Systems Network Architecture

Concepts and Products

GC30-3072-1
File No. 370/4300-30

Second Edition (February 1984)

This is a reprint of GC30-3072-0 incorporating changes released in Technical Newsletter: GN30-3205 (dated 22 September 1981)

Changes are continually made to the information in IBM systems publications. Before using this publication in connection with the operation of IBM systems, refer to the latest *IBM System/370 and 4300 Processors Bibliography,* GC20-0001, for the editions that are applicable and current.

References in this publication to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates.

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

A form for readers' comments has been provided at the back of this publication. If the form has been removed address comments to IBM Corporation, Publications Development, Department E02, P.O. Box 12195, Research Triangle Park, North Carolina, U.S.A. 27709. IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

Preface

This publication introduces the IBM Systems Network Architecture to individuals who need to acquaint themselves with its benefits, its concepts, and the IBM products that are designed for use in SNA networks.

This is the basic publication about Systems Network Architecture (SNA) for managers, system designers, and others involved in making decisions about planning or implementing distributed data processing within an organization.

Note: The descriptions of concepts in this publication apply to the functional capabilities defined by the version of Systems Network Architecture current at the time this edition was published. Not all currently available SNA products have all the functional capabilities of the current version of SNA. 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.

This publication is not a primer on data communication. Although no specific prerequisite reading is suggested, readers of this book are assumed to be familiar with the concepts of data communication through experience with managing, operating, or using data communication systems. Readers lacking this familiarity may wish to avail themselves of a course on data communication concepts. An IBM marketing representative can suggest courses on this subject that are offered by IBM.

Chapter 1 introduces the concepts of a network architecture, explains the benefits of Systems Network Architecture, and briefly introduces the concept of distributed processing.

Chapter 2 explains some basic concepts of SNA. This chapter explains what "end users" are, how end users communicate with one another, how an SNA network is organized, and how data is routed within the network. The chapter then describes the categories of services that SNA provides and concludes by examining the layered structure of SNA.

Chapter 3 relates the layered structure of SNA to the physical and programming components of an SNA network.

Chapter 4 discusses the topics of distributed data processing, distributed applications, job networking, and distributed transaction processing, and summarizes the SNA facilities involved.

Chapter 5 describes some network management capabilities provided by SNA products.

Chapter 6 summarizes many of the information-handling systems, communication controllers and adapters, modems, and data encryption devices that are designed for use in SNA networks.

Chapter 7 summarizes many of the major IBM programs that are related to the control and use of SNA networks. These programs include communication access methods, transaction processing systems, interactive support programs, remote job entry programs, and host-resident support programs.

Both Chapters 6 and 7 refer the reader to introductory publications for each product.

~~Appendix A explains the relationship between SNA, public networks, and~~ international standards.

This book concludes with a Glossary of Terms and Abbreviations related to Systems Network Architecture.

Contents

Figures

This page is intentionally left blank.

Chapter 1. Networks, Distributed Data Processing, and SNA

IBM's Systems Network Architecture is a comprehensive specification for distributed data processing networks. It defines the message formats used within a network, and it defines the rules governing the interaction among components of the network.

This chapter begins by explaining the term *network* and mentioning some requirements for contemporary networks. The chapter then introduces the concept of distributed data processing, defines Systems Network Architecture, and summarizes some of its benefits.

Throughout this publication, references to Systems Network Architecture (abbreviated SNA) apply to the capabilities of the current IBM products designed in accordance with the architecture.

Networks

In a physical sense, a network is a combination of interconnected equipment and programs used for moving information between points where it may be generated, processed, stored, and used. The interconnections may have any of several forms, principally computer channels, telephone lines, microwave links, and satellite links.

In a more abstract sense, the term *network* refers to a *user-application network:* 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 (in the United States and Canada) or telecommunication administrations (in most other countries). This formal definition distinguishes the parts of a network owned and operated by the users of the network (the user-application network) from the parts operated by common carriers or telecommunication administrations, generally referred to as *public networks.*

This publication is concerned with user-application networks, and more specifically, with the SNA network—the part of the user-application network that conforms to the specifications of Systems Network Architecture. (The term *SNA network* is more fully explained in Chapter 2.) The relationship between SNA networks and public networks is described in Appendix A.

Requirements of Contemporary Networks

Early user-application networks were installed to allow data to be entered into or received from a computer at locations more remote from the computer than direct cable connections would allow. The terminals in these remote-access networks fulfilled functions similar to those of locally attached input/output devices in the computer room.

Early applications tended to be concerned with improving the efficiency of clerical tasks incidental to the conduct of the business. Batch applications predominated. Gradually, as technology advanced and the economics of computing changed, more and more applications were added. More aspects of the business came to rely on the use of data processing, and day-to-day operation of the business came to depend increasingly on the availability of the computer. Networks in which terminal operators accessed a single, central computer have expanded to include interconnected computers at many locations.

Further advances in computer technology, accompanied by reductions in the cost of computing power, have made it possible to move some of that power from a central computer to distant controllers and terminals. Known as *distributed data processing,* the concept of dispersing computing power through the network is introduced later in this chapter and described in Chapter 4. In some cases the stores of data—data bases—also are being distributed to parts of the network outside the computer room.

The reduction of processing costs has allowed organizations to economically justify placing more applications on computer-based networks than was possible earlier. This, along with increasing numbers of terminals, has made networks more complex and therefore more difficult to install, manage, maintain, and use. These factors, along with the need for higher reliability, impose stringent requirements on today's networks. Among these requirements are dependability, ease of use, adaptability, and general communication between users.

Dependability

As business, government, and other organizations increasingly rely on fast, accurate information exchange, dependability of network operation assumes greater importance. Not only must the network be available when needed, it must provide its users consistently good response times. Failures, inevitable in a complex, geographically dispersed network, must be quickly identified and corrected. And to the greatest extent possible, errors must be automatically corrected by the network itself, with minimal involvement of its users.

Ease of Use

Most networks are used on the job by people trained to use them. Increasingly, however, networks are serving users who have little or no knowledge of, or interest in, data processing or network operations. Examples of such users include retail and supermarket checkout clerks operating point-of-sale terminals and members of the general public operating banking terminals. Such users must find their terminals easy to use, and they must be able to use them without knowing anything about how the network functions.

Adaptability

Because today's networks can serve so many kinds of applications, network structures must be adaptable to their users' changing needs. Programmers should be able to develop new application programs, or revise existing ones, without being concerned with control of the network. Network control protocols should be independent of the characteristics of the various machines in the network. And it should be possible to change various aspects of a network's operation while it is operating, without disrupting the flow of data through it or causing inconvenience to its users.

Many different transmission facilities are available today, and more are coming. Telephone lines and microwave links have long been the principal means of interconnecting computers and terminals over extended distances. More recently, satellite links are being used for high-traffic paths. New transmission technology is making possible digital links that are often more reliable than the present analog links.

A network designer must be able to select, from the available facilities, those that are appropriate to various parts of the network. A network may, for example, need a combination of telephone lines, microwave links, and satellite links. Neither the operation of the network nor the way its users interact with it should be adversely affected by changes in the facilities used.

As networks become larger and more complex, the question of how to control them assumes increasing importance. Centralizing control of a network's resources at a single site may be appropriate in some cases. In other cases, control of the resources may best be divided among operators at different sites, or among two or more operators at the same site. The network design should be able to accommodate these different levels of control.

General Communication between Users

Networks exist to permit users, or functions specified by users, to communicate. A network of processors and terminals that allows operators to access application programs in one processor but not others limits their ability to communicate. A network that allows an operator at any terminal to communicate with application programs in any processor provides a more general ability for users to communicate. The latter network is more adaptable in fulfilling its users' needs.

In many earlier, pre-SNA IBM networks, programmers who developed application programs had to consider the configuration of the network and the characteristics of the terminals with which the programs communicated. A network design that frees the programmer from such considerations can substantially reduce the effort required to develop application programs. Such a design also eliminates the need to re-code the programs to reflect each change in network configuration (such as types of terminals installed). Programmers can therefore be more productive because they are more able to concentrate on the details of their applications.

Distributed Data Processing

The relationship between data processing and data communication has been gradually changing. No longer is data communication simply a matter of remote access from terminals at distant locations to a central processor. Advances in computing technology have reduced the costs of processing and storage. Consequently, designers and managers of networks can justify more kinds of applications, distribute applications over multiple processors (of varying capabilities), and connect more terminals—typically programmable—to those computers.

As networks have grown more numerous and complex, organizations have had to decide how to assign responsibility for their data processing operations and the associated networks. Some organizations have chosen to centralize their operations in a corporate data processing center, while others have preferred to divide these operations among individual data processing centers at lower levels of the organization—a decentralized approach.

The centralized approach can limit the redundancies—in equipment, programming, space requirements, and people—that often characterize separate data processing operations. A distinct cost advantage can result from sharing these resources. Furthermore, a centralized data processing operation is likely to be easier to manage and control, especially when application requirements change often.

The general advantage of a fully decentralized approach is its responsiveness to the needs of users. Responsibility for application development is at the same site as the user department, or perhaps within that department. Users typically have more control over the system and can more readily tailor it to their needs.

However, the decentralized approach may result in redundancy of resources, as noted above. And although it may be appropriate today to have separate data processing operations (and associated networks), the increasing need for free exchange of information through all levels of an organization may lead management to conclude tomorrow that the separate systems should be linked.

Separate, decentralized systems are likely to have incompatibilities in equipment, programming, and standards that will interfere with the task of linking them. These same incompatibilities are also likely to inhibit the transfer of people, data, and programs between the systems.

Distributed data processing is data processing in which interconnected processors and associated application programs cooperate to perform user applications. Distributed data processing allows an enterprise to combine the benefits of centralized and decentralized data processing systems.

A key benefit of distributed data processing is flexibility. Distributed data processing lets management decide about degree of control and about location of data bases, application programs, and processing power without the inherent limitations imposed by the centralized and decentralized approaches. And distributed data processing allows management the flexibility of altering those decisions as changing business conditions dictate, without disrupting network operations.

IBM offers Systems Network Architecture as the basis for distributed data processing.

Chapter 4 presents a more extensive overview of distributed data processing, emphasizing the distributed data processing performed by IBM subsystems.

Systems Network Architecture

Systems Network Architecture is the description of the logical structure, formats, protocols (rules), and operational sequences for transmitting information through networks and controlling their configuration and operation. Chapter 2 introduces some of these concepts. But SNA needs to be understood in a larger sense. Not only is SNA a specification, it is also a means for structuring a network and a set of products with which to assemble such a network.

SNA is a Specification

Systems Network Architecture is a specification governing the design of IBM products used for distributed data processing. It is called an *architecture* because it specifies the operating relationships of those products as part of a system. In this respect SNA is like a computer architecture.

However, unlike a computer system, whose parts are usually confined to a single room or building, the parts of a network are typically dispersed over a considerable geographical area. Networks may span continents and sometimes link two or more continents. Moreover, the parts are usually joined by telecommunication facilities that are not under the control of the network owner and that are subject to intermittent failures from many causes. Furthermore, the flow of data through a network is not under the control of a single computer, often varies in volume, and is subject to unpredictable delays and errors in transmission.

Still another characteristic of networks is the great variety of processors and terminals they can include and the unlimited combinations of applications they can serve.

Finally, networks continually change; new applications are added and existing ones are revised as the changing needs of an organization dictate.

As a specification for distributed data processing, SNA accommodates these factors in a way that involves users of the network and application programmers as little as possible. SNA is designed to minimize the effort required to install and maintain the network and alter its configuration when necessary. Moreover, SNA is a specification for distributed data processing *systems,* as well as for individual products. SNA defines sets of services that allow two or more programs to cooperate, regardless of their location in the network, in performing useful work.

In some cases, one of the programs may be an application program in one processor and the other may be a program, in the same or a different processor, for accessing a resource such as a data base. SNA allows the application program to access a resource in the same way whether the resource is located at the same processor as the application program or at a distant processor. SNA not only simplifies the way in which application programs use resources, but also permits the network manager to redistribute those resources when necessary without affecting the application programs.

In other cases, both (or all) of the programs may be application programs designed to cooperate in fulfilling a distributed application. For some distributed applications, use of these SNA services permit network owners to realize cost savings and performance improvements as compared with the same application in a nondistributed form.

SNA is a Plan for Structuring a Network

SNA clearly defines both the functional responsibilities of each network component and the rules for communication between components. In this way, SNA provides a coherent network structure that can accommodate varied network configurations and user applications.

SNA also defines the basic principles by which a network owner specifies a network, manages the resources of that network, and controls the transmission of data among the users it serves. By defining these principles, SNA allows network owners to effectively manage their network, arrange the network configuration, and distribute the network management and control functions to meet their needs.

SNA is a Set of Products

Since introducing SNA several years ago, IBM has developed and offered numerous distributed data processing products for use in SNA networks. These products are combinations of hardware and programming designed in accordance with Systems Network Architecture. In addition to a large number of terminals for both specific industries and general applications, the SNA product line includes processors, communication controllers and adapters, modems, and data encryption units. Chapter 6 briefly describes many of these products.

The SNA product line also includes a variety of programs and programming subsystems. Some of these are generally applicable to SNA networks; telecommunication access methods (ACF/TCAM, ACF/VTAM, ACF/VTAME) and the network control program (ACF/NCP/VS) are examples. Others are related to a broad range of distributed applications, such as the DPPX programming system for the 8100 Information System and CICS/VS (Customer Information Control System/Virtual Storage). The product line also includes programs for network management, such as Network Communication Control Facility (NCCF). These and many other currently available SNA programs are briefly described in Chapter 7.

Systems Network Architecture is not static and unchanging. Nor is it simply the basis for the current IBM line of distributed data processing products. IBM is committed to SNA as the basis for further development of both hardware and software products. Already a firm design base for networks, SNA will continue to be developed and improved to accommodate new functions that will make networks more effective in meeting the needs of their users.

The Difference between Architecture and Implementation

It is important to understand the distinction between an architecture and specific implementations of that architecture. The developers of SNA have identified a set of principles that apply to distributed data processing products and networks in general and have embodied those principles in the design of SNA. SNA does not specify the complete design of each product in a network and it does not prescribe the network functions that each product must be capable of performing. These aspects of a product are the responsibility of its designers.

SNA *does* prescribe the manner in which a network function is to be performed if the designers of a product choose to include the associated SNA component in their product. This allows equivalent functions in different products to interact in a universally understood manner and eliminates unnecessary re-invention of the same function in different products, while allowing product designers to innovate for specialized applications.

The Benefits of Systems Network Architecture

Systems Network Architecture can help organizations improve their data processing and communication operations. Some of the benefits of SNA are described below.

Ability to Interconnect Diverse Products into a Unified System

Although networks have for many years been assembled from diverse products, the cost in money and effort has often been excessive because of various incompatibilities. Different types of terminals have used dissimilar data link control protocols and often required separate communication facilities. Various kinds of applications have required dissimilar data-base organizations and access methods. Thus, a network assembled from incompatible products has often needed extra resources such as access methods and communication facilities that would be unnecessary except for the incompatibilities.

In contrast, SNA has defined and made uniform the functions, products, and protocols needed in a distributed data processing network. A network can therefore be assembled from various combinations of SNA hardware and software products without the need for redundant data link control protocols, access methods, and other resources.

Independence of Users from Network Configuration

The structure of SNA (described in Chapter 2) allows users of a network to be independent of many of the characteristics of the network and the details of its operation. For example, a programmer can write an application program knowing only the specific device characteristics directly relating to the application. Furthermore, the network users need not be concerned with the inner workings of the network (for example, the details of how data links are controlled). This means that the users are not affected by alterations to the physical configuration of the network, such as changes in device types, relocation of devices, and changes in physical addresses of devices.

Maximum Flexibility in Configuring a Network

With SNA, it is not necessary to install separate links for different kinds of terminals. For example, SNA keyboard/display terminals and SNA keyboard/printer terminals can be attached to the same link, provided that they operate at the same transmission speeds. In addition, it is not necessary to have separate links for dissimilar applications. For example, terminals used for inquiry/response applications can share a link with terminals used for remote-job-entry applications. Message traffic associated with these dissimilar kinds of applications can be interleaved on the link.

Preservation of Investment when the Network Changes or Expands

Any network can change and expand over the years. New applications are added; existing ones are modified or dropped; older terminals and processors give way to newer ones. Such changes, while necessary to keep up with the needs of the organization, are often excessively costly when a network uses incompatible products. Replacing one kind of terminal with another may require that a different type of telecommunication facility be installed. And application programs may have to be drastically—and perhaps expensively—modified.

In contrast, the use of uniform protocols by SNA products means that one type of SNA product can often be substituted for another. The telecommunication facility and the application programs can often remain intact, the original investment preserved. (Some application programs might, however, be altered to take advantage of new features.) And because many SNA products can be programmed for use in several different applications, new applications can often be added without further investment in hardware.

Attaching Non-SNA Products to SNA Networks

SNA networks accommodate certain non-SNA terminals to make easier the transition from an existing network to a network that provides the benefits of SNA. A network owner can preserve an investment in non-SNA terminals while gradually advancing to SNA. Programs are available that serve as an interface between SNA and non-SNA parts of the network. These programs convert the control sequences associated with certain non-SNA terminals to SNA sequences, and vice versa, so that these terminals appear to the SNA network as SNA resources. (Alternatively, various non-SNA terminals can share the transport services of the network with SNA terminals through the use of special transmission headers by SNA access methods and network control programs. These and other SNA programs are described in Chapter 7.) Consequently, the network owner can stop the proliferation of incompatible terminals and minimize the turnover of the inventory of installed terminals.

Chapter 2. The Concepts of SNA

This chapter describes the concepts of Systems Network Architecture and introduces some basic terminology. Beginning with a view of an SNA network as seen by its users, this chapter explains generally how users communicate with other users through the network, then tells how an SNA network is organized. Finally, the concepts of layers, headers, and protocols are explained.

End Users

The appropriate place to begin understanding the concepts of Systems Network Architecture is from the viewpoint of the end user. All networks exist ultimately to serve people. The term *end user* is applied to people who directly interact with the network, as by using a terminal, in order to obtain a service that the network provide—principally, the efficient exchange of data between points in the network.

Often, however, an individual interacts with the network through an application program located within the terminal that individual is using. Such application programs are regarded as being end users of the network rather than part of it, because they help the human user to obtain a service from the network and because they can make decisions that would otherwise have to be made by the human user.

Other application programs are located in computers and typically provide services for people (or other application programs) using the network. Again, these programs are regarded as end users of the network, rather than part of it, whenever they draw upon the services of the network. (See Figure 2-1.)

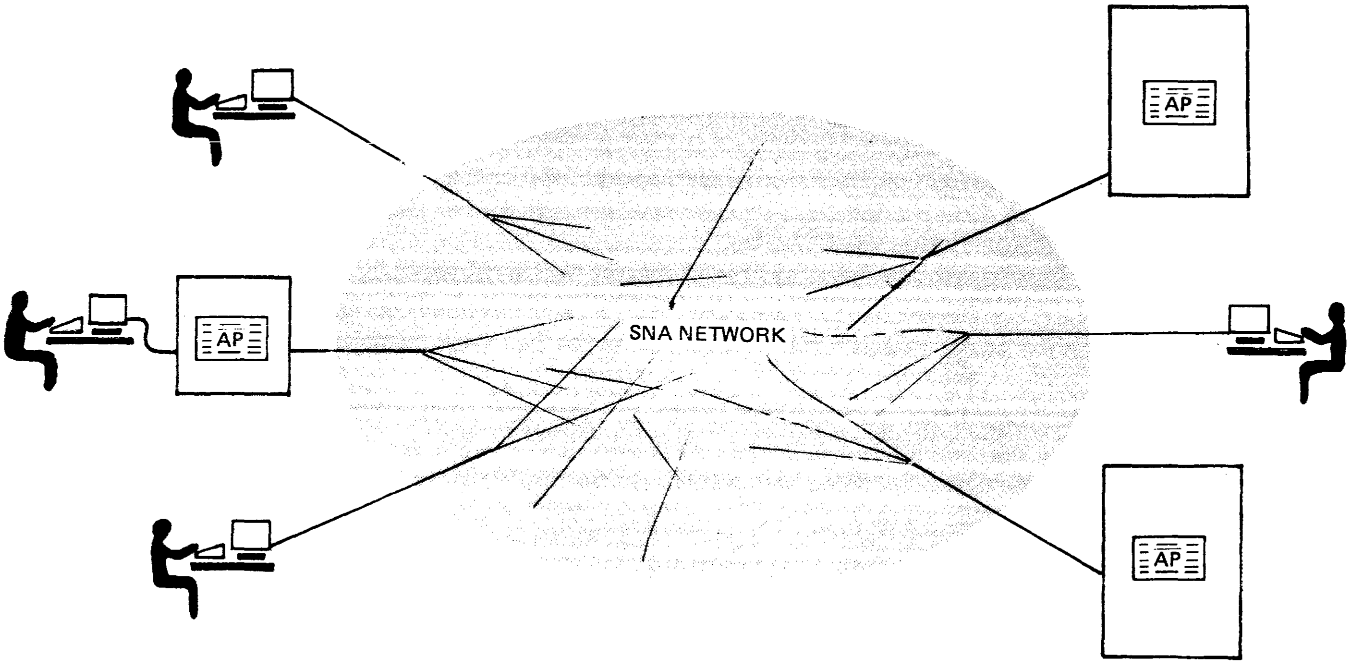


Figure 2-1. Terminal Operators, Application Programs, and the SNA Network

End users of a network are therefore either individuals pr application programs interacting with the network to obtain a service that the network provides. Furthermore, end users are the sources and destinations of application data flowing through the network. The term *application data* refers to data related to applications that the network serves, in contrast to data that is used to control the operation of the network.

Figure 2-2 shows the general relationship between end users, the user-application network, and the SNA network.

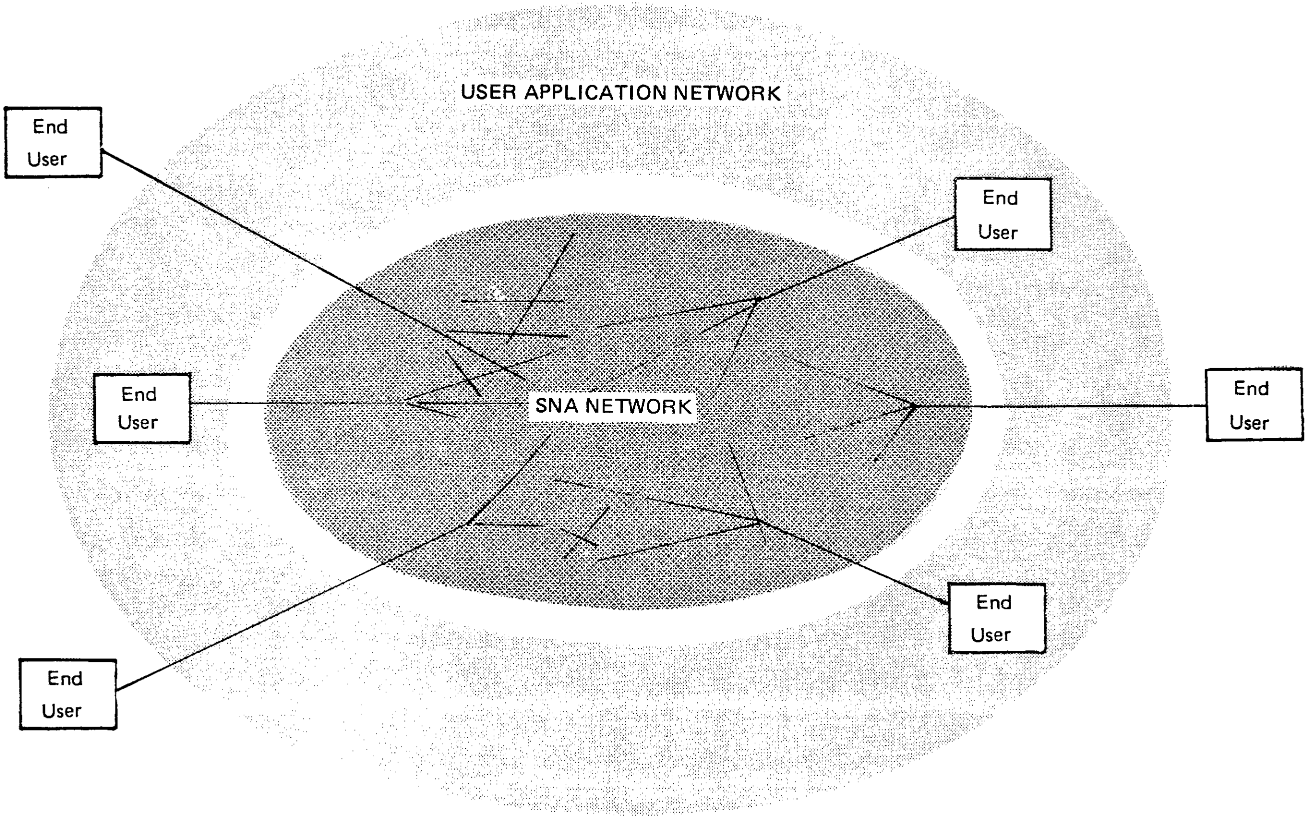


Figure 2-2. End Users, the User Application Network, and the SNA Network

How End Users are Represented to the SNA Network

Because end users are not part of the SNA network, they are not identified to the network. Consequently, there must be something that acts as a bridge between the end user and the network. That bridge is called a *logical unit,* abbreviated LU. Logical units, which are implemented as program code or microcode, provide end users with points of access to the network. Through its LU, an end user gains access to network resources, sends data into the network, and receives data from the network.

Every end user, regardless of location in the user-application network, has a logical unit to permit communication with other end users and to make use of the services of the SNA network. Figure 2-3 shows logical units in relation to end users. (As described later in this chapter, logical units can vary greatly in the functions they perform and in their implementation, depending on the needs of their respective end users.)

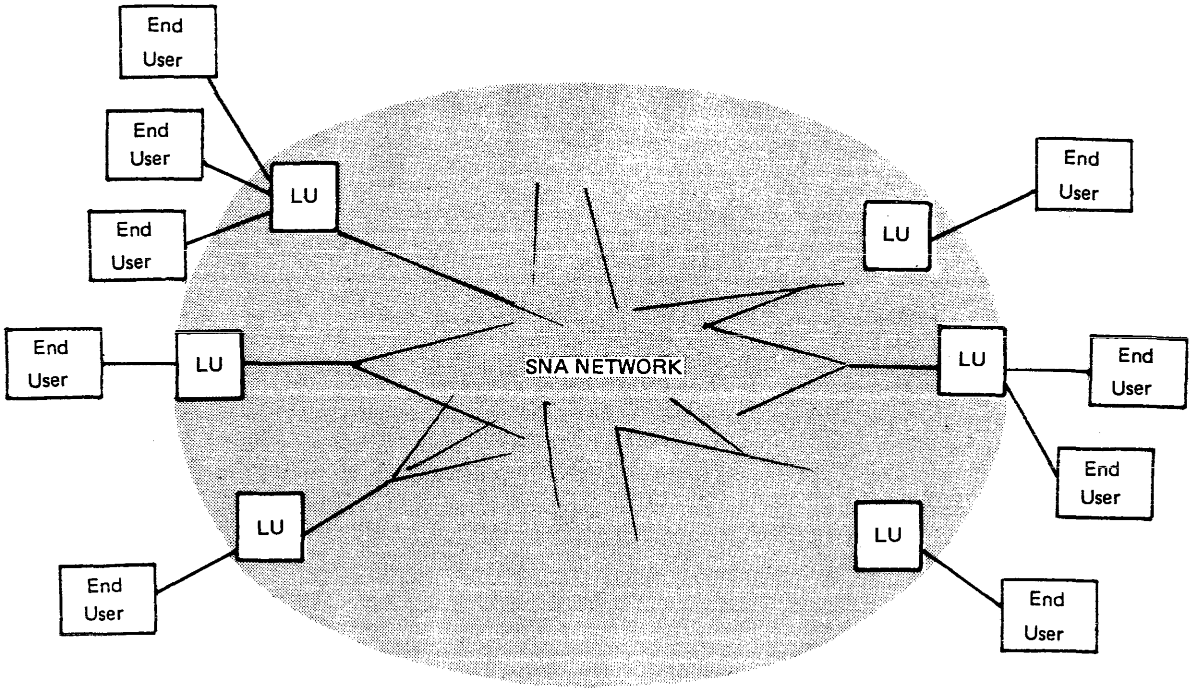


Figure 2-3. Logical Units Representing End Users

The SNA network has been represented thus far in generalized form. Terminals, controllers, and processors are not distinguished, and the actual configuration of links is not shown, because the higher-level concepts of SNA are independent of the configuration. Some other concepts (described later) relate to specific configurations; these are shown in appropriate detail when needed.

How End Users Communicate: Sessions between Logical Units

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*. (See Figure 2-4.) (The term *session partners* is often applied to logical units engaged in a session.)

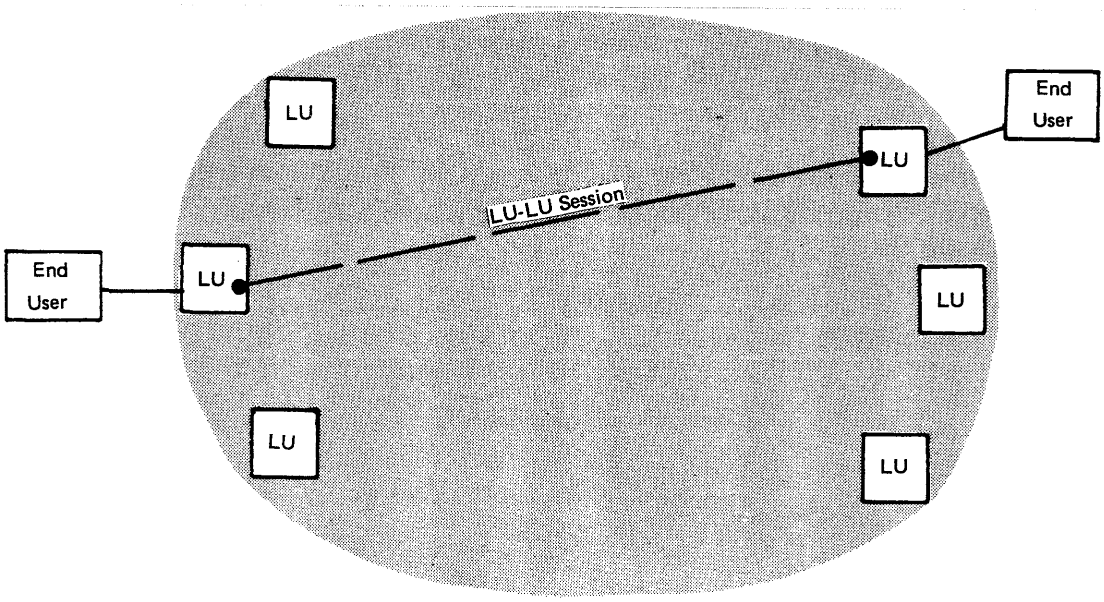


Figure 2-4. A Session between Logical Units

An LU-LU session synchronizes the state of the interaction between the end users. The session is a temporary relationship, or connection, that allows the logical units to exchange data. Activating a session between logical units makes available the appropriate resources such as buffer storage and processor capacity, for the duration of the session. For example, an accounting clerk at a terminal and an accounts receivable application program in a processor might interact via a session. The clerk would initiate the session (through the logical unit in the terminal), use the accounts receivable program to process several transactions, and then end the session.

The exchange of data by end users is subject to a number of procedural rules that the logical units specify before beginning the exchange. These rules represent an agreement between the end users about how the session is to be conducted, where alternatives exist. The rules specify such things as the format of the data, the amount of data to be sent by one end user before the other replies, and the action to be taken if errors occur.

Each logical unit in a network is assigned a *network name* by which it can be specified by another logical unit that intends to initiate session with it. Before the session begins, the SNA network determines the *network address* that corresponds to the network name.

This scheme lets one end user (for example, a terminal operator) establish communication with another end user (for example, an application program) without having to specify where in the network that end user is located. Because this is the case, an application program can be freely moved from one processor to another, without effect on the other application programs or human users that communicate with that application program. (Network names and network addresses are discussed under "Addressing" later in this chapter.)

User data flows between the two logical units in a session as bit sequences that are referred to generically as *message units.* Besides user data, a message unit contains the network addresses of the logical unit that originates the message unit and the logical unit that is to receive the message unit. These are called the origin logical unit and the destination logical unit, respectively.

Activating a Session

A session between a pair of logical units is initiated when one of them (or some other logical unit representing a network operator) issues an activation request. The request to activate a session includes the identities of the logical units with which the session is to be activated.

The process of activating a session consists of several steps; the first is sending a session activation request. The session is successfully activated if a path is available between the logical units, both logical units are capable of meeting the needs of the end users being connected, and the logical units are authorized to communicate with each other. Several factors may determine whether the logical units have this ability. For example, a logical unit representing an application program may enter into session only if that application program is operating. And the session may be activated only if each logical unit is authorized to be in session with the other.

The SNA network can provide different levels of service to sessions. Some sessions, for example, may require faster transmission than others. Or one session may require transmission over a high-security path while another does not. The parameters in the request to activate a session can therefore specify a *class of service* that the network is to provide to that session. (Class of service is described later in this chapter.)

Controlling Data Flow within a Session

Once a session has been activated between a pair of logical units, they can begin to exchange data. Since the exchange proceeds according to agreed-upon rules, data flow can be orderly and the logical units can remain synchronized.

An essential aspect of controlling the flow of data is the sequencing of data exchanges. If the logical units exchange data alternately, at what points in the exchange does the sending logical unit become the receiving logical unit, and vice versa? Does the sending LU expect to receive a response after every message unit it sends, or only after message units that the other LU receives incorrectly? The answers to these and other questions are agreed upon by the session partners when the session is activated.

Another aspect of controlling the flow of data within a session is regulating the rate at which that data flows. A technique called *session-level pacing* regulates the flow of data so that the receiving logical unit in a session does not receive data faster than it can hold or process the data. The sending logical unit transmits a specified amount of data, then awaits an indication from the receiving logical unit signifying its ability to receive more data. (The indication may arrive before the sending logical unit has finished transmitting its data.) Upon receiving that response, the sending logical unit transmits the specified number of message units and once again awaits a response.

Deactivating a Session

A session between a pair of logical units is deactivated when one of them sends a deactivation request, or when some other event outside the session does so. This event might be intervention by a network operator or the failure of some part of the network in the path between the session partners. A logical unit may or may not deactivate a session without first informing its session partner; in either case all data exchanges in the session should, if possible, be successfully completed first. SNA products provide methods for checking for successful completion.

Relationship of End Users and Logical Units

As stated earlier, every end user of a network is represented to the network by a logical unit. The relationship is not necessarily one-to-one; a single logical unit may represent multiple end users (see Figure 2-5).

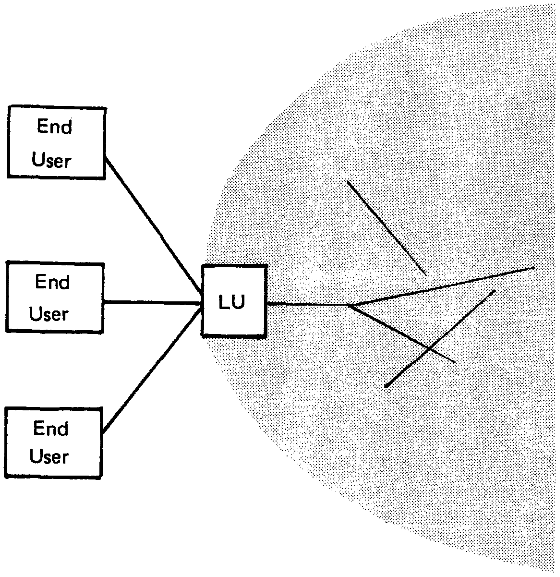
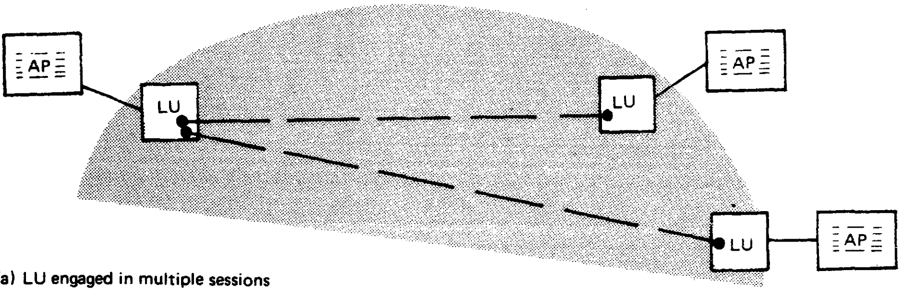
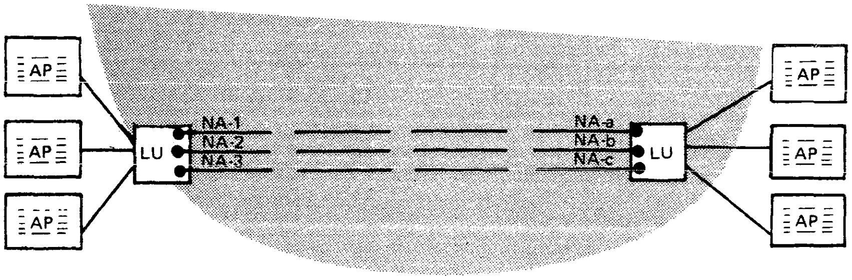


Figure 2-5. Multiple End Users for One Logical Unit

An application may require that one logical unit engage in concurrent sessions with two or more other logical units, as Figure 2-6 (a) shows. Concurrent sessions allow related application programs in one or more processors to interact in a coordinated, interdependent manner. Program synchronization protocols may be used to coordinate the sessions. These protocols are described later in this chapter under "End-User Services."



(a) LU engaged in multiple sessions



(b) LUs engaged in paralled sessions

Figure 2-6. Multiple LU-LU Sessions and Parallel LU-LU Sessions

An application program can also use concurrent sessions to communicate with several terminals at once.

Often it is appropriate for two logical units to engage in a single session. This is the case, for example, when one of them represents a terminal operator and the other represents an application program. On the other hand, the logical units may each represent multiple application programs. In this case it is often desirable to allow many pairs of these programs to be in session simultaneously. *Parallel sessions* make this possible. The pair of logical units engages in as many concurrent sessions as needed for the application programs to communicate with each other.

Figure 2-6 (b) shows parallel sessions. Each session is identified by a unique pair of network addresses; these addresses are assigned automatically as each session is activated. Parallel sessions can be activated and deactivated independently of one another.

Sometimes one of the logical units engaged in an LU-LU session does not represent an end user; instead, that logical unit provides a service for its session partner. The SNA component performing the service is contained entirely within the logical unit. Figure 2-7 shows an LU-LU session serving a single end user.

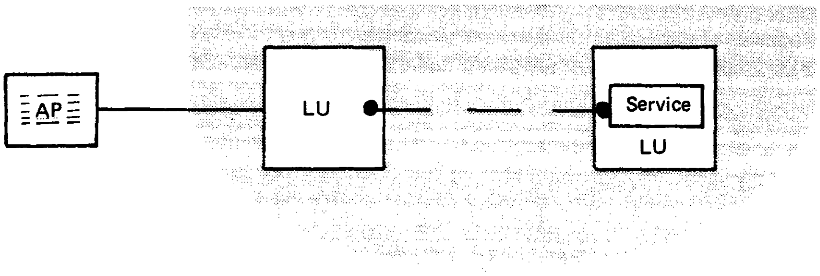


Figure 2-7. LU-LU Session Involving Only One End User

How an SNA Network is Organized and Managed

The logical organization of an SNA network, irrespective of its physical configuration, is divided into two broad categories of components. The first category is called *network addressable units;* the second is called the *path control network.* (See Figure 2-8.)

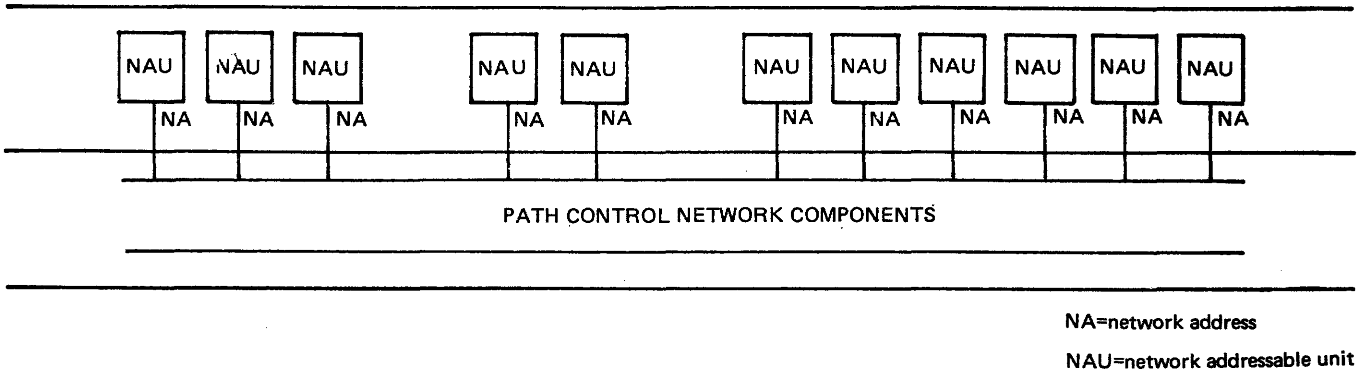


Figure 2-8. Categories of Components in SNA Network

Network Addressable Units

Network addressable units are sets of SNA components that provide services that enable end users to send data through the network and that help network operators to perform network control and management functions. Physically, network addressable units are hardware and programming components within terminals, controllers, and processors. Network addressable units communicate with one another through the path control network, discussed later in this chapter.

Each network addressable unit has an address (sometimes more than one) that identifies it to other network addressable units and to the path control network. It is this address—called a *network address*— that allows the path control network to route data from one network addressable unit to another. The use of network addresses is described later in this chapter under "How Data is Routed through the Path Control Network."

Kinds of Network Addressable Units

SNA networks contain three kinds of network addressable units. The first was introduced earlier in this chapter as the *logical unit* (LU), which represents end users to the network.

The second kind of network addressable unit is the *physical unit* (PU)—not, as its name may suggest, a physical device, but rather a set of SNA components that provide services used to control links, terminals, controllers, and processors in the network. Each terminal, controller, and processor contains a physical unit. The physical unit represents the terminal, controller, or processor to the SNA network; its services manage the physical resources (for example, links) associated with that terminal, controller, or processor.

The third kind of network addressable unit is the *system services control point* (SSCP). This, too, is a set of SNA components. The job of an SSCP is broader than that of physical units and logical units: whereas these units represent machine resources and end users, the SSCP manages the entire SNA network or a significant part of it called a *domain.* (Domains are discussed later in this chapter under "Network Resources, Domains, and Control Points.") The SSCP does not do so by itself. It interacts with other SSCPs and with one or more individuals whose job is to supervise the operation of the network, just as computer operators supervise the operation of computers. This publication refers to these individuals as *network operators,* whether they are responsible for supervising an entire network or only a part of it.

Network operators issue commands to, and receive responses from, system services control points. Many routine, repetitive, supervisory functions, such as activating the network each day, use predetermined sequences of commands and responses that can readily be coded in a program. The term *network operator* therefore refers also to any program that supervises network operation.

The SSCP has three major functions in an SNA network. First, it manages the resources of the network in accordance with commands issued by network operators. Second, it coordinates the activation of sessions between network addressable units. And third, it acts on the physical network when that is required to activate sessions. For instance, a request to activate a session can cause a terminal to be dialed over a switched link connection.

Communication between Network Addressable Units

Just as sessions exist between logical units (LU-LU sessions), they also exist between the other kinds of network addressable units. An SSCP has sessions with logical units (SSCP-LU sessions) to enable end users to access, control, and monitor the processing and communication resources of the network. And an SSCP has sessions with physical units (SSCP-PU sessions) to enable network operators to perform similar functions.

For example, each SSCP in the network uses SSCP-PU sessions to activate links, and logical units use SSCP-LU sessions to request activation of LU-LU sessions. When two or more SSCPs divide control of the network's resources, those SSCPs coordinate their activities via SSCP-SSCP sessions.

Physical units also communicate with other physical units in relationships called *PU-PU flows.* A pair of physical units uses a PU-PU flow in order to transfer a program from a processor through the SNA network to a cluster controller or a terminal.

PU-PU flows are also used in activating, deactivating, and testing routes between nodes. (Routes are discussed later in this chapter under "Paths and Routes.")

SNA Nodes

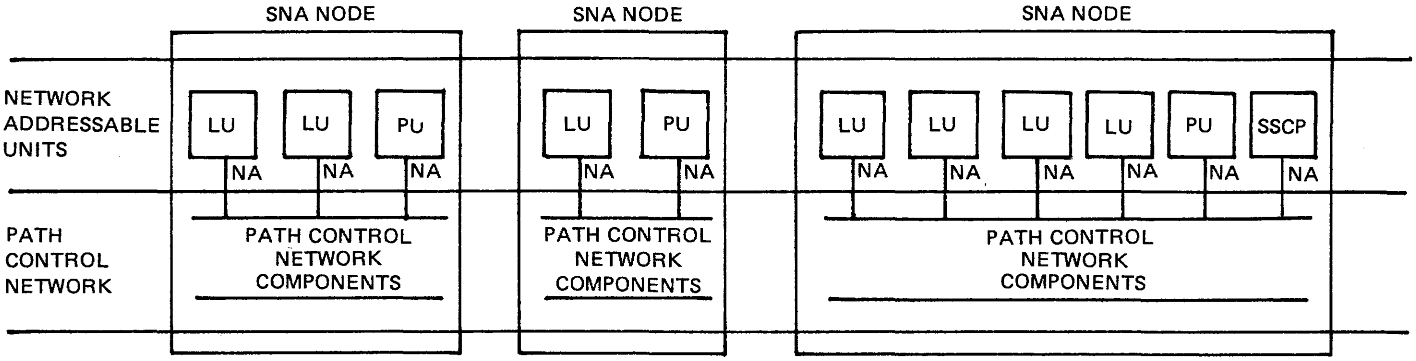
Systems Network Architecture defines a *node* as a point within an SNA network that contains SNA components. Each terminal, controller, and processor that is designed in accordance with SNA specifications can be a node in an SNA network. Each SNA node contains a physical unit that represents that node and its resources to the SSCP.

When the SSCP activates a session with a physical unit (an SSCP-PU session), it makes the terminal, controller, or processor that contains that physical unit an active part of the SNA network. Conversely, the terminal, controller, or processor ceases to be an active part of the network when its physical unit is deactivated. This occurs when the SSCP ends its session with the physical unit.

It is convenient to think of an SNA node as being a terminal, controller, or processor in the network. Strictly speaking, however, the SNA node is contained *within* the machine, and the term *SNA node* refers only to that part of the machine and its programming that conforms to SNA specifications. (Sometimes a machine contains more than one SNA node.)

The relationship of SNA nodes to IBM products is explained further in Chapter 3. The unqualified term *node* in this publication refers to an SNA node.

SNA nodes contain network addressable units, as shown in Figure 2-9; every network addressable unit in the SNA network resides in an SNA node. Besides containing a physical unit, every SNA node can include one or more logical units. Certain nodes can also have an SSCP.



LU=logical unit
NA=network address
PU=physical unit
SSCP=system services control point

Figure 2-9. Network Addressable Units within SNA Nodes

Associated with each node that does not contain an SSCP (that is, each communication controller) is a *physical unit control point* (PUCP). The PUCP contains a subset of SSCP functions needed to activate and deactivate resources associated with the communication controller.

The PUCP interacts only with the physical unit in the same node. When the physical unit in the controller is not in session with any SSCP in the network, the PUCP can interact with the physical unit to activate the network control program and can issue commands to activate links to other nodes.

The Types of SNA Nodes

SNA defines several types of nodes. These types are distinguished by their capabilities within an SNA network and by their logical interrelationships.

Nodes are either *subarea nodes* or *peripheral nodes*. The two kinds differ in the way they interconnect with other nodes and in their ability to route message units through the network. Figure 2-10 shows how subarea and peripheral nodes interconnect.

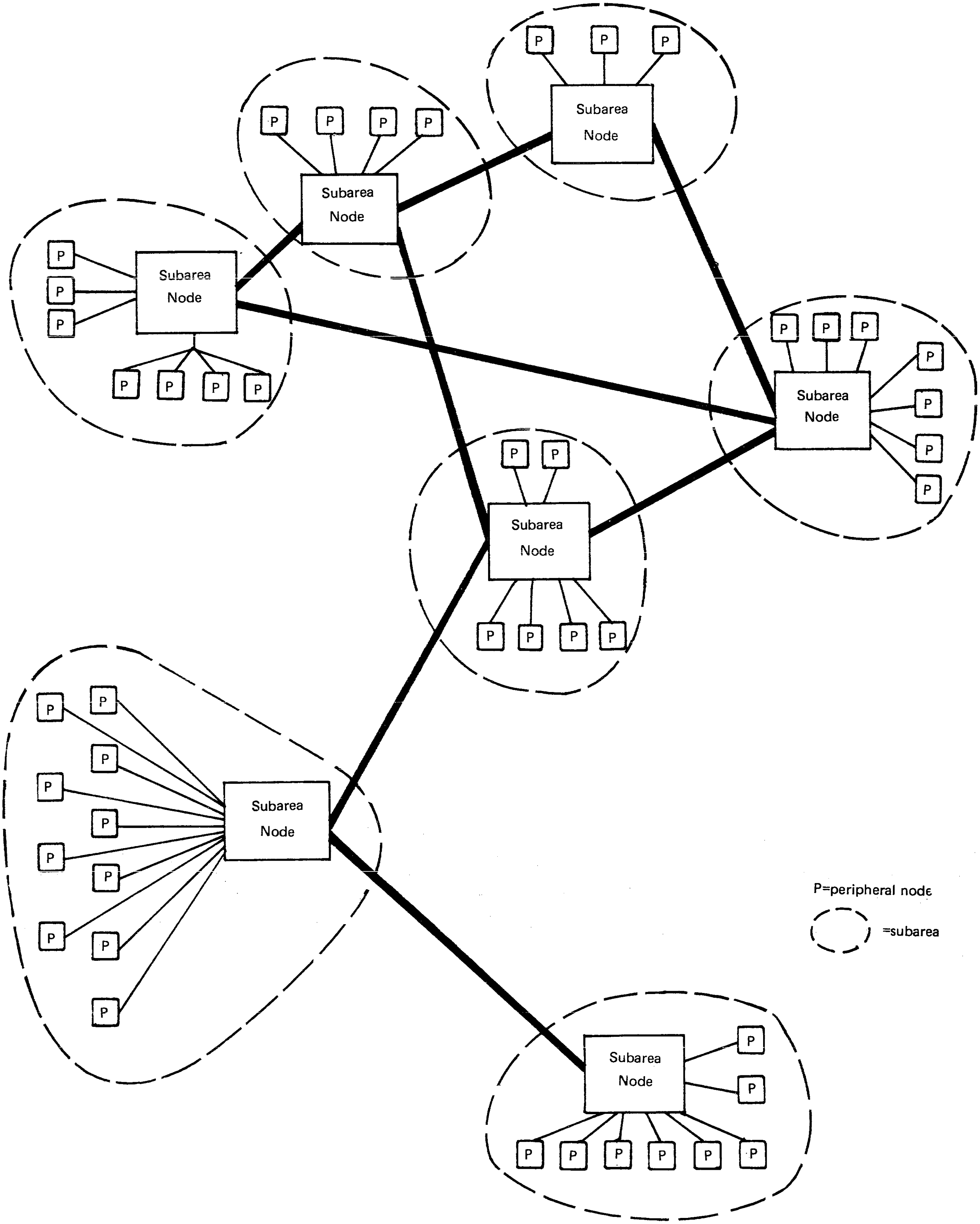


Figure 2-10. Interconnections of Subarea Nodes and Peripheral Nodes

A subarea is a part of the SNA network consisting of a subarea node and all the peripheral nodes attached to it, as Figure 2-10 shows.

~~The origins and destinations of message units are network addressable units.~~ These may be located in a subarea node or a peripheral node. The origin and destination may both be in the same subarea or in different subareas.

A subarea node can receive message units from any origin and move them toward any destination in the network, provided that a transmission group (a group of links) needed to connect the origin and destination are available. To do so, the subarea node examines the network address in each message unit it receives or originates. Using this address and a route number carried in the message unit, the subarea node determines the transmission group to the next node on the path toward the destination and then sends the message unit over that transmission group. (If the destination network addressable unit is located within that subarea node, the message unit is passed directly to that network addressable unit.)

In contrast, a peripheral node can pass message units only between a network addressable unit within that node and the subarea node to which it is attached. The peripheral node is not aware of any part of the network beyond its subarea node and cannot handle the complete network address contained in message units passing between subarea nodes. Instead, the peripheral node uses a simpler forms of address called a *local address.*

The address is called *local* because it is used only by the peripheral node and its attached subarea node. (In its role as manager of a network domain, the SSCP, too, is aware of the local addresses in its domain.)

The peripheral node relies on its attached subarea node to make the required transformations between the complete network address used in data flow between subarea nodes and the local address used for data flow between the subarea node and the peripheral node.

The reasons for the different forms of address are explained later in this chapter under "Addressing." The distinction between subarea nodes and peripheral nodes in terms of their ability to interpret addresses is important in understanding routing and topics discussed later in this chapter under "Configuring and Reconfiguring the Network."

Each subarea node uses a flow-control technique called *virtual-route pacing* on its routes to other subarea nodes (but not on its links to peripheral nodes) to prevent those routes from becoming so congested as to seriously impair traffic flow through the path control network. Virtual-route pacing is described later in this chapter. (A different form of pacing, called *session-level pacing,* is used to prevent overrun of buffers in subarea nodes and peripheral nodes. Session-level pacing is described under "Transmission Control Services" later in this chapter.)

A subarea node may or may not contain an SSCP as one of its network addressable units. If it does, it is referred to as a *subarea node with an SSCP.* If it does not, it is referred to as a *subarea node without an SSCP.* Either type of subarea node can contain one or more logical units.

Logical units that reside in a subarea node are called *subarea LUs,* and logical units that reside in a peripheral node are called *peripheral LUs.* Physical units are similarly distinguished.

How SNA Nodes are Related to the Physical Components of the Network

Up to this point, the organization of an SNA network has been presented in conceptual terms, with little reference to actual distributed data processing products. This is appropriate in order to stress the overall concept of SNA as sets of functions, distributed through the network, that interact in specified ways independent of specific kinds of components. A network does of course consist of actual terminals, controllers, and processors and the links that join them. It is therefore helpful in visualizing an SNA network to have some idea of how SNA nodes are related to these machines.

Figure 2-11 shows a very simple example of an SNA network. A processor communicates, through a communications controller, with a terminal, a cluster controller with attached devices, and another processor. Notice that the processor at the upper left contains an SSCP and is therefore a subarea node with SSCP. The network is controlled from this processor. The processor at the lower left, the terminal, and the cluster controller are peripheral nodes; the communications controller at the top right is a subarea node without SSCP.

Although very simple and not typical, this configuration qualifies as an SNA network: it consists of SNA nodes that contain network addressable units; the network addressable units are linked by a path control network; there is an SSCP to control the resources of the network; and this SNA network serves end users—both people (terminal operators) and application programs.

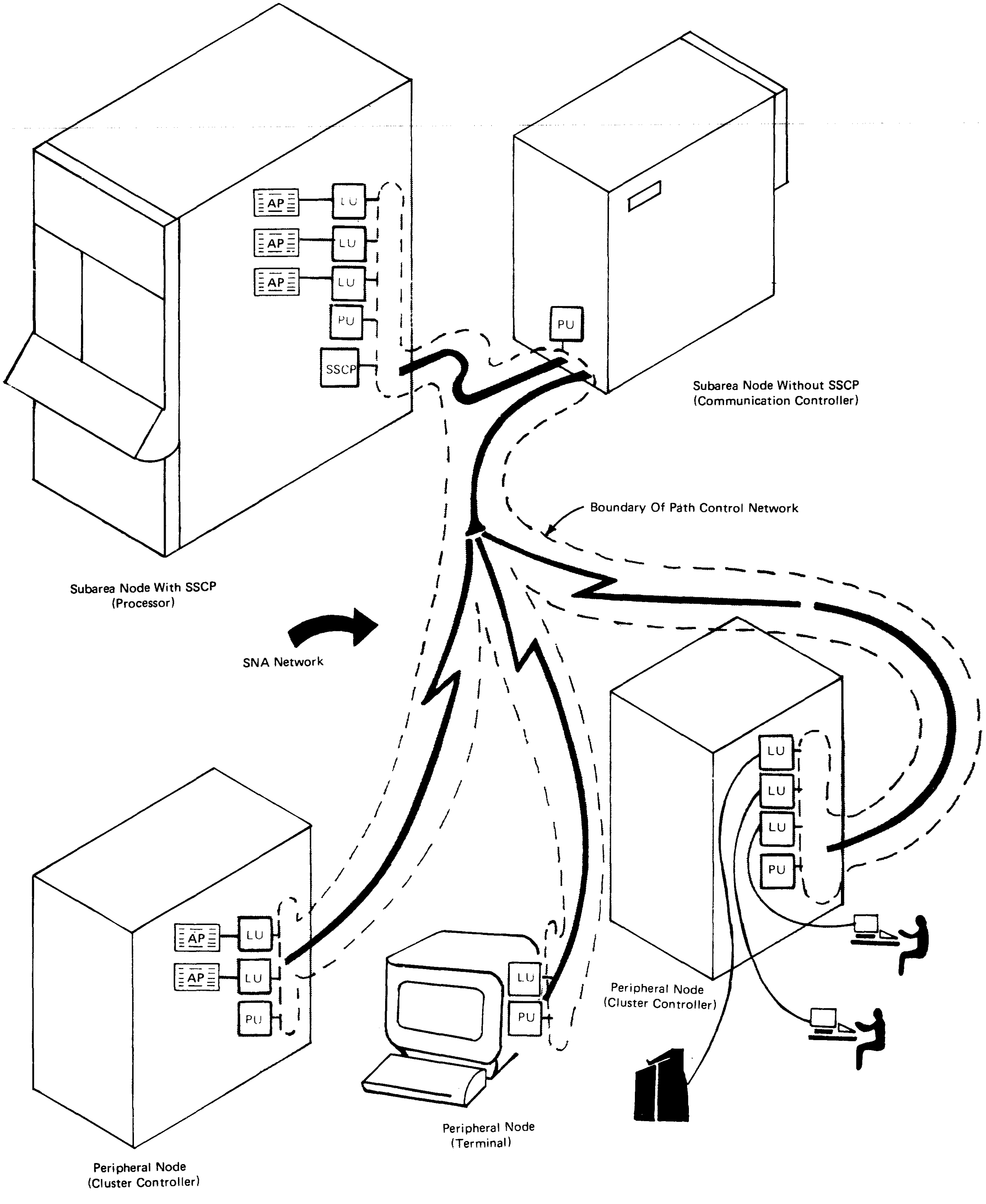


Figure 2-11. Example of an SNA Network

Connections between SNA Nodes

The network components containing SNA nodes are physically connected in various ways. Figure 2-12 shows several kinds of connections between subarea and peripheral nodes. A subarea node may be connected to another subarea node or to a peripheral node located within the same room or building by a System/370 channel (Figure 2-12 [a]). Channels operate at very high data rates; transmission is controlled by channel protocols incorporated into SNA.

A subarea node may be connected to another subarea node or a peripheral node over great distances via public networks furnished by a communication common carrier or telecommunications administration (Figure 2-11 [b], [c], and [d]). Transmission over such links is controlled by Synchronous Data Link Control (SDLC) protocols that are part of Systems Network Architecture; the data rates are usually lower than those of channels. (SDLC is a version of the international standard, HDLC [High-level Data Link Control].)

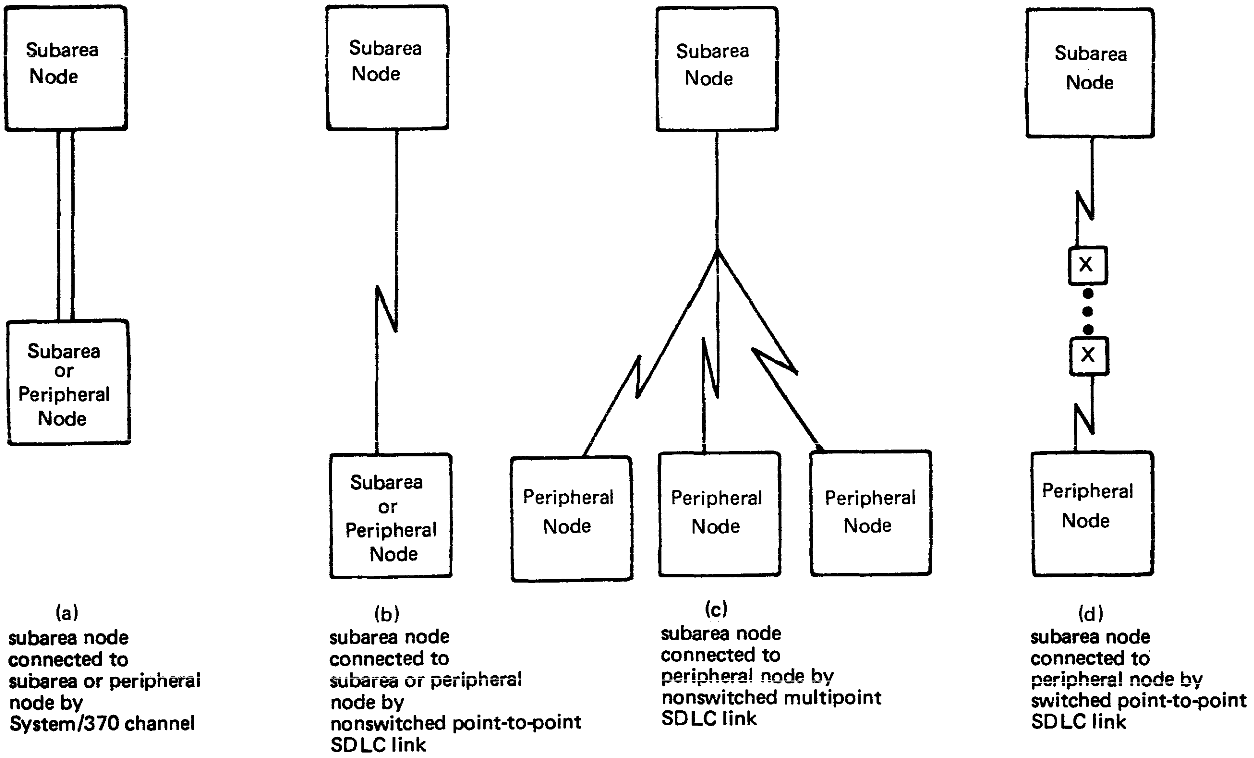


Figure 2-12. System/370 Channel and SDLC Link Connections between SNA Nodes

SDLC links can use various kinds of public network facilities, among them telephone lines, microwave links, and satellite links. These links are characterized as point-to-point or multipoint, and are either permanent (nonswitched) connections or temporary (switched) connections. Other characteristics of public network facilities may vary. One such characteristic is the rate at which they can carry data.

An SNA network may use any of these facilities, in various combinations, depending on such factors as the geographic locations of nodes, transmission rates required, availability of specific facilities where the nodes are located, and costs of various alternative facilities.

An important attribute of an SNA network is the independence of its logical organization both from the physical configurations of SNA nodes and links and from the particular telecommunication facilities used. Thus, although the choices of facilities used will affect the performance of an SNA network, the logical organization of the network does not depend on these choices. Often, the network owner can change the telecommunication facility used by any part of the network without disrupting the network or the applications that use it. For example, application programs served by the network need not be modified when such changes occur.

Parallel Links

Adjacent SNA nodes are usually joined by a single link. However, two subarea nodes can be joined by two or more SDLC links operating concurrently (see Figure 2-13). These links are referred to as *parallel links.*

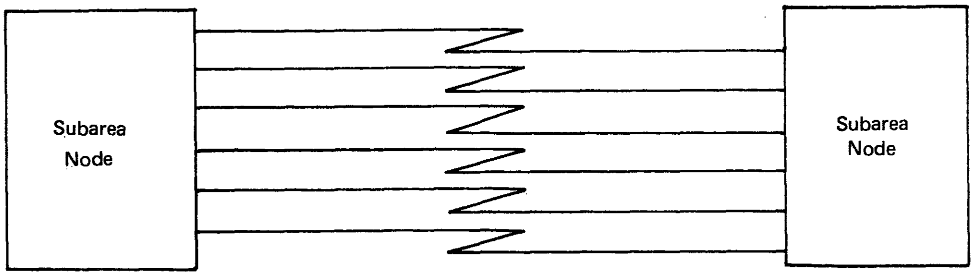


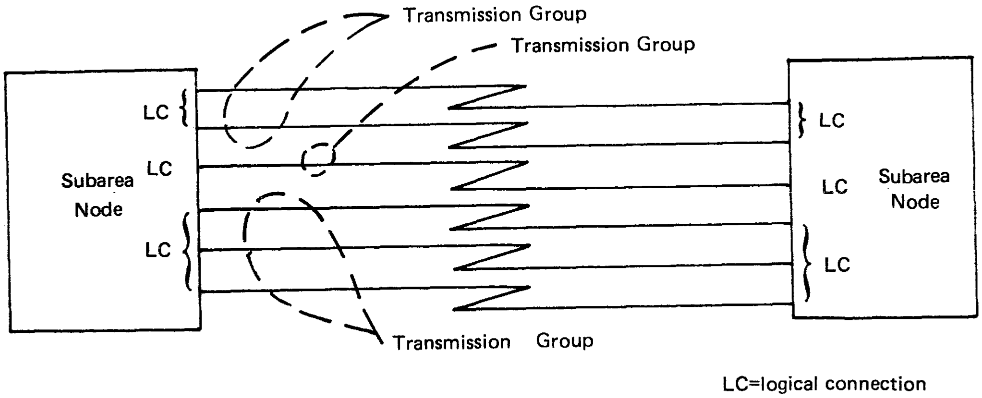
Figure 2-13. Parallel Links between Subarea Nodes

The data traffic flowing between the adjacent nodes is distributed among the links. The distribution is determined partly by the association of links and routes made by the network designer or system programmer. It is also determined partly by the class of service specified for each session when that session is being activated. Routes and class of service are discussed under "Paths and Routes" and "Services of the Path Control Network", respectively, later in this chapter.

Using parallel links may be cheaper than using a single link. This may be the case if the tariff charges for a single high-capacity link are disproportionately greater than the total charges for two or more links of equivalent combined capacity. Parallel links can also be used to provide needed capacity that a single link cannot provide.

Transmission Groups

When adjacent subarea nodes are connected by parallel links, each link is logically specified as a member of a *transmission group.* (See Figure 2-14.)



LC=logical connection

Figure 2-14. Transmission Groups between Subarea Nodes

A transmission group can associate parallel SDLC links of equal data-carrying capacity into a single logical connection whose capacity exceeds that of a single link. Data traffic within sessions using the transmission group is automatically distributed among the active links in the group.

A transmission group of parallel links has better availability than a single link. So long as at least one of its links is operating, the transmission group can carry data. When a link does fail or is taken out of service, data flow within the sessions using that link is automatically redistributed to other active links in the group. The loss of a link therefore does not disrupt the sessions that are currently using it.

Parallel links between a pair of subarea nodes can be placed in one or more transmission groups, as Figure 2-14 shows. (Each link can belong to only one transmission group at a time.) This lets the network designer place links having similar characteristics (for example, high transmission speed and average reliability) in one group and place links having other characteristics in common (for example, medium speed and high reliability) in another transmission group. A link can be transferred from one transmission group to another while the network is operating.

All of the data exchanged within a given session passes over only one of the transmission groups between a pair of adjacent subarea nodes.

A session should be assigned to a transmission group that is suitable for the session. For instance, sessions performing batch data transfers can tolerate the relatively high propagation delays inherent in satellite transmissions. Where both satellite and terrestrial links are available, batch sessions might be assigned to transmission groups containing satellite links. Interactive sessions, on the other hand, would be assigned to a transmission group containing terrestrial links.

Each link in a transmission group is controlled by its own SDLC protocol. This means that individual links can be added to or deleted from the group as necessary. It also means that error statistics can be collected separately for each link. This is of value in identifying a particular link having poor performance.

The term *transmission group* is also applied to single SDLC links and to channels.

Network Resources, Domains, and Control Points

The resources of an SNA network are the machines (terminals, controllers, and processors), the links that interconnect them, and the control programs and application programs they contain. The physical resources of machines are represented and managed by physical units; programs are represented by logical units. The use of these resources is managed by a coordinated set of controls within network addressable units.

A network designer may organize the network resources into subsets called *domains*. A system services control point (SSCP) oversees the management of the resources in a domain. A network whose resources are divided into two or more domains is called a *multiple-domain network*.

A multiple-domain network has three levels of resource management. At the lowest level are resource controls located within physical units and logical units of each SNA node. The controls in physical units activate and deactivate the machine itself (in an SNA sense), each link attached to the machine, and the path control network resources it uses. Activating a machine enables it to participate as an SNA node in the network. Activating a link allows other SNA nodes to be reached over that link.

The controls in logical units activate and deactivate resources they need in order to engage in active sessions with other logical units.

The intermediate level of resource management is performed by the system services control point. The SSCP manages all of the physical units and logical units in its domain and therefore has overall control of all of that domain's resources. The SSCP also obtains error statistics for physical units in its domain, when requested to do so by a network-management application program such as the Network Communications Control Facility (described in Chapter 5). The error statistics are filed and can be viewed by network operators for problem determination purposes.

The highest level of resource management is conducted by cross-domain coordination between system services control points via SSCP-SSCP sessions. SSCPs interact with one another as peers to coordinate the initiation and termination of cross-domain LU-LU sessions.

Some SNA networks may be simple enough that all of their resources can conveniently belong to a single domain. Such a network, called a *single-domain network,* does not have the third level of resource management (via SSCP-SSCP sessions) mentioned above.

Shared Control of Resources

Each resource in a single-domain network is always managed by the same SSCP because the network has only one. In a multiple-domain network, control of resources can be shared by the several SSCPs. An SNA technique called *shared control* allows a network designer to determine which SSCPs are to control the links, physical units, and logical units. Shared control also allows an SSCP to be notified when some other SSCP is no longer active in the network. This can result from planned shutdown of the SSCP or from a failure that prevents the SSCP from controlling some or all of its resources.

For some resources, sharing of control is serial. That is, only one SSCP at a time can control the resource. When that SSCP relinquishes control, another SSCP can assume control. Logical units can only be serially shared. Physical units in peripheral nodes also can only be serially shared.

Control of other resources can be shared by more than one SSCP concurrently. Physical units within subarea nodes can be concurrently or serially shared, as can SDLC links attached to subarea nodes.

Each SNA resource has a *share limit* that specifies the maximum number of SSCPs that can concurrently share control of that resource. The share limit is 1 for resources that can be shared only serially—for example, logical units. Resources that can be concurrently shared have a share limit greater than 1. The network manager can establish the desired share limit for such resources. (The share limit does not apply to physical unit control points [PUCPs]. These control points can always share control of a physical unit or a link with any number of SSCPs.)

A network designer might arrange for shared control of SNA resources for various reasons. One reason is to help maintain the availability of a resource when an SSCP currently managing it becomes unable to do so. (This might happen, for example, upon the failure of a host processor containing an SSCP. In this case SNA resources associated with a node within the domain of the failed SSCP can acquire services from a different SSCP.)

Another reason for shared control is to allow a given resource to be managed from different geographic points at different times of day.

Activating and Deactivating SNA Resources

An SNA network is a collection of logical resources (such as subarea nodes, peripheral nodes, SSCPs, physical units, and logical units) superimposed on a collection of associated physical resources (such as processors, communication controllers, links, cluster controllers, and terminals). The physical resources are usually activated and deactivated manually.

The logical resources are activated after the associated physical resources are activated; conversely, logical resources are usually deactivated before physical resources. SSCPs control the activation and deactivation of logical resources based on network design specifications, operator commands, and requests by end users.

An SSCP performs several functions to activate a network domain. First, it checks that the physical resource associated with each logical resource is active. Second, it checks that the logical resources it has activated are the ones it intended to activate. And third, it becomes a manager of the resources it has activated.

Advantages of the SNA Activation and Deactivation Techniques

The activation and deactivation techniques used by SNA allow the system programmer considerable latitude in specifying how the network is to be activated and deactivated.

A "cascaded" activation capability allows the programmer to increase or decrease the scope of operator commands used to activate resources. Using parameters associated with network definition statements in access methods, the programmer can cause most of the network to be activated automatically; of the remaining resources, selected ones can be activated individually by operator commands. For example, the programmer can specify that when an operator activates a communications controller, the SSCP in the access method will automatically activate all associated links, peripheral nodes, and logical units except for, perhaps, one particular peripheral node and its logical units.

SNA also provides a cascaded deactivation capability by which a network operator can enter a single command that deactivates multiple resources. When reactivating network resources after part of the domain has failed or has been deactivated, the operator can either restore the inactive resources to the status they had just before the failure or deactivation or can restore them to their initial status as defined by the system programmer.

In summary, the SNA network activation and deactivation capabilities are an important technique for network management; they allow the network designer great latitude in customizing the sequence and method of activation to meet the needs of the network.

Configuring and Reconfiguring the Network

Before an SNA network can be made operational, its physical configuration and some of its logical configuration must be defined in control programs that will be executed in subarea nodes. These programs include those containing an SSCP, such as access methods (for example, ACF/TCAM and ACF/VTAM) and processor control programs (for example, DPPX and DPCX for the IBM 8100 Information System), and programs without an SSCP (for example, ACF/NCP/VS). The configuration is defined by system programmers in the form of tables that are loaded into these subarea nodes as part of the access methods and control programs.

The configuration of an SNA network is unlikely to remain exactly as it was originally specified. Both scheduled and unscheduled changes are probable, and SNA products are designed to facilitate these changes with minimal disruption to network operation.

Scheduled Changes

Scheduled changes can be accomplished in either of two ways. One way is to use the same technique used for initially configuring the network. That is, a system programmer re-codes one or more of the tables that define the configuration as necessary to reflect the changes. Then the tables are loaded (again, as part of the access method or network control program) into the affected nodes. This activity requires that resources controlled by the node be deactivated for as long as it takes the revised programs to be loaded into the nodes.

The other way to make a scheduled change is through dynamic reconfiguration. In this method, a network operator adds peripheral nodes and logical units to the network and deletes them from the network. The operator makes these changes while the network is running; resources need not be deactivated.

Once the configuration of the network is established, SNA provides the network operator with additional control over the state of the configuration. By communicating with the SSCP, the network operator can activate and deactivate entire nodes, resources within nodes, and links between nodes.

Unscheduled Changes

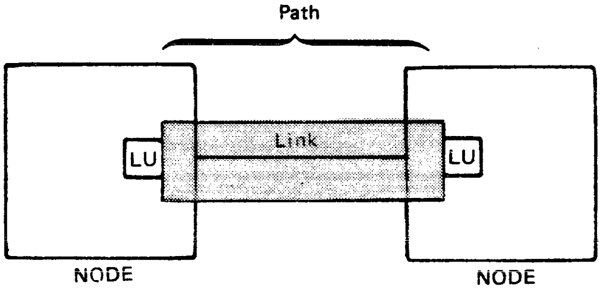
Unscheduled changes in configuration are inevitable. Links between nodes may fail or their performance may become severely degraded. And nodes or resources within nodes may fail. Network operators can sometimes circumvent these failures by reconfiguring the network. As noted above, network operators can dynamically add, delete, activate, and deactivate certain resources.

The SSCP helps in this effort by automatically notifying network operators of link and node failures. SNA products provide functions that display the current status of network resources, accumulate and display error statistics, and perform certain problem determination actions, as described in Chapter 5.

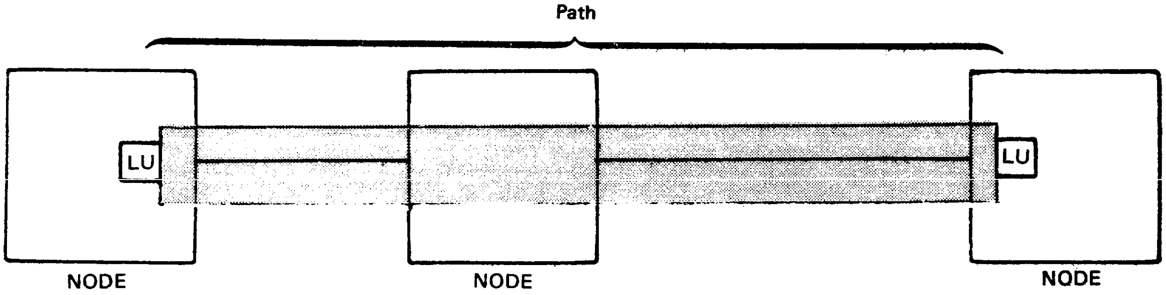
A network can be configured so that parallel links are available between certain nodes, and the configuration may include multiple routes between some subarea nodes. SNA is designed to allow a network operator to substitute one link or route for another without disrupting sessions between logical units not directly affected by the failed link or route.

How Data is Routed within the SNA Network

In order for end users to be able to communicate with one another, there must be established between their respective logical units a path over which the LUs can communicate. After describing this path as a sequence of physical connections, this section explains it as a logical connection between the LUs. As shown in Figure 2-15 (a), the path may consist of a single link if the logical units are in adjacent nodes. The link can be a channel or an SDLC link. (If the adjacent nodes are both subarea nodes, the path can consist of multiple, parallel SDLC links.) Most networks, however, also serve end users at nodes that are connected by a series of other nodes and links, as shown in Figure 2-15 [b].



(a) Path between LUs in Adjacent Nodes



(b) Path between LUs in Non-adjacent Nodes

Figure 2-15. Paths between Logical Units

Regardless of the number of nodes in the path between pairs of logical units (or other network addressable units), a routing scheme is needed for delivery of message units from the logical unit that originates them to the logical unit that is to receive them.

The routing scheme in SNA depends on an addressing structure that is consistent throughout the network.

Addressing

Each SSCP, physical unit, and logical unit has a network address that identifies it to other network addressable units and to the path control network. (Some logical units in subarea nodes can have more than one network address.) Some addresses are assigned when the network configuration is defined by the system programmer. Other addresses are assigned while the network is operating.

For example, while the network is operating, addresses for physical units and logical units reached over switched links are dynamically assigned when the dialed connection is made. As another example, network addresses are dynamically assigned when new physical units and logical units are added to the network through a technique called dynamic reconfiguration.

Each message unit flowing in a session contains the address of the destination network addressable unit. This destination address is examined by each node in the path; routing tables in the node determine the link over which the message unit must be transmitted to reach the next node on the way to its destination.

Each network address is divided into two parts: a *subarea address* and an *element address*. These are used as follows.

Each domain of the SNA network is divided into subareas. As explained earlier, a subarea consists of a subarea node, all of the peripheral nodes (if any) attached to that node, and the resources (such as links and network addressable units) associated with the subarea node and its peripheral nodes. That is, a subarea is either (1) a communication controller, all cluster controllers and terminals attached to it, and associated SNA resources, or (2) a processor containing an SSCP, all cluster controllers and terminals attached to it, and associated SNA resources.

The subarea address is the same for all network addressable units in the subarea. The element address part of the network address is unique to each network addressable unit within that subarea.

Each subarea node has a routing table that reflects the configuration of nodes and links attached to it. The table specifies how the path control function (part of the path control network) in that node is to route message units it receives. The table in each node contains a list of all destination subareas that can be reached from that node. Associated with each listed destination is the address of the transmission group and the physical address of the subarea node on that transmission group to which the message unit must be sent next in order to proceed toward its destination.

Each message unit proceeds through the network from one subarea node to the next until it reaches the subarea node corresponding to the subarea part of its destination's network address. The message unit has then reached its destination subarea. Next, path control in that subarea node examines the element address part of the network address. Using a supplemental table called an element routing table, path control passes the message unit to its destination network addressable unit. This network addressable unit may be in that subarea node or in an attached peripheral node. Before forwarding the message unit to its destination in the peripheral node, path control converts its network addresses to local addresses that are meaningful only to the peripheral node and its subarea node. This conversion is performed by a component called *boundary function.*

Peripheral nodes are insensitive to network addresses. This fact greatly simplifies the task of reconfiguring the network. For example, network addressable units within separate peripheral nodes can have identical local addresses. When it becomes necessary to change the physical connections between subarea nodes, only the routing tables in those nodes need to be updated with the revised network addresses. Local addresses are not affected by the updates to the routing tables.

Network Names

Network addresses and local addresses are used only within the SNA network. It would be inconvenient for end users to have to specify actual addresses of destination logical units when sending message units to them. Moreover, any change to a physical network configuration, or any relocation of an end user (application program) to a different node, would require that end users specify different addresses to reach the destination logical units.

End users therefore refer to network addressable units by their *network names.* A network name can be one to eight characters convenient for the end user to use in identifying a network addressable unit; each network addressable unit within a domain must have a unique name. Different end users may sometimes refer to a given logical unit by different names, or they may not know the actual network name of the logical unit. In such cases they can specify, in a request to activate a session, an *uninterpreted name.* The SSCP in that logical unit's domain translates the uninterpreted name to the corresponding network name.

Upon receiving a session activation request from a logical unit, the SSCP examines the request for the network names of the logical units that are to engage in the session. Using a service called a network directory service, the SSCP determines the network addresses corresponding to the network names specified. The SSCP gives these addresses to the logical unit that activates the session. This may be the logical unit that requested the session or a different one.

Paths and Routes

A basic objective of SNA is that end-user data received by the path control network from an origin LU must be delivered to the destination LU without error or any change in content.

SNA networks transfer message units through the path control network without regard to the content of the end-user data they carry. That is, the routing functions within the path control network do not examine or alter this data. Any change in the method of transmission between nodes therefore has no effect on end users. The path control network does use data link control protocols specific to the kinds of links (channels, nonswitched SDLC links, switched SDLC links), forming the path between the origin and destination logical units, but SNA isolates end users from such considerations.

Another objective of SNA is that the path control network must deliver multiple message units to their destination in the same order that it received them from their origin. This relieves end users from having to buffer and resequence the message units.

All message units submitted to the path control network within the course of a single session are transmitted to their destination over the same path—that is, the same sequence of nodes and transmission groups (or single links)—between the nodes containing the network addressable units. Figure 2-16 shows a path between two end users, one an application program and the other a terminal operator. The path spans the entire distance between the logical units representing the end users.

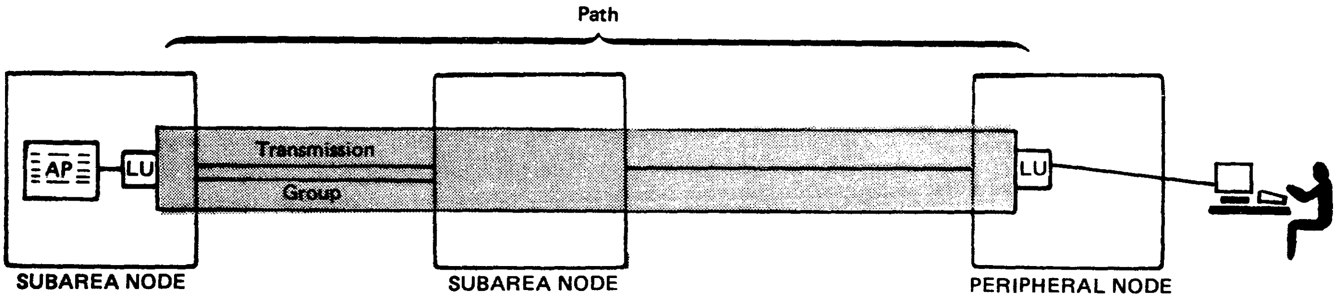


Figure 2-16. **Path between Logical Units for Application Program and Terminal Operator**

Routing Data between Subareas

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

- Minimize the transmission time that each message requires
- Maximize the number of messages that two subareas can exchange in a unit of time
- Minimize the cost of the physical components along the route
- Maximize the availability of the route
- Minimize the loss or retransmission of data due to physical errors occurring in physical components along the route
- Minimize the time needed to detect an inoperative route and reactivate the path over another route, if one is available

Compromises are involved among certain of these objectives. For example, one might minimize the transmission time of messages on a route by assigning it relatively few messages per unit of time. This would achieve the first objective above at the expense of the second.

SNA defines *explicit routes* and *virtual routes* as follows.

Explicit Routes

The components in the path between the two subarea nodes involved in a session are called the *explicit route,* as Figure 2-17 shows. The remaining components in the path—from one of the subarea nodes to the peripheral node—are called the boundary function and *peripheral link.* (If both session partners are located in subarea nodes, the path consists only of the explicit route; there is no boundary function or peripheral link.)

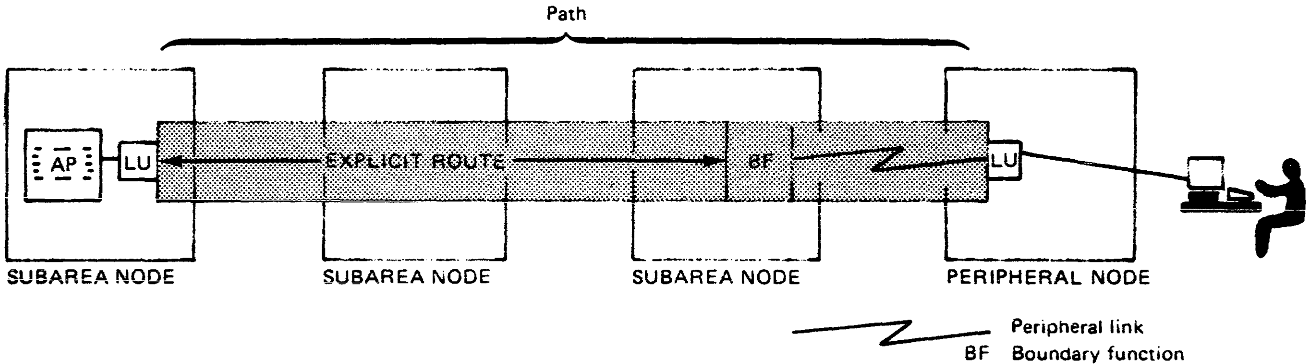


Figure 2-17. Explicit Route and Peripheral Link in a Path

Multiple Active Routing

An important capability of SNA networks is their ability to provide more than one route between any two subarea nodes. This capability, called ***multiple active routing,*** allows traffic loads for different sessions to be distributed over several routes. Providing multiple routes in a network also increases the probability that a route will be available when an attempt is made to establish a session. Figure 2-18 shows three explicit routes between subarea nodes.

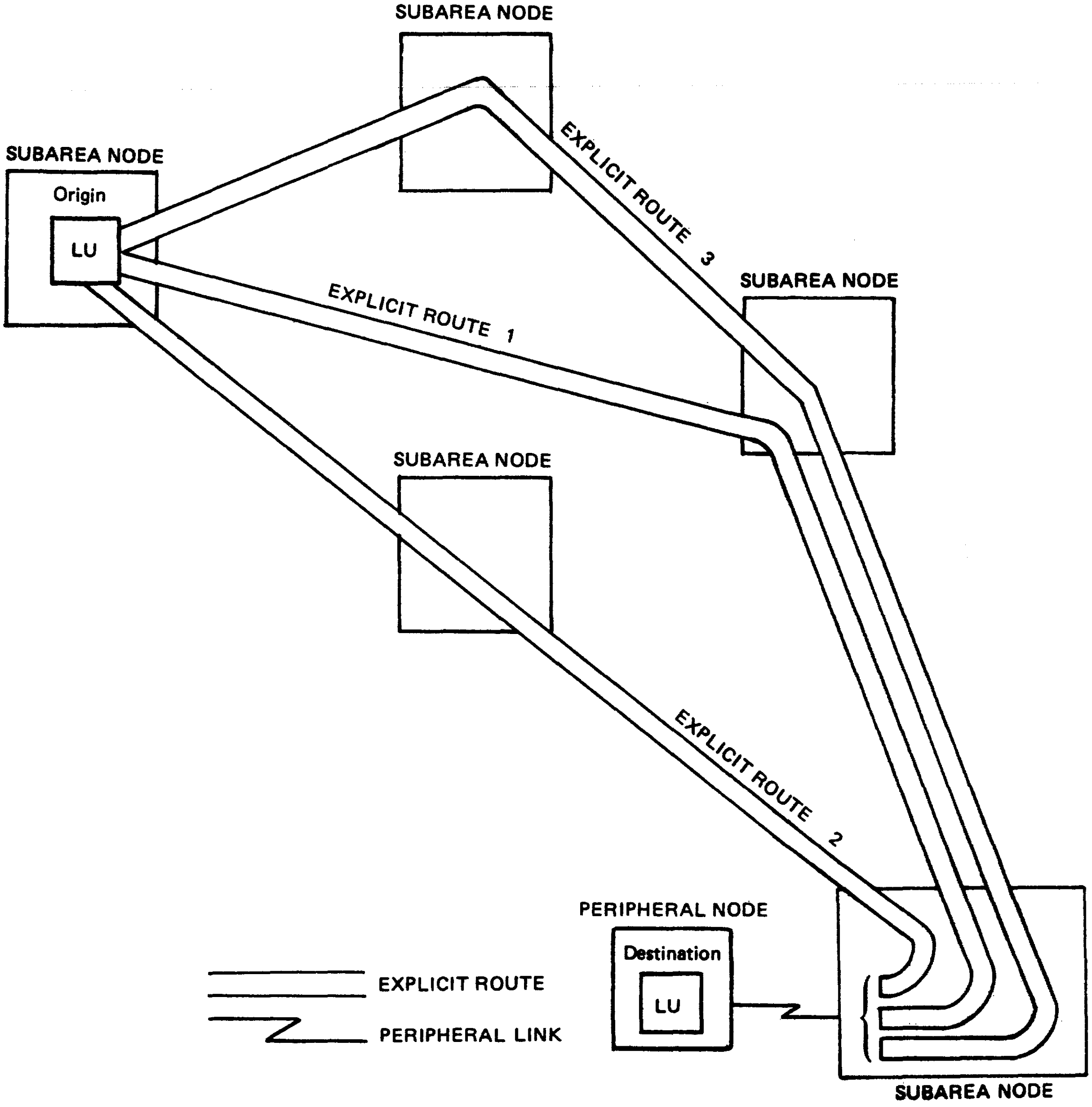


Figure 2-18. Multiple Active Routes between Subarea Nodes

If a route being used by a session becomes inoperative for any reason (as, for example, the failure of a transmission group or a subarea node), the path control network detects the fact and automatically notifies all of the session partners (LUs, PUs, and SSCPs) engaged in sessions over that route. The sessions may be reestablished over one of the remaining active routes, if any, or over a newly activated route. This SNA capability is one of the ways in which the availability of a network to carry traffic may be enhanced.

Although Figure 2-18 shows only three explicit routes, as many as eight may actually be defined between any two subarea nodes.

Besides improving network availability and allowing loads to be balanced, explicit routes provide control over the physical routing of session traffic for other reasons. For example, some routes may be more secure than others and therefore appropriate for use in transferring sensitive information.

Virtual Routes

The multiple routing capability also involves a routing and flow-control technique called virtual routing. Virtual routing associates flow control properties with explicit routes. Virtual routing also controls the integrity of data flow on a route by assigning sequence numbers to message units sent onto the route and verifying sequence numbers of message units received from the route.

A *virtual route* logically connects the end subarea nodes that are involved in a session. Each virtual route is assigned to an underlying explicit route; more than one virtual route may be assigned to the same explicit route. Figure 2-19 shows a virtual route between logical units in two subarea nodes. (Multiple virtual routes can use the same underlying explicit route.)

Figure 2-17 shows that the path between session partners in a subarea node and a peripheral node includes an explicit route, a boundary function, and a peripheral link. At a higher level, the path consists of one of the virtual routes using the explicit route and a *route extension*. The route extension is the part of the path that consists of a boundary function and a peripheral link. (If both session partners were located in subarea nodes, no boundary function or peripheral link would be present and the path would consist entirely of the virtual route.)

A virtual route is identified by the subarea addresses at each end of the route, a virtual route number, and one of three transmission priorities—high, medium, and low. Message units for a given session flow through the network at one of these priorities.

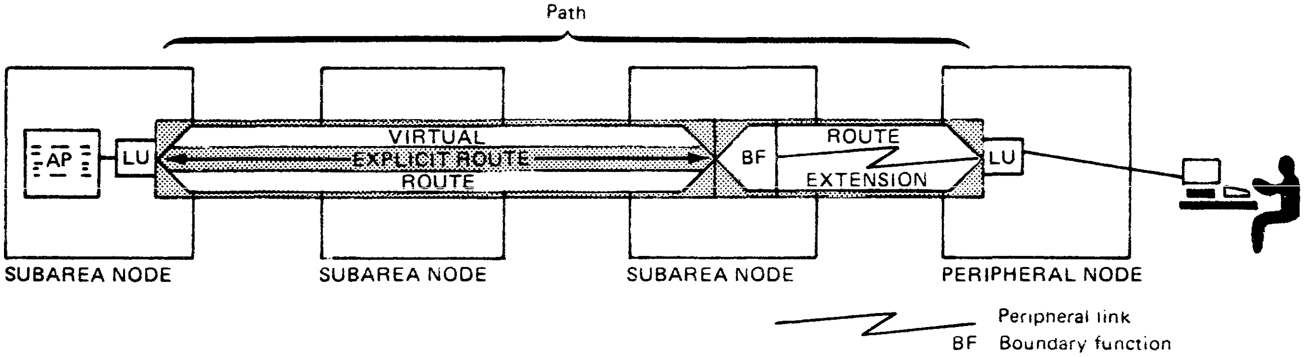


Figure 2-19. Virtual Route and Route Extension in a Path

Transmission Priority

Each subarea node has queues for message units transmitted on sessions. The message units flow over a route in a series of steps. Each is sent over a transmission group to the next subarea node, placed on a queue in that node, then transmitted over a transmission group to the next node, again placed on a queue, and so on until it reaches its destination.

At each queuing point there is an opportunity to favor some message units by transmitting them ahead of others. Message units having high transmission priority are transmitted ahead of those having medium or low priority, and message units having medium priority are transmitted ahead of those having low priority. The lower-priority message units are retained in queues until the higher-priority ones are sent. (This priority technique is not used on channels, however; the much higher data transmission rate over channels, as compared with SDLC links, makes use of this technique unnecessary.)

The use of three levels of transmission priority helps assure that desired applications (such as interactive applications) continue to experience good response times even when the traffic load over routes the application uses is heavy. (SNA assures, however, that low- and medium-priority message units are not retained in queues for excessive periods; such message units are eventually sent even when the route experiences sustained peak loads.)

The system services control point associates lists of specific virtual routes with class-of-service designations that the end users specify when initiating the sessions. (Class of service is described later in this chapter under "Services of the Path Control Network.") During session activation, a list is passed to an LU for use in selecting a particular virtual route.

Once assigned to a specific virtual route, a session continues to use that route until the session is deactivated.

There is a fourth level of transmission priority, called *network priority.* This level is higher than the three mentioned above and allows virtual-route pacing responses to be transmitted ahead of all other traffic. Virtual-route pacing is discussed later in this chapter.

Activation and Deactivation of Explicit and Virtual Routes

Access methods and network control programs in subarea nodes activate explicit routes and virtual routes, and deactivate virtual routes as needed. The network operator enters commands to activate or deactivate physical network resources. The virtual routes and underlying explicit routes are automatically activated only when they are needed to carry session traffic. Neither explicit routes nor virtual routes are directly activated by the operator, and the operator need not know which virtual routes are active at any given moment.

A virtual route is deactivated automatically when all sessions assigned to it have ended.

Benefits of the SNA Routing Technique

Listed earlier under "Routing Data between Subareas" were some routing objectives a network designer might consider. SNA helps designers meet those objectives in the following ways.

Rather than specifying, in each subarea node, the entire sequence of links making up a path, SNA distributes the path specification over all of the nodes in the path. Each node therefore contains only a part of the path specification. This technique saves storage space in the individual nodes and helps users to reconfigure the network.

SNA avoids the rigidity of always assigning sessions to the same route by assigning sessions to a virtual route during session activation.

SNA provides transmission priorities for virtual routes, parallel links in transmission groups, multiple explicit routes between subareas, and virtual-route pacing to prevent congestion that could delay messages. This allows the network designer to minimize the transmission time of messages through the network.

SNA provides multiple links, multiple explicit routes, and virtual-route pacing to allow the network designer to maximize the number of messages exchanged between two subareas in a given period of time.

SNA provides virtual-route pacing and data link control capabilities that can facilitate efficient use of path components and can help designers to minimize the cost of those components.

SNA allows availability of paths to be enhanced through use of multiple links in a transmission group and multiple explicit routes between subareas. SNA products also provide facilities for informing session partners when their sessions are disrupted because of route failure and for allowing them to reestablish disrupted sessions over alternate routes.

SNA provides data link control techniques for detecting and recovering from errors as they occur to minimize loss or retransmission of data because of those errors. SNA products provide network management facilities for identifying error-prone path components by collecting and displaying error statistics for such components.

SNA products provide facilities for informing network operators when routes fail, for informing session partners when route failures have disrupted their sessions, and for allowing them to reestablish disrupted sessions over alternate routes. These facilities minimize the time needed to detect and reactivate an inoperable path.

The Services that an SNA Network Provides

SNA networks provide two broad categories of services, as shown in Figure 2-20. One category is the services of the path control network. These services fulfill the fundamental purpose of any network: to transmit data quickly and accurately between network locations, regardless of how distant they are from one another. Services of the path control network are described later in this chapter.

The other category is NAU services, which facilitate the exchange of data by pairs of end users connected to each other through the path control network. This category also includes services that allow the network to coordinate its activities, such as resource allocation, through its nodes.

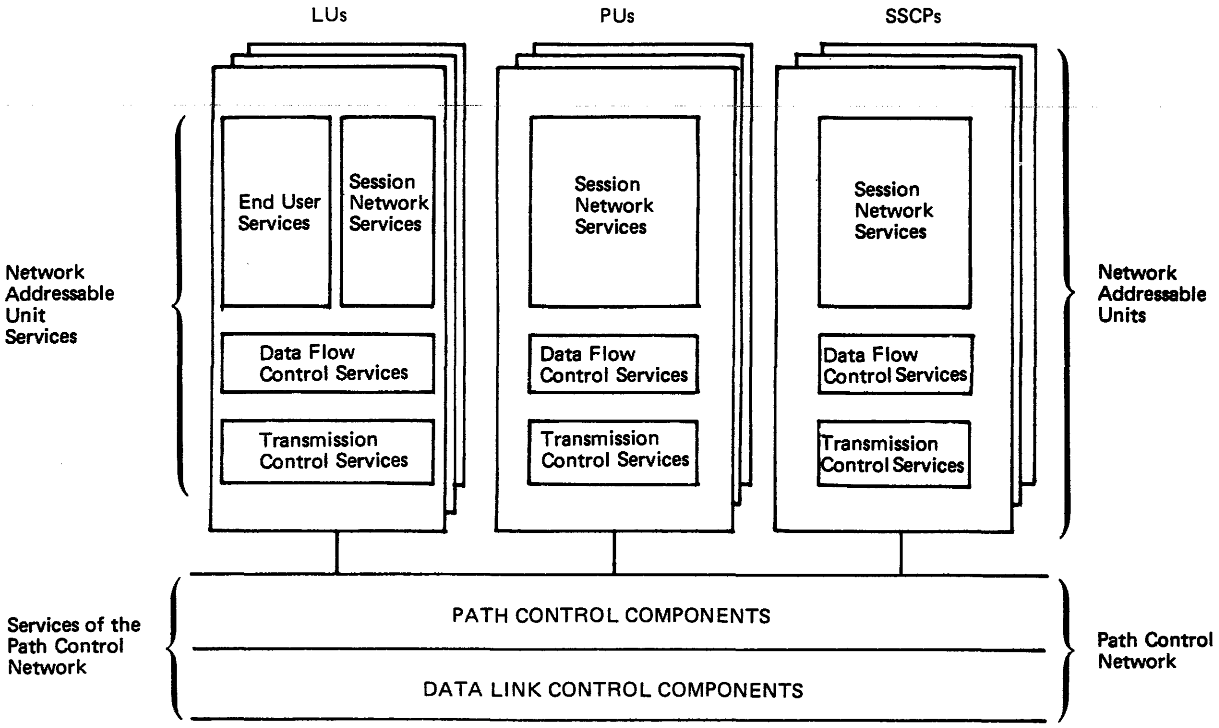


Figure 2-20. Distribution of Services within SNA Network

NAU Services

NAU services used in exchanging data between end users are referred to as *end-user services.* These are the principal services provided by logical units. NAU services that allow the network to coordinate its activities are called *session network services.* These are located in all three kinds of network addressable units. Each network addressable unit also contains data flow control services and transmission control services (discussed later in this chapter), which lie between the other NAU services and the path control network. Figure 2-20 shows the services mentioned.

End-User Services

End-user services aid the exchange of data during sessions between logical units representing application programs and terminal operators. The two categories of end-user services are *session presentation services* and *application-to-application services.*

Session Presentation Services: When terminal operators and application programs communicate interactively, much of the logic of the application programs is concerned with the formatting and displaying of data to the operator. A variety of session presentation services can perform these functions on behalf of the application program.

Session presentation services at both ends of an LU-LU session can work together to match the needs of one end user to those of the other. An example is the common situation in which the same application program communicates with several kinds of terminal devices, such as displays, printers, and card punches. Often it is desirable for the program to send out data in a common format, regardless of the kind of device that is to receive it. But each device requires different format control characters within the data it receives from the application program. Session presentation services can perform the data transformation necessary to resolve the conflicting formatting requirements.

Other examples of session presentation services are *compression* and *compaction.* The compression function recognizes strings of repeated characters in a data stream and replaces them with a control byte indicating the number of repetitions and the character that is to be repeated. (Repeated space characters are typically represented only by the control byte.) This one- or two-byte sequence takes less time to transmit than the character string it represents. Compaction is a technique for transmitting two data characters using the same number of binary digits required for a single character; this technique is applied to the most frequently sent characters. By reducing the amount of data sent during an LU-LU session, compression and compaction make possible more efficient use of links between nodes.

Another example of session presentation services is as follows. Some SNA application subsystems allow application programs to specify a set of display screen formats appropriate for the data that the application programs will send to a terminal; each format is identified by a symbolic name. When sending data, an application program needs to specify to the logical unit only the name of the format, rather than having to supply all the format control information within each screen full of data it sends. Session presentation services within the logical unit at the terminal do the necessary formatting, based on the format name the application program specified.

Application-to-Application Services: SNA defines services for sessions joining two transaction processing systems (such as CICS/VS) in different nodes of the network. These services, accessed using application program interfaces, allow application programs in the different nodes to communicate without being concerned with the details of the network protocols.

Application-to-application services provided by some SNA products also allow an application program to gain access to a data base without being aware of the location of that data base in the network. That is, the application program uses the data base in the same manner whether it is at the same node as the application program or at some other node.

Another service some SNA products provide to applications is a set of protocols for synchronizing activities of two or more interdependent programs.

Synchronization is especially useful in transaction processing applications. Such applications frequently consist of multiple-step actions performed by separate programs, often executed in different processors. All of the actions must be completed as a unit because only the combined result of all the actions is meaningful. An example is the need to synchronize updates to correlated records in a distributed data base.

The synchronization protocols allow a program to reserve a resource for its exclusive use and cause its logical unit to record all activity involving the resource. When the application program has successfully completed all processing related to a specific unit of work, the record of changes is erased and the work unit is referred to as being "committed." At that moment the processing has reached a synchronization point.

If, on the other hand, not all of the related actions were successfully completed, the processing steps already performed are cancelled and the state of the resource is restored to the previous synchronization point.

Session Network Services

Session network services, located in system services control points (SSCPs), logical units, and physical units, are divided into the categories of *configuration services, network operator services, session services,* and *maintenance and management services.*

Configuration Services: This category of session network services is responsible for controlling the resources associated with the physical configuration of the SNA network. This function includes activating and deactivating links between nodes. Configuration services are also used to load programs into SNA nodes and to dump the storage contents of SNA nodes. Configuration services are used by SSCP-PU sessions; they are invoked by the network operator. Configuration services allow the operator to start up the network by activating its nodes and links in the manner described earlier in this chapter. Configuration services also allow the operator to alter the configuration of an active network, to restart parts of it that have failed, and to shut down the network at the end of an operating period.

Configuration services within the SSCP maintain tables containing the network name, network address, and status (activated or deactivated) of each link and each network addressable unit within the domain of the SSCP.

Network Operator Services: Network operator services facilitate communication between network operators (including programmed operators) and SSCPs. Examples of such communication are operator commands for starting and stopping the SNA network, activating and deactivating network resources such as links, physical units, and logical units, and collecting error statistics from nodes in the network.

Session Services: These services help to activate and deactivate sessions upon request. Session services are distributed between the SSCPs and the logical units of a network.

A session between two logical units can be initiated by one of the two logical units that are to participate in the session, by a different logical unit, by a network operator, or by predefined start-up procedures. In any case, session services (located in the SSCP and the logical units) cooperate to activate and deactivate the sessions. Session services are used by SSCP-SSCP and SSCP-LU sessions.

A major function of session services in the SSCP is to convert the network names provided by logical units that are initiating a session into the corresponding network addresses by which the logical units are known within the network. Some logical units can participate in multiple sessions concurrently. The number of concurrent sessions is, however, limited by the availability of resources used by the logical unit. Session services therefore enforce this session limit.

Maintenance and Management Services: Maintenance and management services allow an SSCP to conduct various tests that may help to determine whether the link to a node has failed or the node itself has failed, and to determine the cause of the failure. These services also help the SSCP gather error statistics and test results from nodes. SSCP-PU and SSCP-LU sessions use maintenance and management services.

Data Flow Control Services

Maintaining the integrity of the data flow within a session between a pair of network addressable units is a complex task. Although some parts of this task are performed by transmission control services (discussed in the next section), most are performed by data flow control services.

Some of the functions that data flow control performs are as follows.

Send/Receive Modes: SNA defines the sessions between any pair of network addressable units as bidirectional, that is, capable of transmitting message units in both directions concurrently. While this may be appropriate for some end users, for many it is not. Data flow control therefore provides a choice of send/receive modes. One is the concurrent bidirectional flow just mentioned—called *full-duplex mode.*

Another is *half-duplex flip-flop mode.* In this mode the network addressable units alternate sending data to each other. The network addressable unit that is sending at a given moment can change the direction of data flow by signaling the other to begin sending. This is usually the appropriate mode for such applications as inquiry-response, in which a terminal operator at one node is using an application program at the same or a different node to access data from a data base.

The third kind of send/receive mode is *half-duplex contention.* In this mode either logical unit can begin sending message units. After it has begun, the logical unit continues until it has finished sending all the message units of a chain (chains are discussed below). Then the other logical unit can begin sending. If both logical units try to begin sending simultaneously—a situation referred to as *contention*—then one or the other, by prior agreement, is allowed to prevail.

Chaining: Chaining allows related message units that are to be transmitted in the same direction to be logically grouped into a single, larger unit called a *chain.* Because the data is related, an error detected in any message unit in the chain causes the rest of the message units in the chain to be ignored. The chain is ended quickly so that recovery can begin.

Brackets: Whereas chains are sequences of related message units to be transmitted together in a single direction, *brackets* are related sequences of message units or chains of message units that flow in both directions between a pair of logical units and that represent a particular unit of work. Indicators within message units tell when a bracket is beginning and when it is ending. Using brackets allows applications that process sequences of transactions on a given session (for example, airline seat reservation applications) to keep the data related to one transaction separated from the data related to other transactions.

Response Options: Message units transmitted through the network are acknowledged by the logical units that receive them; the acknowledgment indicates whether they were received error-free or contained errors. Data flow control provides several options for applications to use in handling these acknowledgments, which are called *responses.* For example, an application might require that the receiving logical unit send a response only if any message unit in a chain contains an error. This is a negative response.

Other Functions: Data flow control allows end users to directly influence the flow of message units between logical units. One data flow control function, for example, allows an end user to temporarily stop the flow of data between logical units without actually ending the session between them.

Transmission Control Services

Once a session is active, transmission control keeps track of the status of that session, helps pace the flow of data within it, and assures the correct sequencing of message units within the session. Transmission control receives message units from the path control network and routes them to the appropriate points within the network addressable unit. Some message units are destined for end users, while others are intended for the data flow control or transmission control functions.

Because message units received from the path control network are stored in buffers, which are limited resources, it may be necessary to regulate the flow of message units within a session to avoid overrunning the buffers. This regulation, referred to as *session-level pacing,* is another service of transmission control. Session-level pacing allows logical units having widely divergent processing speeds and buffering capabilities to communicate via LU-LU sessions with minimal danger of overrunning the buffers.

Session-level pacing allows a logical unit to send data to its session partner only as fast as the session partner can receive and process that data without overloading its resources. The logical unit sends a fixed number of message units, called a *pacing group,* then waits to receive a pacing response indicating that the session partner is ready to receive more data. Then the logical unit sends the next pacing group. If the session partner sends the pacing response early (before having received all the message units in the pacing group), the sending logical unit can begin sending the next pacing group immediately after the current one.

Pacing can occur in one stage or two stages and in either or both directions of data flow, as agreed by the session partners when the session is activated. (Or they may agree not to use pacing.) In two-stage pacing from a host logical unit to a peripheral logical unit, the first stage occurs between the origin subarea node and the boundary function of the subarea node (the "boundary node") to which the peripheral node is connected; the second stage occurs between the boundary node and the peripheral node.

The size of the pacing group is established when the session is activated. It may be different for the two directions of flow within the session, and it may be different for the two stages of two-stage pacing.

Session-level pacing is used for data traffic flowing in SSCP-SSCP sessions as well as in LU-LU sessions.

Transmission control also constructs headers and appends them to message units it is delivering to the path control network for transmission. These headers contain control information such as the indicators for the chaining, brackets, and pacing functions mentioned earlier.

As a security measure, transmission control can encipher end-user data it passes to the path control network, and decipher end-user data it receives from the path control network, in a process called *session-level cryptography.* This process involves the use of a data encryption algorithm and session cryptography keys. Logical units can specify whether encipherment of end-user data is to be mandatory or selective. In mandatory encipherment, transmission control enciphers all message units containing end-user data before passing them to the path control network. In selective encipherment, transmission control enciphers only those message units in which an enciphered-data indicator is set.

Services of the Path Control Network

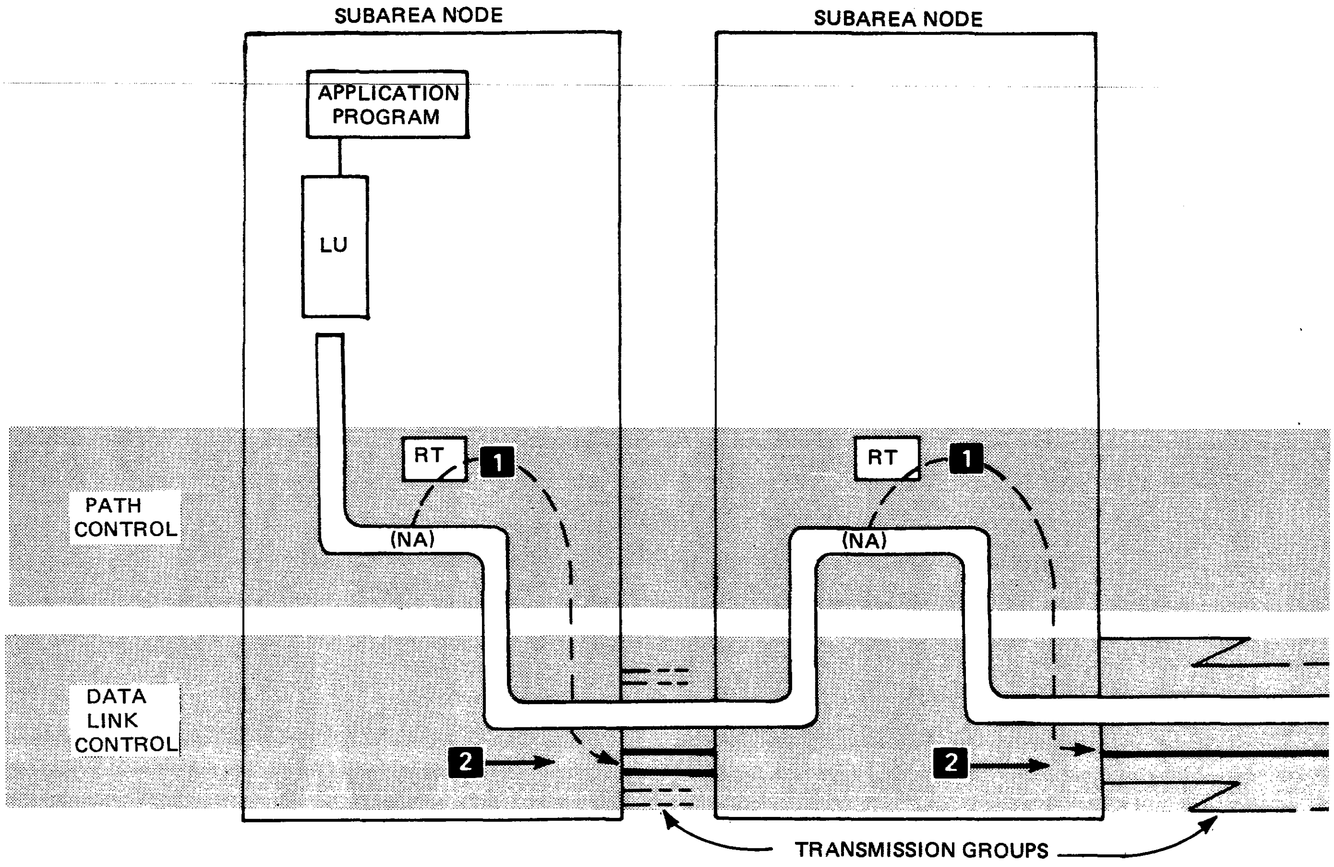
Routing and flow control are the major services provided by the path control layer of the path control network, whereas transmitting data over individual links is the major service provided by the data link control layer of the path control network. Figure 2-20 shows the path control network divided into these two parts.

Routing

Upon receiving message units from a network addressable unit, virtual-route and explicit-route functions within path control determine the next transmission group those message units must pass over in order to proceed toward their destination. The message units are then passed to data link control, which transmits them to the next node.

Upon receiving the message units, path control in the next node repeats the process by selecting a transmission group that will move the message units to the next following node and passing them to data link control for transmission to that node.

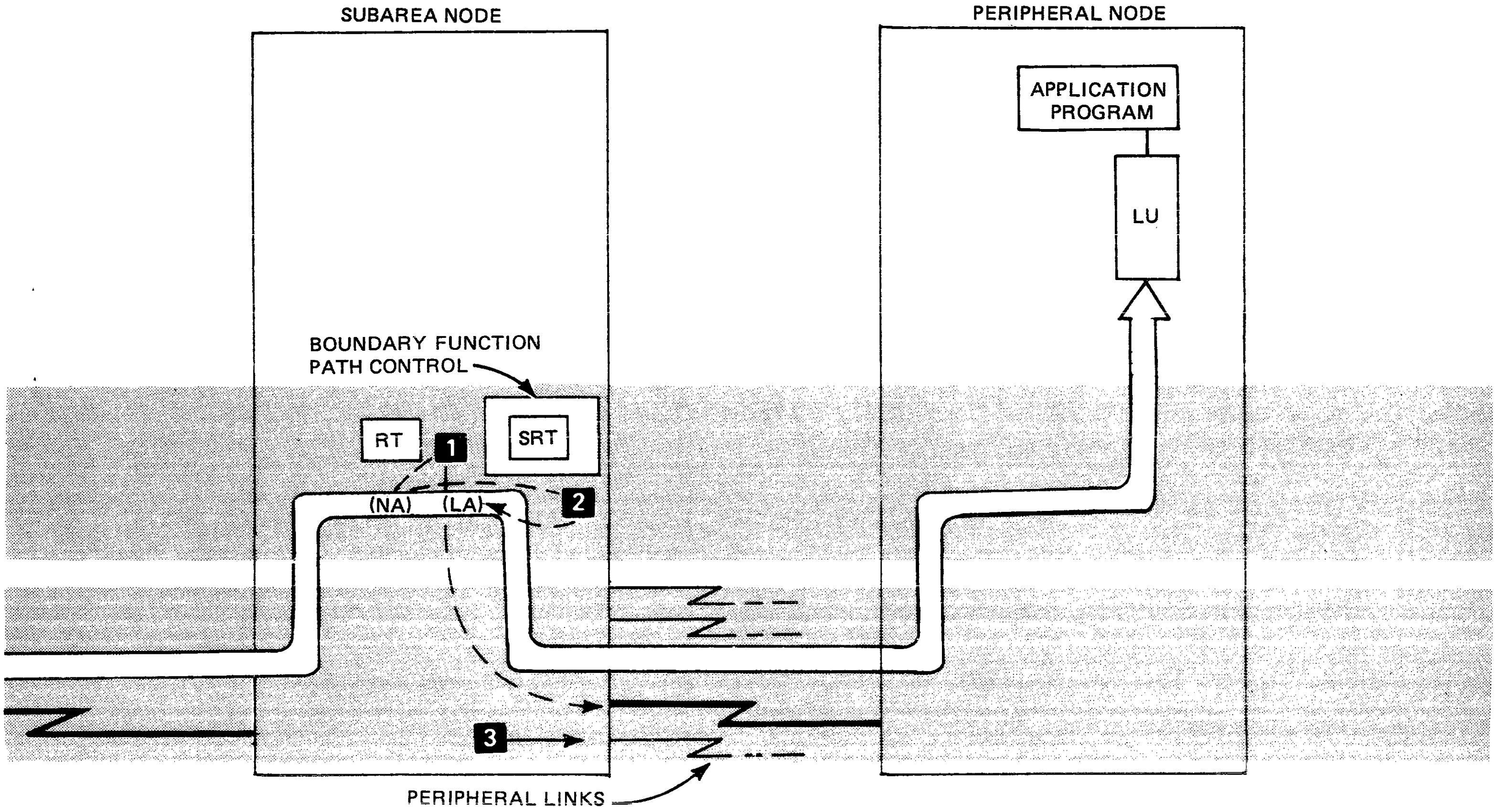
If the message units are to be delivered to a logical unit in a peripheral node, a special form of path control called *boundary function path control* converts the network address in the message units to the local address understood by the peripheral node. Figure 2-21 shows this sequence.



1 Select link and node using routing table (RT) and network address (NA)

2 Transmit to next subarea node

Figure 2-21. Routing by Path Control and Transmission by Data Link Control



1 Select link and node using routing table (RT) and network address (NA)

2 Transform network address (NA) to local address (LA) using supplemental routing table (SRT)

3 Transmit to peripheral node

For message units received from a peripheral node, path control in the attached subarea node consults its routing tables to determine where to send them. If their destination is in that subarea node, path control passes them to transmission control in the destination network addressable unit. On the other hand, if the destination network addressable unit is in some other node, path control determines the appropriate link to move the message units toward their destination; it passes them to data link control for transmission over that link.

It is important to recognize that these routing services are internal to the path control network; the network addressable units (and end users) originating the message units do not specify actual network addresses. Instead, they specify the network *names* of the network addressable units that are the destination of message units. Thus an end user can send data to another end user without knowing where in the user-application network that end user is located.

Another path control service is segmenting of messages. For reasons of transmission efficiency and error recovery it is sometimes useful to divide a single message unit into several segments. Path control's *segmenting* function can perform this service for the end user, and different segment sizes can be specified for each link or transmission group in the path between origin and destination network addressable units.

A related function called *blocking* groups message units into larger message units. Blocking can sometimes increase the efficiency of data transfer through transmission groups—as, for example, by reducing the number of channel input/output operations that data link control needs to execute in order to transmit a group of message units.

Class of Service

End users initiating a session can request by name a particular *class of service* from the SNA network for that session. Some sessions, for example, may require a fast response time, while others may require more secure routes or more reliable connections. Such factors, along with transmission priority requirements, determine which of several possible virtual routes are appropriate for the session.

During session activation, the SNA network resolves the requested class of service to a list of one or more virtual routes that meet the class-of-service requirements. The network then assigns the session to the first virtual route in the class-of-service list that can be (or already is) activated. If the end user does not designate a class of service, a default value is assigned.

Virtual-Route Pacing

The amount of data that a network can transfer between sources and destinations during a given interval is limited. Many interrelated factors determine the limit for any particular network; these factors chiefly include the data transmission rates of individual links and the capacity of SNA nodes to receive, process, store, and send data. For reasons of cost, most user application networks are designed to accommodate the average expected traffic loads rather than the occasional peak loads. Therefore a technique for handling the peak loads is essential to avoid severely degrading the network performance.

SNA provides a flow-control mechanism to prevent virtual routes from becoming saturated with peak-load traffic they cannot handle. Called *virtual-route pacing,* this mechanism limits the flow of data from an origin subarea node onto a virtual route to a degree that prevents the route from becoming congested. Message units prevented from entering the network accumulate in a queue at the entrance to the virtual route until the route is once again able to handle them.

As in the case of session-level pacing, described earlier, data is transmitted over a virtual route in groups of message units called pacing groups. The size of the virtual-route pacing group is initially set when the route is activated. Thereafter, fluctuations in the length of the queues associated with the transmission groups used by the route cause automatic adjustments, within predefined limits, of the pacing group size.

Virtual-route pacing affects the flow of data in all sessions concurrently assigned to each virtual route that joins two subareas. This contrasts with session-level pacing, which applies only to data flowing in individual LU-LU and SSCP-SSCP sessions.

Data Link Control Services

The job of data link control is threefold. First, it establishes logically full-duplex connections over each physical full-duplex or half-duplex link in the network so that sessions can use these links. Second, it transfers data over the links. And third, it detects errors occurring during the data transfer and corrects them by retransmission, if possible; if not, data link control reports them to upper levels of the SNA network so that corrective action can be taken.

Data link control has two forms. One is the channel data link control, as used for System/370 channels. The other is Synchronous Data Link Control (SDLC), which is used by links that employ serial-by-bit transmission.

Data link control is responsible for managing and performing error recovery actions for all the varieties of link configurations—such as loops, nonswitched point-to-point, switched point-to-point, and nonswitched multipoint—in such a way as not to require the involvement of upper layers of the SNA network. This isolation of data link control functions makes it possible to change the kinds of link connections between nodes without affecting end users.

The Layered Structure of SNA

The SNA network has been described earlier as a two-part structure—network addressable units and the path control network—with each part spanning all the nodes in the network. And earlier this chapter identified the several groups of related services within each part.

Because each group of services interacts in a specified way with adjacent groups, they are referred to, and depicted, as *layers* of the SNA network. Figure 2-22 names the layers within the nodes in the network.

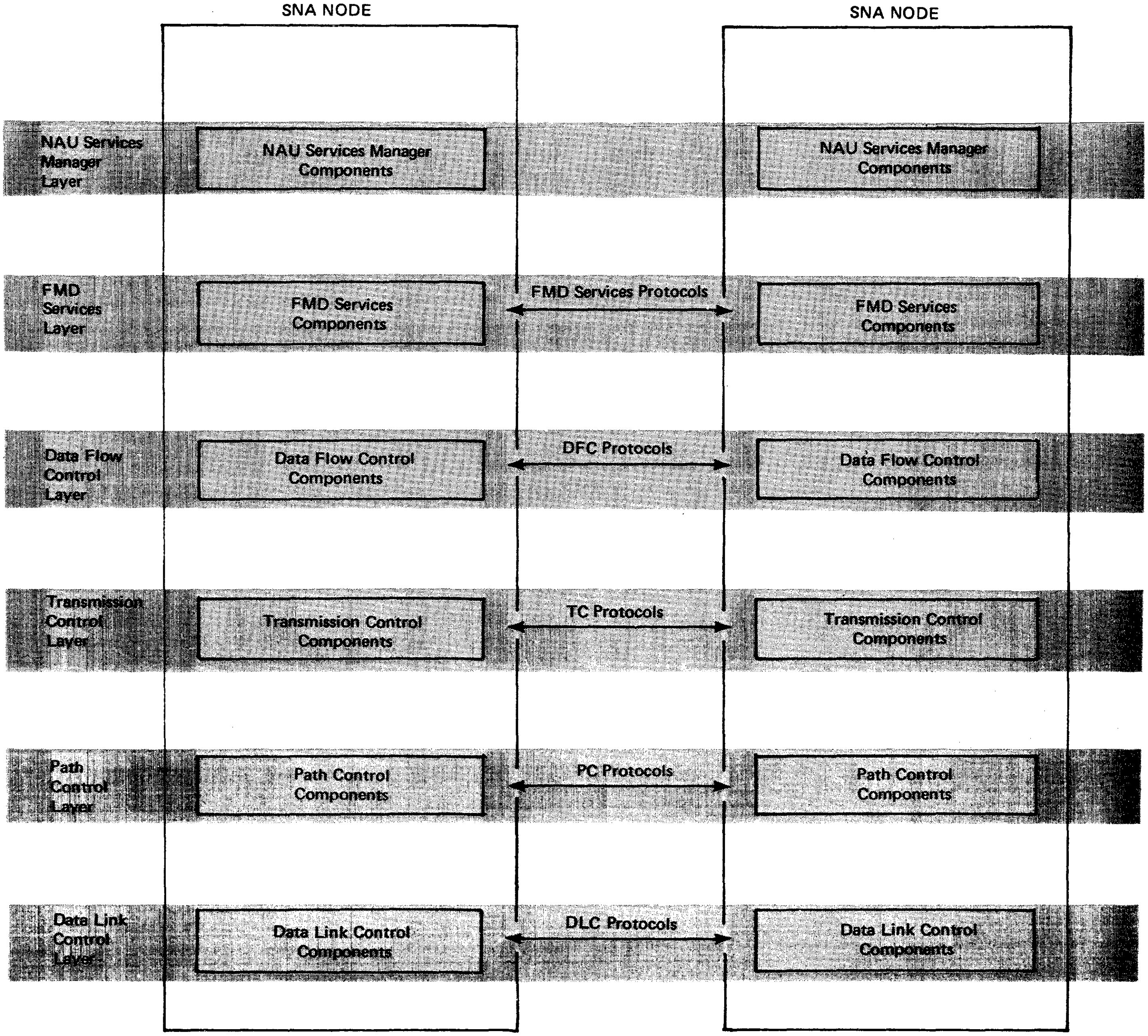


Figure 2-22. Communication within Layers via Peer Protocols

The names of the layers identify the services they perform, as described earlier under "NAU Services" and "Services of the Path Control Network." The top two layers—NAU services manager and FMD (function management data) services—were not identified earlier; together they provide the session presentation services, application-to-application services, and session network services described above under "NAU Services."

This layered structure is significant in two ways. First, equivalent functions in a given layer, but within different nodes, interact in a strictly defined way regardless of the kind of node or machine that contains the functions. For example, path control in any node interacts with path control in any other node according to precise rules. This consistency of interaction within each layer is what makes an SNA network a unified system that behaves predictably regardless of the particular combination of machines and programs of which it consists.

Second, the precisely defined boundary between layers largely isolates changes made in one layer from affecting functions performed in other layers. New functions can be added to a layer or existing ones enhanced. So long as the layer continues to interact with adjoining layers as it did before the changes, those layers are unaffected.

As an example, when new telecommunication facilities become available, changes to the data link control layer will likely be required to accommodate them. The changes would, however, be isolated from the upper layers. Network owners can therefore take advantage of the new telecommunication facilities without having to undergo the expense and effort of recoding their application programs.

The presence of a layer within the various nodes, and the prescribed interaction between the nodes, do not mean that the functions are distributed equally between the nodes. Some products, like those implementing subarea nodes, have the resources (such as storage) to allow a relatively greater functional capability within each layer. Other products, such as those implementing peripheral nodes, have fewer resources and can therefore have less capability.

SNA Layers and Functional Subsets

Each layer of an SNA network can contain a very large number of functions, not all of which are needed by, or are appropriate for, each SNA product. Therefore a limited number of subsets of these functions have been defined for each layer. These subsets are groups of functions that have been identified as generally useful for a variety of end users and products.

For example, session presentation services are available in several subsets that provide different kinds of device control and data formatting functions. Some are appropriate for keyboard/display terminals, others for keyboard/printer terminals. End users establishing a session between their logical units specify the subsets of functions that the session will use.

Classifying SNA Products By Types of LU-LU Session

SNA products are classified according to the type of LU-LU session in which they can participate. Each type designates a particular subset of SNA functions that that product can perform when its LU is in session with an LU in a another product. Some of the SNA functions in the subset are mandatory; others are optional. In general, products that participate in a particular type of LU-LU session are used for a specific kind of communication, such as that between an application program and a general-purpose terminal, or between an application program and a display terminal, or between two application programs. The LU-LU session type designation is a convenient means of classifying SNA machines and programs according to the subsets of SNA functions they can perform.

The products containing two logical units that the network designer intends to communicate must both be capable of participating in the same type of LU-LU session in order for that communication to occur.

Some SNA products can participate in more than one type of LU-LU session.

Advantages of LU-LU Session Type Classifications

By knowing the LU-LU session type or types of an SNA product, a network designer can determine a great deal about the kinds of application in which the product can be used and about the kinds of SNA functions that product can perform. This information is useful in selecting a set of SNA products to use in a particular application. It is also useful in problem determination, since by knowing which functions a particular logical unit can perform, those responsible for problem determination can focus upon problems associated with those functions.

Profiles and Usage Fields

SNA logical units tell their session partners what SNA functions they provide via the command that causes LU-LU sessions to be activated. Related sets of these functions are grouped together in what are called *profiles*. Specific profiles are associated with specific LU-LU session types and specify session options for presentation services and for data flow control and transmission control functions.

Profiles are assigned numbers so that they can be specified in the session activation command. Both session partners must use the same number.

In addition to profiles, the session activation command contains *usage fields* for further specifying session options for presentation services and for data flow control and transmission control functions. The LU-LU session type constrains the combination of profiles and usage values that the logical units may use during the session. A given logical unit can participate only in sessions that draw from its allowable profiles. A product is thus characterized by the session type (or types) that it supports.

SNA Protocols and Headers

In a network, as in other contexts, a protocol is an agreement between two components about how they will interact. An SNA protocol consists of a set of commands and responses that have defined formats, are exchanged in defined sequences, and result in specific actions.

Communication within an SNA Layer

Systems Network Architecture is specified as protocols through which communication occurs between pairs of equivalent ("peer") components located within each SNA layer and within a single SNA node or, more typically, between separate nodes. Sometimes referred to as peer protocols, these protocols coordinate network operations through each of the layers regardless of the types of nodes or how many of them the network contains.

The communication these protocols make possible is called peer-to-peer communication. Figure 2-22 represents peer-to-peer communication via peer protocols in a general way; not all SNA nodes, however, include all of the layers shown. Examples of equivalent components are those that do routing within the path control layer and chaining within the data flow control layer.

Peer protocols are meticulously formulated so that distributed data processing products, regardless of kind, can interact through an SNA network, provided that they contain one or more SNA nodes that conform to the protocols.

Peer protocols use parameters passed between equivalent components in the separate nodes. The parameters are placed in headers successively prefixed to the message units by the various layers of a node; the headers are successively deleted by the various layers of the destination node. (Some parameters are carried in the message unit itself.) The same action occurs in each intervening node, except that only the lower two layers (intermediate routing node) or lower three layers (boundary node) are involved (except for session activation and deactivation requests and responses). Figure 2-23 shows the use of headers to carry parameters from node to node.

In the discussion so far, the term *message unit* has been broadly used to represent any unit of data passing from an origin network addressable unit to a destination network addressable unit. In fact, various combinations of data and control information can flow between network addressable units and each combination has a different name. The elementary message unit passing between network addressable units is the *request/response unit,* abbreviated RU.

In general, request units (often referred to as requests) contain end user data, or control information, or both. Response units (responses) are usually acknowledgments to requests; a given request may or may not be acknowledged by a response. RU is the general term for the message unit that, accompanied by a request/response header, is passed between network addressable units. Figure 2-23 shows the headers being successively added (and then removed) by the lower three layers.

The parameters by which functions in the FMD services, data flow control, and transmission control layers communicate are placed either in the RU or in the *request header* that transmission control appends to the RU. (The FMD services and data flow control layers pass their parameters down to the transmission control layer for inclusion in the request header.)

Though (in the case of the request header) parameters for three layers may be packaged together, the effect within each layer is the same as if they were independently transmitted. With minor exceptions, the parameters exchanged within a layer are generated or examined only by functions within that layer; no other layer sees them and that layer does not see parameters exchanged within other layers. Thus the independence of one layer from another—a principal objective of SNA—is practically achieved.

Communication between SNA Layers

Besides specifying the interlayer parameters that must be exchanged between nodes, SNA specifies the functional relationships between adjacent layers within a node and the information that must be exchanged by those layers. However, because each node is contained in a single machine such as a processor, controller, or terminal, it is unnecessary for SNA to specify the format for the exchange of parameters. The formats are determined by the designer of the product; as long as the interlayer protocols conform to the rules specified by Systems Network Architecture, the unit will function as an SNA node.

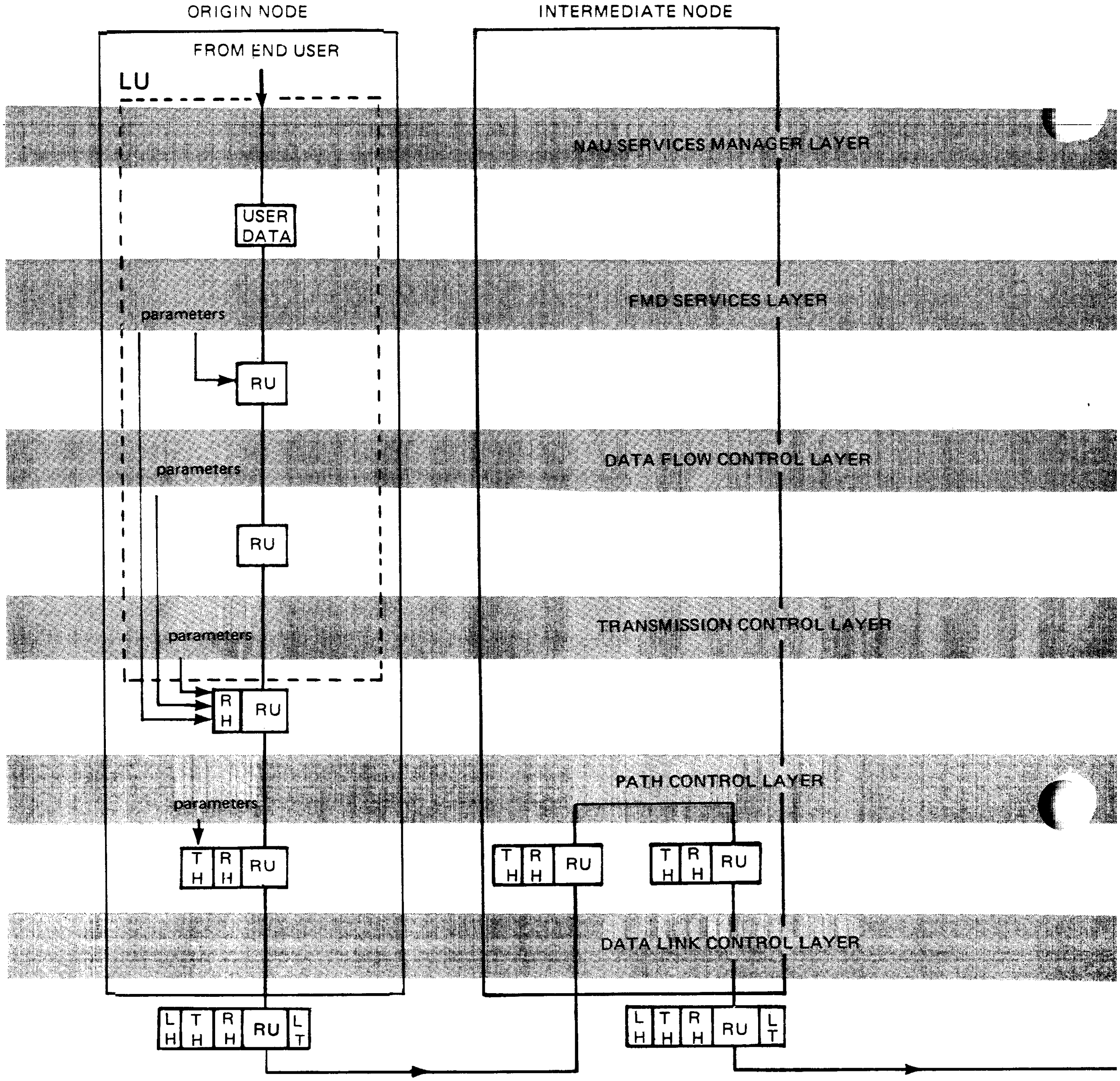
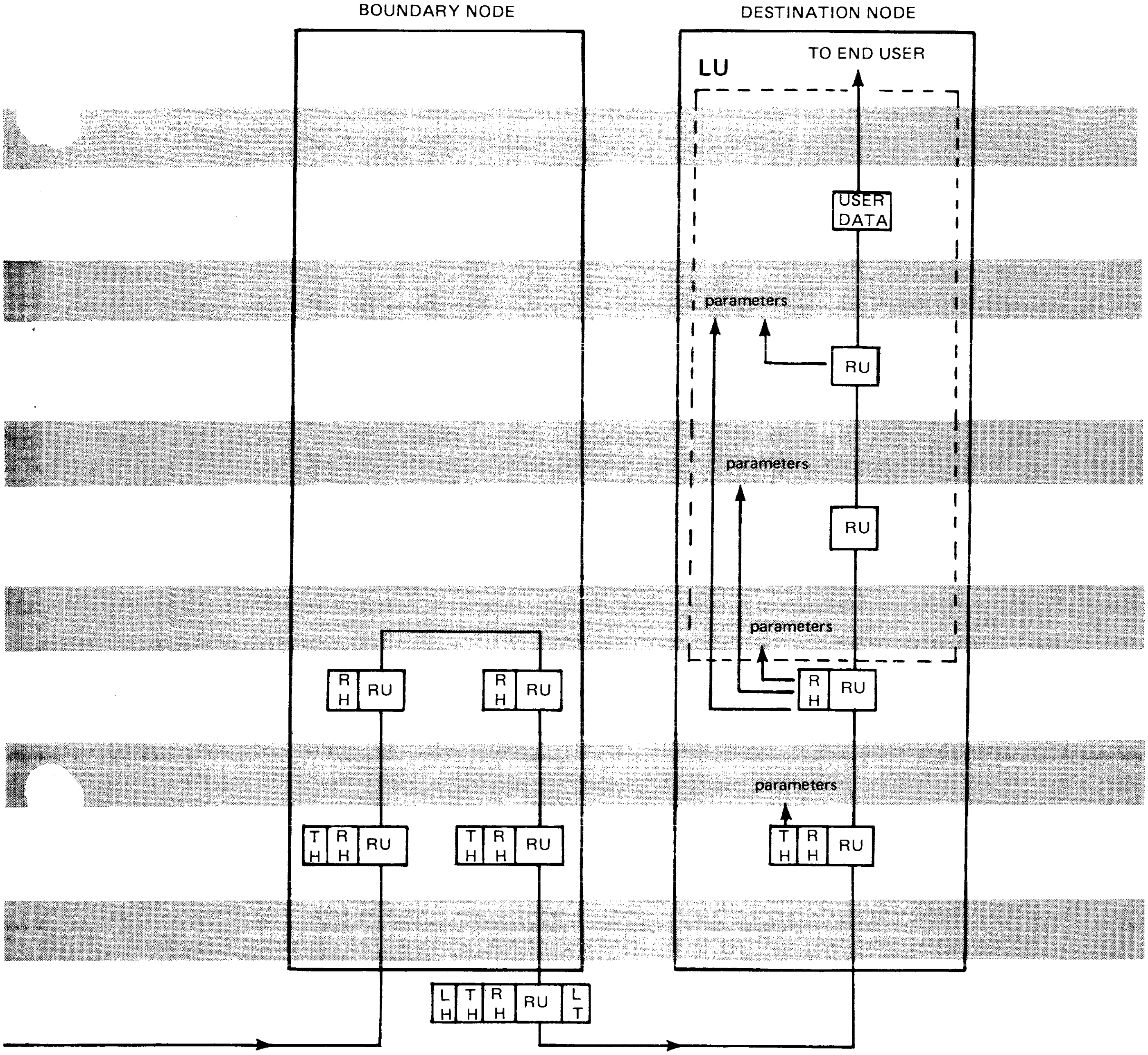


Figure 2-23. Communication within Layers via Message Headers



LH - Link Header
LT - Link Trailer
RH - Request/Response Header
RU - Request/Response Unit
TH - Transmission Header

This page intentionally left blank.

Chapter 3. Relationship of SNA Layers, Nodes, and IBM Products

This chapter explains the general relationship between the functional layers of Systems Network Architecture and the nodes and products in which the layers are implemented.

Chapter 2 identified three types of SNA nodes:

- A subarea node with an SSCP
- A subarea node without an SSCP
- A peripheral node

These architectural designations are independent of any particular product implementation. That is, Systems Network Architecture specifies the functions of each type of node but does not prescribe the kind of products, such as terminals and cluster controllers, in which each type of node is implemented. These architectural designations do correspond to the current IBM product categories as follows.

A subarea node with an SSCP is contained in a processor. Examples of processors that can contain this kind of node are System/370 processors, the 4300 processors, the 8100 processors, System/34 processors, and System/38 processors. (These processors may also contain other kinds of nodes.) In these products, subarea nodes with an SSCP are implemented within an SNA access method (such as ACF/TCAM, ACF/VTAM, or ACF/VTAME) or within the operating system or other control program that the processor executes. A processor that contains a subarea node with an SSCP is often called a *host node.*

A subarea node without an SSCP is typically contained in IBM 3705-II Communications Controllers. The node is implemented within the network control program (ACF/NCP/VS) that the 3705-II executes. A 3705-II controller that contains a subarea node without an SSCP is often called a *communication controller node.*

A peripheral node is contained in a variety of terminals (both programmable and nonprogrammable), cluster controllers, and systems. Examples are the IBM 5250 Information Display System, IBM 3270 Information Display System, IBM 5520 Administrative System, IBM 3767 Communications Terminal, terminals of the IBM 3770 Data Communication System, IBM 3600 Finance Communication System, IBM 6670 Information Distributor, IBM 3730 Distributed Office Communication System, and IBM System/32.

Some IBM products can implement more than one SNA node. For example, some SNA networks are configured with a System/370 processor that executes two SNA access methods (such as ACF/TCAM and ACF/VTAM), each of which contains a subarea node with an SSCP.

Each of the three types of SNA node has a specific combination of layers and types of network addressable unit, as follows.

Subarea Node with an SSCP

A subarea node with an SSCP, as shown in Figure 3-1, contains a physical unit (PU) and one or more logical units (LUs) in addition to the system services control point (SSCP). Each LU is a point of access for one or more application programs or application subsystems with which other logical units can communicate.

In general, application programs gain access to the network through an application subsystem such as CICS/VS or IMS/VS. In this case, the subsystem implements part of the LU or LUs for the application programs; the other part of each LU is implemented by the access method. If, in contrast, no application subsystem is used, the entire LU may be implemented by the access method, or part of it may be implemented by the application program. The boundary between the part of the LU inside the application subsystem or program and the part inside the access method depends on the product and is not prescribed by SNA.

Each LU can engage in concurrent sessions with one or more other LUs, in the same node or different nodes.

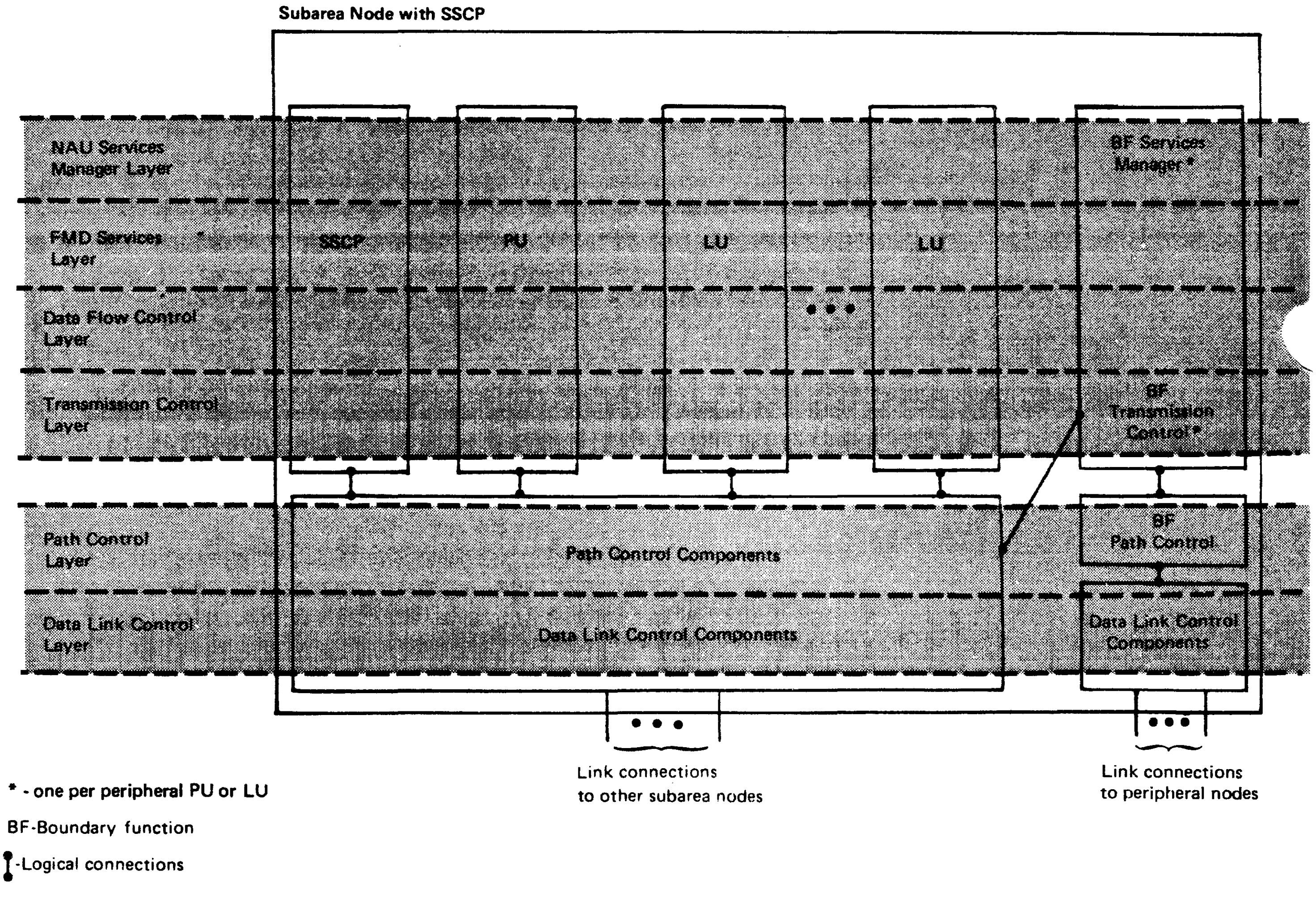


Figure 3-1. Network Addressable Units in a Subarea Node with an SSCP

A subarea node with an SSCP includes boundary function if any peripheral nodes are attached to the processor channel or communication adapter.

Figure 3-2 shows the typical relationship of an SNA node with an SSCP to an SNA access method and an application subsystem. The access method contains the SNA node's SSCP, its PU, and the part of each of its LUs that is concerned with SSCP-LU sessions. It also contains the path control and data link control components and boundary function components, as well as non-SNA functions outside the SNA node.

The application subsystem contains the parts of the LUs that are not within the access method (these are the parts concerned with LU-LU sessions) as well as non-SNA functions outside the SNA node. (In some applications, no application subsystem is used. In this case the application program itself contains part of the LU or LUs if the SNA protocols at this level are used.)

The term *SNA product node* refers to one or more IBM software products that implement an SNA node. In Figure 3-2, the SNA product node consists of an SNA access method (for example, ACF/VTAM or ACF/TCAM) and an application subsystem (for example, IMS/VS or CICS/VS). User-provided application functions lie outside the SNA product node.

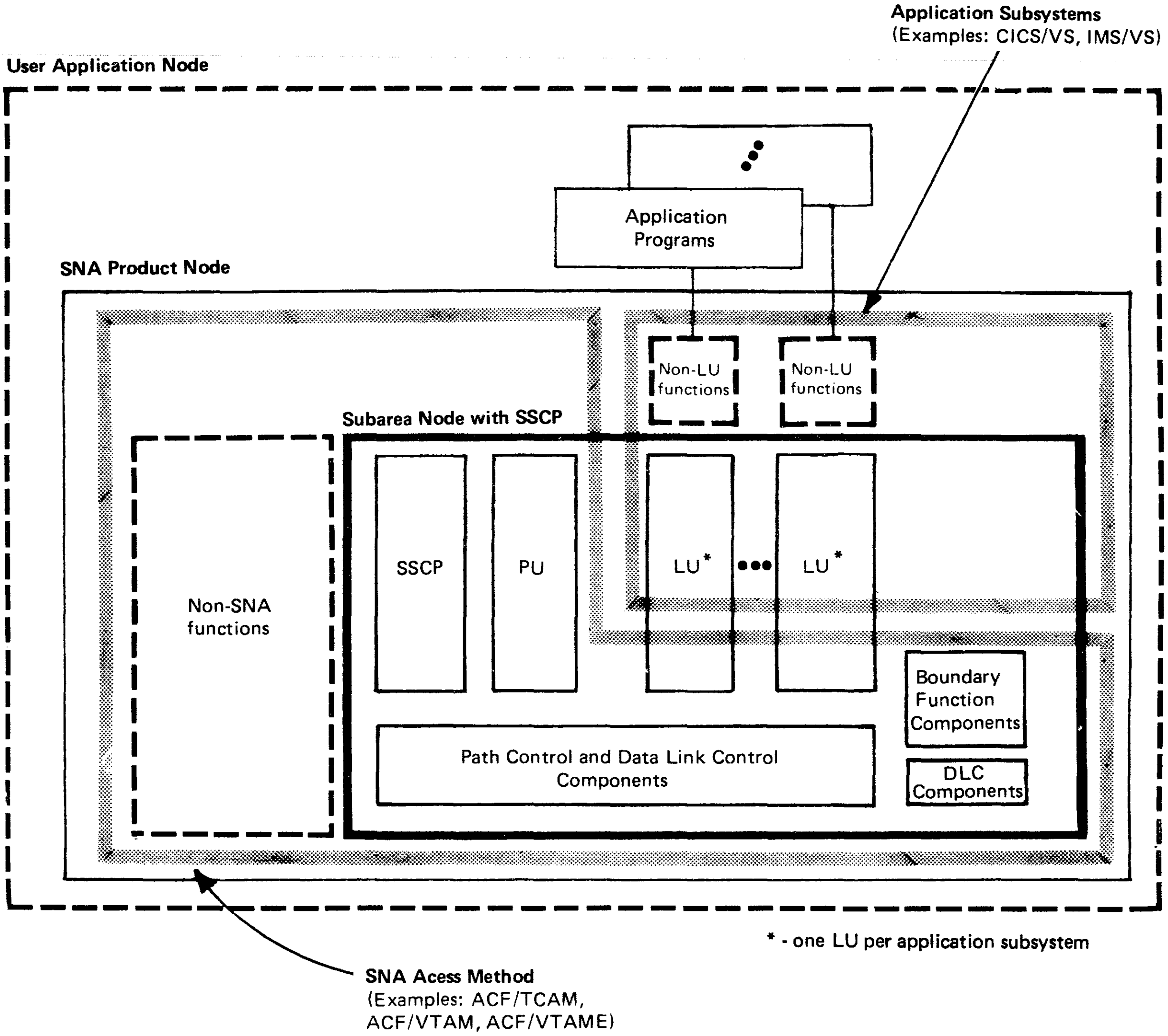


Figure 3-2. Relationship of Subarea Node with an SSCP to the SNA Product Node and User Application Node

The term *user application node* refers to the combination of user-provided application programs and one or more SNA product nodes. Figure 3-2 shows a user application node containing application programs and a single SNA product node. The SNA product node, in turn, contains SNA programs (access method and one or more application subsystems) that implement an SNA node. (Some processors could contain more than one access method and application subsystem, and more than one SNA node. In this case the SNA product node consists of all the SNA programs and the SNA nodes they implement.)

Subarea Node without an SSCP

A subarea node without an SSCP, as shown in Figure 3-3, contains a physical unit and a physical unit control point (PUCP). If this node is implemented in ACF/NCP/VS in a 3705-II communications controller, these are the only network addressable units in the node. If the controller also contains the Network Terminal Option (NTO) program product, the node also contains logical units. These LUs, through conversion routines in NTO, represent certain start-stop terminals to the SNA network as if they were peripheral nodes. (SNA product nodes also support transport of BSC and start-stop traffic through the network using a transmission header specific to this kind of traffic.)

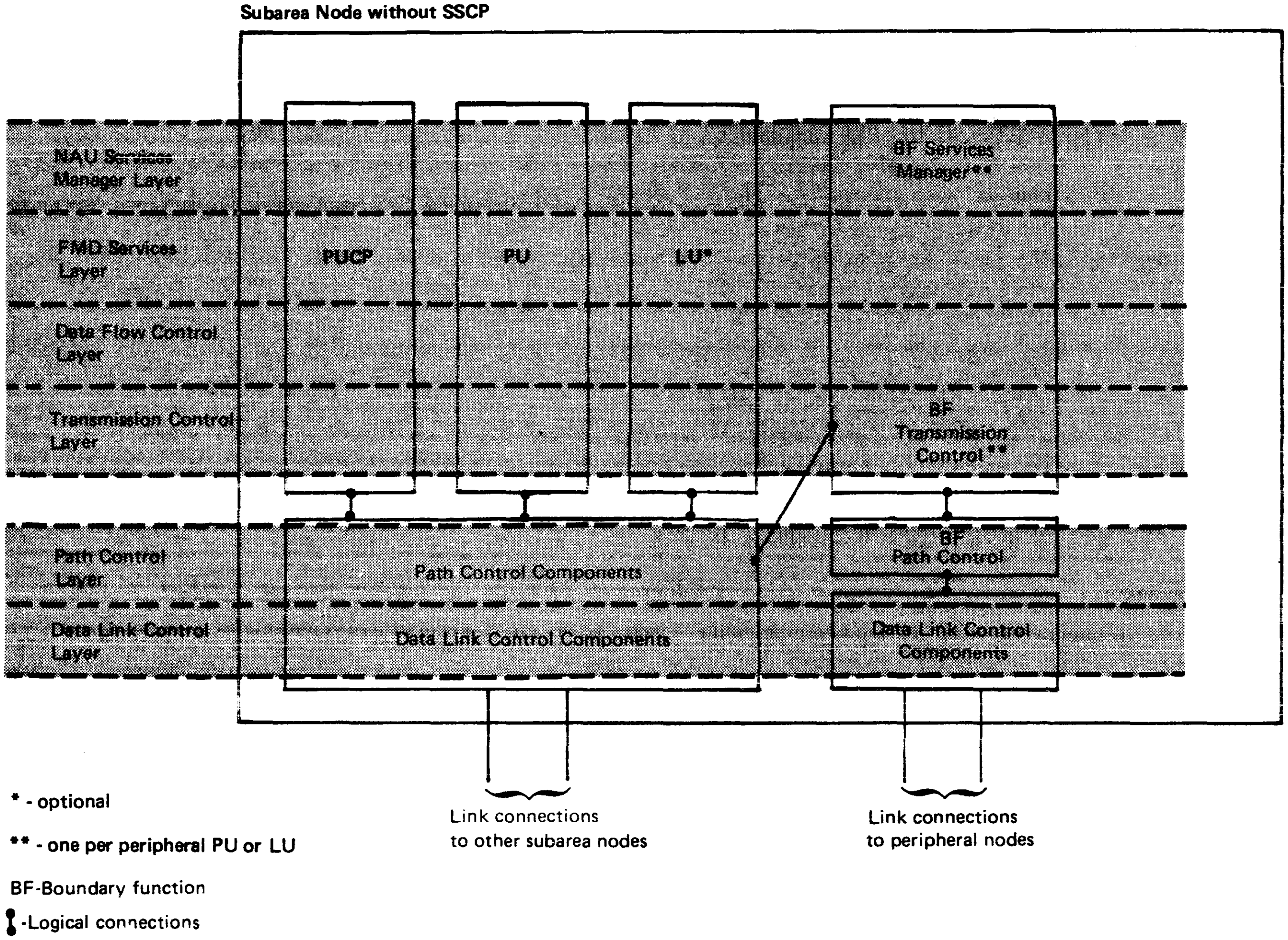


Figure 3-3. Network Addressable Units in a Subarea Node without an SSCP

A subarea node without an SSCP includes intermediate routing function (transmission control and path control) for any other subarea nodes without an SSCP that are attached to it by SDLC links.

A subarea node without an SSCP includes boundary function (transmission control and path control) for all directly attached peripheral nodes.

Figure 3-4 shows the relationship of an SNA node without an SSCP to an SNA network control program (ACF/NCP/VS) in a 3705-II communication controller. The network control program contains a PU, a PUCP, path control and data link control components, and boundary function components. It may also contain one or more LUs (if the Network Terminal Option is present) and non-SNA components such as control functions for non-SNA terminals. In Figure 3-4, the SNA product node consists of ACF/NCP/VS. Because, typically, no user-written programs are executed within the communication controller, the user-application node coincides with the SNA product node.

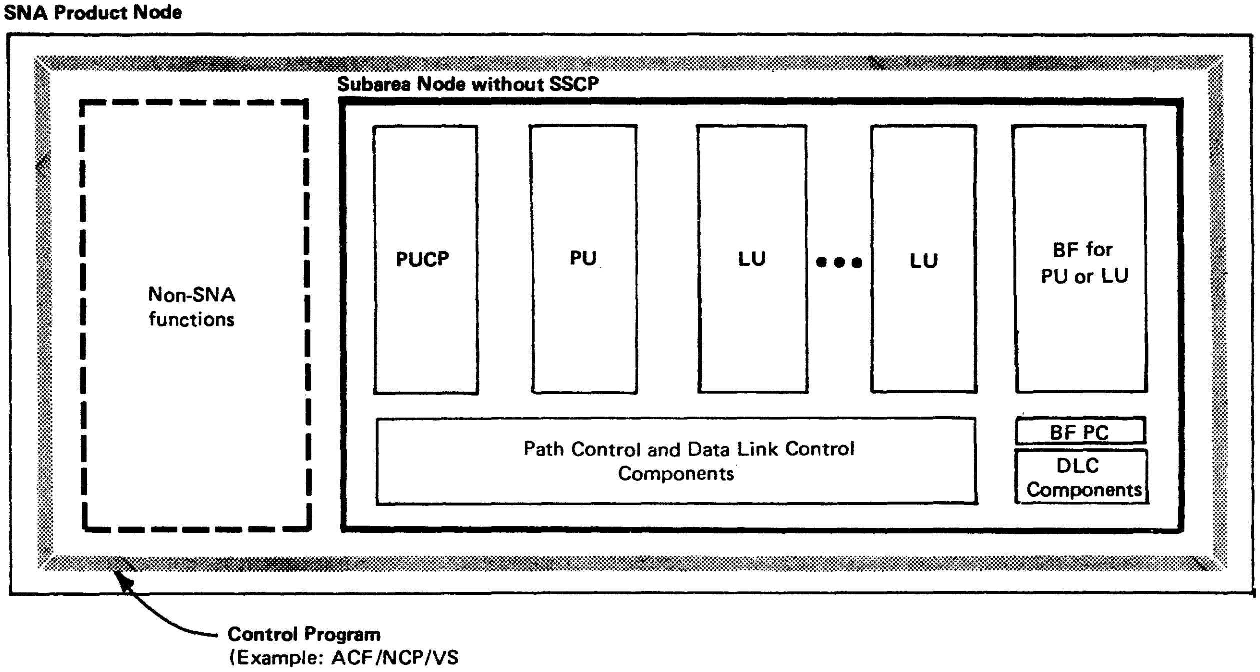


Figure 3-4. Relationship of Subarea Node without an SSCP to the SNA Product Node

Peripheral Node

A peripheral node, as shown in Figure 3-5, contains a PU, a PUCP, and one or more LUs.

Some peripheral nodes are limited to a single LU; others may have as many as 255 LUs. Each LU is a point of access for one or more application programs or application subsystems, or for a directly attached device with which other logical units can communicate. An application program or subsystem may be partly inside the node, containing part of an LU, and partly outside. The boundary between the part of the LU inside the application program or subsystem and the part in the control program depends on the product and is not prescribed by SNA.

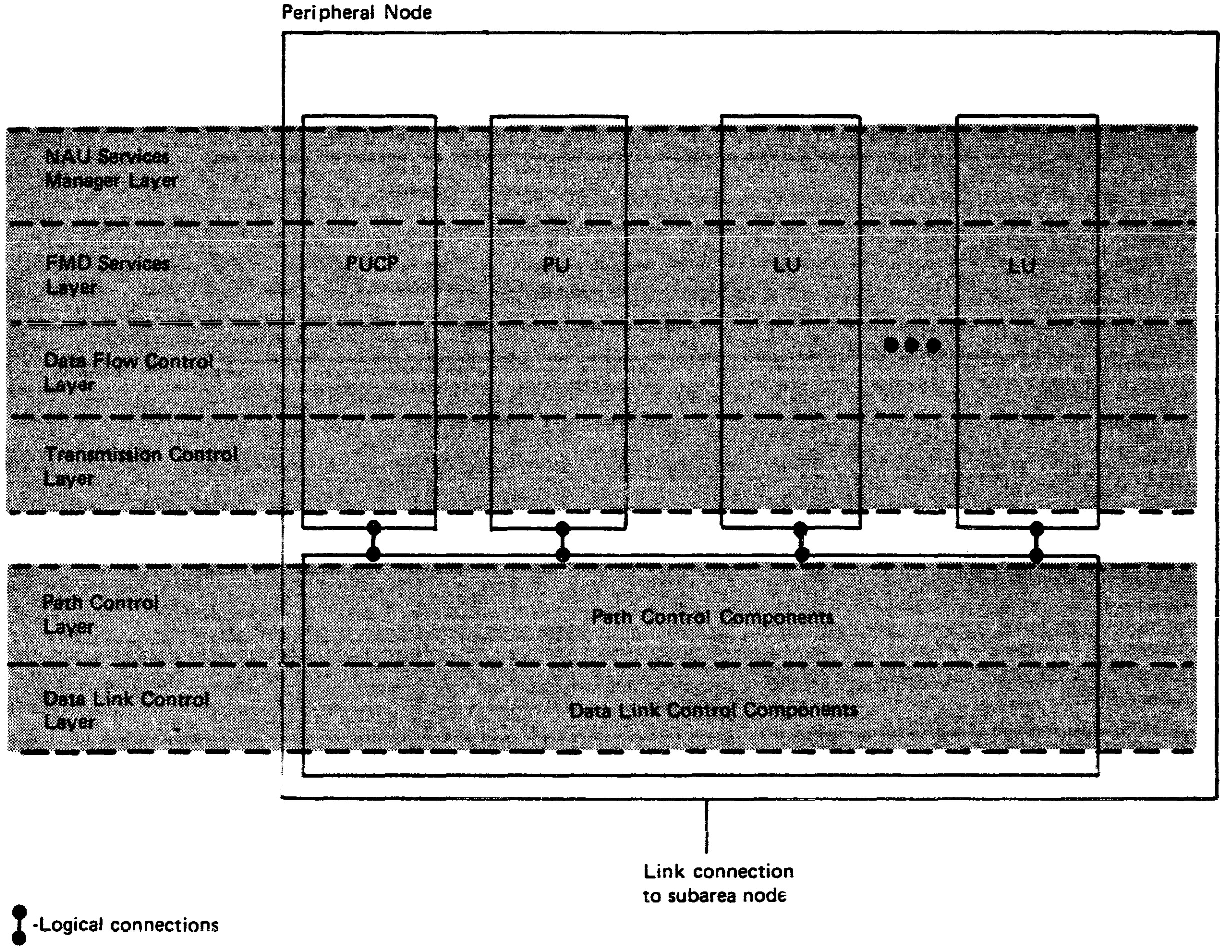


Figure 3-5. Network Addressable Units in a Peripheral Node

Figure 3-6 shows the typical relationship of a peripheral node to a control program and an application subsystem. The control program, whether implemented in hardware or software, or a combination, contains the SNA node's PU, its PUCP, and part of its LUs. The LUs may represent application programs or directly attached devices, or both. The control program also contains the path control and data link control components and may contain non-SNA functions. In Figure 3-6 the SNA product node consists of the control program and an application subsystem, and the user application node consists of the SNA product node and the user-written application programs.

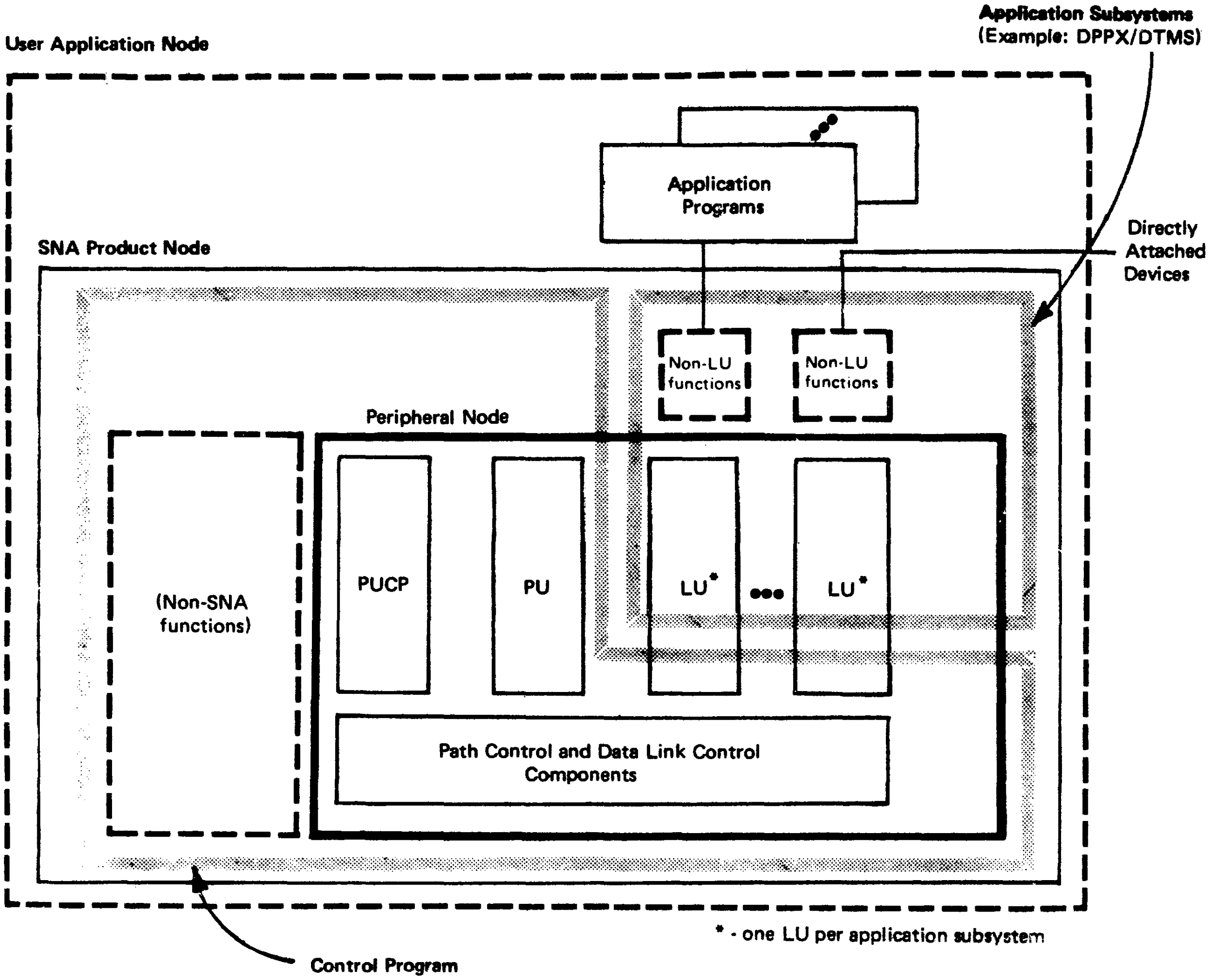


Figure 3-6. Relationship of Peripheral Node to the SNA Product Node and User Application Node

Chapter 4. Using IBM Subsystems to Perform Distributed Data Processing

Many enterprises need to distribute processing power, application programs, and files among several processors interconnected into a network. This chapter describes how SNA provides for distributed data processing via IBM application subsystems.

Overview of Distributed Data Processing in an SNA Network

This overview first defines basic terms associated with distributed data processing, and then relates distributed data processing to centralized and decentralized data processing. The overview next describes various levels of distributed data processing, and ends with descriptions of two major types of distributed data processing implemented by IBM subsystems.

Some Distributed Data Processing Definitions

An *application* consists of a defined set of functions provided for a defined group of users by one or more related application programs and data resources (such as files or data bases). An accounting application and a personnel application are examples of applications. The application programs may be located on a single processor, or may be distributed among two or more interconnected processors. If the programs are distributed among processors, the application is a *distributed application,* and the programs in the interconnected processors are performing *distributed data processing.*

Distributed data processing, then, is data processing in which application programs distributed among interconnected processors cooperate to perform distributed applications for the users of a network.

The processors over which an application is distributed may vary in power and complexity. For example, one may be a System/370 computer, while another may be a System/34 computer, and yet another may be a terminal containing a microprocessor.

Centralized, Decentralized and Distributed Data Processing

Distributed data processing allows an enterprise to combine the benefits of centralized and decentralized data processing systems.

In a centralized system, a single processor and associated resources fill the data processing needs of the enterprise. A centralized system tends to be responsive to needs of the enterprise by allowing the enterprise to maintain a high level of control over application development and processor operation. However, a centralized system may become unresponsive to needs of user departments, particularly as demand on the system grows.

In a decentralized system, multiple processors uniquely meet the data processing needs of individual user departments. A decentralized system tends to be responsive to these needs, because each user department may exercise direct control over application development and processor operation for its processor. However, a decentralized system may be unresponsive to enterprise-wide information needs due to application-program and data-base incompatibilities, and inability to communicate among processors.

Distributed data processing permits networks to be developed that are responsive to the requirements of both the enterprise and individual users. In distributed data processing networks, individual applications may be designed with a greater or a lesser degree of centralization, depending on the needs of the enterprise.

Levels of Distributed Data Processing in an SNA Network

The advent of distributed data processing was a fundamental reason for the creation of SNA, and most SNA networks perform some level of distributed data processing as it is defined above. This section describes the levels of distributed data processing which may be implemented in an SNA network.

In a distributed application, one or more of the following types of resources are distributed over multiple, interconnected processors:

- data resources
- application programs
- processing power

Data resources include files, data bases, and queues. Application programs may be user supplied, or they may be program products or field-developed programs supplied by IBM.

Figure 4-1 shows a network in which none of these resources is distributed. Application programs and data resources (files) are located at a single host processor, and are accessed by users at terminals attached to the processor via a communication controller. Though these terminals provide remote access to the processor, they do not do local user-oriented processing; their only function is to serve as input/output ports for users of the system.

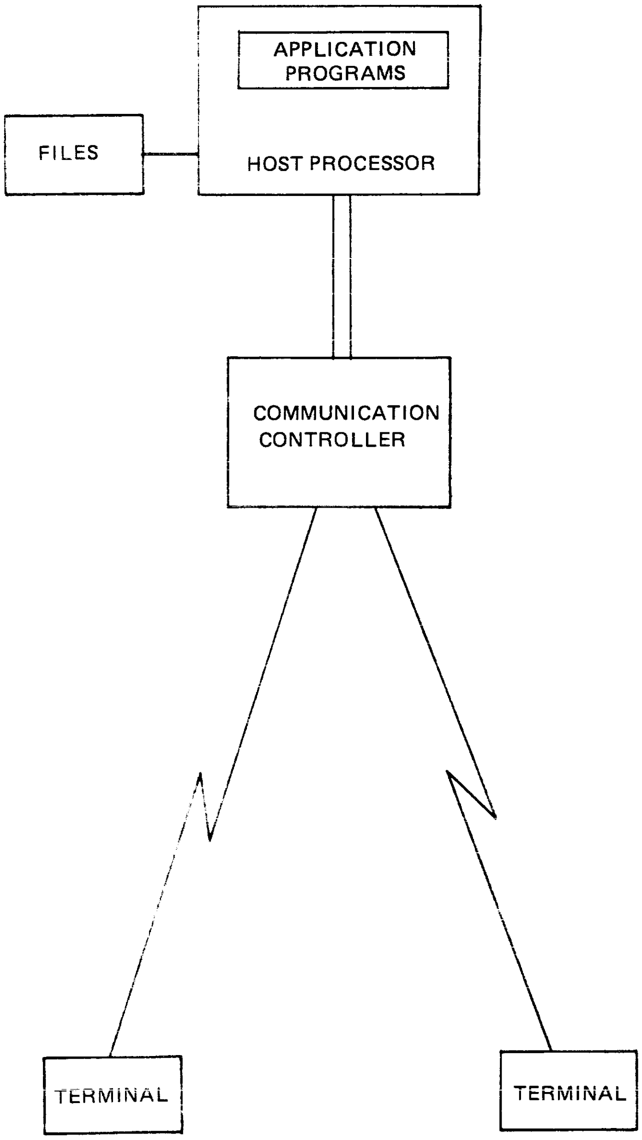


Figure 4-1. Computer System with No Distribution of Resources

Figure 4-2 shows a network consisting of a host processor and two distributed processors connected through a communication controller. The distributed processors can execute application programs but are subordinate to the host processor. The host processor might be an IBM System/370 computer, while the distributed processors might be smaller computers (such as IBM 4300 processors) or cluster controllers (such as the IBM 8100 Information System with DPCX, and the IBM 3600 Finance Communication System). The network illustrated in Figure 4-2 has an elementary level of distributed data processing in that files, application programs and processing power have been distributed among the controlling host processor and the subordinate distributed processors. From its beginning, SNA has supported this level of distributed processing.

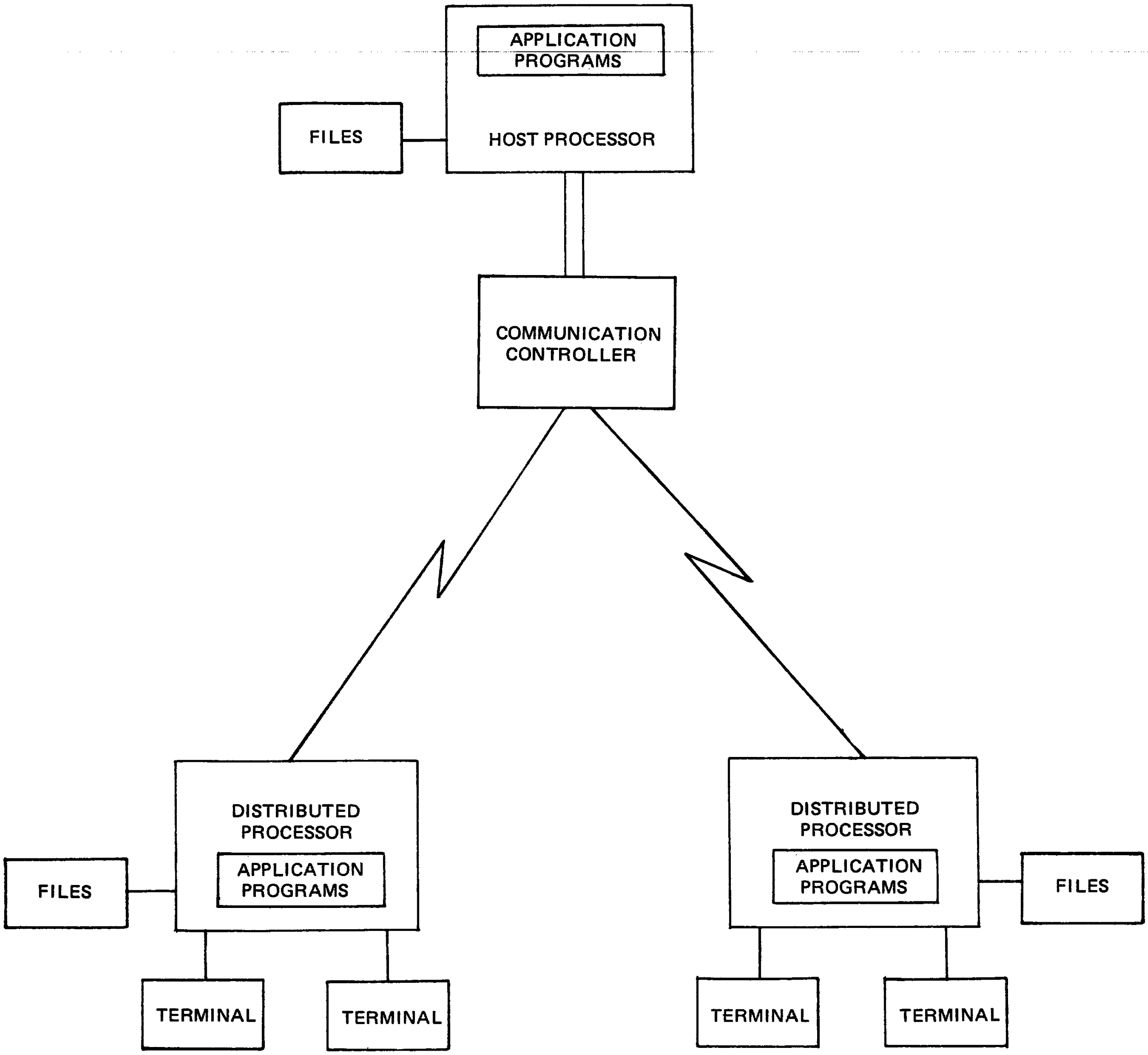


Figure 4-2. Distribution of Resources between a Host Processor and Attached Distributed Processors

Figure 4-3 shows a somewhat more complex network with a higher level of distributed data processing. In this figure, two groups of processors of the type shown in Figure 4-2 are interconnected by one or more transmission groups between their communication controllers. As a result, the resources of each group of processors are available to users at terminals located in the other group of processors.

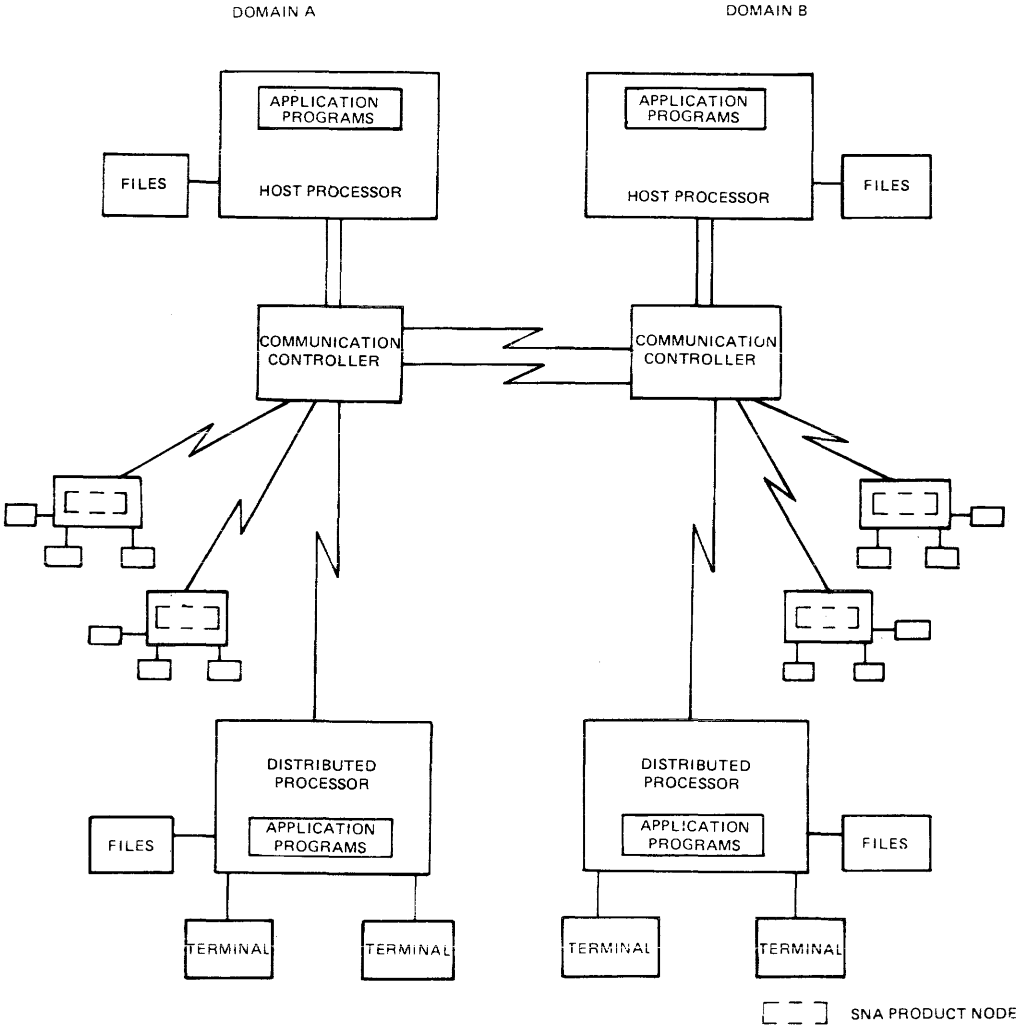


Figure 4-3. Distribution of Resources between Two Groups of Processors

Figure 4-4 superimposes the SNA nodal structure on the configuration illustrated in Figure 4-3. Each of the processors contains an SNA product node plus application programs. Each product node contains an LU, and two of the product nodes contain SSCPs. Each SSCP controls the LUs in its domain and controls the initiation of sessions with its LUs. In this way, the SSCPs indirectly control user access to files, application programs, and processing power in their domains.

In Figure 4-4, a terminal user in Domain A can access applications in Domain B either by causing LU C to go directly into session with LU B (using the path between LU C and LU B), or by causing LU C to go into session with LU A and then relying on LU A to initiate a session and transfer data between LU A and LU B. In either case, the cross-domain session is set up by means of negotiations between SSCPA and SSCPB.

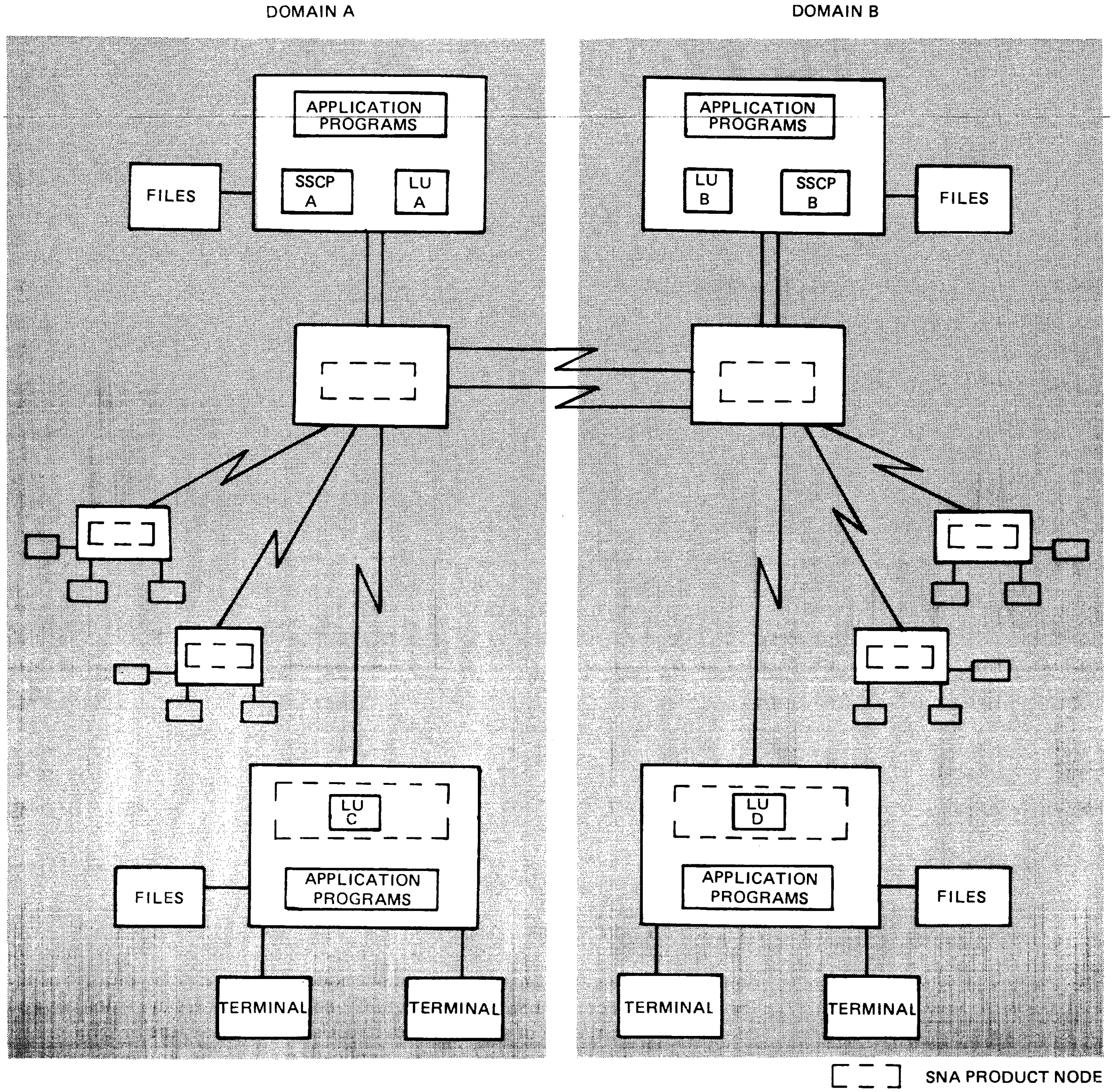


Figure 4-4. Clusters of Processors Showing SNA Product Nodes

SNA products provide the level of distributed processing illustrated in Figures 4-3 and 4-4 by means of a subset of SNA formats and protocols known collectively as Advanced Communication Function (ACF). ACF, which is implemented in ACF/VTAM, ACF/TCAM, and ACF/NCP/VS, allows terminal operators to access application programs and program products located in other domains.

ACF allows SNA products to activate and manage sessions between LUs located in different domains, but does not itself allocate the distributed resources (such as files and application programs) that are made available via LU-LU sessions. Such allocation can be managed either by IBM-developed subsystems or by application programs.

In Figure 4-5, two subsystems are added to the network shown in Figure 4-4. In this configuration, each subsystem controls access to distributed resources located at the SNA product node in which the subsystem resides. Terminal operators communicate with their local subsystem via an LU-LU session; that subsystem communicates with other subsystems to give the terminal operator access to resources controlled by the other subsystems.

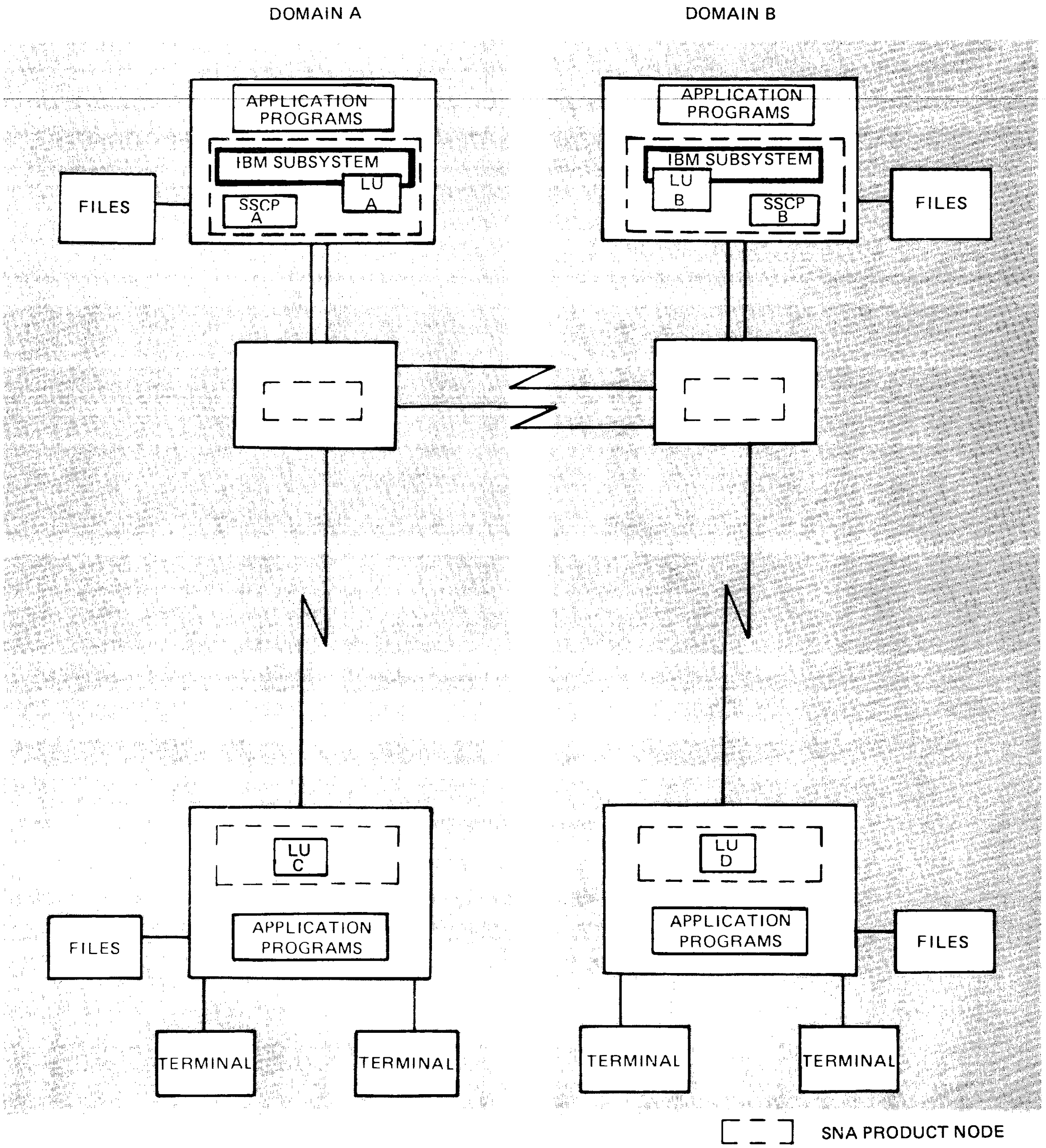


Figure 4-5. Subsystem-to-Subsystem Communication

Just as the SSCP located in an SNA access method manages the nodes and links in a domain, these IBM subsystems manage the application-oriented resources for the SNA product nodes in which they are located. By controlling access to files, application programs, and processing power at the SNA product node in which they are implemented, these subsystems facilitate the design of distributed applications and provide a higher level of distributed data processing support than ACF alone.

Chapter 2 describes the SNA capabilities that permit implementation of the distributed data processing levels illustrated in Figures 4-2 through 4-4. The remainder of this chapter discusses the use of subsystems to provide distributed data processing in an SNA network.

Types of Distributed Data Processing Involving Interconnected Subsystems

In an SNA network, interconnected subsystems may be used for two types of distributed application. These application types are *job networking* and *distributed transaction processing.*

In a job networking application, batch jobs submitted to one subsystem may be sent to another subsystem for execution; results are returned to the originating subsystem. This might be done to balance processor utilization in the network, or to access a program or file that is located at another processor. In a job networking application the unit of work is the *batch job,* a collection of one or more application programs and files that are submitted and executed together.

In a distributed transaction processing application, transactions entered by terminal operators are processed by multiple application programs under the control of multiple cooperating subsystems called transaction processing systems.

A *transaction processing system* (TPS) is a subsystem that supervises the sharing of resources for processing multiple transactions concurrently. Transaction processing systems are designed to support interactive applications in which requests (called transactions) submitted by people at terminals are processed as soon as they are received, with results being returned to the user in a relatively short time. Whereas in job networking the entity being processed is the batch job, in distributed transaction processing the entity being processed is the *transaction,* a request for service that is processed by one or more application programs in order to accomplish a particular result. A batch job is generally longer in duration than a transaction, and generally requires more processor resources than a transaction.

When transaction processing systems are linked by an SNA network, they can share their resources with one another, thereby facilitating the implementation of distributed transaction processing applications.

A subset of SNA formats and protocols known collectively as *intersystem communication* governs the interactions of transaction processing systems. By defining a common way for application programs to communicate even when they are under the control of different TPSs, intersystem communication allows you to construct a distributed transaction processing application using individual application programs as building blocks.

Job Networking

Job networking is a facility for transmitting batch jobs from one host processor to another. The jobs being transmitted may include the following components:

- job-control language (JCL)
- system input (SYSIN) data sets
- system output (SYSOUT) data sets
- job-oriented operator commands

In a simple form of job networking, two host processors connected by a data link could constitute a "job network." A job could enter the network through any local card reader, remote job entry station, or internal interface at either host processor. Once entered, the job could either be executed locally or be transmitted to the other host processor for execution. Upon completion of execution, the output could be printed or punched on local devices or remote job entry stations attached to the host processor that executed the job, or it could be transmitted to the other host processor for output processing on its local or remote output devices.

In a more general job network consisting of many host processors and links, jobs could be submitted anywhere in the network and routed to any processor for execution. Likewise, the output could be routed to any output device or group of output devices in the network for output processing.

Job networking is available in an SNA network through the Network Job Entry Facility for Job Entry Subsystem 2 (NJE for JES2), an IBM program product that runs under MVS. In addition, the Job Entry Program (JEP) and File Transfer Program (FTP) are program products providing transmission of batch jobs and files between distributed DOS/VSE systems and a host system.

NJE for JES2 provides somewhat more function than does JEP/FTP. For more information on NJE for JES2, see the publication *Network Job Entry Facility for JES2 General Information,* GC23-0010. For more information on JEP/FTP, see the publication *Job Entry Program and File Transfer Program General Information,* GH12-5129.

Functions and Advantages of Job Networking

A job network allows an enterprise to perform the following functions:

- move batch jobs
- move job output
- facilitate hardware or software migration

Moving Jobs in a Job Network

A job can be entered into the network at any location and be transmitted to the location having the data sets that are required to successfully run the job. Perhaps the job requires the processing power or the storage of a high-speed, large-capacity computer located at another site rather than the smaller computer that may be available locally.

Or a job may be transmitted to a location that has a special processor hardware feature (such as an emulator) that is not available on the local system.

In some cases, specific applications are supported only at certain sites within an enterprise. Job networking permits jobs to be routed from other sites in the enterprise to the system where an application is supported. In other cases, you might be able to justify more easily the cost of a program product by collecting jobs that use the program from all over the job network.

Moving Job Output

The most obvious reason for transmitting a job's system output data sets is to get them to their appropriate locations. For example, a report program may be run, and copies of its output may be distributed automatically via the network.

System output data sets may be sent to locations having special output equipment. An example is a printer that can print reports using multiple character sets on the same page.

Assisting Migration

The ability to connect host processors together can help an installation to convert from one hardware or software system to another. As new products become available, an enterprise has the option of installing the new products at specific locations in the job network. Selected jobs can then be routed to the new product to accomplish an orderly transition to that product. The migration to be accomplished may be as minor as the transition to a new level of an existing application, or it may be as major as the conversion to a new type of operating system.

SNA Facilities Used in Job Networking

The SNA job networking facilities use ACF/VTAM or (in the case of JEP executing in an IBM 4331 processor) ACF/VTAME for access to the network. JEP has certain restrictions with respect to configurations supported, but NJE for JES2 is able to participate fully in a multiple-domain network. With NJE for JES2, host processors in domains that are not physically adjacent can transmit jobs and output to one another via intermediate NCPs; intermediate host nodes do not need to store and forward the information. NJE users are able to fully utilize the functions provided by ACF, including routing across domains, multiple routes between subareas, and the use of transmission groups to provide parallel links between adjacent subareas.

Distributed Transaction Processing

SNA intersystem communication functions permit you to develop transaction processing applications in which different parts of the application are performed at different transaction processing systems (TPSs). The TPSs are interconnected via an SNA network.

Intersystem communication functions have been implemented in CICS/VS and IMS/VS. Although such support is not yet available, it is IBM's direction to support intersystem communication with the ability to initiate transactions between DPPX based systems and CICS/VS and IMS/VS in an SNA network.

Though CICS/VS and IMS/VS provide different subsets of intersystem communication function, both provide a common core of function that allows both to be used together in a distributed transaction processing network.

TPSs implementing SNA intersystem communication functions perform two major services:

- They allow you to distribute resources (such as files and queues) throughout the network, and provide a means for easily accessing such resources
- They allow you to distribute application programs for transaction-processing (hereafter called transaction processing programs) throughout the network, and provide a means for these transaction processing programs to communicate and cooperate with each other in processing transactions.

The first service is called the *remote resource access capability.* The second is called the *TPP conversational capability.*

The remote resource access capability allows a transaction processing program at one TPS to access data resources (data bases, files, or queues) located at another TPS. In addition, this capability provides access to remote transaction processing programs that are to be executed asynchronously, allowing a transaction processing program at one TPS to schedule the execution of a transaction processing program located at another TPS. Remote program scheduling is considered to be a remote resource access capability rather than a TPP conversational capability because the scheduled transaction processing program is not involved in a conversation with the scheduling transaction processing program.

In remote resource access, a user-supplied transaction processing program at one TPS communicates with IBM-developed code that is part of another TPS. When the transaction processing program makes access requests for remote resources, its TPS reformats such requests and routes each to the appropriate remote TPS. IBM-developed code in the receiving TPS performs the requested function and returns to the originating TPS data and status information associated with the request.

An example of remote resource access occurs when a transaction processing program at one TPS updates a data base controlled by a different TPS. A TPS with the remote resource access capability may make such remote updating transparent to the updating transaction processing program. If the system programmer previously defined the location of the remote data base to the local TPS, the executing transaction processing program need not know whether the data base is local or remote.

To facilitate common types of remote resource, SNA defines *model programs* that are implemented by TPSs. Model programs provide access to remote DL/1 data bases, remote queues, and remote transaction processing programs that are to be asynchronously executed. Model programs are discussed in more detail below.

TPSs support subsets of the remote resource access capabilities described here. For information on the remote resource access capabilities supported by a particular TPS, see the product documentation for that TPS.

Whereas the IBM subsystem supplies the code that performs remote resource access, the application programmer supplies the programs that use the TPP conversational capability.

For conversations between programs, the TPS supplies an interface allowing the application programmer to invoke conversational services in the TPS. When a transaction processing program invokes this interface to request a conversation with another transaction processing program located at a remote TPS, the local TPS assigns the local transaction processing program to an SNA session between the local and the remote TPS, and sends a request to the remote TPS asking it to associate the appropriate transaction processing program with the same SNA session. Once both transaction processing programs have been associated with the SNA session, they may send each other data using the conversational facilities provided by their TPSs. When they are finished conversing, one of the transaction processing programs signals its TPS to end the conversation. This TPS notifies the other TPS that the conversation has ended, and each TPS disassociates its transaction processing program from the SNA session, thereby making the SNA session available for use by other transaction processing programs.

Advantages of Using SNA Intersystem Communication

The major advantage of SNA intersystem communication in a distributed transaction processing environment is that it makes it easier for a network designer to distribute and redistribute applications within a network. Using TPSs that implement SNA intersystem communication functions, the network designer can locate resources such as transaction processing programs and files at the TPS where they fit best. These resources are also available to transaction processing programs running at other TPSs in the network. Transaction processing can thus be divided according to the enterprise's need among multiple cooperating TPSs located at different user application nodes.

SNA intersystem communication also provides different transaction processing programs with a uniform method for communicating with each other. Using intersystem communication, transaction processing programs associated with CICS/VS and IMS/VS can communicate with each other in the same network.

SNA intersystem communication defines protocols for linking synchronization points between communicating transaction processing programs so that changes made by a transaction processing program associated with one TPS are synchronized with those made by another transaction processing program associated with another TPS. Use of these protocols makes in possible, for example, for a transaction processing program to synchronize updates to a remote file with those made to a local file.

Intersystem Communication Functions for Distributed Transaction Processing

To provide distributed resource access and conversational capability to transaction processing programs, TPSs implement session-level functions and model programs defined by SNA.

At the session level, SNA specifies the rules to be observed by TPSs and associated transaction processing programs in communicating with each other. LU-LU sessions between TPSs that observe these rules are called Type 6 LU-LU sessions.

Model programs provide common functions that are defined by SNA and implemented by TPSs in order to assist transaction processing programs in performing distributed applications in a consistent way.

Session-Level Functions

Session-level functions consist of session-management functions and conversation-management functions. Session-management functions provide support for the initiation, maintenance, and termination of LU-LU sessions between TPSs. Conversation-management functions allow transaction processing programs associated with different TPSs to communicate over an LU-LU session.

Among the session-management functions are the capability to have multiple LU-LU sessions (called parallel sessions) between the same two TPSs, and the use of a negotiable Bind Session command to support peer-level communication among TPSs.

Among the conversation-management functions are the use of a special message unit header to name the program with which a requesting transaction processing program wants a conversation, and the use of the half-duplex flip-flop data flow control protocol for conversations between transaction processing programs.

One important conversation management function makes it easier to synchronize changes made by cooperating transaction processing programs. Many distributed transaction processing applications require that two transaction processing programs communicating across a type 6 LU-LU session coordinate the changes they make to their resources. This is particularly true when the cooperating programs update data bases.

SNA provides a protocol for coordinating events between two cooperating transaction processing programs. Under this protocol, the programs establish periodic synchronization points. When a synchronization point is reached, both TPSs are notified that all work since the last synchronization point has been successfully completed. If conversation between the two transaction processing programs is disrupted before a synchronization point is reached, both TPSs cancel data base changes made since the previous synchronization point, so that files used by both transaction processing programs will be at the same level.

Intersystem communication also defines protocols that TPSs can use to resynchronize after a disruption has occurred.

Model Programs

Model programs are defined by SNA and may be implemented by TPSs in order to perform distributed transaction processing in a consistent manner. Three model programs facilitate remote resource access by transaction processing programs, while a fourth handles the routing of system messages received from another TPS. The model programs discussed in this section have been implemented by either CICS/VS or IMS/VS, or both.

Two model programs are provided for remote data resource access. These are the DL/1 model program and the queue model program. A third model program, the scheduler model program, permits asynchronous execution of a remote transaction processing program.

The DL/1 model program gives a TPS and hence its transaction processing program access to a DL/1 data base located at a different TPS.

When a transaction processing program wishes to access a remote data base, its TPS associates it with an LU-LU session between the local TPS and the remote TPS controlling the data base. The local TPS then sends to the remote TPS a command that causes the remote TPS to attach the DL/1 model program to the LU-LU session. The local TPS then sends the transaction processing program's access commands to the DL/1 model program. The DL/1 model program executes these commands, accesses the DL/1 data base as directed, and returns appropriate data to the local TPS. The local TPS then passes the data to the transaction processing program.

The *queue model program* gives a TPS and hence its transaction processing program access to a queue located at another TPS.

The purpose of the queue model program is to allow a transaction processing program to retrieve, replace, and add queue records to a queue controlled by a remote TPS. The manner in which access to the queue is achieved is similar to that described above for the DL/1 model program.

The *scheduler model program* allows a transaction processing program located at one TPS to schedule the execution of a transaction processing program at another TPS. The scheduled program is executed asynchronously, and is not associated with the LU-LU session over which the scheduling was performed.

In addition to the model programs for remote resource access, SNA defines a *system message model program* to handle status messages and error messages that originate at remote TPSs, directing these messages to the appropriate destination at the local TPS. Examples of messages that might be handled by this model program are IMS/VS broadcast messages and messages indicating that an asynchronously scheduled transaction processing program has terminated abnormally.

This page intentionally left blank.

Chapter 5. Network Management Capabilities of SNA

Network management is the task of keeping a network running efficiently and responsively once it has been designed and implemented. The magnitude of this task is generally proportional to the size of the network and to the number and complexity of its applications.

The following are elements of network management:

- Processing management—the task of controlling the normal operation of network resources
- Problem determination—the task of isolating problems in a network and determining their source
- Problem management—the task of detecting, recording, tracking, and resolving network problems
- Change management—the task of planning, coordinating, tracking, and implementing changes in the network configuration
- Performance management—the task of quantifying, measuring, improving, and reporting network performance

IBM provides a number of program products and field-developed programs that are helpful in managing SNA networks. This chapter briefly summarizes some IBM offerings in each of the areas listed above. (These summaries do not reflect all the capabilities of the products, indicate dependencies between products [such as corequisite products or operating system requirements], or distinguish different versions or releases of products. An IBM marketing representative can provide detailed information on the capabilities and requirements for use of the products described). The suitability or usability of any product described must be established by the user.

Note: This chapter may contain references to or information about IBM products (machines or programs), programming, or services that have not been 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.

The Network Control Center Approach to Network Management

Various approaches can be taken to the task of managing a network. When networks are large and complex, the many items and events to be tracked and monitored, and the need to coordinate changes and manage problems, can be effectively addressed by using the *network control center* approach.

A network control center brings together at one site the appropriate skills, tools, information, and procedures required to manage an SNA network. The objectives of such a center are to operate the network, to maintain the required level of service to end users, to assist in the planning and migration process, and to ensure end-user satisfaction.

Among the possible components of a network control center are the following:

- A user help desk, which provides end users with a single point of contact for obtaining application-oriented assistance, general systems information, and responses to requests on procedural matters
- A network operations component, which is responsible for operating the network, monitoring network status, assisting in network-oriented problem-determination tasks, and maintaining network resources

- A technical support component, which provides second-level support on technical matters to such functions as network operations and the user help desk
- A problem management component that is responsible for tracking and handling network problems
- A change management component that is responsible for notification, review, approval, scheduling, implementation, and coordination of all changes for the network
- A network configuration and inventory component that maintains network configuration and inventory data, vendor information, and information on problem determination tools
- An end-user education and information component that keeps end users informed on how to use applications and features and how to report and record problems

For a detailed presentation of the the network control center concept, see the publication *Network Control Center Guide* (G226-3551).

Processing Management

There must be some means for effectively managing the variety of resources that make up a distributed data processing network. In SNA networks, each system services control point (SSCP) controls and monitors the resources in its own domain. (SSCPs and domains are discussed in Chapter 2.) In order to control its part of the network effectively, each SSCP needs some link between itself and the people who are responsible for defining and operating the network.

The network designer communicates with an SSCP via parameters specified by the system programmer who defines the program in which the SSCP resides. Some of the uses of these parameters include identifying the resources of the domain to its SSCP, indicating which of them are to be activated when the domain is activated, and specifying whether a host logical unit may initiate a session with a peripheral logical unit.

Network operators communicate with SSCPs via operator commands and replies to these commands. SNA access methods support an extensive array of operator commands that allow designated network operators to activate and deactivate network resources and display their status.

Access-method definition parameters control the initial activation status of the resources of a domain, while operator commands control the status of the resources after they are first activated.

The operator-control mechanisms of the SNA access methods were originally designed for control of single-domain networks. The current release of ACF/TCAM includes extended operator-control facilities that permit centralized management of multiple-domain networks in which all host processors use ACF/TCAM as the SNA access method.

Network Communications Control Facility

Network Communications Control Facility (NCCF) is a program product that enhances operations control in ACF/TCAM, ACF/VTAM, and ACF/VTAME environments. NCCF provides a program base for network management by supplying network operations, access method services, operating system services, data storage facilities, and facilities that permit users to tailor NCCF to meet the requirements of their installations. NCCF allows operators to issue operator-control commands from a central operator terminal (or from multiple, distributed operator terminals) to host processors participating in a multiple-domain network. The operator terminals may be either channel attached or link attached and need not be dedicated solely to NCCF use. Each host processor may contain ACF/TCAM, ACF/VTAM, or ACF/VTAME as its SNA access method.

NCCF gives network operators convenient and effective control of the network. Its major functions are as follows:

- A capability for the operator to enter operator commands to be executed at any host processor in the network
- Support for recording of and access to network problem determination data
- Support for user-written command processors that perform special functions in response to user-defined operator commands
- Support for the execution of user-defined command lists
- Support for interdomain communication between network operators
- Control of the network from one or many local or remote operator terminals
- A capability for assigning each network operator a span of control, allowing that operator to control only a designated subset of the resources of the network

Besides these operator-control functions, NCCF also provides communication and data-base facilities for collecting, storing, and retrieving data about network errors. These facilities make NCCF the base for the Network Problem Determination Application (NPDA), a program product that aids in the task of problem determination, as described in this chapter under "Network Problem Determination."

NCCF is IBM's major product offering in network management. NCCF operates as an ACF/VTAM or ACF/TCAM application program in its own partition under OS/VS1 or address space under MVS. Under DOS/VSE, NCCF is executed as a DOS/VSE task in its own partition or as a subtask in the ACF/VTAM partition.

For more information on NCCF, see the publication *Network Communications Control Facility General Information,* GC27-0429.

VSE/Operator Communications Control Facility

The VSE/Operator Communications Control Facility (VSE/OCCF) program product provides facilities to reduce the complexity of system operation of a computer running under DOS/VSE. VSE/OCCF can minimize the required operator interaction with the VSE system console by intercepting messages from DOS/VSE and application programs and by responding automatically with predefined actions. VSE/OCCF, in conjunction with NCCF, can route VSE console traffic to an NCCF operator located at a different site. The facilities of VSE/OCCF enable an NCCF operator at one site to control multiple VSE systems.

For more information on VSE/OCCF, see the *VSE/OCCF General Information* manual, GC33-6113.

Network Problem Determination

Problem determination refers to the task of isolating and defining the cause of a failure within the network. This task must be done quickly so that end users whose sessions are disrupted by the failure can reestablish them and resume their work. Problem determination also requires that the cause of the failure be identified with sufficient precision to allow the equipment or software vendor ~~or the application programmer to fix the problem. Individuals performing~~ problem determination therefore collect data in the form of such documents as program listings and dumps, traces, and console printouts.

Many IBM program products, including ACF/TCAM, ACF/VTAM, ACF/NCP/VS, and IMS/VS, provide diagnosis guides to aid in identifying and isolating problems.

Network Problem Determination Application

The Network Problem Determination Application (NPDA) program product helps in the task of on-line problem determination. NPDA is an IBM network-management application that collects, stores, and monitors network problem determination data and allows users to display this data at one or more terminals designated as NCCF operator terminals. An NCCF operator can use a terminal to display, for example, specific problem determination data related to a specific failure, such as a link failure, that the SNA access method has reported. NPDA helps the NCCF operator to identify a hardware component that is causing problems. NPDA also provides suggested user actions for specific network errors.

NPDA is a set of NCCF command processors and uses NCCF services. Working with NCCF, NPDA collects, organizes, and displays statistics about errors associated with channels, communication controllers, terminals, modems, and links. NPDA prompts the NCCF operator to help in problem determination and displays network errors at several levels of detail. At the most detailed level, the program displays information about a single error, including a statement of its probable cause. NPDA operates with IBM's microprocessor-based modems (the IBM 3863, 3864, and 3865 modems) to help establish probable causes of network errors and to display formatted modem test results.

To support NCCF and other network management programs, SNA has defined a communication network management interface that is implemented by both ACF/TCAM and ACF/VTAM. Through this interface, NCCF operators can issue network-management requests to SNA nodes to obtain statistical maintenance data for use in problem determination. NPDA displays this data to the NCCF operator who requested it. Operators can request and display data for nodes in their own domains or in other domains.

For more information on NPDA, see the publication *Network Problem Determination Application General Information,* GC34-2010.

Threshold Analysis and Remote Access Feature

A feature of the NPDA program product called Threshold Analysis and Remote Access (TARA) is a program product for use in networks that contain IBM 3600 Finance Communication Systems. TARA, which is composed of a set of NCCF command processors, records, analyzes, and displays system management data collected through the system monitor of a 3600 system's operating system.

TARA provides alert messages to NPDA based on user-defined loop-quality and transaction-response-time thresholds. Alert messages are provided when critical errors occur within the controller of a 3600 system, outages occur in the loop that attaches devices to the controller, the loop quality becomes degraded, the transaction response time becomes degraded, or a user application in the controller detects certain error conditions. The alert messages, which may be generated by either the control program within the 3600 or by the TARA program product, are sent to NPDA for presentation to the NCCF operator.

TARA also provides facilities for centralized network operator control of 3600 system monitor functions.

For specific information about TARA, see the *TARA General Information* manual, GC34-2055.

Network Error Management Facility

Another problem determination aid is the Network Error Management Facility (NEMF) field-developed program. NEMF collects and stores in a data base information about errors detected in channels, data links, communication controllers, cluster controllers, and terminals. From this data base, a user can display such information as a cumulative count of temporary errors and permanent errors, a description of the most recent error, and the probable cause of the error. NEMF operates as an application transaction in a CICS/VS system and uses a VSAM data set.

When running under DOS/VS, NEMF provides error information for resources controlled by ACF/NCP/VS as well as for channel-attached devices. When running under OS/VS1 or OS/VS2, NEMF provides error information only for channel-attached devices.

For specific information about NEMF, see the availability notice titled *Network Error Management Facility,* GB21-2527.

Network Problem and Change Management

Problem management is the process of detecting, recording, tracking, and resolving problems involving network resources. *Change management* is the process of planning, coordinating, tracking, and implementing changes to a network. Information/Management and Account Network Management Programs are a program product and a field-developed program, respectively, that provide problem-management and change-management functions.

Information/Management

The Information/Management feature of the Information/System Release 2 program product provides interactive applications for problem, change, and system-configuration management. It is a licensed feature of Information/System Release 2, and is used with Information/System Release 2 to support the system management processes of:

- Reporting, tracking and resolving problems detected at the data processing installation.
- Planning, coordinating, and monitoring changes for the data processing installation.
- Maintaining information about the system inventory of hardware and software components that make up the data processing installation.

Information/Management brings the following benefits to users:

- Greater operator productivity. Keystroke savings are provided by response-chaining the powerful search facilities.

- Usability by operators of various skill levels. Inexperienced users simply respond to prompts. Experienced users can reduce their interactions by using the response-chaining facility. Stored response chains can be provided to clerical and inexperienced users to perform standard tasks. The on-line tutorials and other on-line help facilities are available to all users to provide instructions, usage examples, data formats, etc.
- Easier duplicate-problem recognition. The powerful search facility, coupled with structured descriptions of problem symptoms, makes it easy to recognize problems that have previously occurred at the installation. In the case of suspected IBM software problems, easy access of the EWS (early warning system) file in the Information/MVS data base makes the the task of recognizing duplicate problems even easier.
- Integrated systems management applications. Information/System Release 2, in conjunction with the Information/Management and Information/MVS features, the NCCF/Network Problem Determination Application (NPDA), and the full compatibility with the Interactive Problem Control System (IPCS) products, provides an integration of system management tools.

  Users from the various functional areas in a user's DP organization, such as network operations, systems programming, and systems operations, can, through a single NCCF or TSO terminal (without logoff/logon):
  - Utilize the problem, change, and system-configuration management applications of Information/Management.
  - Utilize the EWS file and other files containing valuable technical information of Information/ MVS.
  - Transfer, with a single NCCF/NPDA command, network problem information gathered by NPDA to Information/Management for inclusion in an automatically opened problem record.
  - Manage all problem information provided by all three of the IPCS products (that is, MVS/IPCS, VSE/IPCS, and VM/IPCS Extension).

- No DB/DC dependency. With terminal support provided by NCCF and TSO, the need to have CICS/VS or IMS/VS running in order to support system management activities has been eliminated.

For specific information about Information/Management, see the *Information/System Release 2 General and Preinstallation Information* manual, GC34-2027.

Account Network Management Programs

To assist SNA users in performing problem management and change management functions, IBM provides the *Account Network Management Programs* (ANMP), a field-developed program that runs under CICS/VS in DOS/VS1, OS/VS1, and OS/VS2 environments. ANMP consists of an integrated set of interactive application programs and batch report programs. ANMP contains three major components: *Problem Management Application, Change Management Application, and Network Configuration Application.*

The *Problem Management Application* provides an on-line capability for recording and tracking network problems. Preformatted displays enable system and network incidents to be consistently reported, assigned, and resolved. This application offers flexible browse and search capabilities and user-defined edit tables. It can create history files and print batch reports.

The Problem Management Application can serve as a focal point for recording and tracking many kinds of problems involving equipment, software, network connections, and application programs. It provides an on-line capability for initial entry and updating of information about problems and their solutions.

A user of this application initially enters information about a problem by creating a single-screen problem record and placing in this record symptomatic and administrative information about the problem. Additional information may be added as acquired; the problem record is thus a chronology of the problem-solving process. If a problem originally assigned to one user is reassigned to another, the related problem record can be transferred to the other user, who can continue updating it. The Problem Management Application can highlight, on a display terminal, problems that have not been solved within a specified interval.

The user can browse through the problem data base by searching on specific fields of display screens (except for comments and descriptions). For example, the user can ask the Problem Determination Application to display all unsolved priority-2 software problems assigned to the system programming group.

The *Change Management Application* provides preformatted displays that permit users to coordinate network changes such as installing and relocating equipment and additions such as new application programs, program update tapes, engineering changes, and feature changes. This application offers flexible browse and search capabilities and user-defined edit tables with an approver/reviewer facility. It can create history files and print batch reports.

With this application, users can:

- Coordinate change across computer systems in an installation
- Learn of impending changes that may affect them
- Learn of the current status of changes in progress

The Change Management Application provides the facilities necessary to develop an approver/reviewer system in a data processing installation. For each installation-defined type of change, the user can build a list of approvers and reviewers. Each change entered becomes associated with the appropriate list of approvers and reviewers. Each of these people can display all pending changes in his area by issuing a single command.

This application has a search capability that allows a user to display, in summary or detailed form, changes that meet specified criteria. For example, the user might display all changes that are to be implemented in the current week or changes pertaining to IMS/VS. The user can also display selected change activities in the order they were entered into the system.

The *Network Configuration Application* provides an on-line data base of components in the network. This application can create, update, and display records containing such details about network components as name, characteristics, location, telephone number, and vendor information. The configuration file displays component connections, and the batch report facility produces a network map, component detail listings, and a record of change activity.

This application is a network inventory management tool that can be used to maintain an up-to-date view of the network as it presently exists and as it is planned to be in the future. This information can be displayed or printed.

The Network Configuration Application can assist users to identify network problems. By making vital information available to individuals responsible for problem determination efforts, the application can minimize the time needed to make repairs. This information includes, for example, the location of each terminal in the network, its characteristics, and the names and telephone numbers of individuals responsible for that terminal.

By providing timely information about the network, the Network Configuration Application can help the operations staff to communicate with users quickly and efficiently about changes in the network. The application can also help users to evaluate future plans for the existing network, such as determining when expansion features will be required. It also permits centralized dispatching of support and maintenance people, which contributes to closer management and control of the entire network.

For more information on ANMP and its components, see the availability notice titled *Account Network Management Programs,* GB21-2521.

Network Performance Management

Performance management is the process of measuring, reporting, and improving the performance of a network.

Network Performance Analyzer

The *Network Performance Analyzer* (NPA) is a field-developed program that collects network operating data for analysis. NPA data can highlight the causes for degradation of performance, such as excessive traffic at certain periods or insufficient line capacity. The data can also help users to isolate performance problems induced by high error rates or wide fluctuations in message traffic.

Using NPA data, the operations staff can tune the network for greater efficiency and potentially improved response times. The data can also be helpful in allowing the staff to determine if the network can be expanded and still maintain satisfactory response times. The Network Performance Analyzer can help users identify unused network capacity, thus allowing the network to accommodate increased transaction volumes, more terminals, or more applications.

NPA data can either be displayed as it is collected or be reviewed later, and data of particular interest can be monitored, on-line or off-line, for exceptions to user-defined limits.

NPA is particularly helpful in evaluating a rapidly changing network: it can assist planners in establishing new networks or maintaining existing networks at high levels of efficiency.

The Network Performance Analyzer consists of an ACF/TCAM or ACF/VTAM application program and an extension to ACF/NCP/VS. NPA runs under OS/VS1 or MVS. (In a multiple-domain network, only one access method [ACF/TCAM or ACF/VTAM] has an NPA application program; this program collects data from every ACF/NCP/VS with which the access method can communicate.)

Chapter 6. Summary of SNA Machines

This chapter summarizes the information-handling systems, the 3705-II Communications Controllers, the modems, and the data encryption devices that can operate in an SNA network.

Note: This chapter may refer to or contain information about IBM products (machines or programs), programming, or services that have not been announced in your country. Such references or information must not be construed to mean that IBM intends to announce such IBM products, programming, or services in your country.

Cross-Industry Information-Handling Systems

Described below are the following information-handling systems that may be used in diverse kinds of industries and organizations:

- 3270 Information Display System
- 3730 Distributed Office Communication System
- 3767 Communication Terminal
- 3770 Data Communication System
- 5250 Information Display System
- 5280 Distributed Data System
- 5520 Administrative System
- 6670 Information Distributor
- 8100 Information System
- Series/1
- System/32
- System/34
- System/38

When appropriately configured, these systems can communicate with System/370, 4331, 4341, 8130, and 8140 processors. (Not all types of the information-handling systems listed above can be connected to all of the processors mentioned. IBM marketing representatives can supply specific information on valid configurations.)

3270 Information Display System

The IBM 3270 Information Display System is a family of products that can be tailored to meet the needs of applications requiring visual display and keyboard entry of alphanumeric information. The 3270 offers users a large selection of components and configurations. Also available are a great variety of features that improve performance, provide additional operational capability, and permit expansion of the display system. Some models of display stations and printers provide high-quality, multi-color images for presenting alphanumeric and graphic data.

3270 systems can be connected to System/370 and other IBM processors over switched and nonswitched links.

Typical applications that can use a 3270 system are:

- Data entry applications
- Transaction processing applications
- Inquiry and update applications
- Interactive program development applications

See the *IBM 3270 Introduction* manual (GA27-2739).

3730 Distributed Office Communication System

The IBM 3730 Distributed Office Communication System is a distributed document-preparation system that is used primarily by office personnel such as secretaries, typists, and clerks. It helps offices to reduce the time and space-consuming clerical activities of creating, updating, storing, and retrieving documents.

The 3730 can operate as a stand-alone unit, attached to a 3790 Data Communication System, or connected to a processor through an SNA network; the capabilities of the 3730 differ in each case.

See the *IBM 3730 Introduction* manual (GA33-3021).

3767 Communication Terminal

The IBM 3767 Communication Terminal is an operator-oriented, desk-top, general-purpose keyboard/printer for SNA networks.

Applications for the 3767 include:

- Interactive problem-solving applications
- Inquiry/response applications
- Low-volume data-entry applications
- Low-volume output-printing applications
- Interactive program-development applications

See the *IBM 3767 Component Description* manual (GA27-3096).

3770 Data Communication System

The IBM 3770 Data Communication System is a family of communication terminals that offers combinations of keyboards and printers in a desk console arrangement. Stand-alone printers, card readers, and card punches are also available.

Several configurations of the 3770 system can be used to perform a variety of interactive, remote job entry, and offline batch processing applications. Such uses include data collection, formatting, checking, and editing; interactive data entry; searching and updating files; batch data transmission; and creation of source documents.

See the *IBM 3770 Introduction* manual (GA27-3144).

5250 Information Display System

The IBM 5250 Information Display System is a family of work stations (displays and printers) that are designed for interactive data entry and inquiry applications. The displays and printers can be connected locally or remotely to a variety of processors. Both typewriter-like and data-entry keyboards are available.

The 5250 display system is adaptable to many different applications. Up to 24 user-selectable command functions can be assigned to the top row of keys on the keyboards; these command functions can be combined as needed to suit the various applications.

See the *IBM 5250 Introduction* manual (GA21-9246).

5280 Distributed Data System

The IBM 5280 Distributed Data System is a versatile, diskette-based system used for collecting, validating, storing, and processing data for a wide range of business applications. The 5280 can be used to capture high volumes of source data, to send data to and receive data from a variety of processors, and to develop and execute user-written programs. The data stations, printers, and control units can be placed in the normal office environment.

Depending on the data formats used, diskettes can be exchanged with other IBM devices and systems, such as the System/32, System/34, System/38, Series/1, 3770, 3790, and 8100.

The 5280 can be programmed to perform extensive data validation and editing during the data-entry operation, thus helping data-entry operators to capture data accurately and efficiently. The 5280 can also be used as a remote batch or remote job entry (RJE) terminal system.

See the *IBM 5280 General Information* manual (GA21-9350).

5520 Administrative System

The IBM 5520 Administrative System is designed specifically to improve office productivity through automated handling of documents. The 5520 combines in one system text processing, document management, and document distribution capabilities. A document can be created at a display terminal, revised, electronically stored, and distributed to the desired recipients anywhere there is another connected 5520 system or compatible communicating office equipment such as the IBM 6670 Information Distributor. Recipients may view the document on a display terminal or print it on an attached printer.

The 5520 system consists of an IBM 5525 System Unit and attached display stations, printers, and magnetic card units. The 5525 system unit can communicate over switched or nonswitched links with other 5525s, compatible office communicating equipment, and System/370 processors.

See the *IBM 5520 Introduction* manual (GC23-0702).

6670 Information Distributor

The IBM 6670 Information Distributor is a printer for word-processing and data-processing applications that serves as a text processor, a communication terminal, and a convenience copier. The printer includes a laser printhead, a magnetic card unit, a processor, and a control panel.

The 6670 can be used to produce and distribute documents, produce repetitive letters using pre-stored text and format instructions, and enter data into and receive data from properly programmed computing systems.

When used as a text processor and a communication terminal in an SNA network, the 6670 can format and print documents from magnetic cards, and can exchange data with other 6670s and with System/370 processors.

The 6670 can operate as a remote work station for remote job entry to System/370 processors. Information such as sales orders, warehouse receipts, management and production reports, inventory lists, and source programs can be recorded on magnetic cards and transmitted by the 6670 to a computer for text processing or merging with other files. When the processing or merging is completed, the computer can transmit the resulting document, such as a sales report or product list, to the 6670 for printing.

~~An easy-to-use operator control language, together with sets of commands and~~ instructions, allows users to control such 6670 functions as changing type styles, setting margins, numbering pages, and changing line spacing.

See the *IBM 6670 General Information* manual (G544-1006).

8100 Information System

The IBM 8100 Information System is a distributed data processing system that combines the benefits of both centralized and decentralized data processing. The 8100 system includes the 8130 and 8140 processors and the 8101 Storage and Input/Output Unit, and is supported by two operating systems: Distributed Processing Program Executive (DPPX) and Distributed Processing Control Executive (DPCX). A very wide range of terminal types (including some BSC and start-stop terminals) can be attached to an 8100 system. The combination of hardware, software, and the large variety of terminal types it supports makes the 8100 suitable for many distributed processing applications in almost every industry.

An 8100 system can operate either as a stand-alone system or as part of a network of interconnected 8100 systems. An 8100 system can communicate with System/370, 4331, and 4341 processors through a 3705-II communications controller. A multi-use communication loop adapter permits a variety of printers, card I/O devices, and terminals to be attached to an 8100 processor.

To ease the selection and configuring of an 8100 Information System, IBM has defined several configurations for typical user application environments. The pre-configured systems are structured so that each makes use of a specific set of 8100 system functions.

See the *IBM 8100 Introduction* manual (GA27-2875).

Series/1

The IBM Series/1 is a family of small, general-purpose computers having a variety of data-processing input/output devices and input/output attachments including disk and diskette storage units, magnetic tape units, printers, display stations, sensor input/output units, and communication units. SDLC, binary synchronous communication (BSC), and start-stop communication lines can be attached to the Series/1.

A variety of features (for example, teletypewriter adapters) allows users to attach their own input/output devices and instrumentation to a Series/1 processor.

Series/1 meets the needs of users who require either a single small computer or multiple small computers. Most Series/1 units are designed for standard 483-millimeter (19-inch) rack mounting.

Series/1 is suitable for realtime, sensor-based applications such as energy management and controlled-access security systems as well as for conventional data processing.

See the *IBM Series/1 System Summary* manual (GA34-0035) and the *IBM Series/1 Programming System Summary* manual (GC34-0285).

System/32

The IBM System/32 is one of the smaller general-purpose data processing systems (about the size of an ordinary office desk) for small enterprises or branch-office locations of larger companies. This system is most useful for both business and problem-solving applications such as:

- Accounts payable and receivable
- Inventory control
- Scientific calculation
- Word processing

The System/32 is designed to be installed by the first-time user of data processing.

See the *IBM System/32 Introduction* manual (GC21-7582).

System/34

The IBM System/34 is a general-purpose data processing system for the distributed data processing environment. System/34 provides the high level of function demanded by sophisticated users; at the same time, System/34 is designed to be installed by first-time users of data processing. An easy-to-use command language, a simplified means for selecting tasks, and operator guidance contribute to the suitability of the System/34 for environments lacking a high degree of data processing expertise. For example, System/34 helps users to tailor its facilities to their operating environment by leading them through a question-and-answer session from its console. This configuration process takes just a few minutes.

A comprehensive set of IBM-developed and -tested utility programs accomplish many common tasks such as creating and maintaining files and creating reports from files, without requiring user programming.

A variety of available program products make the System/34 suitable for many business and manufacturing applications. For example, MAPICS (Manufacturing Accounting and Production Information Control System) is a group of workstation-oriented applications that include order processing and accounting, financial, production control and costing, and material requirements planning applications.

See the *IBM System/34 Introduction* manual (GC21-5153).

System/38

The IBM System/38 is a general-purpose data processing system designed to significantly improve users' productivity in developing, maintaining, and enhancing applications for the interactive workstation environment. The system permits direct attachment of many workstations (displays and printers) as well as remote attachment of workstations over SDLC links.

The workstations can be placed where needed in the organization (offices, plants, warehouses, etc.) so that company personnel can share a common data base and the processing power of a computer.

System/38 is specifically designed to fulfill the requirements unique to the workstation environment. An example is the ability for workstation users to share programs, data files, and system resources without significant delays to any user.

System/38 can support a range of environments, from one consisting largely of batch processing to one that makes extensive use of interactive workstation applications. System/38 is designed to manage a mixture of batch and interactive work, providing both fast response to workstation users and good throughput for batch jobs.

See the *IBM System/38 Introduction* manual (GC21-7728).

Industry-Specific Information-Handling Systems

Applications for industry-specific information-handling systems vary depending upon the processing requirements of the industry. The following systems are available for use in the finance, manufacturing, retail, and supermarket industries:

- 3600 Finance Communication System
- 3630 Plant Communication System
- 3650 Retail Store System
- 3650 Programmable Store System
- 3660 Supermarket System
- 3680 Programmable Store System

An IBM marketing representative can provide information about the configuration requirements for each system.

3600 Finance Communication System

The IBM 3600 Finance Communication System is used to control the operations of a financial institution. For example, tellers can use this system to debit and credit checking and savings accounts, post interest, and record loan payments. Accountants can use it to maintain a record of cash flow through the financial institution. IBM-supplied programs assist tellers or merchants in processing payments and credit card transactions at the point-of-sale. A 3600 system can be tailored to process each transaction as required by a financial institution's specific operation.

See the *IBM 3600 System Summary* manual (GC27-0001).

3630 Plant Communication System

The IBM 3630 Plant Communication System is used to collect, process, and distribute information related to manufacturing operations. Data to be processed is collected from punched-hole badges, punched cards, magnetic striped documents, or terminal keyboards. Data entry from sensors connected to user equipment and actuators attached to plant floor terminals is also possible.

See the *IBM 3630 System Description* manual (GA24-3652).

3650 Retail Store System

The IBM 3650 Retail Store System consists of devices that can be used on the selling floor, in the credit office, and in the receiving room. The host processor is usually located in a central site and communicates with the system's store controller at a remote store location. For example, a store manager can request that a sales analysis application be executed at the host processor and that reports from this analysis be transmitted to, and printed at, the store.

See the *IBM 3650 Retail Store System Introduction* manual (GA27-3075).

3650 Programmable Store System

The IBM 3650 Programmable Store System permits a large retail or supermarket chain's purchasing or receiving, accounting, and retail-store sales departments in each of the chain's stores to access the services provided by a central data processing center. User-written application programs define the configuration best suited for a specific business requirement.

The 3650 Programmable Store System can operate as:

- A user-programmed retail system
- A user-programmed supermarket system
- A user-programmed retail and supermarket system
- A fixed-function retail system
- A combination user-programmed retail and supermarket system and a fixed-function retail system.

The host processor (and the 3705 communication controller, if used) are usually located at the central data processing center. The store controller is located at a remote store and is link-attached to the communication controller or processor. Input (such as information required to prepare a purchase order) can be entered at a store from a display terminal, a 3767 terminal, or a point-of-sale terminal, then transmitted to the host processor. After processing by an application program, the output (such as a prepared purchase order) can be returned to the store from which the input data originated, verified on a display terminal, and then printed.

See the *IBM 3650 Programmable Store System Introduction* manual (GA27-3163).

3660 Supermarket System

The IBM 3660 Supermarket System improves supermarket operations by providing efficient customer checkout services and host processing functions. A 3660 system is available as a key-entry system or as a scanning system. Depending on the type of system selected, the checkstand operator can use the checkout scanner or the keyboard to send item price information to the host processor.

Usually, each supermarket has a store controller and point-of-sale terminals. The point-of-sale terminals perform the same checkout functions as cash registers, plus many additional functions.

See the *IBM 3660 Introduction* manual for the Scanning System (GA27-3076) or for the Key-Entry System (GA27-3111).

3680 Programmable Store System

The IBM 3680 Programmable Store System consists of point-of-sale terminals used to perform sales transactions and administrative functions for many kinds of retail operations. Some examples are:

- Drug stores
- Electronics stores

- Lumber and building supply stores
- Restaurants
- Supermarkets
- Department stores

The 3680 system can perform a wide variety of applications in a store environment. User-written application programs are used to implement the specific store application. Examples of the functions that an application program can provide are:

- Quick and accurate transaction processing at the point of sale.
- Collection and storage of data for inventory control, audit procedures, and management reports. Data can be collected from either the keyboard or the magnetic wand reader. (The magnetic wand reader is a hand-operated input device that can read encoded data from a credit card's magnetic stripe.)
- Price and credit checking inquiries.

See the *IBM 3680 Introduction* manual (GA27-3199).

Communication Controller and Communication Adapters

This part of the chapter summarizes the 3705-II Communications Controller and features of processors generally called communication adapters.

3705-II Communications Controller

The IBM 3705-II Communications Controller is a programmed control unit available in many models that vary in the size of storage they contain and in the number of communication lines they can control. The storage contains a control program and buffers for holding incoming and outgoing data traffic. The control program is loaded into the 3705-II from a host processor over a channel or an SDLC link. Once loaded, the control program interacts with an access method in a host processor to provide the physical management of the network. (In some configurations, the control program may interact with more than one access method in one or more host processors.)

A 3705-II may be attached to as many as four host processor channels. Various interface hardware features permit a variety of synchronous and asynchronous links, operating at one of several different data rates, to be attached to the 3705-II. Both nonswitched and switched links are supported. The links can attach the 3705-II to other 3705-IIs and to a variety of terminals, controllers, and data processing systems.

See the *IBM 3705 Introduction* manual (GA27-3051).

Communication Adapter Features

A number of processors, such as the 4331 Processor and processors for the 3600 Finance Communication System, 8100 Information System, System/34, and System/38, are connected to SNA networks through a feature typically called a Communication Adapter or Integrated Communication Adapter, rather than through a 3705-II Communications Controller. These adapters, in conjunction with the access method or control program in the processor, are equivalent to a 3705-II, with its network control program (ACF/NCP/VS), in connecting the processor to an SNA network and enabling it to communicate through the network with other SNA machines. For information on these adapters, see the appropriate publications referred to in introductory manuals for the respective processors and systems (for example, the *System/38 Introduction* manual).

Modems

Modems are devices that modulate and demodulate signals transmitted over communication facilities. As such, modems are key elements of an effective SNA network. This part of the chapter summarizes three advanced-function, microprocessor-based modems:

- 3863 Modem
- 3864 Modem
- 3865 Modem

These modems operate in conjunction with network management program products in the host processor and in the 3705-II Communications Controller. The network management programs are summarized in Chapter 5.

The modems and their supporting program products allow a network operator to isolate most problems to either the communications controller, the line, the line interface, a cluster controller, a terminal, or a modem. Therefore, a trained specialist or diagnostic equipment at the central site is not required. A network operator can diagnose problems from the same operator terminal used to control network operation. The operator can respond to system error messages, initiate diagnostic tests, or analyze network error data.

3863, 3864, and 3865 Modems

The IBM 3863 and 3864 Modems operate at 2400 bits per second and 4800 bits per second, respectively, over switched or nonswitched lines. The IBM 3865 Modem operates at 9600 bits per second over nonswitched lines. All three modems are available in two models and use either the BSC or SDLC line protocol. All three modems support switched network backup for nonswitched lines and provide an auto-answer capability for both switched operation and switched network backup.

Each modem can be customized through use of a set of optional features. These optional features make selection of the proper configuration for a network easier. Examples of the optional features include:

- Multiterminal Fanout feature
- Data Multiplexing feature
- Extended Diagnostic feature
- Rack Mount Adapter feature

See the *IBM 3863, 3864, and 3865 Introduction and Site Preparation Guide* manual (GA27-3200).

Data Encryption Devices

A data encryption device enciphers or deciphers digital data transmitted over a communication line. This part of the chapter summarizes the 3845 and 3846 data encryption devices.

3845 and 3846 Data Encryption Devices

The IBM 3845 and 3846 Data Encryption Devices are used in pairs in an SNA network—one at each termination of a line. The 3845 and 3846 are each available in several models. Depending on the model, the 3845 or the 3846 can be used with either the BSC or SDLC protocol or the business machine clock.

The 3845 can be placed on a table top or shelf-mounted between a data terminal equipment (DTE) and a data communications equipment (DCE). A mounting panel accessory is available for the 3846 (only) so that up to four 3846s can be mounted on a rack and positioned, in the same way as the 3845, between a DTE and a DCE.

See the *IBM 3845 and 3846 General Information* manual (GA27-2865).

Chapter 7. Summary of SNA Programs

This chapter summarizes many of the program products that can be executed in the processors and communication controllers that can be used in an SNA network. Generally, the programs described are those directly concerned with operating SNA networks (for example, ACF/TCAM, ACF/VTAM, ACF/VTAME, and ACF/NCP/VS) and major programs and subsystems (such as transaction processing systems and remote job entry programs) that help end users move data, transactions, and jobs through SNA networks. Another category of programs directly related to SNA networks—Network Management Aids—is described in Chapter 5.

The summaries in this chapter are necessarily brief and general. These summaries do not reflect all of the capabilities of the products, indicate dependencies between products (such as corequisite products or operating system requirements), or distinguish different versions or releases of products. (An IBM marketing representative should be consulted for detailed information on the capabilities and dependencies of the products described.) The suitability or usability of any product described must be established by the user.

This chapter does not describe many other IBM program offerings that, though used in installations that have SNA networks, do not directly concern the control or use of these networks. Examples are operating systems, language compilers, program generators, and many application programs and subsystems that can be executed in processors that are attached to SNA networks.

Note: This chapter may contain references to or information about IBM products (machines and programs), programming, or services that have not been 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.

Programs that Control SNA Networks

This part of the chapter summarizes many of the programs that reside in the processors, systems, and communication controllers summarized in Chapter 6.

Access Methods

Access method programs are primarily responsible for routing data between host processor storage and the input/output devices in the SNA network.

The access methods summarized are:

- Advanced Communications Function for the Telecommunications Access Method (ACF/TCAM)
- Advanced Communications Function for the Virtual Telecommunications Access Method (ACF/VTAM)
- Advanced Communications Function for VTAM Entry (ACF/VTAME)

ACF/TCAM

The ACF/TCAM program product is an SNA access method that controls communication between network resources (both logical units and non-SNA devices). ACF/TCAM directly controls the transmission of data to and from channel-attached devices and uses ACF/NCP/VS to forward data to and receive data from devices attached by communication lines. Network resources can communicate without knowledge of intermediate connections such as communication lines and communication controllers.

To provide this communication, the ACF/TCAM base program product:

- Permits the use of network resources by name, without specific knowledge of their locations or addresses
- Controls allocation of network resources
- Permits sharing of such network resources as communication lines, communication controllers, and terminals
- Establishes, controls, and terminates sessions involving logical units and non-SNA devices in its domain
- Transfers data between network resources (logical units and non-SNA devices)
- Queues data it receives from and sends to the network (except for data exchanged with application subsystems)
- Passes data directly between the network and application subsystems such as CICS/VS and IMS/VS (that is, does not queue the data)
- Permits the ACF/TCAM operator to monitor and alter operation of the domain
- Permits the network configuration to be changed while the network is operating
- Attempts to detect and correct problems in network operation
- Permits use of the Time Sharing Option (OS/VS2 MVS only) through ACF/TCAM

The Multisystem Networking Facility is an optional feature of ACF/TCAM that enables network resources (logical units and non-SNA devices) in one domain to communicate with network resources in other domains in network configurations that have multiple host processors and multiple access methods. To provide this cross-domain communication, ACF/TCAM with its Multisystem Networking Facility:

- Establishes, controls, and terminates sessions between network resources (logical units and non-SNA devices) in the ACF/TCAM domain and network resources in other domains
- Allows control of one host processor's communication controller(s) to be shared by or transferred to another host processor

ACF/TCAM is available for MVS and OS/VS1 systems. It supports channel-attached SNA and non-SNA devices, as well as applicable SNA and non-SNA devices attached by a loop adapter. Through ACF/NCP/VS, ACF/TCAM supports SDLC-attached SNA devices and non-SNA devices.

For information on ACF/TCAM facilities, see the *ACF/TCAM General Information: Introduction* manual (GC30-3057). For information on ACF/TCAM concepts, see the *ACF/TCAM General Information: Functional Description* manual (GC30-3131).

ACF/VTAM

The ACF/VTAM program product is an SNA access method that controls communication between logical units in a network. ACF/VTAM directly controls the transmission of data to and from channel-attached devices and uses ACF/NCP/VS to forward data to and receive data from devices attached by communication lines. Logical units can communicate without knowledge of intermediate connections such as communication lines and communication controllers.

To provide this communication, the ACF/VTAM base program product:

- Permits the use of logical units by name, without specific knowledge of their locations or addresses
- Controls allocation of network resources
- Permits sharing of such network resources as communication lines, communication controllers, and terminals
- Establishes, controls, and terminates sessions between logical units in its domain
- Transfers data between logical units
- Permits the ACF/VTAM operator to monitor and alter operation of the domain
- Permits the network configuration to be changed while the network is operating
- Attempts to detect and correct problems in network operation
- Permits use of the Time Sharing Option (OS/VS2 MVS only) through ACF/VTAM (TSO/VTAM)

The Multisystem Networking Facility is an optional feature of ACF/VTAM that enables logical units in one domain to communicate with logical units in other domains in network configurations that have multiple host processors and multiple access methods. To provide this cross-domain communication, ACF/VTAM with its Multisystem Networking Facility:

- Establishes, controls, and terminates sessions between logical units in the ACF/VTAM domain and logical units in other domains
- Allows control of one host processor's communication controller(s) to be shared by or transferred to another host processor

ACF/VTAM is available for MVS, OS/VS1, and VSE systems. It supports channel-attached SNA devices and non-SNA 3270 terminals, as well as applicable SNA and non-SNA devices attached by a loop adapter. Through ACF/NCP/VS, ACF/VTAM supports SDLC-attached SNA devices and BSC 3270 terminals. (Non-SNA devices other than BSC 3270s are supported only in conjunction with the Network Terminal Option (NTO) program product or with user routines and user-written network addressable units that execute as part of ACF/NCP/VS.)

For information on ACF/VTAM facilities, see the *ACF/VTAM General Information: Introduction* manual (GC27-0462). For information on ACF/VTAM concepts, see the *ACF/VTAM General Information: Concepts* manual (GC27-0463).

ACF/VTAME

The ACF/VTAME program product is an SNA access method that controls communication between logical units in a network. ACF/VTAME directly controls the transmission of data to and from channel-attached devices and uses a communication adapter to forward data to and receive data from devices attached by communication lines. ACF/VTAME enables logical units in one domain to communicate with logical units in other domains in network configurations that have multiple host processors and multiple access methods. Logical units can communicate without knowledge of intermediate connections such as communication lines, communication adapters, or communication controllers.

To provide this communication, the ACF/VTAME program product:

- Permits the use of logical units by name, without specific knowledge of their locations or addresses
- Controls allocation of network resources
- Permits sharing of such network resources as communication lines, communication adapters, and terminals
- Establishes, controls, and terminates sessions between logical units in its domain, and sessions between logical units in the ACF/VTAME domain and logical units in other domains
- Transfers data between logical units
- Provides intermediate-node routing to other domains
- Permits the ACF/VTAME operator to monitor and alter operation of the domain
- Permits the network configuration to be changed while the network is operating
- Attempts to detect and correct problems in network operation

ACF/VTAME is available for use with the VSE system. It supports channel-attached SNA devices and non-SNA 3270 terminals, as well as applicable SNA and non-SNA devices attached by a loop adapter. Through a communication adapter, ACF/VTAME supports SDLC-attached SNA devices and BSC 3270 terminals.

For information on ACF/VTAME facilities, see the *ACF/VTAME General Information: Introduction* manual (GC27-0438). For information about ACF/VTAME concepts, see the *ACF/VTAME General Information: Concepts* manual (GC27-0451).

Transaction Processing Systems

Transaction processing systems (TPS) control sessions between interactive application programs and the devices to which the application program is connected. Transaction processing systems also permit users to change or expand application programs that examine and maintain files on a large data base. Whenever a device such as a display sends a request for use of the operating system, a transaction processing system identifies the application program required for the job. Typically, the transaction processing system loads the application program into the operating system (if it is not already loaded), and starts a task to execute that application program. A session can then be activated between the application program and the device. When the session has been completed, the transaction processing system ends the task and returns the device to a standby state. During execution of the application program, the transaction processing system transmits data between the program and the device.

This part of the chapter summarizes four virtual storage transaction processing systems:

- Information Management System/Virtual Storage (IMS/VS)
- Customer Information Control System/Virtual Storage (CICS/VS)
- DPPX Data Base and Transaction Management System (DPPX/DTMS)
- Airline Control Program/Transaction Processing Facility (ACP/TPF)

Information Management System/Virtual Storage (IMS/VS)

The Information Management System/Virtual Storage and the features available for use with it provide a variety of services for the application

programs that can operate in an SNA network.
Examples of applications that can use IMS/VS are:

- Payroll record applications
- Personnel record applications
- Manufacturing bill of material applications
- Inventory control applications
- Accounts receivable applications
- Transaction processing applications

See the *IMS/VS General Information* manual (GH20-1260).

Customer Information Control System/Virtual Storage (CICS/VS)

The Customer Information Control System/Virtual Storage provides a variety of services for the application programs that can operate in an SNA network. Examples of applications that can use CICS/VS are:

- Inquiry applications
- Inquiry-with-update applications
- Data-entry applications
- Batch-processing applications
- Message-switching applications

All of these kinds of applications can be handled concurrently by CICS/VS.

See the *CICS/VS General Information* manual (GC33-0066).

DPPX Data Base and Transaction Management System (DPPX/DTMS)

The DPPX Data Base and Transaction Management System (DPPX/DTMS) operates under control of the DPPX/Base operating system. DTMS handles transaction processing and manages data bases in an 8100 Information System.

When DTMS is used, system programmers are not required to write instructions that (1) establish domain membership for an application program, (2) establish a connection to a terminal, and (3) send data to and receive data from a terminal. DTMS performs these functions.

See the *DPPX/DTMS General Information* manual (GC26-3915).

Airline Control Program/Transaction Processing Facility (ACP/TPF)

ACP/TPF is a reliable, highly responsive, performance-oriented transaction processing system for realtime, transaction-driven applications that must accommodate high message rates. ACF/TPF systems are characterized by thousands of terminals dispersed over a large geographic area, where each location may have from one to several hundred terminals. ACP/TPF provides realtime inquiry and update to a large, centralized data base, where message length is generally short in both directions and response time is generally less than three seconds.

ACP/TPF is applicable to any online, transaction-oriented application that requires fast response time from a large number of terminals. Some examples of applications are airline seat reservations, hotel reservations, car rental reservations and billing, credit authorization and verification, and loan payment processing.

See the *ACP/TPF Application* manual (GH20-2140).

Interactive Support Programs

Interactive support programs are programs that interact with other operating system programs to support specific program functions.

This part of the chapter summarizes three interactive support program products. Each can interact with an OS/VS system control program and a transaction processing system to provide a user service. The program products summarized are:

- Time Sharing Option (TSO)
- Virtual Storage Personal Computing (VSPC)
- Virtual Machine/VTAM Communications Network Application (VM/VCNA)

Time Sharing Option (TSO)

TSO is a full-function time sharing system that provides interactive computing for the large-system environment. TSO is a component of the OS/VS2 (MVS) operating system.

TSO supports an extensive range of terminals through ACF/TCAM and ACF/VTAM. Terminals may be shared between TSO and other ACF/TCAM or ACF/VTAM applications. Interfaces to high-speed remote job entry (RJE) facilities in JES2 and JES3 are provided.

TSO has a comprehensive, yet easy-to-use edit capability for creating and manipulating data, programs, and JCL (job control language) files.

TSO satisfies the needs of the following people and gives them the full power of a large computer through a terminal:

- System programmers: for maintaining system libraries, catalogs, and procedure libraries
- Application programmers: for developing new applications (batch, interactive, and data base/data communication [DB/DC]) and for maintaining existing application programs
- Programming librarians: for creating, maintaining, and controlling development-support and production libraries
- Problem solvers who need full operating system facilities
- End users of interactive programs

As part of MVS, TSO offers the widest range of computer functions, through the widest range of terminals, to the widest range of users of any IBM interactive computing system.

See the *OS/VS2 Systems Programming Library: TSO* manual (GC28-0629).

Virtual Storage Personal Computing (VSPC)

Virtual Storage Personal Computing provides functions tailored for users who do not have extensive data processing knowledge. VSPC provides easy-to-use commands that permit a user to control system resources, maintain user profiles, and develop source programs. A set of conversational remote job entry commands permit the VSPC user to submit batch jobs for processing. The VSPC AID prompting facility permits the creation of user-defined commands.

Programming languages that can be used with VSPC functions to develop source programs are APL, BASIC, and FORTRAN.

See the *VSPC General Information* manual (GH20-9070).

Virtual Machine/VTAM Communications Network Application (VM/VCNA)

The Virtual Machine/VTAM Communications Network Application (VM/VCNA) allows users at display or keyboard/printer terminals in an SNA network to logon to and use the facilities of VM/370 and Conversational Monitor System (CMS) as virtual-machine console operators.

VM/VCNA allows SNA users access to VM/CMS and allows VM/370 to better participate in an SNA network.

VM/VCNA is an ACF/VTAM or ACF/VTAME application that is executed in a VM/SP virtual machine. (VM/SP is a program product that extends the function of VM/370.)

See the *VM/VCNA General Information* manual (GC27-0501).

Remote Job Entry Programs

Remote job entry (RJE) programs for SNA networks are data management program components of an MVS or DOS/VSE system control program. Jobs can be submitted to an operating system for processing from remotely located input devices when RJE programs (subsystems) are installed. This part of the chapter summarizes eight SNA Remote Job Entry subsystems:

- OS/VS1 Remote Entry Service for Job Entry Subsystem 1 (RES/JES1)
- OS/VS Job Entry Subsystem 2 (JES2) with Remote Job Entry
- OS/VS2 Network Job Entry (NJE) for JES2 (JES2/NJE)
- OS/VS2 Job Entry Subsystem 3 (JES3) with Remote Job Processing
- MVS/Information Distributor Workstation Support (MVS/IDWS)
- DOS/VSE Virtual Storage Extended/POWER (VSE/POWER)
- DPPX Remote Job Entry Workstation Facility
- OS/VS and DOS/VSE Job Entry Program (JEP) and File Transfer Program (FTP)

OS/VS1 Remote Entry Services for Job Entry Subsystem 1 (RES/JES1)

OS/VS1 Remote Entry Services for Job Entry Subsystem 1 extends the functions of OS/VS1 JES1. RES/JES1 provides OS/VS1 batch processing facilities to users at remote devices. Thus, instead of collecting jobs and taking them to the host processor location so that the network operator can enter the jobs into the computer, the remote user can submit a job or a batch of jobs directly into the system from a terminal. A subset of network operator commands permits a remote user to inquire about and manipulate his or her own jobs.

See the *OS/VS1 Planning and Use Guide* manual (GC24-5090).

OS/VS Job Entry Subsystem 2 (JES2) with Remote Job Entry

Remote Job Entry is a standard facility of JES2 that accepts job streams submitted from remote work stations. Job output may be returned to the submitting work station, printed or punched at the central site, or routed to some other destination. The operator at the work station may use JES2 commands to inquire about or manipulate jobs, output, or devices.

See the *JES2 Introduction* manual (GC23-0002).

Network Job Entry (NJE) for JES2 (JES2/NJE)

JES2/NJE has all the facilities of JES2 including Remote Job Entry (RJE). In addition, it transmits to another NJE system selected jobs, in-stream data sets, system output (SYSOUT) data sets, operator commands and messages, and job accounting information. The SNA network, a channel-to-channel connection, or a binary synchronous communication (BSC) link may be used for the transmission.

See the *JES2/NJE General Information* manual (GC23-0010).

OS/VS2 Job Entry Subsystem 3 (JES3) with Remote Job Processing

Remote Job Processing (RJP) is a standard facility of JES3. JES3 controls job scheduling and job flow in a loosely coupled multiprocessing complex of up to eight MVS processors. RJP provides the facility to submit jobs from a remote work station to any processor in the complex. The job output may be returned to the submitting work station, printed or punched at the central site or routed to any other work station. A console authorization scheme allows the user to define the scope of the JES3 commands that may be entered from the console at a remote work station. The operator at the work station may be restricted to inquiring about and manipulating only work and devices associated with that work station. Or the operator may be allowed to inquire about and manipulate work and devices associated with other work stations.

See the *JES3 Introduction* manual (GC28-0607).

MVS/Information Distributor Workstation Support (MVS/IDWS)

MVS/Information Distributor Workstation Support (MVS/IDWS) provides remote job entry support for IBM 6670 Information Distributors on SDLC links in a way that enhances standard RJE support. Some of the benefits of MVS/IDWS are:

- Users with limited data processing skills can submit work by issuing simple, English-like commands instead of MVS job control language statements to generate MVS jobs.
- Users can protect private or confidential output by specifying a key or password for output when they submit work. IDWS holds the protected output at the host processor until the user tells IDWS (by including the key in an IDWS command) that he or she is ready to receive it.
- A 6670 can be used for word processing tasks during the day and as a computer printer at night because IDWS automatically connects to the 6670 and sends it output created at the host processor. IDWS keeps logs and audit trails for all work station communication to help users to separate or disseminate the output and to understand errors that may have occurred when the 6670 was unattended.

See the *MVS/IDWS General Information* manual (GC23-0031).

DOS/VSE Virtual Storage Extended/POWER (VSE/POWER)

DOS/VSE Virtual Storage Extended/POWER is the spooling subsystem for DOS/VSE. VSE/POWER provides automatic staging of unit record input and output and also controls priority scheduling of all programs with which it communicates.

The *Remote Job Entry* feature permits remotely located users to submit jobs, or enter commands to the host operating system and have the output returned to either the originating input device or an alternate device.

The *Shared Spooling* feature permits VSE/POWER files to be shared between two or more programs executing in different devices.

See the *VSE/POWER General Information* manual (GH12-5128).

DPPX Remote Job Entry Workstation Facility

DPPX/RJE allows an IBM 8100 Information System processor to operate with one or more remote job entry (RJE) work stations concurrently with other DPPX application programs. DPPX/RJE runs as an application program under the control of the Distributed Processing Programming Executive (DPPX).

See the *DPPX/RJE Workstation Facility General Information* manual (GC30-3053).

OS/VS and DOS/VSE Job Entry Program (JEP) and File Transfer Program (FTP)

The OS/VS and DOS/VSE Job Entry Program and File Transfer Program allow a DOS/VSE operating system to appear as a remote work station to an OS/VS RJE subsystem or another DOS/VSE job entry subsystem. As a result, JEP and FTP permit both output from jobs and output from files to be routed to either the originating host processor or an alternate host processor. JEP and FTP operate in association with each other.

See the *General Information* manual for the JEP and FTP programs (GH12-5129).

Host-Resident Programs that Support Programs in Other SNA Nodes

This part of the chapter summarizes three host-resident program products that support programs residing in other SNA nodes. They are:

- Systems Support Programs (SSP) for ACF/NCP/VS
- Distributed Systems Executive (DSX)
- Host Command Facility (HCF)

System Support Programs for ACF/NCP/VS

System Support Programs for ACF/NCP/VS are a set of programs used to prepare, install, and use ACF/NCP/VS. The system support programs execute as non-system tasks under control of an OS/VS or DOS/VS operating system. The system support programs consist of:

- A procedure for generating ACF/NCP/VS
- A communications controller assembler
- Loader and dump utilities
- A dynamic dump utility
- Trace Analysis Program (ACF/TAP)

See the *ACF/NCP/VS General Information* manual (GC30-3058).

Distributed Systems Executive (DSX)

The Distributed Systems Executive (DSX) is a set of System/370 programs and files that store, manage, and distribute software modules and data in a distributed data processing network containing one or more System/370 processors and 8130 or 8140 processors or 3790 communication system controllers. DSX provides simple, comprehensive, and effective data and network management functions for the DPCX operating system and users of the IBM 3790 Communication System.

See the *DSX General Information* manual (GH20-2149).

Host Command Facility (HCF)

The Host Command Facility is used as a host support program for System/370, 4300, and 8100 processors. The Host Command Facility permits an operator at the System/370 or 4300 Processor console to perform the following functions at an 8130 or 8140 processor:

- Control the operations of the DPPX operating system.
- Use most operation and service functions available at an 8130 or 8140 location. The exceptions are functions that require manual intervention (for example, mounting a tape or inserting a diskette).
- Perform operator-oriented tasks related to either DPPX or DPCX problem determination and isolation.

See the *HCF General Information* manual (GC27-0453).

Programs Resident in the 3705-II Communications Controllers

Programs resident in a 3705-II subarea node interact with access method programs in a host operating system to control data transfer through an SNA network. The access method programs are summarized under "Access Methods" in the "Programs that Control SNA Networks" part of this chapter.

This part of the chapter summarizes two program products that reside in the 3705-II:

- Advanced Communications Function for Network Control Program/Virtual Storage (ACF/NCP/VS)
- Network Terminal Option (NTO)

ACF/NCP/VS

ACF/NCP/VS is an Advanced Communications Function program product for the 3705-II in an SNA network. ACF/NCP/VS uses SNA formats and protocols to interact with an ACF access method (that is, ACF/TCAM, ACF/VTAM, and/or ACF/VTAME) to control the operation of a network. Release 3 of ACF/NCP/VS provides communication control functions for a single-domain network or a multiple-domain network. The communication control functions include:

- Polling and addressing of stations on multipoint links
- Dialing and answering stations over a switched network
- Recognizing and reacting appropriately to device control characters
- Translating machine codes to appropriate line and device codes for data transmitted using the binary synchronous link protocol
- Dynamically allocating buffers from controller storage as data is received from either a host processor or a station
- Selecting communication line speeds
- Maintaining error records for the 3705-II hardware, the links and devices attached to the 3705-II, and the network control program

Release 3 of ACF/NCP/VS also provides program functions that permit it to operate in a multiple-domain network.

See the *ACF/NCP/VS General Information* manual (GC30-3058).

Network Terminal Option (NTO)

The Network Terminal Option program product extends the capabilities of ACF/NCP/VS in a 3705-II communications controller to allow an SNA network to support a select group of non-SNA terminals. NTO is executed along with ACF/NCP/VS in the 3705-II and operates with ACF/VTAM Release 2 or Release 3 or ACF/TCAM Version 2 Release 3. The non-SNA terminals that NTO supports are:

- IBM 2740 Model 1 Communications Terminal
- IBM 2741 Communications Terminal
- IBM 3101 Communications Terminal
- IBM 3767 Communications Terminal in 2741 compatibility mode
- Western Union Teletypewriter Exchange Service (TWX Models 33/35)
- World Trade teletypewriter terminals (WTTY), nonswitched only

See the *NTO General Information: Introduction* manual (GC38-0297).

Network Design Aids

Programs used as design aids enable a data processing administrator to evaluate the performance of the interactive program elements of an SNA network before they become active. The Teleprocessing Network Simulator (TPNS) is one such program.

Teleprocessing Network Simulator (TPNS)

The Teleprocessing Network Simulator is a testing package that enables users to test and evaluate application programs in a simulated network environment in order to reduce the testing time and to simplify the maintenance of a network.

Applications using a network can be tested better, faster, and possibly less expensively, reducing the risk of failure in both function and schedule. When used to predict network behavior under an expected future load, TPNS can help users to detect potential performance inadequacies in time to take corrective action.

TPNS can be used for testing in both single-domain and multiple-domain networks in MVS, SVS, and VS1 environments.

See the *TPNS General Information* manual (GH20-1907).

Network Management Aids

The key activities involved in managing a network are:

- Processing Management
- Problem Determination
- Problem and Change Management
- Performance Management

These activities are facilitated by a number of IBM program products and field-developed programs. These programs are summarized in Chapter 5, "Network Management Capabilities of SNA," and are therefore not included in this chapter.

This page intentionally left blank.

Appendix A. SNA, Public Networks, and International Standards

Any SNA network that extends beyond the property boundaries of the network owner must rely on communication services furnished by public networks. These are networks established by communication common carriers (in the United States and Canada) or telecommunication administrations (in most other countries) to provide nonswitched, circuit-switched, or packet-switched services.

Until recently, practically all nonswitched and circuit-switched services were provided with facilities originally designed for voice transmission—that is, the telephone network. This has been changing as several new specialized data transmission services have become available in the past few years, and more such services are in the planning stages. Many of these specialized services use digital transmission techniques instead of the analog techniques used for voice transmission. The digital techniques offer some major advantages, including significantly reduced error rates, increased efficiency (especially at high data rates), and, in some cases, lower transmission costs.

These new public networks are being designed to conform to a number of international telecommunication standards and recommendations developed by organizations such as the International Organization for Standardization (ISO); the International Telegraph and Telephone Consultative Committee (CCITT), an activity of the International Telecommunication Union; and the American National Standards Institute (ANSI).

Systems Network Architecture is compatible with a number of telecommunication standards and recommendations. SNA support for some of the current international and national standards and recommendations is as follows:

- CCITT V.24 and V.25 (for telephone [analog] networks)
- CCITT X.21 (for digital, point-to-point and multipoint, private line and circuit-switched services)
- ISO HDLC (high-level data link control) (SDLC provides the HDLC subset for unbalanced normal class of procedure)
- ANSI ADCCP (advanced data communication control procedure) (SDLC provides the ADCCP subset for unbalanced normal class of procedure)
- Packet-switched networks using CCITT X.25

Not all SNA products support all of these interface standards, and the support offered varies from country to country. Specific information should be obtained from IBM marketing representatives.

This page intentionally left blank.

Glossary

This glossary includes terms defined in Systems Network Architecture; these are prefixed by "In SNA,". Also included are some terms used in this publication that are not specific to SNA. For definitions of terms not appearing in this glossary, see *IBM Data Processing Glossary,* GC20-1699.

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 communications 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.

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).

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. See also *path control network, network addressable unit.* (2) The component that provides these capabilities.

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.

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.

communication adapter. An optional hardware feature, available on certain processors, that permits communication lines to be attached to the processors.

communication controller node. 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.

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).

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.

cross-domain. In SNA, pertaining to control or resources involving more than one domain.

cryptography. The transformation of data to conceal its meaning.

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 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.

distributed data processing. Data processing in which some or all of the processing, storage, and control functions, in addition to input/output functions, are situated in different places and connected by transmission facilities. Contrast with *remote access data processing.*

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

element address. In SNA, a value in the element address field of the network address identifying a specific resource within a subarea. See also *subarea address.*

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.

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, (REX), virtual route.*

flow control. In SNA, the process of managing the rate at which data traffic passes between components of the network. The purpose of flow control is to optimize 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 nodes, nor leave the receiver waiting for more message units. See also *pacing, session-level pacing, virtual route (VR) pacing.*

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.

function management (FM) 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 at the session partner and control the way that 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, a program or device).

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).

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.

intermediate routing function. In SNA, a path control capability within a subarea intermediate function.
A path control capability within 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.

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 also *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).

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).

link station. In SNA, the combination of hardware and software that allows a node to attach to and provide control for a link.

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.

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 the system services control points (SSCPs). An LU is capable of supporting 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).*

LU. Logical unit.

LU-LU session. In SNA, 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.

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.*

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).

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

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 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.*

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 *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 name. In SNA, the symbolic identifier by which end users refer to a network addressable unit (NAU), a link station, or a link.

network operator. In SNA, a person or program responsible for controlling the operation of all or part of a network.

network services (NS). In SNA, the services within network addressable units (NAUs) that control network operation via SSCP-SSCP, SSCP-PU, and SSCP-LU sessions.

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, cluster controllers, or terminals. Nodes can vary in routing and other functional capabilities. See also *node type, peripheral node, subarea node.*

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, session-level pacing, virtual route (VR) pacing.*

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 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 (NAU) 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.

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).

PC. Path control.

peripheral link. In SNA, a link that connects a peripheral node to a subarea node. See also *route extension (REX).*

peripheral 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.

peripheral 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 PU, physical unit type, subarea PU.*

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.

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.*

remote access data processing. Data processing in which certain portions of the input/output functions are situated in different places and connected by transmission facilities. Contrast with *distributed data processing.*

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 FM header, 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. (1) In SNA, a message unit that acknowledges receipt of a request; a response consists of a response header (RH) and optionally a response unit (RU). (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.

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 comprise the portion of a path between a subarea node and a network addressable unit (NAU) in an adjacent peripheral node.

routing. 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.

RU. Request/response unit.

SDLC. Synchronous Data Link Control.

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.*

Note: For routing purposes, each session is identified by the network (or local) addresses 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). Contrast with *session deactivation.*

session deactivation. In SNA, the process of exchanging a session-deactivation request response between network addressable units (NAU). Contrast with *session activation.*

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 that the receiver can process them. See also *pacing, virtual-route (VR) pacing.*

session limit. In SNA, the maximum number of concurrently active LU-LU sessions a particular logical unit (LU) can support.

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 partner. In SNA, one of the two network addressable units 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 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.*

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.

share limit. In SNA, the maximum number of control points that can concurrently control a network resource.

single-domain network. In SNA, a network with one system services control point (SSCP). Contrast with *multiple-domain network.*

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, boundary function components, and the path control network.

SNA node. In SNA, a node that supports SNA protocols.

SSCP. System services control point.

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-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.

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, links, and adjacent link stations (in attached peripheral or subarea nodes) 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 LU. In SNA, a logical unit in a subarea node.

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 *peripheral node.*

subarea PU. In SNA, a physical unit in a subarea node.

Synchronous Data Link Control (SDLC). In SNA, a discipline for managing synchronous, code-transparent, serial-by-bit information transfer over a link connection. Transmission exchanges may be duplex or half duplex over switched or nonswitched links. The configuration of the link connection may be point-to-point, multipoint, or loop. SDLC conforms to subsets of the Advanced Data Communication Control Procedures (ADCCP) of the American National Standards Institute and High-level Data Link Control (HDLC) of the International Standards Organization.

system services control point (SSCP). A focal point within an SNA network for managing the configuration, coordinating network operator and problem determination requests, and providing directory 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). In 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 Display Station, 3767 Consumer Transaction Facility, 3614 Communications Terminal, and 3624 Consumer Transaction Facility.

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 header (TH). In SNA, control information, optionally followed by a basic information (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 (PIU).*

user-application network. 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 communication 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 intra-node sessions. A virtual route between distinct subarea nodes imposes a transmission priority on the underlying explicit route, provides flow control through virtual-route pacing, and provides data integrity through sequence numbering of path information units (PIUs). See also *explicit route, path, route extension (REX).*

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.*

Index

Systems Network Architecture
Concepts and Products

Order No. GC30-3072-1

READER'S COMMENT FORM

This form may be used to communicate your views about this publication. They will be sent to the author's department for whatever review and action, if any, is deemed appropriate. Comments may be written in your own language, use of English is not required.

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.

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 would like a reply, complete the following (Please Print):

Your Name ______________________ Date __________
Company Name ______________________
Department ______________________
Street Address ______________________
City ______________ State __________ Zip Code ______

Thank you for your cooperation. No postage stamp is necessary if mailed in the U.S.A. (Elsewhere, an IBM office or representative will be happy to forward your comments.)

Note: Staples can cause problems with automated mail sorting equipment.
Please use pressure sensitive or other gummed tape to seal this form.

Cut or Fold Along This Line

GC30-3072-1

ADDITIONAL COMMENTS:

Cut or Fold Along Line

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

IBM®

SNA Concepts and Products (File No. S370/4300-30) Printed in U.S.A. Order No. GC30-3072-1