuing points in the half-sessions and for half-session support in boundary functions.

**explicit route (ER).** In SNA, the path control network components, including a specific set of one or more transmission groups, 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.

**explicit-route length.** In SNA, the number of transmission groups in an explicit route.

**EXR.** Exception request.

**FID.** Format identification.

**field-formatted request.** In SNA, a request that is encoded into fields, each having a specified format such as binary codes, binary counts, bit-significant flags, and symbolic names; a format indicator in the request/response header (RH) for the request is set to zero. Synonymous with formatted request. Contrast with character-coded request.

**first speaker.** In SNA, the LU-LU half-session defined at session activation as: (1) able to begin a bracket without requesting permission from the other LU-LU half-session to do so, and (2) winning contention if both half-sessions attempt to begin a bracket simultaneously. Contrast with bidder. See also bracket protocol.

**flow control.** In SNA, 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; that is, to neither overflow the buffers at the receiver or at intermediate routing

nodes, nor leave the receiver waiting for more message units. See also pacing, session-level pacing, virtual-route (VR) pacing.

**FMD.** Function management data.

**FMD services layer.** In SNA, the layer within a half-session that routes FMD requests and responses to particular NAU services manager components and that provides session network services or session presentation services, depending on the type of session.

**FMH.** Function management header.

**format identification (FID) field.** In SNA, 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 in accordance with the types of nodes between which they pass.

**Note:** There are six FID types:

- FID0, 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.
- FID1, used for traffic between adjacent subarea nodes when either or both nodes do not support explicit-route and virtual-route protocols.
- FID2, used for traffic between a subarea node and an adjacent PU type 2 peripheral node.
- FID3, used for traffic between a subarea node and an adjacent PU type 1 peripheral node.
- FID4, used for traffic between adjacent subarea nodes when both nodes support explicit-route and virtual-route protocols.
- FIDF, 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.

**formatted request.** Synonym for field-formatted request.

**formatted system services (FSS).** A facility that provides certain system services as a result of receiving a field-formatted command, such as an INITIATE or TERMINATE command. Contrast with unformatted system services.

**frame.** In Synchronous Data Link Control (SDLC), a basic link unit (BLU).

**FSS.** Formatted system services.

**function management data (FMD).** In SNA, an RU category used for end-user data exchanged between logical units (LUs) and for requests and responses exchanged between network services components of LUs, PUs, and SSCPs.

**function management data (FMD) services.** In SNA, a generic term for session network services and session presentation services, both of which process FMD requests and responses.

**function management header.** In SNA, one or more headers, optionally present in the leading request units (RUs) of an RU chain, that allow one half-session in an LU-LU session to: (1) select a destination as the session partner and control the way the end-user data it sends is handled at the destination, (2) change the destination or the characteristics of the data during the session, and (3) transmit between session partners status or user information about the destination (for example, whether it is a program or a device).

**Note:** FM headers can be used by LU types 0, 1, 4, and 6.

**function management (FM) profile.** In SNA, a specification of various data



flow control protocols (such as RU chains and data flow control requests) and FMD options (such as use of FM headers, compression, and alternate codes) supported for a particular session. Each function management profile is identified by a number.

**half-session.** In SNA, a component that provides FMD services, data flow control, and transmission control for one of the sessions of a network addressable unit (NAU). See also primary half-session, secondary half-session.

**host node.** A subarea node that contains a system services control point (SSCP); for example, a System/370 computer with OS/VS2 and ACF/TCAM.

**immediate-request mode.** In SNA, an operational mode in which the sender stops sending request units (RUs) on a given flow (normal or expedited) after sending a definite-response request chain on that flow until that chain has been responded to. Contrast with delayed-request mode. See also immediate-response mode.

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

**IMS/VS.** Information Management System/Virtual Storage.

**initial chaining value.** In SNA, an eight-byte pseudo-random number used to verify that both ends of a session with cryptography have the same session cryptography key. The initial chaining value is also used as input to the the Data Encryption Standard (DES) algorithm to encipher or decipher data in a session with cryptography. See also session cryptography seed.

**Note:** A new initial chaining value is selected for each session.

**initiation.** See LU-LU session initiation. See also session-initiation request.

**intermediate routing function.** In SNA, 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.

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

**Note:** A subarea node may be a boundary node, an intermediate routing node, both, or neither, depending on how it is used in the network.

**IPR.** Isolated pacing response.

**isolated pacing response (IPR).** In SNA, a response to a session-level pacing request that is sent independently of any particular request (without any correlation of sequence numbers) and that signals the readiness of the receiver to receive an additional pacing group.

**Note:** An IPR may be sent on either the expedited or normal flow and is particularly useful when no other response is available; for example, when operating under no-response protocols.

**JES3.** Job Entry Subsystem 3.

**key-encrypting key.** In SNA, a key used in sessions with cryptography to encipher and decipher other keys. Contrast with data encrypting key.

**layer.** In SNA, 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 NAU services manager layer, FMD services

layer, data flow control layer, transmission control layer, path control layer, data link control layer.

**link.** In SNA, the combination of the link connection and the link stations joining network nodes; 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.** In SNA, the physical equipment providing two-way communication between one link station and one or more other link stations; for example, a communication line and data circuit terminating equipment (DCE). Synonymous with data circuit.

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

**link station.** In SNA, the combination of hardware and software that allows a node to attach to and provide control for a link. See also adjacent link station, primary link station, secondary link station.

**link test.** In SNA, a test in which one link station returns data received from another link station without changing the data in order to test the operation of the link.

**Note:** Three tests can be made; they differ in the resources that are dedicated during the test. A link test, level 0 requires a dedicated subarea node, link, and secondary link station. A link test, level 1 requires a dedicated link and secondary link

station. A link test, level 2 requires only the dedicated link station.

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

**local address.** In SNA, an address used in a peripheral node in place of a network address and transformed to or from a network address by the boundary function in a subarea node. See also network address.

**local session identification (LSID).** In SNA, a field in a FID3 transmission header that contains an indication of the type of session (SSCP-PU, SSCP-LU, or LU-LU) and the local address of the peripheral logical unit (LU) or physical unit (PU).

**logical unit (LU).** In SNA, 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), peripheral LU, physical unit, system services control point, primary logical unit, secondary logical unit.

**logical unit (LU) services.** In SNA, capabilities in a logical unit to: (1) receive requests from an end user and, in turn, issue requests to the system services control point (SSCP) in order to perform the requested functions, typically for session initiation; (2) receive requests from the SSCP, for example to activate LU-LU sessions via Bind Session requests; and (3) provide session presentation and other services for LU-LU sessions. See also physical unit (PU) services, SSCP services.

**LSID.** Local session identification.

**LU.** logical unit.

**LU type 0.** In SNA, a type of LU that uses SNA-defined protocols for transmission control and data flow control, but uses end-user or product-defined protocols to augment or replace FMD services protocols. For example, an LU for an application program using IMS/VS and an IBM 3600 Finance Communication System in which the operator of the 3600 terminal is updating the passbook balance for a customer's savings account.

**LU type 1.** In SNA, a type of LU for an application program that communicates with single- or multiple-device data processing terminals 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 3767 Communication Terminal in which the terminal operator is correcting a data base that the application program maintains. The data stream is the SNA character string (SCS).

**LU type 2.** In SNA, a type of LU for an application program that communicates with a single display terminal in an interactive environment, using the SNA 3270 data stream. For example, an LU for an application program that uses IMS/VS and an IBM 3277 Display Station, in which the 3277 operator is creating and sending data to the application program.

**LU type 3.** In SNA, 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 3284 Printer attached to an IBM 3791 Controller.

**LU type 4.** In SNA, a type of LU for: (1) an application program that communicates with a single- or multiple-device data processing or word processing terminal 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 (SCS) for data processing environments and Office Information Interchange Level 2 for word processing environments.

**LU type 6.** In SNA, a type of LU for an application subsystem that is to communicate with another application subsystem in a distributed 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 services manager.** In SNA, an SNA component that provides a logical unit (LU) with network services and end-user to end-user services. The LU services manager provides services for all half-sessions within the LU.

**maintenance services.** In SNA, one of the types of network services in system services control points (SSCPs) and physical units (PUs). maintenance services provide facilities for testing links and nodes and for collecting and recording error information. See also configuration services, management services, network services, session services.

**management services.** In SNA, one of the types of network services in system services control points (SSCPs) and logical units (LUs). Management services forward requests for network data, such as error statistics, and deliver the data in reply. See also configuration services, maintenance services, network services, session services.

**mandatory cryptographic session.** A cryptographic session in which all outgoing data is enciphered and all

incoming data is deciphered. Contrast with selective cryptographic session.

**master cryptography key.** In SNA, a unique cryptography key provided by installation management for each node of the network. The system services control point (SSCP) uses the master cryptography keys to encipher session cryptography keys in order to protect their identities while sending them through the network to the session partners.

**message segment.** In ACF/TCAM, that portion of a message that is contained within a single request unit.

**message unit.** In SNA, a generic term for the unit of data processed by any layer; for example, a basic information unit (BIU), a path information unit (PIU), a request/response unit (RU).

**meta-implementation.** In SNA, an architectural (or design) description in a form similar to actual implementations of an architecture; for example, one that uses a programming language to specify a human- or machine-executable model that follows the architectural rules, thereby defining those rules.

**multiple-domain network.** In SNA, a network with more than one system services control point (SSCP). Contrast with single-domain network.

**multiple explicit routes.** In SNA, two or more explicit routes between subarea nodes used to accommodate changes in network conditions such as traffic loads or route failures and to offer different classes of service.

**Multisystem Networking Facility.** An optional feature of ACF/TCAM and ACF/VTAM that permits these access methods, together with ACF/NCP/VS, to control a multiple-domain network.

**NAU.** Network addressable unit.

**NAU services.** In SNA, the functions provided by the NAU services manager layer and the FMD services layer.

**NAU services manager layer.** In SNA, the layer that: (1) controls network operations via LU-LU, SSCP-LU, SSCP-PU, and SSCP-SSCP sessions, and (2) coordinates end-user interactions on LU-LU sessions. See also configuration services, session services, maintenance services, management services.

**NC.** Network control.

**NCP node.** A subarea node that contains an ACF/NCP program but does not contain a system services control point (SSCP).

**negative response.** In SNA, 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.** In SNA, a capability that allows two LU-LU half-sessions to negotiate the parameters of a session when the session is being activated.

**network.** In data processing, a user-application network. See also path control network, public network, SNA network, user-application network.

**network address.** In SNA, 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 transforms local addresses to network addresses and vice versa. See also local address, network name.

**network addressable unit (NAU).** In SNA, 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 (PC) 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 configuration tables.** In ACF/TCAM and ACF/VTAM, the tables through which the system services control point (SSCP) interprets the network configuration.

**network control (NC).** In SNA, a request/response unit (RU) category used for requests and responses exchanged between physical units (PUs) for such purposes as activating and deactivating explicit and virtual routes and sending load modules to adjacent peripheral nodes. See also data flow control, function management data, session control.

**network name.** In SNA, 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.** In SNA, a person or program responsible for controlling the operation of all or part of a network. See also node operator.

**network services.** In SNA, the services within network addressable units (NAUs) that control network operation via SSCP-SSCP, SSCP-PU, and SSCP-LU sessions. See configuration services, maintenance services, management services, session services.

**network services header.** In SNA, a 3-byte field in an FMD request/response unit (RU) flowing in an SSCP-LU, SSCP-PU, or SSCP-SSCP session. The network services header is used primarily to identify the network services category of the RU (for

example, configuration services, session services) and the particular request code within a category.

**node.** In SNA, an endpoint of a link or a junction common to two or more links in a network. Nodes can be distributed to host processors, communication controllers, or terminals. Nodes can vary in routing and other functional capabilities. See also node type, peripheral node, subarea node.

**node operator.** In SNA, a person or program responsible for controlling the operation of a node via the physical unit control point (PUCP). See also network operator.

**node type.** In SNA, a designation of a node according to the protocols it supports and the network addressable units (NAUs) that it can contain. Four types are defined: 1, 2, 4, and 5. Type 1 and type 2 nodes are peripheral nodes; type 4 and type 5 nodes are subarea nodes. See also physical unit type.

**no response.** In SNA, a value in the form-of-response-requested field of the request header (RH) indicating that no response is to be returned to the request, whether or not the request is received and processed successfully. Contrast with definite response, exception response.

**normal flow.** In SNA, a data flow designated in the transmission header (TH) that is used primarily to carry end-user data. The rate at which requests flow on the normal flow can be regulated by session-level pacing. Contrast with expedited flow.

**Note:** The normal and expedited flows move in both the primary-to-secondary and secondary-to-primary directions. Requests and responses on a given flow (normal or expedited) usually are processed sequentially within the path, but the expedited-flow traffic may be moved ahead of the normal-flow traffic within the path at queuing points in the

half-sessions and for half-session support in boundary functions.

**NS.** Network services.

**OAF.** Origin address field.

**OAF'.** Origin address field prime

**OEF.** Origin element field.

**origin address field (OAF).** In SNA, a field in a FID0 or FID1 transmission header that contains the address of the originating network addressable unit. Contrast with destination address field. See also format identification (FID) field, local session identification (LSID), origin address field prime (OAF'), origin element field (OEF), origin subarea field (OSAF).

**origin address field prime (OAF').** In SNA, a field in a FID2 transmission header that contains the local address of the originating network addressable unit (NAU). Contrast with destination address field prime (DAF'). See also origin address field (OAF), format identification (FID) field.

**origin element field (OEF).** In SNA, a field in a FID4 transmission header that contains an element address, which combined with the subarea address in the origin subarea field (OSAF), gives the complete network address of the originating network addressable unit (NAU). Contrast with destination element field (DEF). See also format identification (FID) field.

**origin subarea field (OSAF).** In SNA, a field in a FID4 transmission header that contains a subarea address, which combined with the element address in the origin element field (OEF), gives the complete network address of the originating network addressable unit (NAU). Contrast with destination subarea field (DSAF). See also format identification (FID) field.

**OSAF.** Origin subarea field.

**other-domain resource.** In SNA, a resource owned by a domain other than the domain in which it is known only by its network name and its associated system services control point (SSCP).

**Note:** A resource does not span domains, as may an LU-LU session.

**pacing.** In SNA, a technique by which a receiving component controls the rate of transmission of 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.** In SNA, (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 on the normal flow in one direction in 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.** In SNA, (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.** In SNA, 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. See also isolated pacing response.

**parallel links.** In SNA, two or more links between adjacent subarea nodes.

**parallel sessions.** In SNA, 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.** In SNA, the series of path control network components (path control and data link control) that are traversed by the information exchanged between two network addressable units (NAUs). A path consists of a virtual route and its route extension, if any. See also explicit route.

**path control (PC) layer.** In SNA, 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, data link control layer, transmission control (TC) layer.

**Note:** The unit of control information built by the sending path control component is the transmission header (TH), attached to the BTU; the TH is interpreted by the receiving path control component. The path control layer in subarea nodes consists of explicit route control, transmission group control, virtual route control, and boundary-function path control.

**path control (PC) network.** In SNA, the part of the SNA network that includes the data link control and path control layers. See also boundary function, SNA network, user-application network.

**path information unit (PIU).** In SNA, 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.** In SNA, a link that connects a peripheral node to a subarea node. See also route extension (REX), subarea link.

**peripheral logical unit (LU).** In SNA, a logical unit in a peripheral node.

**peripheral node.** In SNA, 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 physical unit (PU).** In SNA, a physical unit in a peripheral node.

**physical unit (PU).** In SNA, 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. See also peripheral physical unit, subarea physical unit.

**Note:** An SSCP activates a session with the physical unit in order to indirectly manage, through the PU, resources of the node such as attached links and adjacent link stations.

**physical unit control point (PUCP).** In SNA, a component that provides a subset of system services control point (SSCP) functions for activating the physical unit (PU) within its node and its local link resources. Each peripheral node and each subarea node without an SSCP contains a PUCP.

**physical unit (PU) services.** In SNA, the components within a physical unit (PU) that provide configuration services and maintenance services for SSCP-PU sessions. See also logical unit (LU) services, SSCP services.

**PIU.** Path information unit.

**PLU.** Primary logical unit.

**point-to-point line.** A link that connects a single remote link station to a node; it may be switched or non-switched.

**positive response.** In SNA, a response indicating that a request was successfully received and processed. Contrast with negative response.

**presentation medium.** In SNA, a medium that is shared by the logical unit and the terminal operator. Examples of presentation media are the paper in a printer, the screen of a display, a magnetic card that is inserted into and removed from a terminal, and magnetic tapes or disks that can be mounted on and removed from the terminal. Keyboards and control panels are part of the presentation medium when they allow the operator to alter information on the medium or control its operation.

**presentation services (PS).** See session presentation services.

**primary half-session.** In SNA, the half-session that sends the session activation request. See also primary logical unit. Contrast with secondary half-session.

**primary link station.** In SNA, the link station on a link that is responsible for the control of that link. A link has only one primary link station. All traffic over the link is between the primary link station and a secondary link station. Contrast with secondary link station.

**primary logical unit (PLU).** In SNA, the logical unit (LU) that contains the primary half-session for a particular LU-LU session. Contrast with secondary logical unit.

**Note:** A particular logical unit may contain primary and secondary half-sessions for different active LU-LU sessions.

**prime compression character.** In SNA, the character selected to be represented, whenever it occurs in a string, by a single encoded control byte. Other compression characters require that each of their strings be represented by a control byte followed by the character itself. Therefore, a prime compression character string is represented by one byte, whereas other compression character strings are represented by two bytes.

**protocol.** In SNA, 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.

**PU.** Physical unit.

**public network.** A network established and operated by communication common carriers or telecommunication Administrations for the specific purpose of providing circuit-switched, packet-switched, and leased-circuit services to the public. Contrast with user-application network.

**PUCP.** Physical unit control point.

**PU-PU flow.** In SNA, the exchange between physical units (PUs) of network control requests and responses.

**PU services manager.** In SNA, an SNA component that provides network services for all half-sessions within the physical unit (PU).

**PU type.** See physical unit type.



**receive pacing.** In SNA, the pacing of message units that a component is receiving. See also send pacing.

**reply.** In SNA, 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.** In SNA, a message unit that signals initiation of a particular action or protocol. For example, INITIATE SELF is a request for activation of an LU-LU session.

**request header (RH).** In SNA, a request unit (RU) header preceding a request unit.

**request unit (RU).** In SNA, 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).** In SNA, 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).** In SNA, a generic term for a request unit or a response unit.

**response.** In SNA, (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).** In SNA, 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 isolated pacing response, negative response, pacing response, positive response.

**response unit (RU).** In SNA, 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 (REX).** In SNA, 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.** In SNA, the forwarding of a message unit along a particular path through a network as determined by parameters carried in the message unit, such as the destination network address in a transmission header.

**RU.** Request/response unit.

**RU chain.** In SNA, a set of related request/response units (RUs) that are consecutively transmitted on a particular normal or expedited data flow. The request RU chain is the unit of recovery: if one of the RUs in the chain cannot be processed, the entire chain is discarded.

**Note:** Each RU belongs to only one chain, which has a beginning and an end indicated via control bits in request/response headers within the RU chain. Each RU can be designated as first-in-chain (FIC), last-in-chain (LIC), middle-in-chain (MIC), or only-in-chain (OIC). Response units and

expedited-flow request units are always sent as only-in-chain.

**same-domain LU-LU session.** In SNA, an LU-LU session between logical units (LUs) in the same domain. Contrast with cross-domain LU-LU session.

**SC.** Session control.

**SCB.** String control byte.

**SCS.** SNA character string.

**SDLC.** Synchronous Data Link Control.

**secondary half-session.** In SNA, the half-session that receives the session-activation request. See also secondary logical unit. Contrast with primary half-session.

**secondary link station.** In SNA, any link station on a link, using a primary-secondary protocol, that is not the primary link station. A secondary link station can exchange data only with the primary link station; no data traffic flows from one secondary link station to another. Contrast with primary link station.

**secondary logical unit (SLU).** In SNA, the logical unit (LU) that contains the secondary half-session for a particular LU-LU session. Contrast with primary logical unit.

**Note:** A logical unit may contain secondary and primary half-sessions for different active LU-LU sessions.

**secondary station.** See secondary link station.

**segmenting of BIUs.** In SNA, 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.

**Note:** When segmenting is not done, a PIU contains a complete BIU.

**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.** In SNA, pacing of message units that a component is sending. See also receive pacing.

**session.** In SNA, 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. Sessions compete for network resources such as the links within the path control network. See half-session, LU-LU session, SSCP-LU session, SSCP-PU session, SSCP-SSCP session. See also LU type, PU-PU flow.

**Note:** For routing purposes, each session is identified by the network (or local) address of the session partners.

**session activation.** In SNA, 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.** In SNA, 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 and ACTPU. Contrast with session deactivation request.

**session control (SC).** In SNA, (1) one of the components of transmission control. Session control is used to purge data flowing in a session after an unrecoverable error occurs, to resynchronize the data flow after such an error, and to perform cryptographic verification. (2) An RU category used for requests and responses exchanged between the session control components of a session and for session activation/deactivation requests and responses.

**session count.** In SNA, (1) the number of currently active LU-LU sessions for a particular logical unit. (2) The number of currently active sessions for a particular virtual route.

**session cryptography key.** In SNA, a data encrypting key used to encipher and decipher function management data (FMD) requests transmitted in an LU-LU session that uses cryptography.

**session cryptography seed.** In SNA, an 8-byte, non-zero, pseudo-random number used to verify that both half-sessions have the same session cryptography key, and used thereafter as the initial chaining value.

**Note:** The secondary half-session generates the session cryptography seed upon receiving the BIND request specifying session-level cryptography, enciphers the seed under the session cryptography key received in the BIND request, and sends the seed to the primary half-session in the BIND response. Upon receiving the seed, the primary half-session transforms and re-enciphers the seed under the session cryptography key and returns the modified seed in the Cryptography Verification request.

**session deactivation.** In SNA, the process of exchanging a session deactivation request and response between network addressable units (NAUs). Contrast with session activation.

**session deactivation request.** In SNA, a request that deactivates a session between two network addressable units (NAUs); for example, UNBIND and DACTPU. Contrast with session activation request.

**session initiation.** See LU-LU session initiation.

**session-initiation request.** In SNA, an Initiate or logon request from a logical unit (LU) to a system services control point (SSCP) that an LU-LU session be activated.

**session limit.** In SNA, the maximum number of concurrently active LU-LU sessions a particular logical unit (LU) can support.

**session-level pacing.** In SNA, a flow control technique that permits a receiving half-session to control the data transfer rate (the rate at which it receives request units) on the normal flow. 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 network services.** In SNA, network services that are performed on a half-session by half-session basis, rather than for the network addressable unit (NAU) as a whole.

**session parameters.** In SNA, the parameters that specify or constrain the protocols (such as bracket protocol and pacing) for a session between two network addressable units (NAUs).

**session partner.** In SNA, one of the two network addressable units (NAUs) having an active session.

**session presentation services.** In SNA, a component of the FMD services layer that provides, within LU-LU sessions,

services for the application programmer or terminal operator such as formatting data to be displayed or printed.

**session sequence identifier.** In SNA, an identifier in the sequence number field that may uniquely identify a request unit, typically on the expedited flow, until (and if) that request unit is responded to. Unlike a session sequence number, the identifier is not necessarily updated sequentially.

**session sequence number.** In SNA, a sequentially incremented identifier that is assigned by data flow control to each request unit on a given normal flow of a session, typically an LU-LU session, and is checked by transmission control. The identifier is carried in the transmission header (TH) of the path information unit (PIU) and is returned in the TH of any associated response. Contrast with session sequence identifier, virtual-route sequence number.

**session services.** In SNA, 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, maintenance services, management services.

**session termination.** See LU-LU session termination.

**shared control.** In SNA, sequential or concurrent control of network resources - physical units (PUs), logical units (LUs), links, link stations, and their associated resources - by two or more control points. See also concurrent control count, share limit.

**share limit.** In SNA, the maximum number of control points that can concurrently control a network resource. See concurrent control count, shared control.

**single-domain network.** In SNA, a network with one system services control point (SSCP). Contrast with multiple-domain network.

**SLU.** Secondary logical unit.

**SNA.** Systems Network Architecture.

**SNA character string (SCS).** In SNA, a data stream composed of EBCDIC controls, optionally intermixed with end-user data, that is carried within a request/response unit.

**SNA network.** In SNA, 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 node.** In SNA, a node that supports SNA protocols.

**SNA station.** A station that supports SNA protocols.

**SNA terminal.** A terminal that supports SNA protocols.

**SNF.** Sequence number field.

**SSCP.** System services control point.

**SSCP ID.** In SNA, a number that uniquely identifies a system services control point (SSCP). The SSCP ID is used in session activation requests sent to physical units (PUs) and to other SSCPs.

**SSCP-LU session.** In SNA, 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.** In SNA, 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 services.** In SNA, the components within a system services control point (SSCP) that provide configuration, maintenance, management, network, and session services for SSCP-LU, SSCP-PU, and SSCP-SSCP sessions. See also logical unit (LU) services, physical unit (PU) services.

**SSCP services manager.** In SNA, an SNA component that provides network services for all the half-sessions of the system services control point (SSCP).

**SSCP-SSCP session.** In SNA, a session between the system services control point (SSCP) in one domain and the SSCP in another domain. An SSCP-SSCP session is used to initiate and terminate cross-domain LU-LU sessions.

**station.** (1) A link station. (2) One or more computers, terminals, devices, and associated programs at a particular location.

**string control byte (SCB).** In SNA, an optional control byte in the SNA character string (SCS) data stream that identifies how end-user data is compressed or compacted. See also compaction, compression.

**subarea.** In SNA, 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.** In SNA, a value in the subarea field of the network address

that identifies a particular subarea. See also element address.

**subarea link.** In SNA, a link that connects two subarea nodes. See also peripheral link.

**subarea LU.** In SNA, a logical unit in a subarea node. Contrast with peripheral LU.

**subarea node.** In SNA, 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 support for peripheral nodes. See also subarea link, node type.

**subarea PU.** In SNA, a physical unit in a subarea node.

**sync point.** Synonym for synchronization point.

**synchronization point.** In SNA, a point at which the processing of end-user data is checkpointed. Synonymous with sync point.

**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 full-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 services control point (SSCP).** In SNA, a focal point within an SNA network for managing the configuration, coordinating network operator and problem determination requests, and providing directory support and other

session services for end users of the network. Multiple SSCPs, cooperating as peers with one another, can divide the network into domains of control, with each SSCP having a hierarchical control relationship to the physical units and logical units within its own domain. See also 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.

**Note:** The purpose of the layered structure of SNA is to allow the ultimate origins and destinations of information—that is, the end users—to be independent of, and unaffected by, the way in which the specific SNA network services and facilities used for information exchange are provided.

**TC.** Transmission control.

**terminal node.** A peripheral node that is not user-programmable, having less processing capability than a cluster controller node. Examples are the IBM 3277, 3767, 3614, and 3624.

**termination.** See LU-LU session termination.

**TGID.** Transmission group identifier.

**TH.** Transmission header.

**transmission control (TC) layer.** In SNA, 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 control has two components: the connection point manager and session control.

**transmission group.** In SNA, a group of links between adjacent subarea nodes,

appearing as a single logical link for routing of messages.

**Note:** A transmission group may consist of one or more SDLC links (parallel links) or of a single System/370 channel.

### **transmission group identifier**

**(TGID).** In SNA, a set of three values, unique for each transmission group, consisting of the subarea addresses of the two adjacent nodes connected by the transmission group, and the transmission group number (1-255).

**transmission header (TH).** In SNA, 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.** In SNA, 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.

**transmission services (TS) profile.** In SNA, a specification in a session activation request (and optionally, in the responses) of transmission control (TC) protocols (such as session-level pacing and the usage of session-control requests) to be supported by a particular session. Each defined transmission services profile is identified by a number.

**TS.** Transmission services.

**unformatted request.** Synonym for character-coded request.

**unformatted system services (USS).** A system services control point (SSCP) facility that translates a character-coded request such as a LOGON or LOGOFF request, into a

field-formatted request for processing by formatted system services and translates field-formatted replies and responses into character-coded requests for processing by a logical unit.

**uninterpreted name.** In SNA, a character string that a system services control point (SSCP) can convert into the network name of a logical unit (LU).

**user-application network.** In SNA, a configuration of data processing products (such as processors, controllers, and terminals) established and operated by users for the purpose of data processing or information exchange, which may use services offered by common carriers or telecommunication Administrations. Contrast with public network.

**virtual route (VR).** In SNA, 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 intranode 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 (REX).

**virtual-route identifier (VRID).** In SNA, a virtual-route number and a transmission-priority number that, when combined with the subarea addresses for the subareas at each end of a route, identify the virtual route.

**virtual-route (VR) pacing.** In SNA, a flow control technique used by the virtual-route control component of path control at each end of a virtual route to control 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.

**virtual-route sequence number.** In SNA, a sequential identifier assigned by the virtual-route control component of path control to each path information unit (PIU) that flows over a virtual route. It is stored in the transmission header of the PIU. Contrast with session sequence number, session sequence identifier.

**VR.** Virtual route.

**VRID.** Virtual-route identifier.

**window.** Synonym for pacing group.

**window size.** Synonym for pacing-group size.





## INDEX

### A

access method, SNA 4-21  
  and subarea node 1-2  
  network addresses assigned by 2-7  
  operator commands 2-16  
  routes activated and deactivated  
  by 4-16  
ACF/NCP 1-4, 4-21  
  and network addresses 2-7  
ACF/TCAM 1-2  
  and network activation 2-9  
  and pacing-group size 4-20  
  and Programmed Cryptographic  
  Facility 5-21  
  and session reinitiation 4-25  
  checkpoint/restart capability  
  of 5-25  
ACF/TCAM options 2-22  
ACF/VTAM 1-2  
  and network activation 2-9  
  and pacing-group size 4-20  
  and Programmed Cryptographic  
  Facility 5-21  
ACF/VTAM options 2-21  
ACTIV operand 2-22  
Activate Connect In (ACTCONNIN)  
  request 2-15  
Activate Cross-Domain Resource  
  Management (ACTCDRM) request 5-26  
Activate Link (ACTLINK) request 2-15  
Activate Logical Unit (ACTLU)  
  request 2-15, 5-11, 5-26  
Activate Physical Unit (ACTPU)  
  request 2-15, 2-16, 5-26  
Activate SNA Resource basic operator  
  command 2-22  
activating a cross-domain LU-LU session  
  (RU sequence) 5-33  
activating a same-domain LU-LU session  
  (RU sequence) 5-31  
activating an SSCP-SSCP session (RU  
  sequence) 5-32  
activation capabilities, benefits  
  of 2-19  
activation options 2-21  
activation overview 2-2-2-9  
activation requests 1-4, 5-26

ADDRESS operand of LINE macro  
  (ACF/NCP) 3-14  
adjacent link stations 2-5, 3-10, 3-12  
  example of data transmission  
  between 3-10  
Advanced Communications Function vi  
Advanced Communications Function/Network  
  Control Program  
  See ACF/NCP  
Advanced Communications  
  Function/Telecommunications Access  
  Method  
  See ACF/TCAM  
Advanced Communications Function/Virtual  
  Telecommunications Access Method  
  See ACF/VTAM  
application host 2-18  
automatic network shutdown 2-14

### B

basic information unit (BIU) 3-9  
basic link unit (BLU) 3-5, 3-10  
basic operator commands 2-22, 3-13  
basic transmission unit (BTU) 3-8  
BBI  
  See begin-bracket indicator (BBI)  
BCI  
  See begin-chain indicator (BCI)  
begin-bracket indicator (BBI) 5-7,  
  5-13, 5-14  
begin-chain indicator (BCI) 5-7, 5-8,  
  5-10  
between-brackets state 5-15  
Bid request 5-13, 5-14  
bidder 5-13, 5-14, 5-15  
Bind Failure (BINDF) request 5-4  
Bind image 5-4  
Bind image table 5-4  
Bind Session (BIND) request 2-16  
  See also Bind Session parameters  
  and Bind image 5-4  
  and bracket protocol 5-13  
  and class of service 4-14  
  and error-recovery  
  responsibilities 5-24  
  and profiles 5-26

- and session-level pacing 5-23
- receiver of (secondary LU) 5-5
- sender of (primary LU) 5-5
- Bind Session parameters 5-13, 5-15, 5-24
- BIU
  - See basic information unit (BIU)
- BLU
  - See basic link unit (BLU)
- boundary function (BF) 1-5, 4-6
  - See also node, boundary-function (BF) and address transformation 2-2
- boundary-function (BF) component 1-1, 1-5, 3-9
- boundary-function (BF) logical unit 3-9
- boundary-function (BF) path control 3-9
- boundary-function (BF) physical unit 3-9
- bracket indicators
  - See begin-bracket indicator (BBI)
  - See end-bracket indicator (EBI)
- bracket protocols 1-7, 5-13, 5-15
- bracket termination 5-13, 5-14
- brackets 5-13
- BTU
  - See basic transmission unit (BTU)
- BUILD macro (ACF/NCP) 4-28
- busy primary link station (SDLC sequence) 3-22
- busy secondary link station (SDLC sequence) 3-21

## C

- Cancel request 5-14
- cascaded activation of SNA resources 2-9, 2-13, 2-19
- cascaded deactivation of SNA resources 2-19
- CDI
  - See change-direction indicator (CDI)
- CDRM profile 5-26
- chaining of RUs
  - See RU chain
- Change Window Indicator (CWI) 4-21
- Change Window Reply Indicator (CWRI) 4-21
- change-direction indicator (CDI) 5-7, 5-15
- channel, data
  - See data channel
- Chase request 5-13
- CICS/VS 1-4, 5-25

- class of service (COS) 4-6
  - and list of virtual routes 4-13, 4-29, 4-30
  - class of service (COS) name 4-14, 4-30, 5-4
  - class of service (COS) table 4-29, 4-30
    - specifying in session-initiation request 5-4
- Clear request 5-18
- cluster controller 2-1
- CMC host
  - See communication management configuration
- CMI
  - See compression indicator (CMI)
- communication controller node
  - See node, NCP
- communication management configuration (CMC) 2-16, 2-19
  - host 2-18
- communication using brackets in half-duplex flip-flop mode (RU sequence) 5-38
- communication using half-duplex contention protocols (RU sequence) 5-40
- communication using half-duplex flip-flop protocols (RU sequence) 5-41
- compaction indicator (CPI) 5-30
- compression indicator (CMI) 5-30
- concurrent sharing of SNA resources 2-15
- congestion in network
  - See SNA network, congestion in
- congestion indicators, setting 4-36
- Connect Out (CONNOUT) request 2-15
- connection, link
  - See link connection
- Contact request 2-15, 2-16
- contention loser 5-15
- contention state 5-15
- contention winner 5-15
- control block, link station
  - See link station control block
- Control Initiate (CINIT) request 5-4
- control modes
  - See request/response control modes
- control options 2-21
- COS macro (ACF/TCAM) 4-30
- COS macro (ACF/VTAM) 4-29
- COS operand (ACF/TCAM IEDBENT macro) 4-30
- COS operand (ACF/VTAM MODEENT macro)
- COSEND macro (ACF/TCAM) 4-30

COSEND macro (ACF/VTAM) 4-29  
 COSTAB macro (ACF/TCAM) 4-30  
 COSTAB macro (ACF/VTAM) 4-29  
 CPI  
   See compaction indicator (CPI)  
 Cross-Domain Initiate (CDINIT)  
   request 5-4  
 cross-domain LU-LU session  
   See LU-LU session, cross-domain  
 cross-domain takedown sequence (RU  
   sequence) 5-37  
 cryptographic algorithm 5-20, 5-21  
 cryptographic key 5-21  
 cryptography 5-20  
 Customer Information Control System/VS  
   (CICS/VS)  
   See CICS/VS  
 CWI  
   See Change Window Indicator (CWI)  
 CWRI  
   See Change Window Reply Indicator  
   (CWRI)

## D

data channel 3-2, 3-4  
 data compaction 5-20, 5-29  
 data compression 5-20, 5-29  
 data encryption 5-21  
 Data Encryption Standard 5-21  
 data flow  
   expedited-flow 5-8  
     and immediate-request mode 5-12  
     and RU chains 5-10  
   managing 5-6, 5-30  
   normal-flow 5-8, 5-16  
     and brackets of RU chains 5-13  
     and RU chains 5-10  
     and send/receive modes 5-15  
   regulating 1-5, 4-17  
   synchronizing 4-19  
 data flow control (DFC) component 1-7,  
   5-5, 5-6  
 data handling 5-18  
 data handling protocols 5-18  
 data link control (DLC) components 1-4,  
   1-5  
 data link control layer 3-1, 3-10  
 data security 5-20  
 data stream 5-28  
   SNA character string (SCS) 5-28  
   3270 data stream 5-29  
 data transmission between nodes

data transmission from node to node  
   See transmitting data from node to  
   node  
 Deactivate Cross-Domain Resource  
   Management (DACTCDRM) request 5-25  
 Deactivate Logical Unit (DACTLU)  
   request 5-25  
 Deactivate Physical Unit (DACTPU)  
   request 5-25  
 Deactivate SNA Resource basic operator  
   command 2-22  
 deactivating a cross-domain LU-LU  
   session (RU sequence) 5-35  
 deactivating a same-domain LU-LU session  
   (RU sequence) 5-34  
 deactivation capabilities, benefits  
   of 2-19  
 deactivation options 2-21  
 deactivation overview 2-2-2-9  
 deactivation requests 1-4, 5-25  
 definite response 1 indicator  
   (DR1I) 5-7, 5-8  
 definite response 2 indicator  
   (DR2I) 5-7, 5-8  
 definite-response chain 5-10  
 delayed-request mode 5-12  
 delayed-response mode 5-12  
 destination network address  
   See network address, destination  
 destination subarea address 1-10, 3-10  
 disconnecting a secondary link station  
   (SDLC sequence) 3-17  
 display surface 5-29  
 domain 1-2, 1-13  
   allocation of resources to 2-15  
   control of 2-15  
 DR1I  
   See definite response 1 indicator  
   (DR1I)  
 DR2I  
   See definite response 2 indicator  
   (DR2I)  
 Dump 3705 NCP Storage basic operator  
   command 3-13  
 dynamic reconfiguration 2-13

## E

EBI  
   See end-bracket indicator (EBI)  
 ECI  
   See end-chain indicator (ECI)

## EDI

See enciphered-data indicator (EDI)  
element field (of network address) 2-2  
enciphered-data indicator (EDI) 5-7  
end user v, 4-10, 5-1  
end-bracket indicator (EBI) 5-7, 5-13  
end-chain indicator (ECI) 5-7, 5-8,  
5-10

## ERI

See exception-response indicator  
(ERI)  
error handling for routes 4-22, 4-26  
error recovery 5-24  
error, definition of 5-24  
exception request (EXR) 3-9, 5-7  
exception-response chain 5-10, 5-11  
exception-response indicator (ERI) 5-7  
expedited flow  
See data flow  
expedited-flow request  
See data flow, expedited-flow  
explicit route (ER) 1-10, 4-6, 4-8-4-13  
activating 2-24, 2-31, 4-16  
active 4-16  
as part of path 4-3  
deactivating 2-32  
defining to ACF/NCP 4-28  
defining to ACF/TCAM 4-29  
defining to ACF/VTAM 4-28  
disrupted 4-22  
length of 4-16  
multiple 4-10, 4-13, 4-26  
operational 4-16  
structure of 1-7, 1-11  
Explicit Route Operative (ER-OP)  
request 4-29  
explicit-route control (ERC) 3-6  
explicit-route number 1-10, 4-14, 4-28  
EXR  
See exception request (EXR)

## F

FID4 transmission header 4-14  
and virtual-route pacing 4-20, 4-21,  
4-36  
field-formatted request 5-9  
first in chain (FIC) 5-7, 5-10  
first receiver 5-15  
first sender 5-15  
first speaker 5-13, 5-14, 5-15  
flow-control algorithm 4-17  
FM services  
See function management (FM) services

FMD request unit 5-9  
FMD services component 1-7, 5-5, 5-6  
form-of-response requested  
indicators 5-7, 5-14  
frame check sequence, SDLC 3-5  
frame, SDLC 3-5, 3-12  
See also link header, SDLC  
See also link trailer, SDLC  
full-duplex mode 5-16  
function management (FM) header 5-9,  
5-19, 5-20  
types of 5-19  
function management (FM) services  
profile 5-26  
usage field 5-26

## G

GROUP macro (ACF/NCP) 2-21

## H

half-duplex contention mode 5-15  
half-duplex flip-flop mode 5-15  
half-session 1-7, 5-5-5-24  
components of 1-7, 5-5  
host node  
See node, host

## I

IBM 3270 data stream  
See data stream, 3270 data stream  
IBM 3705 Communications Controller 1-4,  
2-1, 3-4  
loading an NCP into (RU  
sequence) 2-29  
IBM 5973-LO2 Network Interface  
Adapter 3-4  
identifier 5-18  
immediate-request mode 5-12  
immediate-response mode 5-12  
IMS/VS 1-4  
checkpoint/restart capability  
of 5-25  
Information Management System/VS  
See IMS/VS  
Initiate Other (INIT-OTHER)  
request 5-2, 5-4  
Initiate Self (INIT-SELF) request 5-2,  
5-4

Inoperative (INOP) request 5-25  
inquiry response (SDLC sequence) 3-27  
interleaved primary link station  
transmissions (SDLC sequence) 3-26  
intermediate routing 3-6  
See also node, intermediate routing  
intermediate routing node  
See node, intermediate routing  
INTRO macro (ACF/TCAM) 4-29  
invalid command (SDLC sequence) 3-23  
IPL a 3705 NCP basic operator  
command 3-13  
isolated pacing response (IPR) 5-8,  
5-22  
ISTATUS operand 2-9, 2-21

## J

Job Entry Subsystem/3 (JES3) 1-4

## L

last in chain (LIC) 5-7, 5-10  
layers, SNA functional  
See SNA functional layers  
LINE macro (ACF/NCP) 2-21, 3-14  
link 3-10  
See also link header, SDLC  
See also link trailer, SDLC  
defining to ACF/NCP 3-14  
defining to ACF/TCAM 3-13  
defining to ACF/VTAM 3-14  
multiple 3-6, 3-12, 4-26  
nonswitched 3-5  
parallel 3-6, 3-12  
switched 2-13, 2-15, 3-5  
types of 3-4  
link connection 3-1, 3-2, 3-13  
link header, SDLC 3-5, 3-12  
link station 1-5, 2-15, 3-1-3-14  
See also adjacent link stations  
address assigned to 2-5  
as component of data link  
control 1-4  
defining to ACF/NCP 3-14  
defining to ACF/TCAM 3-13  
defining to ACF/VTAM 3-14  
primary link station 3-3, 3-13  
secondary link station 3-3, 3-5,  
3-13  
link station control block 2-5  
link station, adjacent

See adjacent link stations  
link stations exchange numbered frames  
(SDLC sequence) 3-19  
link trailer, SDLC 3-5, 3-12  
link, SDLC  
See SDLC link  
LKSTA operand of TERMINAL macro  
(ACF/TCAM) 3-13  
local address 1-5, 3-9  
log-on mode table entry (ACF/VTAM) 4-29  
logical unit (LU) 1-4, 4-10  
See also LU type  
Logical Unit Status (LUSTAT)  
request 5-10, 5-18  
LU macro (ACF/NCP) 2-21  
LU type 5-1, 5-2, 5-27  
and data stream 5-19  
LU-LU session 2-1, 5-1-5-44  
activating 5-2, 5-5  
and class of service 4-14  
notification 4-24, 4-27  
cross-domain 2-16, 5-4  
disruption of 4-22, 4-24  
resynchronizing 5-24  
same-domain 2-16, 5-4

## M

MAXSSCP operand (ACF/NCP BUILD  
macro) 4-28  
message unit 1-5, 3-10, 4-14  
middle in chain (MIC) 5-7, 5-10  
mode name 4-14, 5-4  
mode setting (SDLC sequence) 3-27  
MODEENT macro (ACF/VTAM) 4-29  
modem (modulator/demodulator) 3-2  
monitoring, virtual-route  
See virtual-route monitoring  
multiple SDLC links  
See link, multiple

## N

NAU services manager 1-7  
NCP generation 2-21  
NCP node  
See node, NCP  
negative response to a Poll (SDLC  
sequence) 3-15  
negative response to RU chain 5-11  
negotiable Bind Session request 5-5  
network v

See also public network  
 See also SNA network  
 See also user application network  
 network address 1-5, 2-2-2-7, 3-9, 5-2  
   destination 4-14, 4-16  
 network addressable unit (NAU) 1-1,  
   1-5, 1-7  
   path between NAUs 4-1, 4-6  
 Network Control Explicit Route Activate  
 (NC-ER-ACT) request 4-16  
 Network Control Explicit Route  
 Inoperative (NC-ER-INOP) request 4-35  
 Network Control Explicit Route Operative  
 (NC-ER-OP) request 4-30, 4-31  
 network control program 2-2, 3-2, 4-16  
   See also ACF/NCP  
 network control structure 1-13  
 network name 5-2  
 network operator 2-7  
   altering resource location by 2-13  
   and explicit and virtual routes 4-16  
   control of domain governed by 2-15  
   notified when explicit route  
     fails 4-27  
   notified when transmission group  
     fails 4-22  
   resource activation by 2-9, 2-22,  
     4-16  
   session reinitiation by 5-25  
   use of VARY command by 2-22  
 network services (NS) 5-9  
 network services (NS) header 5-9  
 no-response chain 5-10, 5-11  
 node 1-7, 3-10  
   boundary-function (BF) 3-9, 4-6  
   communication controller 1-5  
   host 1-2, 2-18, 3-10  
   intermediate routing 1-10, 3-1, 4-6,  
     4-19  
   congestion in 4-21, 4-22  
 NCP 1-4, 3-10  
 peripheral 1-4, 2-15  
   activating (RU sequence) 2-26,  
     2-27  
   addresses for resources in 2-7  
   deactivating (RU sequence) 2-34,  
     2-35  
   transmitting data to subarea  
     node 3-9, 3-10  
 SNA 1-1, 3-1  
 subarea 1-2-1-5, 2-2  
   activating (RU sequence) 2-25  
   activating SDLC link between (RU  
     sequence) 2-30

deactivating (RU sequence) 2-36  
 transmitting data to peripheral  
   node 3-9, 3-10  
 transmitting data to subarea  
   node 3-6, 3-9  
 nonnegotiable Bind Session request 5-5  
 nonswitched link  
   See link, nonswitched  
 normal flow  
   See data flow  
 normal-flow request  
   See data flow, normal-flow  
 normal-flow send/receive mode 5-15  
 numbering error in full-duplex exchange  
   (SDLC sequence) 3-24  
 NUMHSAS operand (ACF/NCP BUILD  
   macro) 4-28

## O

one-stage pacing  
   See session level pacing, one-stage  
 only in chain (OIC) 5-7, 5-10  
 OPTION macro (ACF/TCAM) 4-30  
 origin subarea address 1-10

## P

pacing  
   See session level pacing  
   See virtual-route pacing  
 pacing delay  
   See virtual-route pacing, pacing  
     delay  
 pacing indicator (PI) 5-7, 5-8, 5-22  
 pacing response  
   See session level pacing, response  
   See virtual-route pacing, response  
 pacing-group size  
   See session level pacing,  
     pacing-group size  
   See virtual-route pacing,  
     pacing-group size  
 packet-switched data networks 3-4  
 parallel links, SDLC  
   See link, parallel  
 path 1-10, 3-1, 4-1-4-6  
   structure of 1-7, 1-11  
 path control (PC) components 1-4, 1-10  
 path control (PC) functions 3-1  
 path control network 1-1, 1-7, 1-10  
   elements of 1-10

path error 5-12  
 path information unit (PIU) 3-1, 3-2  
     and virtual-route pacing 4-20  
     blocking of 3-5, 3-8  
     sequence numbering of 3-12  
 PATH macro (ACF/NCP) 4-28, 4-29  
 PATH macro (ACF/TCAM) 4-29  
 PATH statement (ACF/VTAM) 4-28  
 peripheral link 1-10, 3-10, 4-6  
     as part of path 4-3  
 peripheral node  
     See node, peripheral  
 physical unit (PU) 1-4  
 physical unit control point (PUCP) 2-1,  
 2-18, 2-21  
 PI  
     See pacing indicator (PI)  
 PIU  
     See path information unit (PIU)  
 positive response to a Poll (SDLC  
 sequence) 3-16  
 positive response to RU chain 5-11  
 presentation medium 5-28  
 presentation services (PS)  
     profile 5-26  
     usage field 5-26  
 prime compression character 5-29  
 priority, transmission  
     See transmission priority  
 profile field 5-26  
 profiles 5-26  
 Programmed Cryptographic Facility  
 program product 5-21  
 protocol  
     bracket 5-13, 5-15, 5-26  
     SDLC 3-13, 3-14  
     session 3-7  
     SNA v, 1-1, 5-24  
     transmission-group 3-7  
     used by logical units 5-1, 5-4-5-30  
 protocol, data handling  
     See data handling  
 protocols for deactivating LU-LU session  
 (RU sequence) 5-43  
 PS  
     See presentation services (PS)  
 PU macro (ACF/NCP) 2-21, 4-28  
 PU-PU flow 5-9  
 public network v  
 PUCP  
     See physical unit control point  
     (PUCP)

## Q

queued responses 5-22  
 Quiesce at End of Chain (QEC)  
     request 5-16  
 Quiesce Complete (QC) request 5-16  
 quiesce protocol 5-16  
 quiescing data flow (RU sequence) 5-42

## R

Ready to Receive (RTR) request 5-13,  
 5-14  
 receive pacing  
     See session level pacing, receive  
     pacing  
 receive sequence number 5-17  
 reconfiguration, dynamic  
     See dynamic reconfiguration  
 Release Quiesce (RELQ) request 5-16,  
 5-17  
 request  
     See request unit (RU)  
 request and response control  
     modes 5-12, 5-13  
 request error 5-12  
 request header (RH) 5-7, 5-11  
 request reject error 5-12  
 request unit (RU) 1-7, 5-9  
     data flow control (DFC) 5-9, 5-10,  
     5-18  
     function management data (FMD) 5-9,  
     5-21  
     network control (NC) 5-9  
     session control (SC) 5-9, 5-10, 5-18  
 request unit (RU) sequences  
     for activating and deactivating  
     resources 2-22  
     for activating and deactivating  
     sessions 5-30  
     for routing 4-30  
     for transferring data 5-30  
 request unit chain  
     See RU chain  
 request unit size 5-6, 5-12  
 request/response control modes 1-7  
 request/response indicator 5-8  
 Reset Window Indicator (RWI) 4-21  
 resource hierarchy, SSCP 2-9, 2-14  
     cascaded activation 2-9, 2-13, 2-19  
     changes in 2-13  
     reactivating resources in 2-14, 2-19  
 resource resolution table 2-21

response  
   See definite response 1 indicator (DR1I)  
   See definite response 2 indicator (DR2I)  
   See exception-response indicator (ERI)  
   See isolated pacing response (IPR)  
   See response type indicator (RTI)  
   See response unit (RU)  
   See session level pacing, response  
   See virtual-route pacing, response  
 response header (RH) 5-8  
 response type indicator (RTI) 5-8  
 response unit (RU) 5-5, 5-11  
 reverse explicit-route number 1-10  
 REX  
   See route extension (REX)  
 RH usage error 5-12  
 route 1-1, 1-10, 4-1  
   See also explicit route (ER)  
   See also routing  
   See also virtual route (VR)  
   specifying 4-27, 4-30  
 route extension (REX) 4-16, 5-25  
 route-availability monitoring option (RAMO) 4-30  
 routing  
   See also explicit route (ER)  
   See also node, intermediate routing  
   See also virtual route (VR)  
   benefits of SNA techniques for 4-26  
   data from subarea to subarea 4-1, 4-39  
   multiple 4-10  
   overview of 4-1-4-26  
   routing information 4-34  
   routing table 3-12, 4-13, 4-14, 4-28  
   segments 4-14  
 Routing Table Generator (RTG)  
   field-developed program 4-28  
 RTI  
   See response type indicator (RTI)  
 RU  
   See request unit (RU)  
   See response unit (RU)  
 RU category 5-7, 5-8, 5-9  
 RU chain 1-7, 5-6, 5-10  
   canceling 5-11  
   chaining control indicators for 5-7  
 RWI  
   See Reset Window Indicator (RWI)

## S

S/370 channel  
   See data channel  
 same-domain LU-LU session  
   See LU-LU session, same-domain  
 SCB  
   See string control byte (SCB)  
 SCS  
   See data stream, SNA character string (SCS)  
 SDI  
   See sense data indicator (SDI)  
 SDLC  
   See Synchronous Data Link Control  
 SDLC frame  
   See frame, SDLC  
 SDLC link 1-4, 3-4  
   See also link  
   activating 2-30  
   components of 3-2  
   configurations of 3-5  
   deactivating 2-32  
   failure of 4-26, 5-25  
   loop configuration 3-5  
   multipoint 3-5  
   point-to-point 3-5  
 SDLC monitor mode function (SMMF) 2-18  
 SDLC protocol  
   See protocol  
 SDLC sequences 3-14  
 secondary link station comes online (SDLC sequence) 3-20, 3-25  
 secondary link station requests  
   connection (SDLC sequence) 3-18  
 secondary link station requests  
   preparation to receive (SDLC sequence) 3-18  
 send pacing  
   See session level pacing, send pacing  
 send sequence number 5-17  
 send/receive mode  
   See normal-flow send/receive mode  
 sense data 5-11  
 sense data indicator (SDI) 5-7, 5-8  
 sequence number  
   See receive sequence number  
   See send sequence number  
 sequence number error 5-12  
 sequence numbers 1-7, 3-9, 3-12  
   and direction of flow 5-9  
   and use of STSN request 5-24  
   assigned to normal-flow requests 5-17



- checked by transmission control 5-5
- serial sharing of SNA resources 2-15
- session
  - See LU-LU session
  - See SSCP-LU session
  - See SSCP-PU session
  - See SSCP-SSCP session
- session initiation 5-2
- session initiation request 5-2, 5-4
- session level pacing 1-5, 3-9, 5-21-5-23
  - one-stage 5-22
  - pacing-group size 5-22, 5-23
  - parameters 5-22
  - receive pacing 5-22
  - response 5-22
  - send pacing 5-22
  - two-stage 5-22
- session limit 5-5
- session network services 5-6
- session outage notification 5-25
- session partner 5-1
- session presentation services 5-6, 5-19
- session reinitiation 4-24, 4-27, 5-25
  - and Unbind user exit 4-30
- session services 5-2
- Session Started (SESSST) request 5-4
- Set and Test Sequence Numbers (STSN) request 5-18, 5-24
- shared control capabilities, benefits of 2-19
- shared control capability 2-15, 2-16
- Shutdown (SHUTD) request 5-17
- Shutdown Complete (SHUTC) request 5-17
- shutdown protocol 5-17
- Signal (SIG) request 5-18
- SMMF
  - See SDLC monitor mode function (SMMF)
- SNA access method
  - See access method, SNA
- SNA functional layers 1-1, 5-4
- SNA network
  - automatic shutdown of 2-14
  - components of 1-1, 1-7
  - congestion in 4-17
    - actions to alleviate 4-22, 4-26
    - effect of 4-21
  - sample 1-1
  - structure of 1-1, 1-7
  - traffic conditions in 4-17
- SNA node
  - See node, SNA
- SNA products 5-1, 5-2, 5-27
  - See also ACF/NCP
- See also ACF/TCAM
- See also ACF/VTAM
- SNA protocol
  - See protocol
- SNA resource 2-9
  - activating 2-1, 2-14, 2-21, 2-22
  - controlling 2-14, 2-19
  - deactivating 2-1, 2-14, 2-21, 2-22
  - sharing control of 2-15, 2-16
    - benefits of 2-19
- SPEED operand of LINE macro (ACF/NCP) 3-14
- SSCP resource hierarchy
  - See resource hierarchy, SSCP
- SSCP-LU session 2-15, 2-16, 5-18
  - and Inoperative requests 5-25
  - and session-initiation requests 5-2
- SSCP-PU session 2-14, 5-18
  - and peripheral node 2-16
  - and route extension failure 5-25
- SSCP-SSCP session 2-16, 5-18
  - reactivation of 2-14
- starting an LU-LU session 5-2
- state error 5-12
- station, link
  - See link station
- status information 5-18
- string control byte (SCB) 5-19, 5-20, 5-29
- subarea 1-5
  - See also node, subarea
  - multiple routes between 4-10
- subarea field (of network address) 2-2
- subarea link 3-13
- subarea node
  - See node, subarea
- switched SDLC link
  - See link, switched
- synchronization point 5-25
- Synchronous Data Link Control 1-4
- system services control point (SSCP) 1-2, 2-14

**T**

- TERMINAL macro (ACF/TCAM) 2-22, 3-13
- TGN operand of PU macro (ACF/NCP) 3-14, 4-28
- TH
  - See transmission header (TH)
- traffic in SNA network
  - See SNA network, traffic conditions in

transmission control (TC)  
 component 1-7, 5-5

transmission group (TG) 1-4, 1-5  
 as part of explicit route 1-10  
 as part of path 1-10  
 defining to ACF/NCP 3-14  
 defining to ACF/TCAM 3-13  
 defining to ACF/VTAM 3-14  
 failure of 4-22  
 multiple SDLC links in 3-6, 3-13,  
 5-25  
 operational 4-16

transmission group (TG) number 3-13

transmission group control (TGC) 3-6,  
 3-7, 3-8

transmission header (TH) 3-9  
 sequence number field in 5-18

transmission priority 4-8, 4-26

transmission services (TS)  
 profile 5-18, 5-26  
 usage field 5-26

transmitting data from node to  
 node 3-1, 3-5  
 path control components for 3-6,  
 3-10

TS  
 See transmission services (TS)

two-stage pacing  
 See session level pacing, two-stage

TYPE operand of PU macro (ACF/NCP) 3-14

## U

Unbind Session (Unbind) request 5-16,  
 5-25

Unbind user exit (ACF/TCAM) 4-30  
 usage field 5-26

user application network v

## V

VARY operator command 2-9, 2-22

version of SNA in this book vi

virtual route (VR) 2-8, 4-8-4-13, 5-25  
 activating 2-31, 4-16  
 and class of service 4-6  
 deactivating 2-24, 2-32  
 defining to ACF/NCP 4-28  
 defining to ACF/TCAM 4-29  
 defining to ACF/VTAM 4-28  
 list of 4-13, 4-14, 5-4  
 loss of 4-13

virtual-route control (VRC) 3-6

virtual-route monitoring 4-13, 4-26,  
 4-30

virtual-route pacing 4-8, 4-19-4-22  
 pacing delay 4-22  
 pacing groups 4-20  
 pacing-group size 4-20, 4-22, 4-29  
 request 4-20  
 response 4-19, 4-21

virtual-route selection exit  
 (ACF/VTAM) 4-29

VRN operand (ACF/TCAM PATH macro) 4-29

VRN operand (ACF/VTAM PATH  
 statement) 4-28

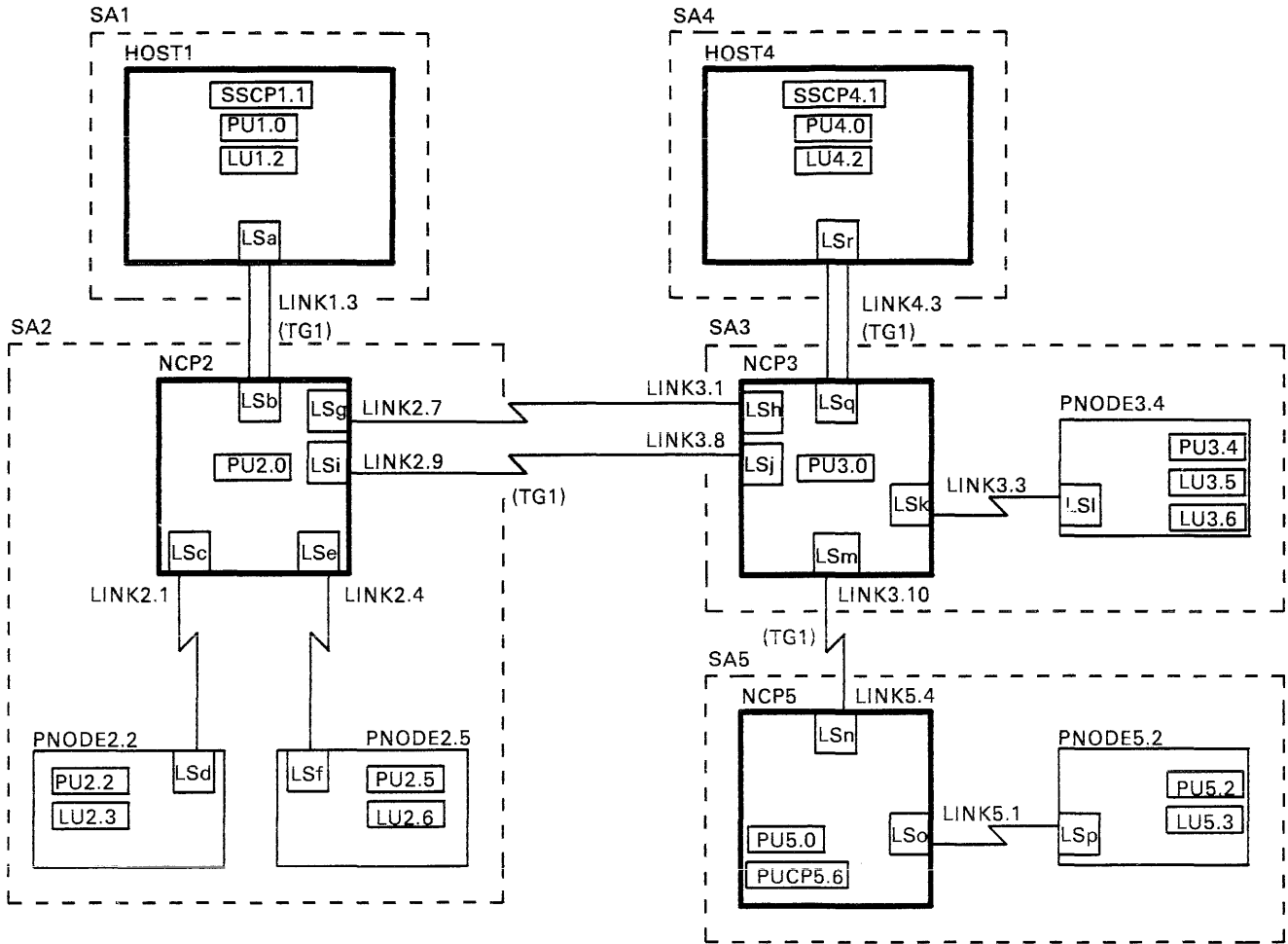
VRPOOL operand (ACF/NCP BUILD  
 macro) 4-28

## X

X.25 circuit 3-4

X.25 interface 3-4

X.25 NCP Packet Switching Interface  
 program product 3-4



Network addresses of representations of link stations in adjacent nodes

Link station	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r
Network address	X	1.4	X	2.2	X	2.5	3.2	2.8	3.9	2.10	X	3.4	5.5	3.11	X	5.2	4.4	X

LSn = link station

Figure 2-1. Sample Network for Description of Component Activation

Systems  
Network  
Architecture

Reader's  
Comment  
Form

Technical Overview

Order No. GC30-3073-0

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.

Your comments will be sent to the author's department for whatever review and action, if any, are deemed appropriate. Comments may be written in your own language; English is not required.

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.)

Cut  
Along  
This  
Line

Reader's Comment Form

Please Do Not Staple

Fold and Tape

Fold and Tape



No Postage  
Necessary  
If Mailed  
In The  
United States

**BUSINESS REPLY MAIL**

FIRST CLASS PERMIT NO. 40 ARMONK, N. Y.

POSTAGE WILL BE PAID BY ADDRESSEE

International Business Machines Corporation  
Dept. E02  
P. O. Box 12195  
Research Triangle Park  
North Carolina 27709



Fold and Tape

Fold and Tape

Please Do Not Staple



Cut  
Along  
This  
Line