IBM

International Technical Support Centers

APPN Architecture and Product Implementations Tutorial

GG24-3669-01

APPN Architecture and Product Implementations Tutorial

Document Number GG24-3669-01

June 1992

International Technical Support Center
Raleigh

Take Note

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

Second Edition (June 1992)

This edition applies to IBM Advanced Peer-to-Peer Networking Architecture and Product Family Overview.

Order publications through your IBM representative or the IBM branch office serving your locality. Publications are not stocked at the address given below.

An ITSC Technical Bulletin Evaluation Form for readers' feedback appears facing Chapter 1. If this form has been removed, comments may be addressed to:

IBM Corporation, International Technical Support Center
Dept. 985, Building 657
P.O.Box 12195
Research Triangle Park, NC 27709

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

Abstract

This document describes Advanced Peer-to-Peer Networking within the Systems Network Architecture. It provides a tutorial on the APPN architectural functions, the relationship between these functions, and a summary of implementations in various products.

The document is intended for system engineers, system planners, system programmers, and network administrators who need to know the APPN functions, the APPN node types, and their interrelation. A basic knowledge of networking concepts and terminology is assumed.

CO, EN (190 pages)

Contents

Figures

Special Notices

This document is intended for system engineers, system planners, system programmers, and network administrators who need to know the APPN functions, the APPN node types, and their interworking. This publication is not intended as the specification of any programming interfaces that are provided by IBM APPN products. See the PUBLICATIONS section of the IBM programming Announcement for IBM APPN products for more information about what publications are considered to be product documentation.

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. Any reference to an IBM product, program, or service is not intended to state or imply that only IBM's product, program, or service may be used. Any functionally equivalent program that does not infringe any of IBM's intellectual property rights may be used instead of the IBM product, program or service.

Information in this book was developed in conjunction with use of the equipment specified, and is limited in application to those specific hardware and software products and levels.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Commercial Relations, IBM Corporation, Purchase, NY 10577.

The information contained in this document has not been submitted to any formal IBM test and is distributed AS IS. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environments do so at their own risk.

The following terms, which are denoted by an asterisk (*) in this publication, are trademarks of the International Business Machines Corporation in the United States and/or other countries:

AS/400
ES/3090
IBM
OS/2
OS/400
PS/2
SAA
Systems Application Architecture
VTAM

The following terms, which are denoted by a double asterisk (**) in this publication, are trademarks of other companies.

DOS is a registered trademark of Microsoft Corporation.

Preface

This document is intended for system engineers, system planners, system programmers, and network administrators who need to know the APPN functions, the APPN node types, and their interrelations. A basic knowledge of networking concepts and terminology is assumed.

The purpose of Advanced Peer-to-Peer Networking is to provide a network that is easy to use, has decentralized network control, but centralized network management, allows for arbitrary topologies, has connection flexibility and continuous operation, and requires no specialized communication hardware.

This document describes Advanced Peer-to-Peer Networking within the Systems Network Architecture. It provides a tutorial on the APPN architectural functions, the relationship between these functions, and a summary of implementations in various products.

The document is organized as follows:

- Chapter 1. "APPN Overview"

  This chapter gives a short introduction to Advanced Peer-to-Peer Networking, positions APPN towards LEN and SNA, and introduces the basic terminology used within APPN.

- Chapter 2. "T2.1 Node Structure"

  This chapter provides details about the structure of T2.1 nodes, gives an overview of control point functions, an introduction to logical units, a description of intermediate session routing, path control and data link control.

- Chapter 3. "Address Space Manager"

  This chapter addresses the functions performed by the address space manager component of a T2.1 control point. A description is given of the TG "address space" concept, the assignment of local-form session identifiers, and the process of BIND segmenting and reassembly.

- Chapter 4. "Configuration Services"

  This chapter addresses the functions performed by the configuration services component of a T2.1 control point. A description is given of transmission groups, links, link stations, connection networks, and virtual routing nodes.

- Chapter 5. "Topology and Routing Services"

  This chapter addresses the functions performed by the topology and routing services component of a T2.1 control point. A description is given of topology databases, resource characteristics, class of service, routing trees and route computation. The chapter also shows how the use of a route selection control vector enables session data and session setup data to be routed through an APPN network.

- Chapter 6. "Directory Services"

  This chapter addresses the functions performed by the directory services component of a T2.1 control point. Explained is how network resources are located, how T2.1 nodes register local resources, and how the use of directory servers and the nonverify function reduce session setup traffic.

- Chapter 7. "Session Services"

  This chapter addresses the functions performed by the session services component of a T2.1 control point. It is explained how fully qualified procedure correlation IDs are generated. Details are given about CP-CP and LU-LU session establishment. The last section of this chapter describes session services extensions.

- Chapter 8. "Network Management"

  This chapter addresses the functions within SNA network management services which have relevance to APPN.

- Chapter 9. "Border Node"

  This chapter addresses the border node function. The reason for the introduction of the border node function is explained, and the difference between border nodes R1 and R2 is depicted.

- Chapter 10. "APPN Implementations"

  This chapter gives some detail as to how APPN functions have been implemented on VTAM and NCP, AS/400, PS/2 and 3174. At the end of the chapter an overview is given of optional functions within APPN and the support for these functions on the four platforms mentioned.

- Appendix A. "APPN VTAM"

  In the last chapter an extensive description is given of the APPN support within VTAM. For readability some overlap exists between this chapter and previous chapters. At the end of the chapter is a discussion of how VTAM supports dependent LUs within an APPN network.

Related Publications

The following publications are considered particularly suitable for a more detailed discussion of the topics covered in this document.

General Publications

GA27-3093	*SDLC General Information Manual*
GA27-3136	*Systems Network Architecture Formats*
GA27-3345	*X.25 Interface for Attaching SNA Nodes to Packet Switching Networks*
GA27-3918	*3174 Planning Guide Configuration Support C*
GC30-3073	*SNA Technical Overview*
GC30-3084	*SNA Transaction Programmer's Reference for LU Type 6.2*
SC30-3098	*SNA Distribution Services Reference.*
SC30-3346	*SNA Management Services Reference*
SC30-3383	*LAN Technical Reference*
SC30-3422	*SNA Type 2.1 Node Reference*
SC31-6807	*SNA File Services Reference*
SC31-6808	*SNA LU Version 6 Reference: Peer Protocols*
SC41-8188	*AS/400 Communications: APPN Guide*
SC52-1110	*NS/2 Installation and Network Administrator's Guide*
G326-0161	*Extended Services for OS/2 Information and Planning Guide*

ITSC Publications

GG24-3433	*Enterprise Networking with SNA Type 2.1 Nodes*
GG24-3662	*NS/2 Installation, Customization, and Operations APPN for OS/2*
GG24-3702	*3174 APPN Implementation Guide*
GG24-3717	*AS/400 APPN with PS/2 APPN, 3174 APPN, 5394, and Subarea Networking*

Non-IBM Publications

ISBN 0-13-166836-6 *Computer Networks*
by Andrew S. Tanenbaum
Prentice-Hall International Editions

Acknowledgements

This publication is the result of a residency conducted at the International Technical Support Center, Raleigh.

The author of the second edition of this document is:

Vokke Kreuk	IBM The Netherlands

The advisor for this project was:

Peter Lenhard	International Technical Support Center, Raleigh

The authors of the first edition of this document are:

Bernd Kampmann	IBM Germany
Martin Numan	IBM The Netherlands

The advisor for the first edition of this document was:

Paul Berdowski	IBM The Netherlands

Thanks to the following people for the invaluable advice and review comments provided in the production of both versions of this document:

Gary Schultz	IBM Research Triangle Park
Michael Allen	IBM Research Triangle Park
Marilyn Beumeier	IBM Research Triangle Park
Rachel Bodner	IBM Research Triangle Park
Ray Boyles	IBM Research Triangle Park
David Bryant	IBM Research Triangle Park
Mark Cossak	IBM Rochester
Jim Fletcher	IBM Research Triangle Park
Doyle Horne	IBM Research Triangle Park
Larry Plank	IBM Rochester
Wolfgang Singer	IBM Austria
Shawn Walsh	International Technical Support Center, Raleigh

Fold and Tape **Please do not staple** Fold and Tape

NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

IBM International Technical Support Center
Department 985, Building 657
P.O. BOX 12195
RESEARCH TRIANGLE PARK NC
USA 27709-2195

Fold and Tape **Please do not staple** Fold and Tape

ITSC Technical Bulletin Evaluation RED000

APPN Architecture and
Product Implementations
Tutorial

Publication No. GG24-3669-01

Your feedback is very important to us to maintain the quality of ITSO redbooks. **Please fill out this questionnaire and return it via one of the following methods:**

- Mail it to the address on the back (postage paid in U.S. only)
- Give it to an IBM marketing representative for mailing
- Fax it to: Your International Access Code + 1 914 432 8246

Please rate on a scale of 1 to 5 the subjects below.
(1 = very good, 2 = good, 3 = average, 4 = poor, 5 = very poor)

Overall Satisfaction ____

Organization of the book ____
Accuracy of the information ____
Relevance of the information ____
Completeness of the information ____
Value of illustrations ____

Grammar/punctuation/spelling ____
Ease of reading and understanding ____
Ease of finding information ____
Level of technical detail ____
Print Quality ____

Please answer the following questions:

a) Are you an employee of IBM or its subsidiaries? Yes____ No____
b) Are you working in the USA? Yes____ No____
c) Was the bulletin published in time for your needs? Yes____ No____
d) Did this bulletin meet your needs? Yes____ No____
If no, please explain:

____

____

What other Topics would you like to see in this Bulletin?

____

____

What other Technical Bulletins would you like to see published?

____

Comments/Suggestions: (THANK YOU FOR YOUR FEEDBACK!)

________________________ ________________________
Name Address

________________________ ________________________
Company or Organization

________________________ ________________________
Phone No.

Chapter 1. APPN Overview

This chapter introduces the basic terminology used with Advanced Peer-to-Peer Networking and puts it in context without going into detail.

1.1 LEN and APPN

A network can be very simple. For example, two PS/2s* connected by a telephone line, as shown in the figure below.

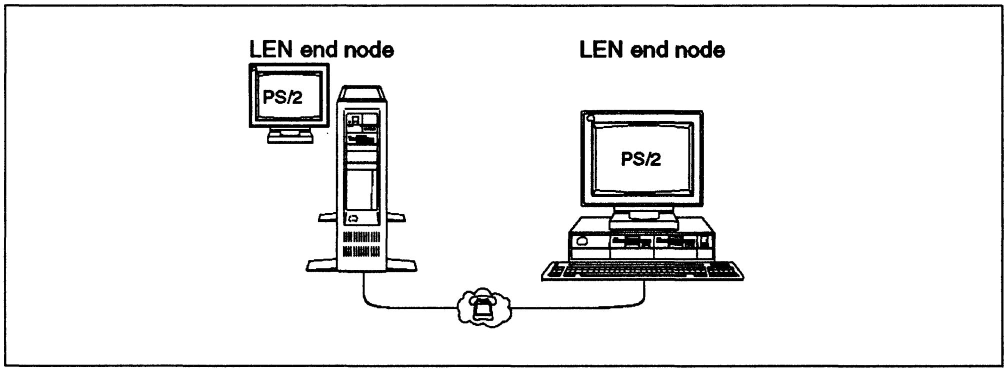

Figure 1. Two PS/2s Forming a LEN Connection

The purpose of connecting these two systems is to exchange data between two end users. An end user can be a person working with this system, a program running on the system, or a printer controlled by the system.

The end user gains access to the network through the logical unit (LU). Before the two LUs are able to exchange data, they must start an LU-LU session. For program-to-program communication, this session would typically be an LU 6.2 session.

In the case above, when the two systems (PS/2s) establish a Low Entry Network (LEN) connection, the two connecting systems are known as ***LEN end nodes***. Using the architectural terms, the configuration above could be drawn as shown in Figure 2.

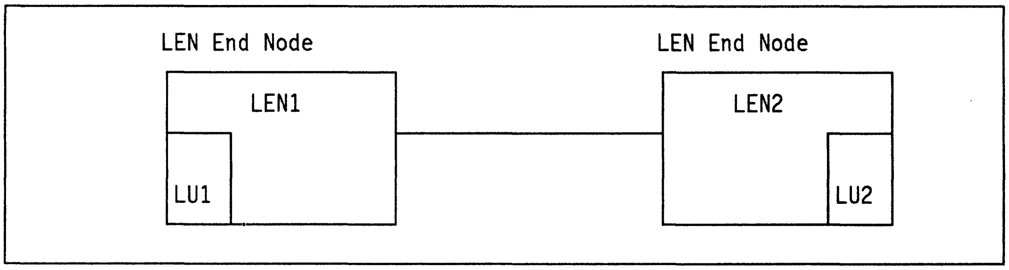

Figure 2. The Basic LEN Connection

Several systems can be configured as LEN end nodes, such as VTAM* and NCP, AS/400* and PS/2.

LEN end nodes provide the minimum functions required to:

- Provide a connection between LEN1 and LEN2
- Establish a session between the LUs named LU1 and LU2
- Transport data

The relation between LEN end nodes is truly peer-to-peer. Either side may activate a connection or start a session to the partner. It should be noted that according to the architecture, there are only *two* adjacent nodes involved in a LEN connection. That is, no matter how many nodes are actually in the network, the LEN connection only recognizes two nodes.

Obviously, there must be functions in addition to LEN to build a network with more than two nodes. One of these functions is the capability to act as an intermediate node; that is, the node can receive data that is not for itself and can pass it on to the destination node. This principle is shown in Figure 3.

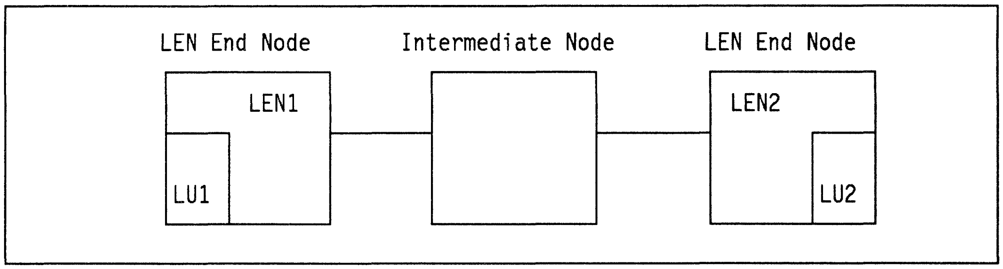

Figure 3. LEN End Nodes Connected to an Intermediate Node

According to the LEN architecture, the relation between LEN end nodes is always a "*two*-node peer relationship." Therefore LUs residing on non-adjacent LEN nodes can only establish sessions and exchange data, because the intermediate node presents itself as a LEN node owning all LUs residing on non-adjacent nodes. As seen from LEN1, the intermediate node is just a normal LEN end node, and LEN2 is not visible at all from LEN1. For LEN1, the LU named LU2 seems to be in the intermediate node.

VTAM and NCP support the LEN end node function and also provide intermediate routing between LEN end nodes. Figure 4 on page 3 gives an example of this configuration with VTAM on an ES/3090* as intermediate node.

The functions of LEN nodes are limited, for example the nodes are not able to exchange topology and configuration data. Additional capabilities are required to reduce the number of definitions and the maintenance effort when building larger networks. For this purpose the Advanced Peer-to-Peer Networking (APPN) architecture has been developed, which has been announced and published as the latest extension to SNA (Systems Network Architecture).

APPN architecture distinguishes two types of nodes:

1. ***APPN End Node***

   The APPN end node is similar to a LEN end node, except that the control point (CP) of the end node exchanges information with the CP in the adjacent network node. The communication over the CP-CP sessions reduces the requirement for network definitions, and thus makes installation and maintenance of the network easier.

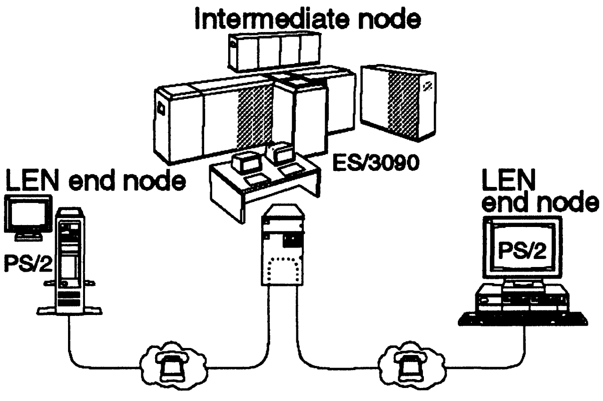

Figure 4. VTAM/NCP Providing the Intermediate Routing Function for LEN End Nodes

2. ***APPN Network Node***

   The APPN network node has intermediate routing functions and provides network services to either APPN or LEN end nodes which attach to the network node. An APPN network node establishes CP-CP sessions with its adjacent APPN network nodes to exchange network topology and resource information. The CP-CP sessions to adjacent APPN end nodes are optional and are required only for APPN end nodes for which the APPN network node provides network services (such as locating resources in the APPN network).

APPN architecture describes the connection of LEN end nodes to APPN network nodes or APPN end nodes.

Figure 5 shows the basic form of an APPN network and gives an example of the services provided by the APPN network node. When LU1 requests a session with LU3, the network node will locate the partner LU and assist in establishing the session.

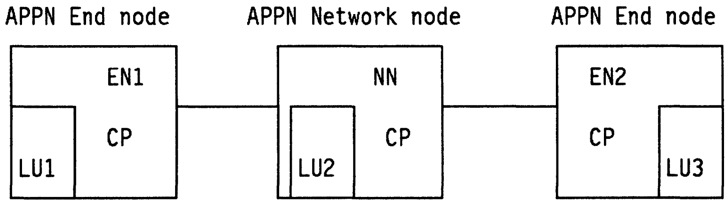

Figure 5. Advanced Peer-to-Peer Networking with Three Nodes

Figure 5 shows the basic form of an APPN network, however, APPN networks can be much more complex. APPN does not limit the number of nodes in an APPN network nor does it explicitly limit the number of intermediate APPN network nodes through which LU-LU sessions are routed. One restriction exists however: the length of the route selection control vector (RSCV) describing a physical session path is limited to 256 bytes, which gives a maximum of between 9 and 30 intermediate nodes on one session path depending on the length of the specific CP names and TG names.

Figure 6 on page 4 shows a backbone structure of APPN network nodes to which end nodes connect. The APPN nodes communicate through CP-CP sessions which are established between adjacent nodes. User sessions can be established from any LU to any LU.

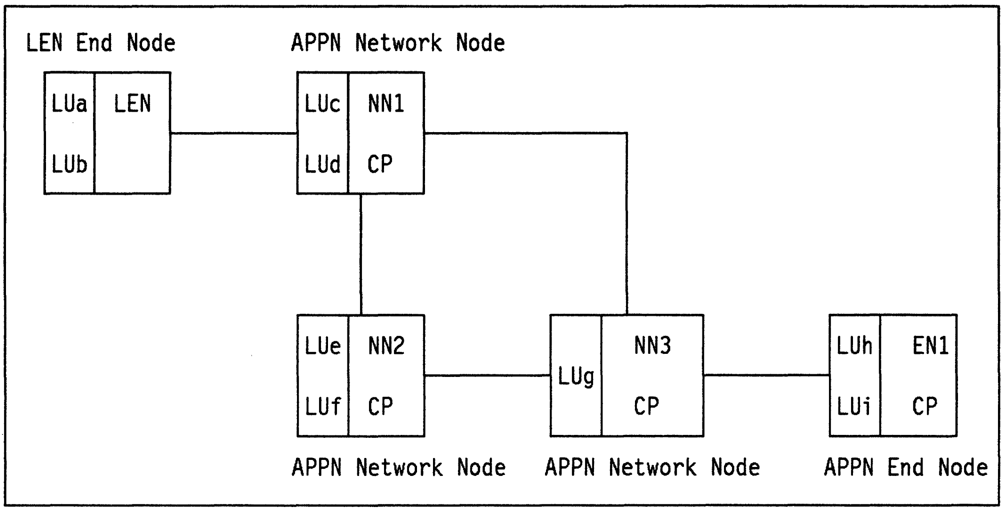

Figure 6. APPN Network with Different Node Types

While the previous figure showed the architectural node types used in the network, the next picture (Figure 7) shows a variety of products, such as VTAM and NCP, AS/400, PS/2, and 3174, connecting through different link protocols.

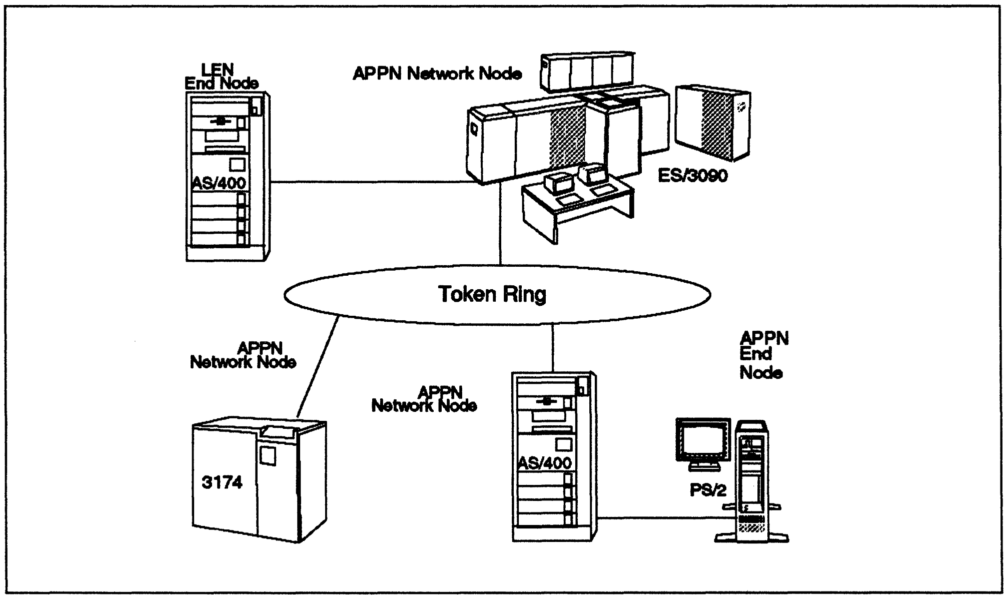

Figure 7. Advanced Peer-to-Peer Networking

Figure 7 depicts a VTAM host, an AS/400, and a 3174 configured as APPN network nodes, a PS/2 configured as an APPN end node and a second AS/400 configured as a LEN end node.

Note: A VTAM configured as a network node, together with all its owned NCPs, is called a *composite network node* (CNN). Within the composite network node, subarea protocols are used (see Figure 8) but to the other APPN or LEN nodes the CNN gives the appearance of an APPN network node. For details about the APPN implementation of VTAM see "APPN VTAM" on page 169.

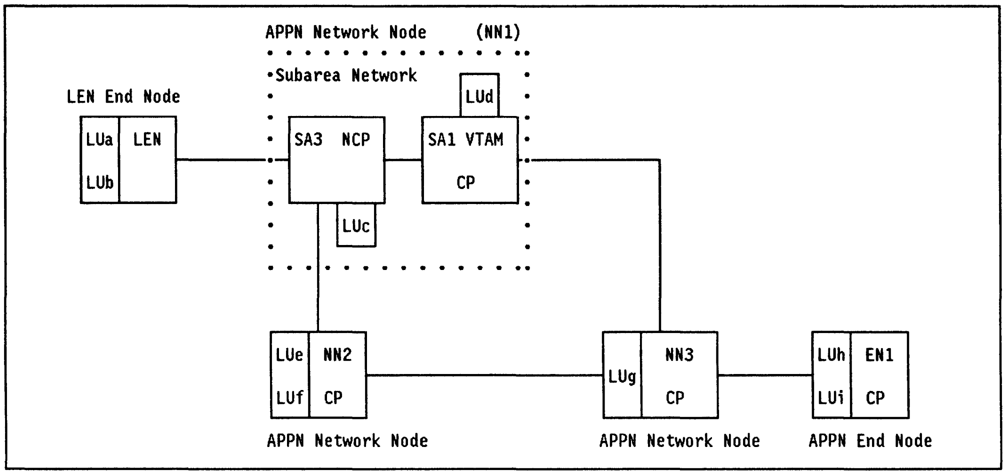

Figure 8. Composite Network Node with APPN Network Node Appearance

You have seen that the APPN architecture defines several types of nodes and that the CPs of these nodes have different scopes of function. The node types are defined more precisely later in this chapter. The CP functions are covered in several chapters from page 18 to page 159. The internal implementation of the functions defined by the architecture may be different in different products. Chapter 8, "APPN Implementations" on page 159, will provide details.

1.2 Names

Resource naming is important as it allows end users to start sessions without knowing the exact location of different resources within the network.

The Network Accessible Unit

In an APPN network, all components that can establish sessions with one another are called network accessible units. Examples are CPs and LUs. The term NAU was previously used as an abbreviation for "network *addressable* units." The terminology has changed with APPN, and now NAUs are represented by names rather than by addresses.

Within an APPN network, the names of network accessible units must be unique. A consistent naming convention must be put into place to ensure the uniqueness of the names. To make the administering of resource names easier, "the network" can be divided into partitions.

Network Identifiers

Each partition of "the network" is given a unique network identification (NETID). NETIDs are 1 to 8 bytes long. The NETID is used throughout SNA, both in the subarea part and the APPN part of a network. Because names of LUs and CPs have to be unique only within the scope of a NETID, they can be assigned and administered independently for each distinct partition of the network.

Registering can help network administrators ensure the uniqueness of the NETID they use. IBM* provides a worldwide registry for network IDs; information on the registration process can be obtained from your IBM representative.

IBM-registered NETIDs have an 8-byte name with the following structure:

cc is the country code (according to ISO Standard 3166)

eeee is the enterprise code (unique within a country)

nn is the network suffix code (unique within one enterprise).

Network Names

A network name is an identifier of a network resource. Each CP, LU, link, and link station in an SNA network has a network name. The network names are assigned through system definition. In an APPN node, the system definition is done using the node operator facility (NOF).

Network-Qualified Names

A network-qualified name identifies both the resource and the network in which the resource is located. It is a concatenation of the network ID and the network name of the resource, for example, names NETA.LUA and NETB.LUA refer to different entities.

1.3 Addresses

The addresses used in SNA are either network or local addresses. Network addresses uniquely identify a resource throughout the subarea network. Local addresses uniquely identify a session on a link. APPN uses local addresses.

Addresses are used for routing. Routing in an SNA network is done by a combination of two things:

- Information carried in the transmission header of the message
- Information stored in the intermediate node.

In an APPN network, routing information is session oriented. The transmission header carries session identifiers that are locally defined for each pair of adjacent routing nodes and are only *temporarily* assigned. They are assigned at session initiation, and released when the session ends. The session initiation request (BIND) carries routing information about the full session path that determines the sequence of links used from origin to destination. The local session identifier stored in each intermediate node in a session path is contained in a session connector and only kept during the life of the session.

The session identifier is associated with

- A particular session
- A transmission group (link) between two nodes.

Figure 9 on page 7 shows a session between two LUs, LU1 and LU2, residing on two non-adjacent APPN end nodes. The session data is routed through two intermediate network nodes. The session can be thought of as a sequence of three session stages or "hops" with a distinct session identifier assigned to each session stage.

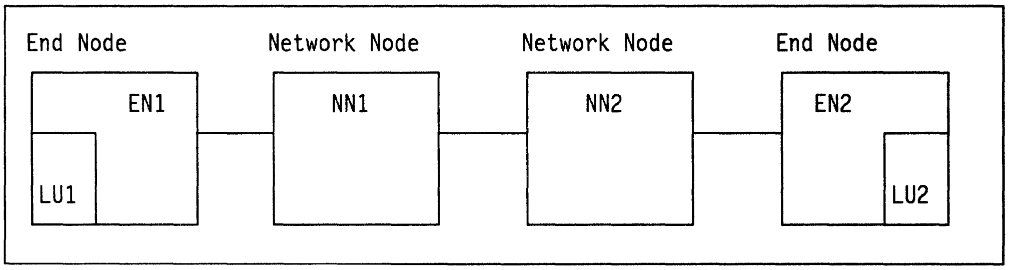

Figure 9. Session with Several Session Stages

Session identifiers vary at different session stages, which is why they are called local-form session identifiers (LFSID). The LFSID is set up during session establishment by the address space manager component of the CP and assigned for the "lifetime" of an LU-LU (or CP-CP) session. Details may be found in "Address Space Manager" on page 33.

Each session is uniquely identified by a network-unique identifier, the fully qualified procedure correlation ID (FQPCID), described in "Fully Qualified Procedure Correlation Identifier (FQPCID)" on page 112.

Note: Traditional subarea networking uses local *and* network addresses. The local addresses are used between peripheral nodes and the boundary functions of VTAM and NCP; the network addresses are used when routing data between subarea nodes and bear no relation to specific sessions.

1.4 Domains

A domain is an "area of control." A domain in an APPN network consists of the control point in a node and the resources controlled by the control point. Consequently, all APPN networks are multi-domain networks.

Though all APPN nodes are peers with respect to session initiations and do not rely on other nodes to control their resources, APPN end nodes and LEN end nodes use the services of network nodes. The domain of an APPN end node or LEN end node contains the node's own (local) resources. The domain of an APPN network node contains its local resources *and* the resources of those nodes that use the network node's services. Thus, the domains of the APPN end nodes and LEN end nodes are included in the domains of their respective network node servers.

Note: Within traditional subarea networking, a domain is the portion of the network managed by a VTAM System Services Control Point (SSCP). Within this document, when using the term domain, we refer to an APPN domain unless explicitly stated differently.

1.5 Node Types

Before and after its announcement in 1986 the LEN end node received many names. Some of the names for the LEN end node that are found in literature are:

LEN end node
LEN node
Peer node
PU type 2.1
PU 2.1
SNA PU 2.1
SNA Type 2.1 node
Type 2.1
T2.1
etc...

All the names mentioned above are synonyms for *LEN end node*. They all refer to the same function set. With the APPN extensions to SNA, two other types of T2.1 nodes, *APPN end node* and *APPN network node*, have been introduced. Throughout this document we will use the term ***T2.1 node*** to refer to any of these three types of nodes, and use the term ***APPN node*** when referring to either an APPN network node or an APPN end node.

1.5.1 APPN Network Node

An APPN network node provides distributed directory and routing services for all LUs that it controls. These LUs may be located on the APPN network node itself or on one of the adjacent LEN or APPN end nodes for which the APPN network node provides network node services. Jointly, with the other active APPN network nodes, an APPN network node is able to locate all destination LUs known in the network.

As part of the ***nonverify function*** an APPN network node may cache resource information and connection information of the APPN end node on which the resource resides within the APPN network node's directory database. If the network node server of the origin LU indicates in the Locate request that the end node's resources need not be verified, this allows the network node server of the destination LU to immediately respond to the Locate request, without forwarding the request to the APPN end node first.

A facility known as ***central resource registration*** allows an APPN network node to register its resources at a central directory server. Once a resource is registered, all APPN network nodes can locate the resource by querying the central directory server instead of using a broadcast search, thus improving network search performance during session establishment.

After the LU is located, the APPN network node is able to calculate the route between origin and destination LU according to the required class of service. All network nodes exchange information about the topology of the network. When two adjacent network nodes establish a connection, they exchange information about the network topology as they know it. In turn, each network node broadcasts this network topology information to other, active and adjacent, network nodes with which it has CP-CP sessions.

Alternatively, if the connection between network nodes is deactivated, then each network node broadcasts this change to all other, active and adjacent, network nodes. An APPN network node that is taken out of service will be declared inactive and, after some time, removed from the topology information in all network nodes together with its routing capabilities to other nodes.

The APPN network node is also capable of routing LU-LU sessions through its node from one adjacent node to another adjacent node. This function is called intermediate session routing.

1.5.2 APPN End Node

An APPN end node provides limited directory and routing services for LUs local to it. The APPN end node can select an adjacent APPN network node and request this network node to be its ***network node server***. If accepted by the network node, the APPN end node may register its local resources at the network node server. This allows the network node server to intercept Locate search requests for resources that are located on the APPN end node and pass the request on to the APPN end node for verification.

As part of the ***nonverify function***, an APPN end node may register its connections to adjacent APPN nodes at its network node server. This allows the network node server to immediately respond to Locate search requests for LUs residing on the APPN end node, without forwarding the request to the APPN end node first.

The APPN end node automatically forwards all Locate search requests for resources unknown to the APPN end node, to its network node server. The APPN network node uses its distributed directory and routing facilities to locate the LU (via directed or broadcast searches or its non-verify function) and calculates the optimal route starting at the APPN end node toward the destination LU.

The APPN end node may have active connections to multiple adjacent network nodes; however, only one of these network nodes at a time can act as its network node server. The APPN end node selects its network node server by establishing CP-CP sessions with an adjacent APPN network node.

On APPN network nodes, the APPN end nodes are categorized as either ***authorized*** or ***unauthorized***. An authorized APPN end node may send registration requests to register local network accessible resources at a network node server, a facility known as ***end node resource registration***, and may, in addition, request that these resources are registered with the central directory server. If during session establishment a network node server does not know where an LU is located, the network node server queries authorized APPN end nodes within its domain which have indicated they are willing to be queried for unknown resources. Network accessible resources on unauthorized nodes require explicit definition at the network node server as part of its system definition or dynamically by the network node server's operator. To avoid explicitly defining resources of authorized nodes at their network node server, the APPN end node should either register its resources or allow the network node server to query the APPN end node for unknown resources.

1.5.3 LEN End Node

A LEN end node provides peer-to-peer connectivity to other LEN end nodes, APPN end nodes, or APPN network nodes. A LEN end node requires that all network accessible resources, either controlled by the LEN end node itself or on other nodes, be defined at the LEN end node. LUs on adjacent nodes need to be defined with the control point name of the adjacent node. LUs on nonadjacent nodes need to be defined with the control point name of an adjacent network node, as LEN end nodes assume that LUs are either local or reside on adjacent nodes.

Unlike APPN end nodes, the LEN end node cannot establish CP-CP sessions with an APPN network node; therefore, a LEN end node cannot register resources at a network node server, nor can it request its network node server to search for a resource, or, to calculate the route between the LEN end node and the node containing a destination resource.

However, indirectly a LEN end node uses the distributed directory and routing services of an adjacent network node by predefining remote LUs, owned by nonadjacent nodes, with the CP name of an adjacent APPN network node. The session activation (BIND) request for that remote LU is sent by the LEN end node to the adjacent network node. The network node, in turn, automatically acts as the LEN end node's network node server, locates the actual destination of the LU, calculates the route to it, and uses this route to send the BIND to its final destination.

1.5.4 Other Node Types

In SNA, a node represents an endpoint of a link or a junction common to two or more links in a network. The LEN end node, APPN end node, and APPN network node are endpoints of a link. Each node has a distinct role in an APPN network.

Besides these node types you will find references in the APPN literature to other node types that are either synonyms for APPN nodes as seen from a subarea network, represent a specific junction in the network, or represent an APPN node with additional functions. The following list does not provide a complete list, but merely highlights the ones found when creating this document:

- Boundary and peripheral node
- Composite node
- Virtual routing node
- Border node

Boundary and Peripheral Node

Within traditional subarea SNA, the resources in a domain of a subarea SNA network are controlled through a hierarchical structure. The nodes that play a role in these networks are categorized as subarea and peripheral nodes. A good example of such an SNA network is a S/370 type mainframe running VTAM and a 3745 communication controller running the Network Control Program (NCP). Both VTAM and NCP are referred to as subarea nodes. The VTAM subarea node includes the control point function, hereafter called the System Services Control Point (SSCP). Like the APPN control point, the SSCP controls all the resources that are in its domain.

Attached to these subarea nodes, or also *boundary nodes*, are the *peripheral* nodes. The peripheral node is either a T2.0 or T2.1 node. The T2.0 node is a traditional hierarchical node that requires the support of an SSCP to establish sessions. The T2.1 node represents either a LEN end node, APPN end node, or APPN network node. Traditional subarea SNA allowed LEN connections only; CP-CP sessions could not be established between VTAM and the T2.1 nodes.

With the introduction of APPN VTAM, a VTAM or a composite network node (subarea network consisting of one VTAM and one or more NCPs) is able to present an APPN image to T2.1 devices. APPN VTAM allows CP-CP sessions with APPN nodes attached to the VTAM or NCP boundary function, to get full APPN connectivity. The term "peripheral" node has lost its value in a network which is truly peer-to-peer.

Composite Node

The term *composite node* is used in some publications to represent a group of nodes that appear as **one** T2.1 node to another T2.1 node. For example, a subarea network that consists of one VTAM host and one or more NCPs consists of multiple nodes, but when connected to an APPN node, appears as **one** logical T2.1 node.

A subarea composite node may appear as either a LEN end node or as an APPN network node. In the former case, the term composite end node is used; in the latter case the term composite network node (CNN) is used. To a CNN, see for example VTAM1/NCP in Figure 10, multiple VTAMs and NCPs may connect using subarea protocols. Session establishment is possible between any LU in the subarea network and any LU in the APPN network. The VTAM host to which APPN nodes attach, or the VTAM host owning the NCPs to which APPN nodes attach, must have implemented APPN VTAM as it is responsible for the conversion of subarea to APPN protocols and vice versa; other VTAMs within the subarea network may be backlevel VTAMs. From the viewpoint of the APPN nodes, LUs owned by VTAMs (for example, VTAM2 or VTAM3) other than the VTAM/NCP providing the boundary function are considered to reside on APPN end nodes.

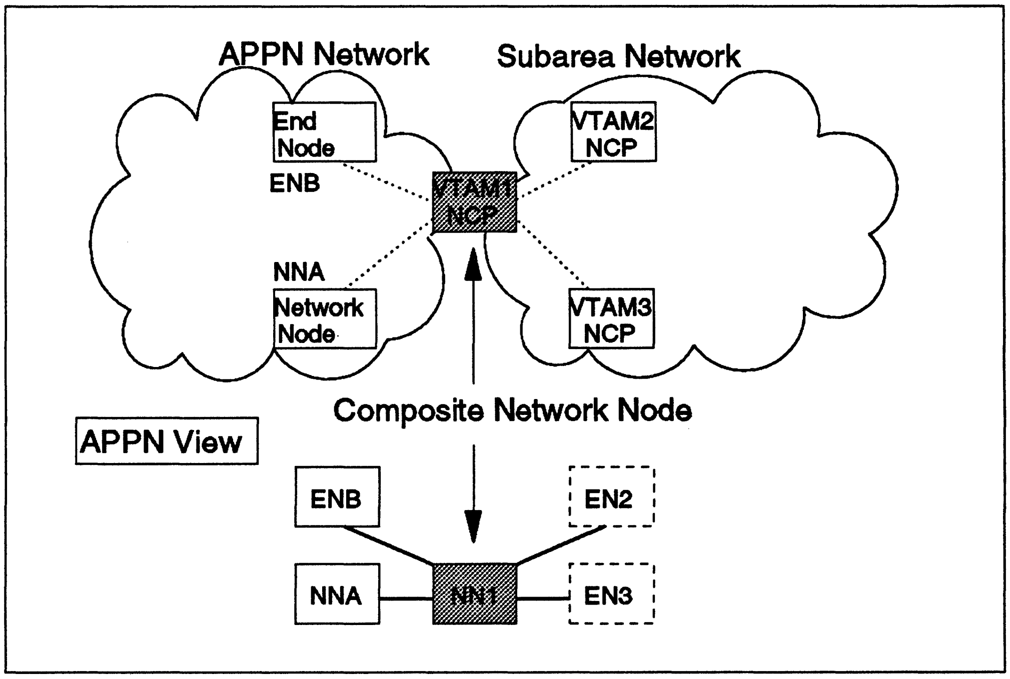

Figure 10. Composite Network Node

Note: Figure 10 shows the basic form of connecting APPN and subarea networks using a composite network node. For more details see "APPN VTAM" on page 169.

Virtual Routing Node

APPN allows APPN nodes to reduce the addressing information stored at each node connected to a shared-access transmission facility (SATF), such as token-ring, by allowing each node to define a virtual routing node (VRN) to represent its connection to the shared facility and all other nodes similarly configured. The SATF and the set of nodes having defined a connection to a common virtual routing node are said to comprise an ***connection network***.

A virtual routing node (VRN) is not a node, but it is a way to define an APPN node′s attachment to a shared-access transport facility. It reduces end node definition requirements by relying on the network node server to discover the common connection and supply necessary link-level signaling information as part of the regular Locate search process; LU-LU session data can then be routed directly, without intermediate node routing, between APPN nodes attached to the SATF. For more information see "Connection Networks and Virtual Routing Nodes" on page 49.

Border Node

APPN architecture does not allow two adjacent APPN network nodes to connect and establish CP-CP sessions when the network nodes do not share the same NETID. The border node function is an additional APPN function to overcome this restriction. It enables network nodes with different NETIDs to connect and allows session establishment between LUs in different NETID ***subnetworks***. Topology information flowing within different NETID subnetworks is prevented from crossing subnetwork boundaries.

The border node function can also be used to partition APPN networks with the same NETID, reducing the flow of topology updates and the storage requirements for the network topology database on network nodes in each of the network partitions. For more information see "Border Node" on page 151.

Chapter 2. T2.1 Node Structure

This chapter provides details on T2.1 nodes. The structure of a LEN end node, APPN end node, or APPN network node is shown in Figure 11 on page 14. The components in the figure are described below:

- Node Operator

  This component defines all information required by the node (for example, on links to adjacent nodes, and on LUs within its domain) and causes activation and deactivation of the node and its resources (for example, links). It may also query the status of a node's resources. See "Node Operator Facility (NOF)" on page 15 for more details.

- Node Operator Facility (NOF)

  The function of this component is to allow communication between the node operator and the control point (CP), intermediate session routing (ISR), and LUs. NOF initializes the CP and ISR components when the node is started. It also performs functions such as the following when requested to do so by the node operator:

  - Defining (creating) and deleting (destroying) LUs
  - Activating and deactivating links
  - Querying the CP and ISR for database and status information

- Application Transaction Program (TP)

  These programs communicate with other local or remote application transaction programs (TPs) to perform user-defined functions. Communication is accomplished by establishing conversations between TPs. Data is then exchanged between the TPs using an LU verb interface.

- Control Point (CP)

  The function of the CP is to manage the resources of the node. It creates the path control (PC) and data link control (DLC) components. The CP also manages session resources and provides facilities such as directories and topology information. The CP is created by NOF when the node is started.

- Intermediate Session Routing (ISR)

  The intermediate session routing (ISR) component is present only in an APPN network node. The primary function of ISR is to route session traffic destined to another node. ISR is created by NOF when the node is started.

- Logical Unit (LU)

  The LU serves as a port into the network for one or more application transaction programs. It establishes sessions with other LUs. Conversations are allocated on these sessions that allow communication between TPs.

- Path Control (PC)

  This component routes message units between LUs that reside within the same node or LUs, ISR, and CPs that reside in adjacent nodes. Messages received by path control from DLC are routed to the appropriate component (CP, LU, or ISR).

- Data Link Control (DLC)

  DLC provides the protocols necessary to assure reliable delivery of messages between link stations in adjacent nodes attached to a common transmission medium. DLC also controls the node attachment to various types of transmission media.

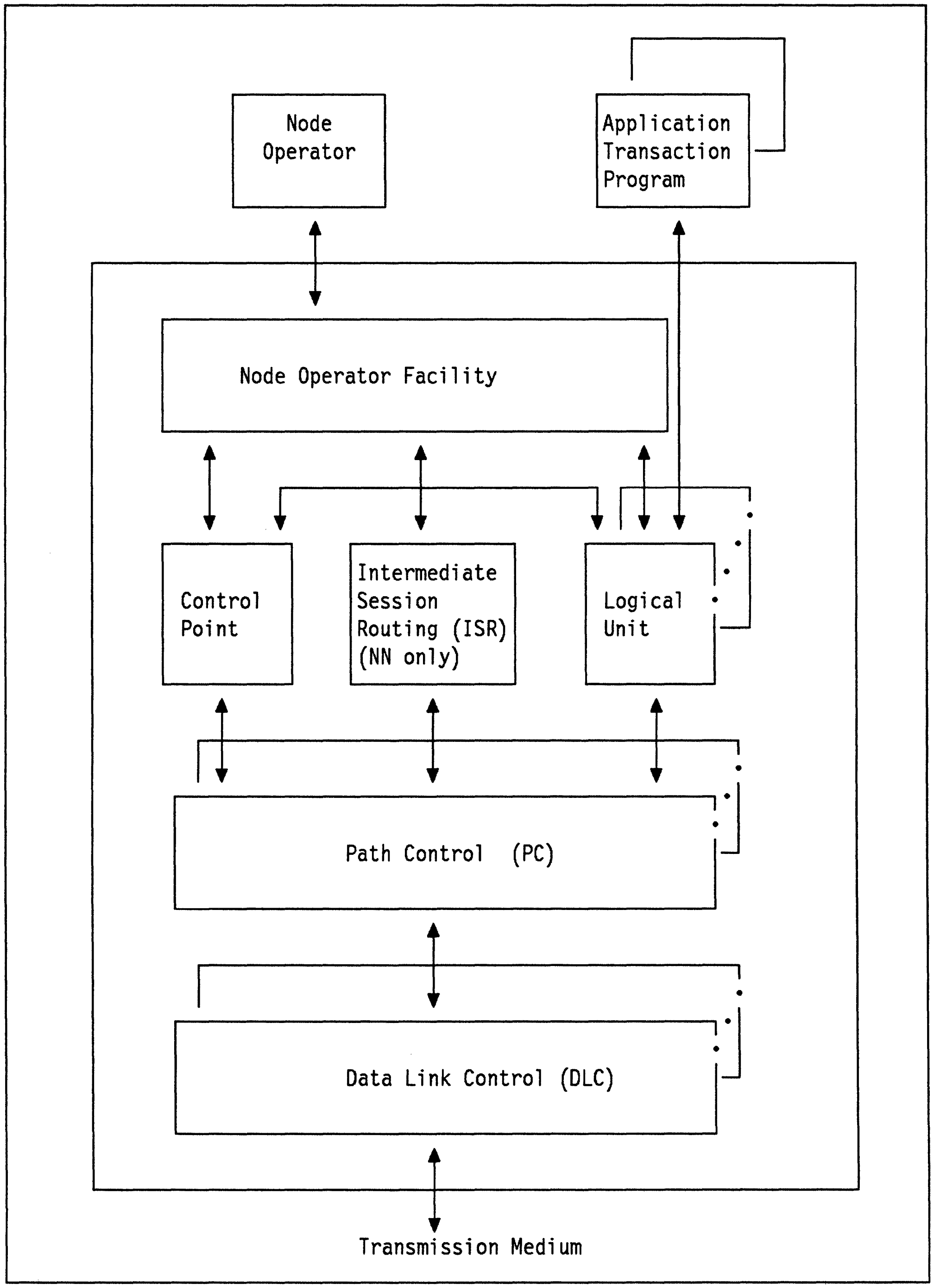

Figure 11. Structure of a T2.1 Node

2.1 Node Operator Facility (NOF)

The node operator facility provides an interface to the T2.1 node so that node operators can control the operation of the node. For example, the node operator may activate and deactivate link stations, define and delete LUs, query the control point about links and another node's resources, and receive diagnostic information.

The node operator can be:

- A human operator using an interactive display to issue commands
- A file containing a list of commands
- A transaction program handling remote requests from a partner transaction program in another node

All three types of node operators make use of a program within the system to interact with the node operator facility. Figure 12 on page 16 illustrates the different cases.

In the case of the ***human operator***, a system-specific dialog manager converts the information entered by the human operator into node operator commands and forwards the commands to the node operator facility. The dialog manager receives the command results from the node operator facility and shows those results on the display in a human readable form.

An implementation-specific file interpreter reads the ***command file***, converts the file records into node operator commands, and forwards the command to the node operator facility. The file interpreter logs or discards the command results after receiving them from the node operator facility. Command files are very useful when a series of commands must be repeated periodically (for example, a command file may be used to load the initial configuration when a node is started). The file interpreter may discard the results because the node operator facility will log commands and their results upon request.

Remote operations of the node are allowed by permitting node operator commands to be issued by ***transaction programs***. The local transaction program receives commands from a partner transaction program, converts the commands from the transaction-specific format into node operator commands, and issues the command. The local transaction program receives the command result from the node operator facility and forwards this result to the remote transaction program.

2.1.1 Node Initialization

At node initialization time the node operator facility creates and initializes the control point components in a controlled manner using installation-defined parameters. The node initialization is started with one or more of the following parameters:

- The type of T2.1 node
- The network-qualified name of the control point
- Whether negotiable link stations are supported
- Whether segment reassembly is supported
- Whether BIND reassembly is supported
- Whether the node's resources should be registered with its network node server (APPN end node only)

- Whether the node's resources should be registered with a central directory server (APPN network node only)
- Whether mapping of mode name to class of service and transmission priority is supported
- The name of the management services log file
- The name of the topology database file
- The name of the class-of-service (COS) definition file
- List of resource types (only LU resource type is currently supported) this node can be searched for by its network node server

In the chapters describing the various control point components, references will be made to the node's initialization parameters.

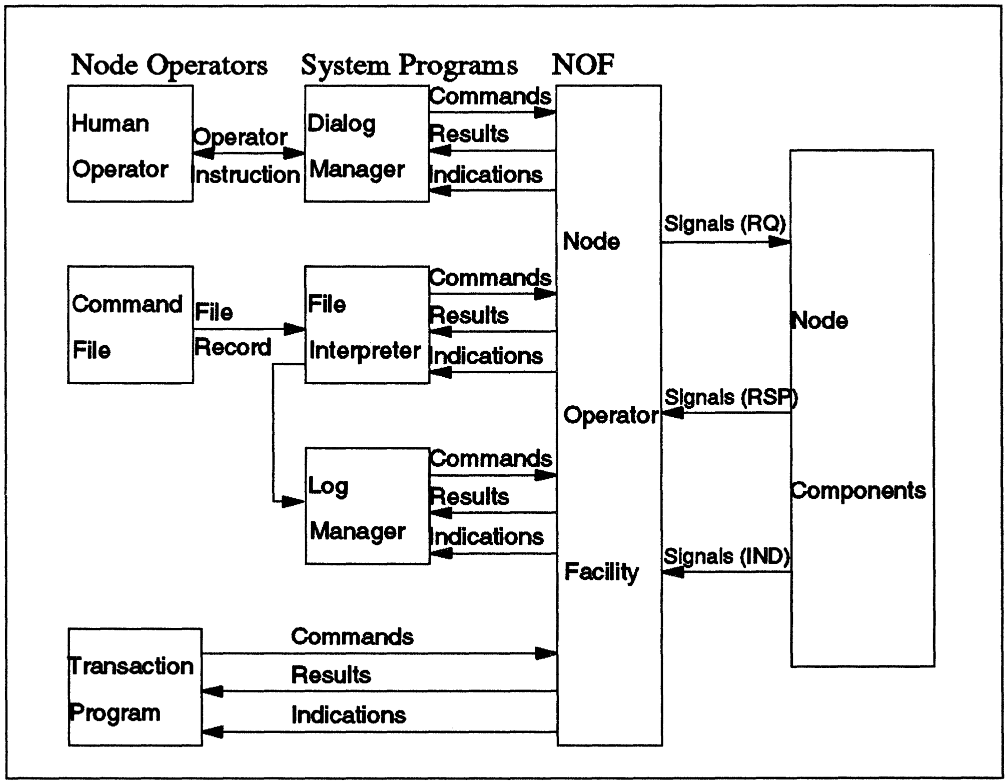

Figure 12. Node Operators. Interaction between node operators, the Node Operator Facility, and the node components.

The components that NOF creates and the order of their creation are:

1. Address Space Manager
2. Session Services
3. Directory Services
4. Configuration Services
5. Management Services
6. Topology and Routing Services
7. Session Connector Manager of ISR
8. Session Manager of the Control Point
9. Session Manager of each LU

2.1.2 Network Operator Facility Commands

The node operator facility interfaces with the control point components to define, change, or delete the node's resources, start and stop transmission groups, or obtain the status of resources. Where appropriate, references are made in this document to specific commands and their function. The available node operator facility commands are:

- Define/Delete adjacent node
- Define/Delete class of service (COS)
- Define/Delete connection network (CN)
- Define/Delete directory entry
- Define/Delete data link control instance
- Define/Delete link station
- Define/Delete local LU
- Define/Delete mode
- Define/Delete partner LU
- Define/Delete port
- Define/Delete TP
- Initialize/Change/Reset session limit
- Query class of service (COS)
- Query connection network (CN)
- Query data link control instance
- Query link station
- Query port
- Query statistics
- Start node
- Start TP
- Start/Stop data link control instance
- Start/Stop link station
- Start/Stop port

For more information on the node operator facility commands see Chapter 3, "Node Operator Facility" within *SNA Type 2.1 Node Reference*.

2.2 Control Point (CP)

The control point (CP) is responsible for managing the T2.1 node and its resources. It activates links to adjacent T2.1 nodes, exchanges CP capabilities when establishing CP-CP sessions with adjacent nodes, and interacts with the node operator through the node operator facility. For its local LUs, the control point finds the partner LU′s location and provides routing information. The services of the control point are described in detail later in this document. They can be categorized as follows:

- Configuration Services (CP.CS)

  In LEN end nodes, APPN end nodes, and APPN network nodes, configuration services manages the links to adjacent nodes.

- Topology and Routing Services (CP.TS)

  In LEN end nodes and APPN end nodes, topology and routing services collects information on links and adjacent nodes. In APPN network nodes, topology and routing services additionally collects and exchanges information on other network nodes and the links between them. For LU-LU sessions, it provides the best route between the two LUs.

- Directory Services (CP.DS)

  The directory services component is responsible for locating network resources throughout the APPN network. On LEN end nodes, directory services only searches network resources defined in its local database. On APPN end nodes, directory services searches network resources in its local database first, but, if the network resource cannot be located, it uses the distributed search facilities provided by the APPN network node with which it has established CP-CP sessions. Although an end node can have active links to more network nodes, it maintains CP-CP sessions only with its current network node server.

  In order to locate network resources, directory services at each node collects resource information from the node operator and maintains this information in the local directory database. On request of an authorized APPN end node for which it provides network node services, directory services at the APPN network node registers the APPN end node′s resources in its local directory database.

- Session Services (CP.SS)

  The session services component is responsible for activating and deactivating the CP-CP sessions that are used by CP components to exchange network information. It is also responsible for maintaining and assigning unique session identifiers to sessions and assisting logical units in activating and deactivating LU-LU sessions.

- Address Space Manager (CP.ASM)

  The address space manager administers addresses that are used by path control to identify each individual session on a particular link. It interacts with LUs and ISR at BIND/RSP(BIND) and UNBIND/RSP(UNBIND) time. Optional features of address space manager are BIND reassembly and adaptive BIND pacing.

- Management Services (CP.MS)

  Management services monitors and controls the node's resources. Upon malfunction it will receive or generate alerts and forward these alerts to the network operator located either at its own node or a focal point node.

2.2.1 CP-CP Sessions

To perform directory services, session services, and topology and routing services, adjacent nodes throughout the APPN network use a pair of parallel CP-CP sessions to exchange network information. All CP-CP sessions use LU 6.2 protocols. Both sessions must be active in order for the partner CPs to begin or continue their interactions. Once the CP-CP sessions are established, the *capabilities* of the control points are exchanged.

Network nodes use CP-CP sessions to keep track of the network topology and also for directory and session services. A network node establishes two parallel sessions with each adjacent network node and with each client APPN end node. An APPN end node establishes two parallel sessions with a single adjacent network node acting as its current server. Between APPN end nodes, CP-CP sessions are not used. A LEN end node does not support CP-CP sessions.

Within APPN nodes, the CP *send* session refers to a CP-CP session that directory services uses to send either a Locate search or a registration flow to a partner CP (for details, see "Directory Services" on page 83). The send session corresponds to the *contention-winner* CP-CP session. An APPN node's CP *receive* session refers to a CP-CP session that directory services uses to receive the Locate search reply or a registration reply; the receive session corresponds to the *contention-loser* session of the pair. On the adjacent node, the CP-CP sessions are matched in the obvious complementary fashion with respect to sending and receiving. Each CP sends the session-activation (BIND) request for its own contention-winning session.

All CP-CP sessions are used to conduct directory searches. In addition, end node to network node CP-CP sessions may be used to register resources and to pass Alerts between management services components. CP-CP sessions between adjacent network nodes are also used to exchange topology information.

During link activation (see "Link Activation" on page 45) APPN network nodes indicate whether they support CP-CP sessions to particular APPN nodes on the link. During link activation APPN end nodes indicate whether or not they support CP-CP sessions, or if they support *and* request CP-CP sessions, over the link. The APPN end node may delay the establishment of CP-CP sessions, for example if they want to select a network node server at a later time.

CP-CP sessions between adjacent APPN network nodes can only be established if both nodes have the same NETID. An APPN end node can establish CP-CP sessions with a network node server that has a different NETID.

2.3 Logical Unit (LU)

The logical unit (LU) serves as a port into the network and acts as an intermediary between the end user of the LU and the network. The LU is engaged in session establishment with one or more partner LUs and manages the exchange of data with partner LUs.

T2.1 nodes support LUs that can both initiate sessions and respond to session initiation requests. The BIND sender is referred to as the primary LU (PLU); the BIND receiver is referred to as the secondary LU (SLU). A session starts when the PLU sends a BIND and the SLU responds with a RSP(BIND) and stops when UNBIND and RSP(UNBIND) are exchanged. The UNBIND may be sent by either LU.

For a discussion of dependent and independent LUs, see "Dependent and Independent LUs."

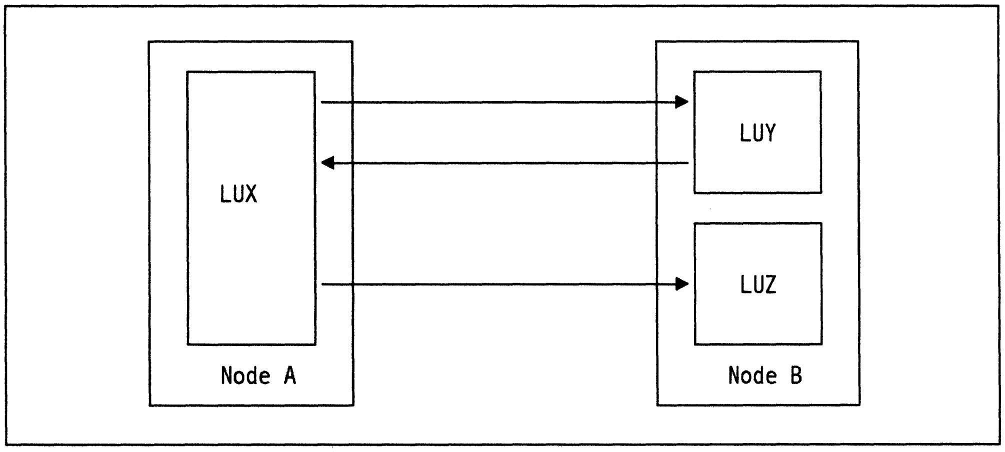

Figure 13. Multiple and Parallel Sessions

Figure 13 shows the distinction between *multiple* and *parallel* sessions. LUX has multiple sessions, two parallel sessions with LUY and one, single session, with LUZ. The direction of the session arrow shows the PLU-SLU relation: in this example, LUX acts as the PLU for the session with LUZ and one of the sessions with LUY. LUX also acts as the SLU for one of the parallel sessions conducted with LUY.

2.3.1 Dependent and Independent LUs

Logical unit types define the sets of functions in an LU that support end-user communication. The most flexible LU type is type 6.2, also known as LU 6.2. LU 6.2, because of its peer-oriented communication support, is particularly suited to today's environment in which processing power is widely distributed throughout a network. That is the reason that LU 6.2 was the only LU type supported in APPN networks.

Earlier LU types evolved within a hierarchical (subarea) network and depend on a system services control point (SSCP) for establishing LU-LU sessions. These are called ***dependent LUs***. An ***independent LU*** is able to activate an LU-LU session without assistance from an SSCP. LU 6.2 is the only LU type that can be an independent LU.

Dependent LUs feature an asymmetry in the roles of the partner LUs, with the host-based LU having the primary role with respect to session activation and recovery. The LU 6.2 avoids this limitation by allowing either partner to assume the primary role and activate communication over a session.

In order to migrate existing subarea networks to APPN there is, however, the need to support the vast amount of dependent LUs (e.g. 3270 type LUs) currently installed in subarea networks. See "Dependent LU Support" on page 183 for a discussion of the evolving support of dependent LUs in APPN networks.

2.4 Intermediate Session Routing (ISR)

At session endpoints it is the role of the LU, in conjunction with control point services, to establish sessions with a session partner and route session data back and forth to the partner LU. If the session partners reside on nonadjacent nodes, session data will pass through intermediate (network) nodes. As these intermediate nodes do not control any of the LU endpoints, LU services cannot be invoked on these nodes. It is the responsibility of intermediate session routing (ISR), to forward session data to the next node along the session path.

The structure of intermediate session routing is shown in Figure 14. The components of ISR are the session connectors (one for each session passing through the node) and a session connector manager.

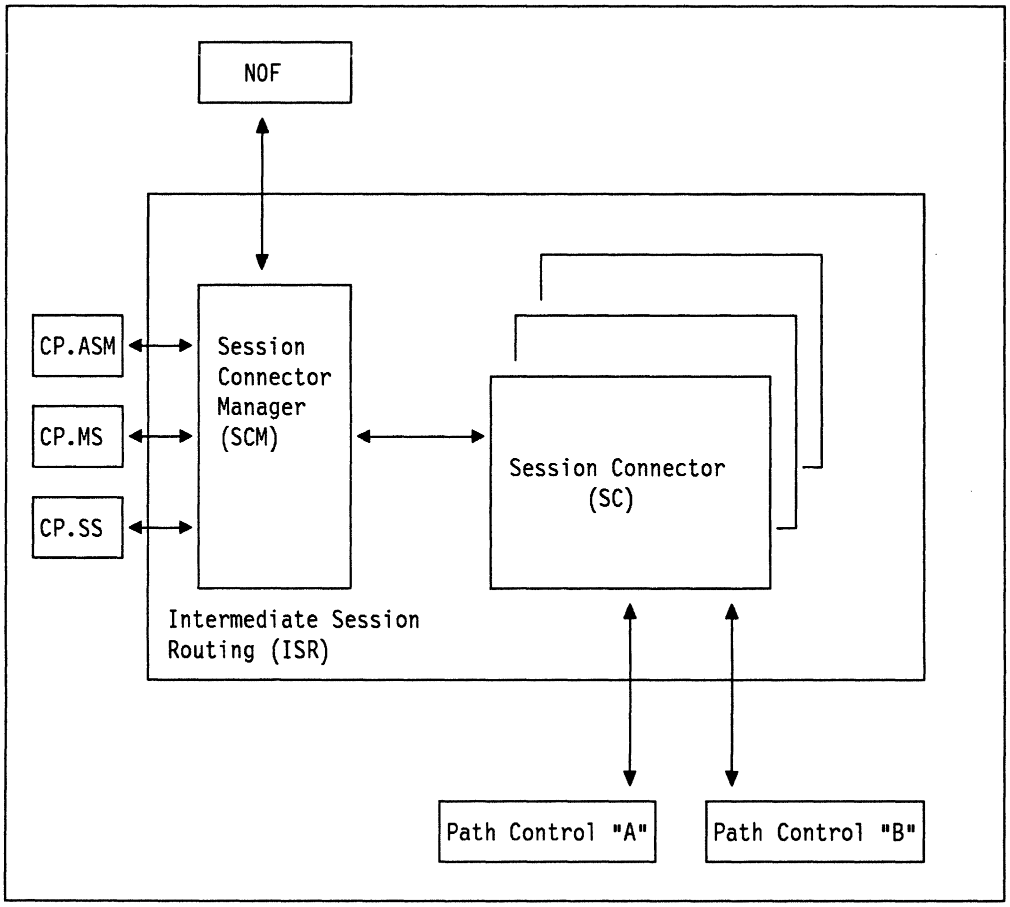

Figure 14. Structure of Intermediate Session Routing

Updates on dynamic node characteristics, which define the ability to perform intermediate session routing, are exchanged among APPN network nodes. Network nodes in an APPN network use node and TG characteristics in choosing routes. Two node characteristics are reported:

- Route-Addition Resistance
- Congestion status

Route-addition resistance defines the desirability for a network node to perform additional intermediate session routing. The higher the value defined for this attribute, the less desirable the node becomes for additional intermediate session routing purposes.

Congestion is determined so that new sessions can be directed away from a node when, for example, 90% of the defined maximum number of sessions using the node as an intermediate node, has been reached. A node is considered no longer congested when the number of routed sessions drops below, for example, 80% of the maximum number. When actually a node signals being congested or no longer congested is determined by implementation and installation definition. Node congestion may or may not allow additional sessions to be routed through a network node but it does indicate to all the other network nodes that alternative routes should be used.

2.4.1 Session Connector Manager (SCM)

SCM manages session connectors for sessions passing through the node. Its main functions are:

- To interface with the address space manager to obtain an LFSID for the TG in the direction of the destination LU.
- To perform intermediate BIND processing. RU sizes within the BIND will be updated if they exceed the maximum RU size allowed for the intermediate node. Session-level pacing will always be set to adaptive pacing, and window sizes will only be changed if an installation has defined specific values for the intermediate node.
- To create a session connector. The session connector will contain, among other parameters, the fully qualified procedure correlation ID (FQPCID) of the session and the LFSIDs used by the session on both the incoming and outgoing TG.
- To connect the session connector to the two path control instances when the session is activated.

For example, see Figure 15 on page 24, and assume an LU at ENA wants to establish a session with an LU at ENC. After having selected an LFSID(i), SCM forwards the BIND request unit, with LFSID(i) in its TH, to NNB. Routing information, to be more specific, the Route Selection Control Vector (RSCV), is contained within the BIND. The address space manager at NNB receives the BIND and passes it to the ISR component, as the destination LU is not located on NNB.

After NNB has changed the BIND according to its installation-defined parameters and a new LFSID(j) has been obtained, the LFSID(j) is entered in the TH (replacing LFSID(i)) and the BIND is forwarded to the next node.

2.4.2 Session Connector

The session connector (SC) connects two stages of a session. The main functions of the SC are:

- The routing of session traffic, in the form of path information units (PIUs), by performing address swapping on the address fields in the transmission header based on the LFSIDs stored during session activation along the route.
- Session-level pacing, either adaptive or fixed, of session data flowing on both stages of the session.

  With **fixed** session-level pacing, the maximum number of messages sent in one window is predefined at BIND time; with **adaptive** session-level pacing, the receiver dynamically adapts the number of messages sent in one window.

  Note: Session data is subject to two forms of flow control, namely **session**-level and **link**-level pacing. (Link-level pacing uses a fixed window size exchanged in XIDs during link activation.) Both techniques are used between adjacent nodes. For more information on congestion control see *SNA Technical Overview*.
- Intermediate reassembly of the inbound (received) basic information unit (BIU) segments (optional in end nodes).

  Note: The reverse process, segmenting a path information unit (PIU) into BIU segments of the appropriate size for the outgoing link, is a function done by path control (optional in end nodes).

2.4.3 Local-Form Session Identifier (LFSID) Swapping

Session traffic, in the form of path information units (PIUs), is routed through an intermediate node by performing address swapping on the address fields in the transmission header (TH), based on the LFSIDs stored during session activation along the route. This is illustrated in Figure 15 on page 24 and explained in the annotations.

For each TG on which a T2.1 node can send and receive message units, a separate path control instance and corresponding address space of *local-form session identifiers* (LFSID) is maintained. Each path control instance handles addresses only from its corresponding address space.

T2.1 nodes associate each session using a given TG with a 17-bit LFSID taken from the address space corresponding to that TG. On a specific TG, adjacent T2.1 nodes use the same LFSID to identify the message flow for a given session; they map the LFSID into the transmission headers (THs) they exchange in a defined way. On each session stage (or *hop*) between the endpoints, each pair of adjacent nodes uses a distinct LFSID to identify the session. An LFSID is assigned, when the BIND flows to activate the session, by each T2.1 node that forwards the BIND on a specific TG. The assignment endures for the life of the session, with *address swapping* in the TH occurring on each hop as subsequent session traffic flows over the route.

"Address Space Manager" on page 33 discusses the LFSID assignment algorithm.

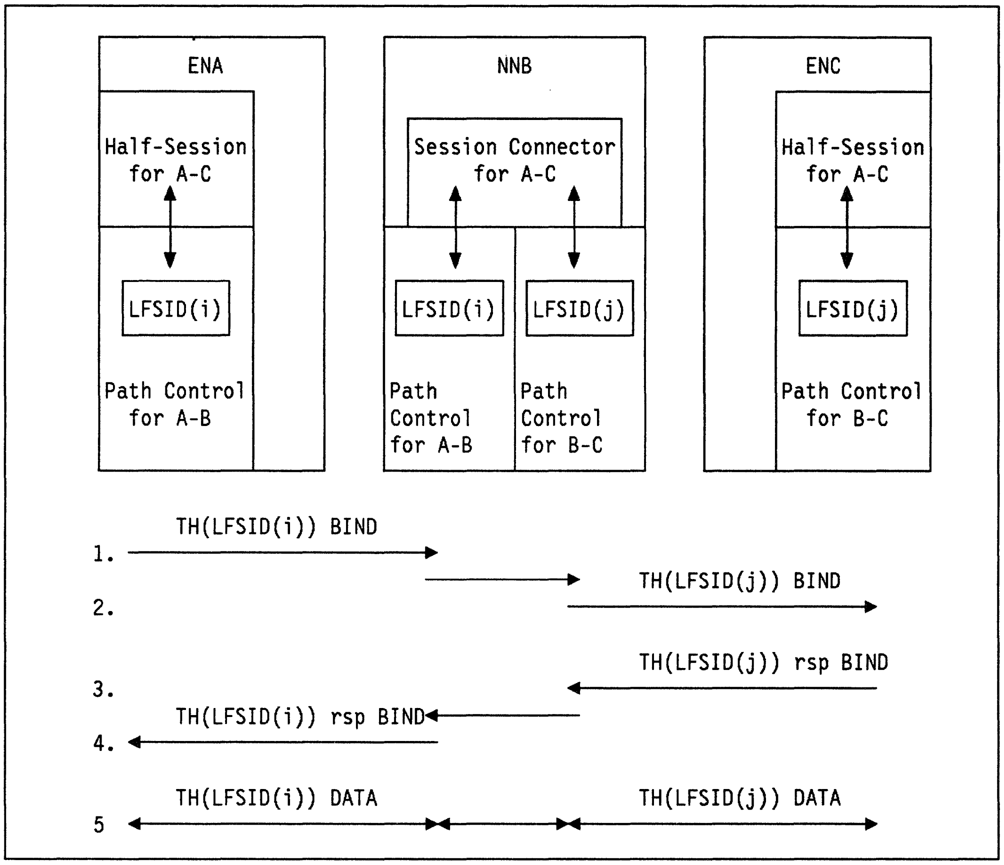

Figure 15. BIND Sets Up Address Swapping.

Annotations:

1. *The half-session in ENA is to be connected with the half-session in ENC to activate an LUA-LUC session. A BIND goes from ENA to NNB, carrying a TH that contains an LFSID created in ENA. In NNB, the BIND invokes upper-layer management components (address space manager and SCM) and creates entries in the newly activated session connector and in both path control components (one for the incoming TG and one for the outgoing TG). NNB created a new LSFID(j) for the session stage to ENC.*

2. *The BIND continues to ENC, but with new address fields, representing LFSID(j), in the TH.*

3. *ENC accepts the BIND and returns a positive response. The LFSID values used in the TH are reversed for the return path at each session stage.*

4. *The response continues to ENA with swapped address values in the TH.*

5. *Now the rest of the PIUs on the session can flow through NNB without rising above the session connector layer. The session PIUs pass through the session connector layer for the pacing function and to switch path control components. The addresses in the THs are swapped as noted in accordance with the information stored at BIND time.*

2.5 Path Control (PC)

The path control component delivers message units (MUs) between session-layer components in the same or different nodes. Session components consist of half-sessions in LUs and CPs (collectively referred to as *network accessible units*, or *NAUs*), as well as session connectors residing in intermediate network nodes. (See Figure 16 and Figure 17 on page 26.) The path control component allows these components to exchange MUs without concern for the underlying configuration of nodes and links.

Path control routes two types of message unit traffic:

- Session traffic. Requests and responses transmitted between paired session components.
- Nonsession traffic. Requests and responses between paired session components, including session activation and deactivation message units, for example, BIND, UNBIND, BIND-pacing Isolated Pacing Message.

The PC components support a transmission priority function for outgoing message units. Higher-priority messages are passed to data link control (DLC) before lower-priority messages.

One PC instance, a *process* initiated by CP configuration services, exists per transmission group (TG). A separate PC instance serves as a connection between logical units (LUs) in the same node. This latter PC instance is called the *intranode* PC; all others are *internode* PCs.

At the DLC layer, a single DLC process may serve multiple adjacent link stations. Each adjacent link station (ALS) is represented by its own ALS identifier and control block within its managing DLC process. A PC instance interacts with DLC using a DLC process ID and ALS designation for its adjacent link station.

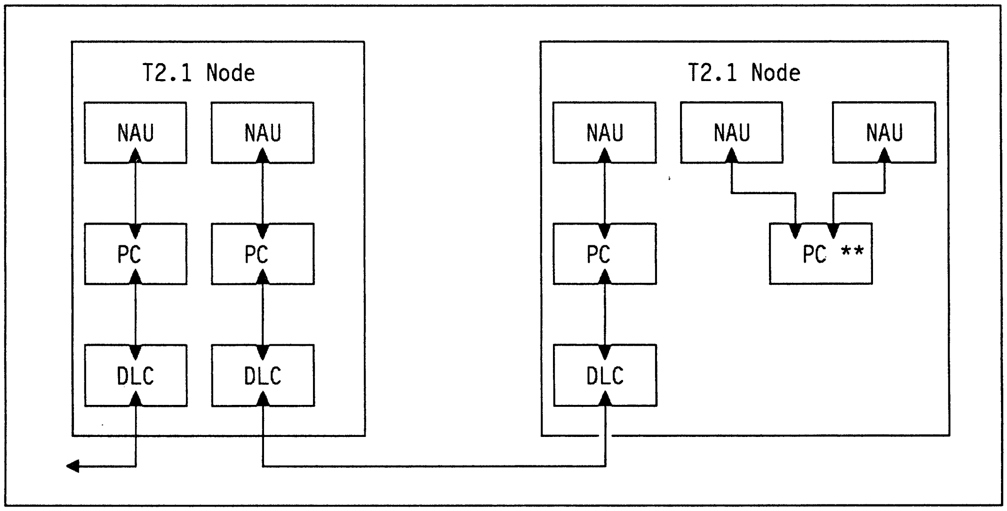

Figure 16. Internode and Intranode (**) Path Control Connections

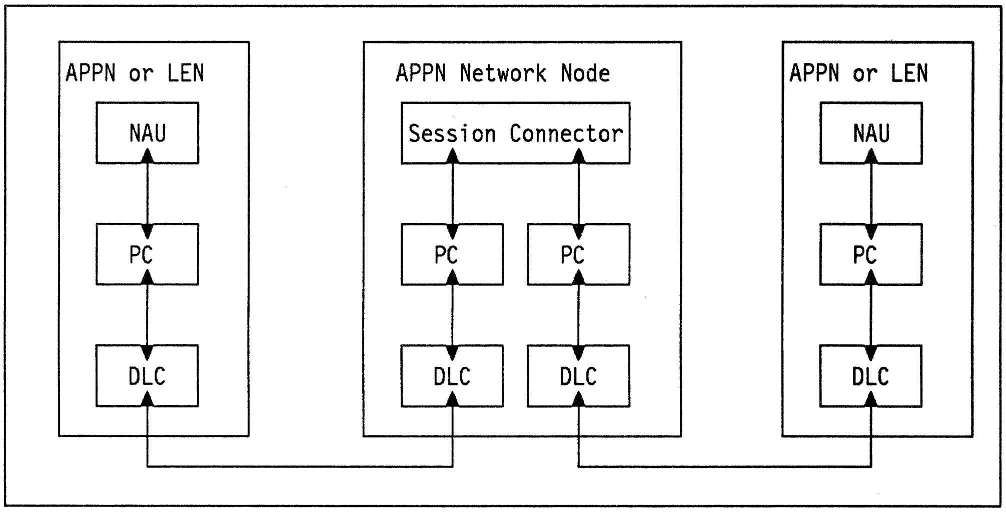

Figure 17. Intermediate Session Routing. The session connector interfaces with two PC instances.

Figure 18 on page 27 shows the structure of a PC instance and its interactions with other components.

The address space manager (ASM) sends session-connection and session-disconnection information to a path control manager, causing it to change the set of half-sessions connected to the path control instance. Each PC instance has its own address space from which the address space manager assigns local-form session identifiers (LFSIDs) to half-sessions connected to that PC instance.

The functions of the path control manager are:

- Session connection and disconnection. When notified, it establishes or breaks a connection to the specified half-session or session connector.
- Data flushing. Upon request, it stops outbound traffic after sending out all pending messages (those residing in path control queues) to data link control. This function is used when a link is being deactivated gracefully.

The path control element is responsible for:

- Message routing. It routes session traffic between session connectors, LU and CP half-sessions and DLC components.

  In order to perform its routing functions, path control maintains awareness of its connected session components. PC tables show the relationship between session components in the node connected to a PC instance and their assigned LFSIDs; PC uses this information to build or interpret TH addresses.
- Message transformation. It converts message units received from DLC to a form that can be processed by the CP and LU, and, conversely, it converts message units received from the CP and LU to a form that can be processed by DLC.
- Segment generation. It generates basic information unit (BIU) segments for outbound session traffic when required (done only if segment generation is supported by the local node and reassembly by the adjacent node). The reassembly of BIU segments into BIUs for inbound message units is performed after the message units have been passed to the session component or to ASM, as appropriate (done only if segment reassembly is

supported by the node). The ASM is invoked when PC has received nonsession data, for example BINDs.

- Error checking. It performs error checking (to find TH errors) on message units received from the data link.
- Transmission Priority. It enqueues outgoing messages to DLC according to session priority. Transmission priority support is optional in end nodes.

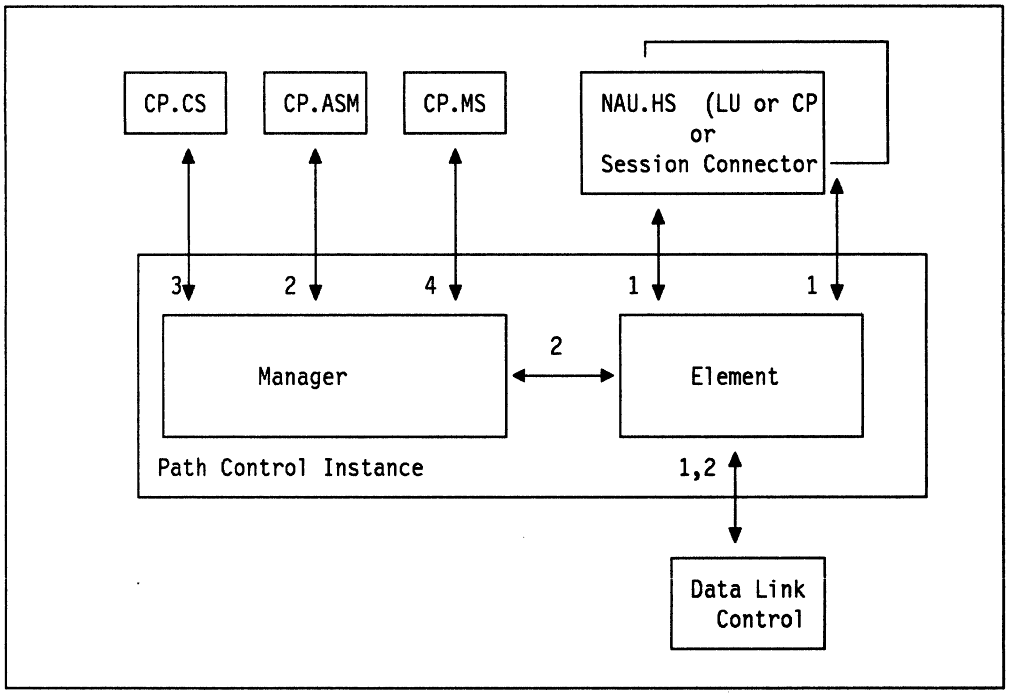

Figure 18. Structure of Path Control.

Annotations:

1. *Session Traffic*
2. *Nonsession Traffic*
3. *Create and Destroy Signals*
4. *Alert Signals*

2.5.1 Segment Generation

Session traffic and nonsession traffic are segmented if segmenting is supported by the node. Segmenting of BIUs into smaller BIU segments is performed by path control in order to transmit message units longer than the maximum size BTU allowed on a particular TG. These segments are reassembled into complete BIUs at the partner node. The LU learns from session services of the segment generation and reassembly capabilities of its node and all adjacent nodes, as well as the maximum BTU size of the TG. With this information, it prevents any message units from being passed to PC that would exceed the maximum BTU size when segment generation is not possible.

Segment Generation: A sender segments a BIU if the link receive buffer in the adjacent node is not large enough to allow the node to receive the whole BIU.

Segments are generated as illustrated in Figure 19. The mapping field in the TH of each PIU is set to indicate whether the BTU contains the first, middle or last segment of the BIU. If the BIU has not been segmented, the Mapping field indicates that the BTU contains a whole BIU. Nodes that do not support segmenting make a mandatory check for a Mapping field value that does not indicate a whole BIU; if such a value is found, the node sends a negative response if possible.

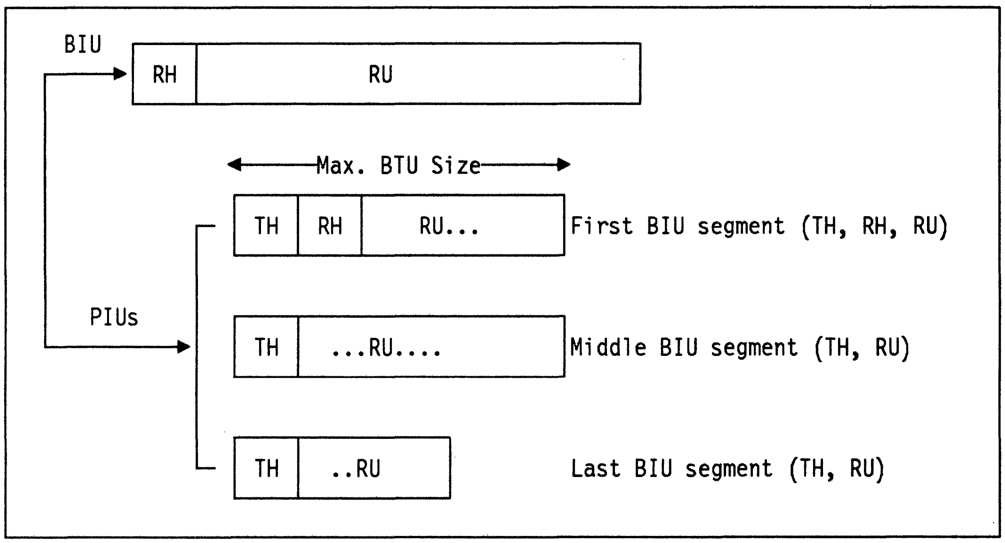
BIU
RH
RU
Max. BTU Size
TH
RH
RU...
First BIU segment (TH, RH, RU)
PIUs
TH
...RU....
Middle BIU segment (TH, RU)
TH
..RU
Last BIU segment (TH, RU)

Figure 19. Length of Segments.

Annotations:

RU	*Request Unit*
RH	*Request Header*
TH	*Transmission Header*
BIU	*Basic Information Unit*
PIU	*Path Information Unit*
BTU	*Basic Transport Unit*

All the segments of a BIND or RSP(BIND) are sent contiguously, not interleaved with other traffic.

Segment Reassembly: Segment reassembly is done according to the Mapping field in the TH of each PIU. Nodes do reassembly on a session basis in order to properly reassemble segments interleaved from different sessions.

2.5.2 Transmission Priority

Transmission priority provides a mechanism for specifying on a session basis the priority (network, high, medium, low) at which all outgoing messages on a session are to be transmitted (except isolated pacing messages, or IPMs, which are always transmitted at network, or highest, priority). The transmission priority is indicated in the message unit passed to PC. The message's priority dictates the order in which PC hands over the messages to DLC.

The PC components provide four queues (one for each transmission priority) for outgoing message units. Implementations may vary in their selection algorithms; for example, higher-priority queues may be served more frequently, or an aging mechanism may be chosen that guarantees a minimum bandwidth for lower-priority traffic.

2.5.3 Routing Actions

To route messages through an SNA network, path control adds a transmission header (TH) to each BIU. SNA path control uses format identifier type 4 headers (FID4) to route messages between subarea nodes and format identifier type 2 headers (FID2) to route messages between T2.0 or T2.1 nodes.

For incoming messages, the addressing information in the TH is used to relate the message to a specific half-session or session connector. For outgoing messages, path control uses the LFSID to generate the appropriate addressing information for the TH.

All T2.1 nodes use FID2 transmission headers for the internode routing of data traffic. A FID2 TH contains three address fields:

- A one-bit OAF′-DAF′ Assignor Indicator (ODAI)
- An eight-bit Destination Address Field prime (DAF′)
- An eight-bit Origin Address Field prime (OAF′)

The mapping between the 17-bit LFSID and the three TH addressing fields is done as follows. Path control uses a one-to-one mapping between the leftmost bit of the LFSID and the ODAI field. Mapping between the remaining 16 bits of the LFSID, composed of two eight-bit fields, SIDH and SIDL, as well as the DAF′ and OAF′ fields, is done according to the following table (see also "Local-Form Session Identifier (LFSID)" on page 34):

Table 1. Mapping of LFSID Fields into TH

1) Mapping of LFSID Fields for Message Unit (MU) flows

	TH Fields	
Direction of MU Flow	DAF′	OAF′
BIND sender to BIND receiver	SIDL	SIDH
BIND receiver to Bind sender	SIDH	SIDL

Note: The ODAI in both the LFSID and TH has its value set by CP.ASM in the BIND sender's node.

2) LFSID to TH Mapping for BIND-pacing IPMs

Direction of Isolated Pacing Message (IPM) flow	TH Fields	
	DAF′	OAF′
From primary link station	SIDL	SIDH
To primary link station	SIDH	SIDL

Note: The ODAI in both the LFSID and TH has the value 0.

Legend:

SIDL	Session identifier low
SIDH	Session identifier high
DAF′	Destination Address Field prime
OAF′	Origin Address Field prime
ODAI	OAF′-DAF′ Assignor Indicator

2.6 Data Link Control (DLC)

The DLC layer is responsible for the node-to-node protocols necessary to assure reliable delivery of information between paired stations in nodes attached to a common communication medium. These protocols are provided for sequencing, acknowledgment, error recovery, and the establishment and maintenance of synchronization between the paired stations.

There is one DLC manager and element for each DLC layer instance (a *process* started by the CP); see Figure 20. DLC provides, currently, protocols for SDLC, X.25, token-ring, ISDN, and S/370 channel connections.

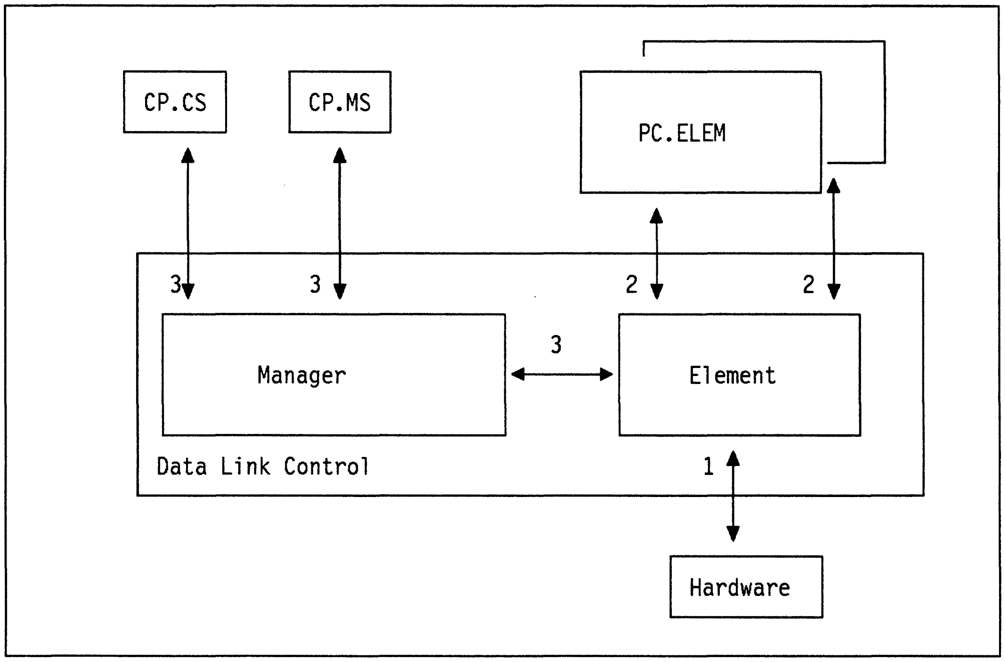

Figure 20. Data Link Control.

Annotations:

1. *The DLC element is the only component that communicates directly with the physical hardware. Both data and hardware control information flow over this interface.*
2. *Session related traffic is passed from path control to the DLC element.*
3. *Nonsession traffic flows over this interface. Requests to establish a switched connection, send a mode-setting command, etc. are performed over this interface.*

The functions of the DLC manager are:

- Activates and deactivates the DLC element
- Activates and deactivates links
- Manages the DLC portion of the CP-DLC protocol boundary
- Coordinates the actions performed by the DLC element in response to the service request from the CP
- Notifies the CP whenever a station or port becomes operative or inoperative

The functions of the DLC element are:

- Exchanges data traffic with adjacent DLC elements, subject to any fixed-window agreements and retransmitting when necessary

- Manages the DLC portion of the PC-DLC protocol boundary
- Transfers data to the physical medium
- For data networks, exchanges data traffic with the data network access data switching exchanges (DSEs)

Chapter 3. Address Space Manager

The address space manager (ASM) is one of the components in the control point (CP) of an APPN network node or APPN end node. ASM's functions include:

- Managing the session address (called *local-form session identifier, or LFSID*) used by a local path control for the routing of session traffic.
- Routing the session-activation messages (BIND, RSP(BIND)) and session-deactivation messages (UNBIND, RSP(UNBIND)) between the session managers or session connector managers (SCMs) and path control components within the node.
- Reassembling segmented session-activation messages (BIND, RSP(BIND)) received by the node into whole messages.
- Performing flow control of the session-activation messages (BIND).
- Notifying the appropriate session managers in the node when a link connection or link station fails.

3.1 Functional Overview

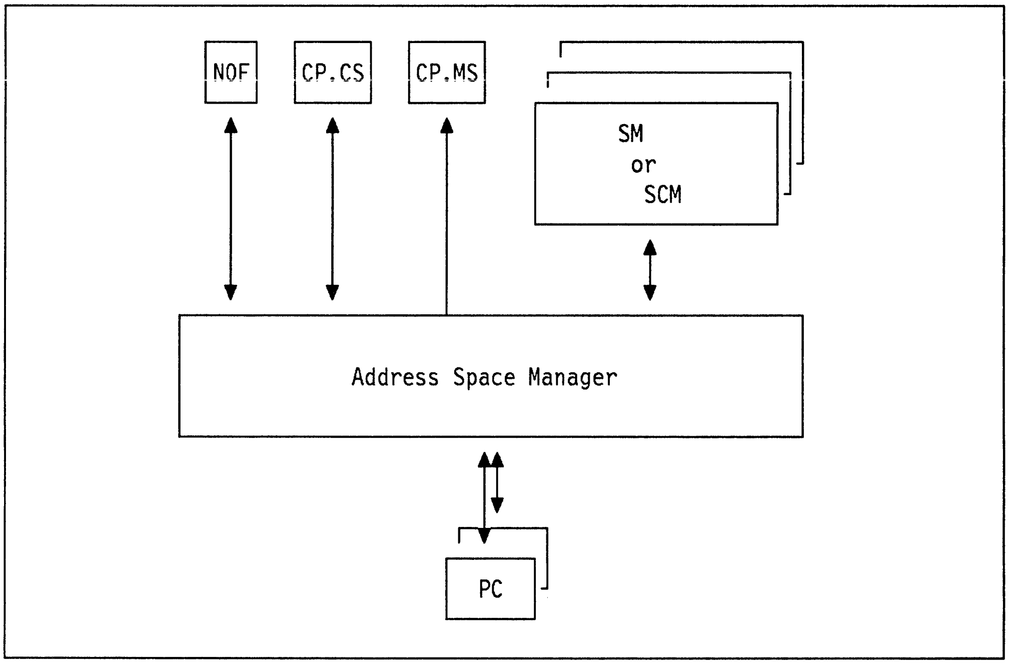

Figure 21. Overview of ASM Interaction with Other Components in the Node

The address space manager is created by the node operator facility at node initialization time. The node operator facility passes the following parameters to the address space manager:

- The name of the control point
- The network ID
- Whether or not BIND reassembly is supported

3.2 Address Space

For each TG attached to the node, ASM defines an address space consisting of 2^{17} (131072) LFSIDs that may be assigned (with some restrictions, see Figure 22 on page 35) to sessions routed over the TG. Each TG is associated with a unique path control instance (*process*), and the identifier for the TG′s path control instance is also used as the identifier for the TG′s address space.

When configuration services (CS) activates or deactivates a TG, it informs ASM. ASM then creates or removes, respectively, the control table it uses to manage the TG′s address space. In this control table, for each LFSID in the address space, ASM saves an indication as to whether or not the LFSID is assigned to a session.

The two nodes connected by a TG share that TG′s address space. When a session is initiated over the TG, ASM in the node that forwards the BIND selects the LFSID. To keep ASM in the two nodes from selecting the same LFSID for two sessions being initiated at the same time by BINDs flowing in opposite directions, the address space is divided into two partitions; ASM in one node selects LFSIDs from one partition, and ASM in the other node selects LFSIDs from the other partition. The partition is determined by the setting of one bit (the ODAI) of the LFSID. See the discussion of ODAI in the next section for more information about partitioning.

3.3 Local-Form Session Identifier (LFSID)

The transmission group between adjacent nodes can be used by multiple sessions. In order to relate the messages to a particular session, adjacent path control instances use unique session identifiers (LFSIDs) in the messages. On each session stage (or *hop*) between two session endpoints, each pair of adjacent nodes uses distinct session identifiers to identify a session; therefore, the term *local-form session identifier (LFSID)* is used.

The LFSID is a 17-bit identifier used by path control to route session traffic; see Figure 22. The LFSID is composed of a 1-bit ***ODAI*** (OAF′-DAF′ Assignor Indicator) field and two 8-bit fields: ***SIDH*** (Session Identifier High) and ***SIDL*** (Session Identifier Low). The ODAI divides the LFSID address space into two distinct partitions. Each ASM in the two nodes connected by a TG selects LFSIDs from that TG′s address space with a different ODAI value, so that the two ASMs never select the same LFSID. The ODAI value determination is a by-product of link station role negotiation during XID exchange. ASM in the node with the primary link station selects LFSIDs with an ODAI value of 0, and ASM in the node with the secondary link station selects LFSIDs with an ODAI value of 1.

The SIDH and SIDL allow ASMs on either end of a TG to assign 2^{16} (65536) session identifiers for each TG. For details see the next section.

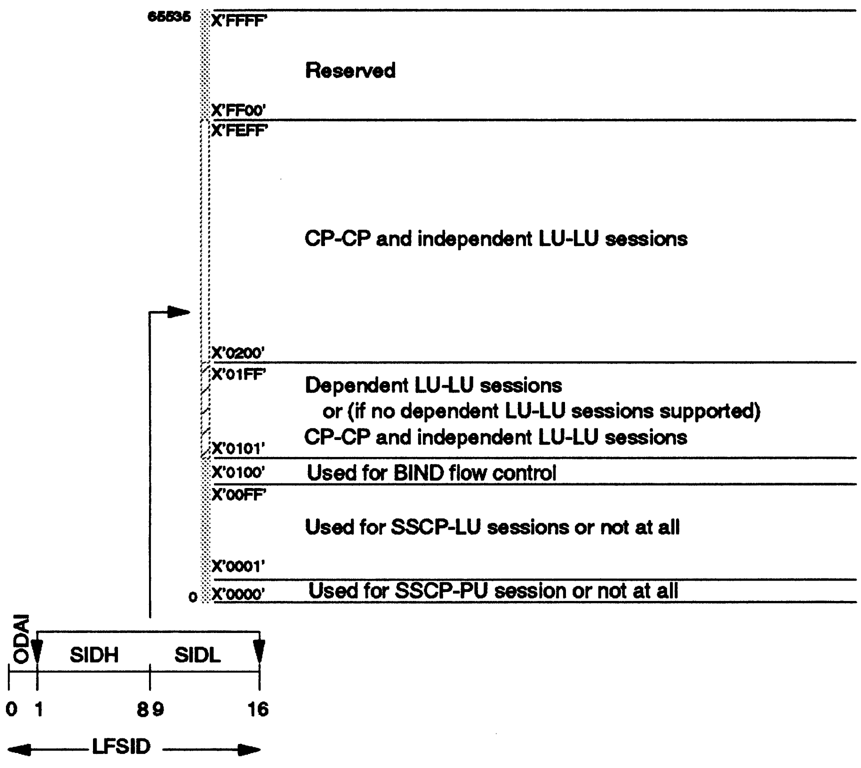
65535
X'FFFF'
Reserved
X'FF00'
X'FEFF'
CP-CP and independent LU-LU sessions
X'0200'
X'01FF'
Dependent LU-LU sessions
or (if no dependent LU-LU sessions supported)
CP-CP and independent LU-LU sessions
X'0101'
X'0100'
Used for BIND flow control
X'00FF'
Used for SSCP-LU sessions or not at all
X'0001'
0
X'0000'
Used for SSCP-PU session or not at all
ODAI
SIDH
SIDL
0 1
89
16
LFSID

Figure 22. Local-Form Session Identifier (LFSID)

3.3.1 Address Space Management

For CP-CP or independent LU-LU sessions, the session manager (SM) components in the CP or LU request an LFSID from ASM. For sessions routed through an intermediate node, the session connector manager (SCM) invokes ASM to obtain an LFSID. ASM selects an LFSID that is not currently in use by another session, assigns it to the particular session and informs the SM or SCM of the assigned LFSID. If the T2.1 node contains dependent LUs, the association of LFSIDs to dependent LUs is made at system-definition time. ASM assigns LFSIDs according to the partitioning of the address space described below; Figure 22 illustrates this partitioning.

1. If dependent LU-LU sessions can use the TG, the LFSID with SIDH = X'00' and SIDL = X'00' is used for the SSCP-PU session; otherwise, this LFSID is not used.

2. If dependent LU-LU sessions can use the TG, the LFSIDs with SIDH = X'00' and SIDL values in the range from X'01' to X'FF' (inclusive) are used for SSCP-LU sessions; otherwise, these LFSIDs are not used.

3. The LFSID with SIDH = X'01', SIDL = X'00', and ODAI=0 is used for BIND flow control.

4. If dependent LU-LU sessions can use the TG, the LFSIDs with SIDH = X'01' and SIDL values in the range from X'01' to X'FF' (inclusive) are used for dependent LU-LU sessions; otherwise, these LFSIDs are used for CP-CP and independent LU-LU session.

5. LFSIDs with SIDH values in the range from X'02' to X'FE' (inclusive) are used for CP-CP and independent LU-LU sessions.
6. LFSIDs with SIDH = X′FF′ are reserved.

This partitioning of the LFSID address space enables a T2.1 node that contains dependent LUs to accept ACTPU, ACTLU, and BIND requests from a VTAM or NCP boundary function.

Each T2.1 node forwarding a BIND request (that is, the node owning the PLU and each intermediate node), assigns the lowest valid LFSID value available from the appropriate LFSID address space. Available LFSIDs include those that are released when sessions are deactivated. The LFSIDs assigned to a session, one LFSID on each TG connecting two nodes, are valid only while a session is active. Path control informs ASM that an UNBIND or RSP(UNBIND) has been sent, or when session managers inform ASM that the session activation has failed.

The two nodes connected by a TG share that TG′s address space. When a session is initiated over the TG, ASM in each node that forwards the BIND selects an LFSID for the TG between two adjacent nodes. Path control on either side of the TG inserts this session identifier in the transmission header of all the basic information units (BIUs) for that session.

Note: The usage of LFSIDs is similar to the usage of *logical channels* within X.25. Session identifiers allow path control instances on two adjacent nodes to multiplex data on TGs connecting the nodes, and relate the data received to specific half-sessions or session connectors (for intermediate session routing). The session identifiers have *local* significance only. If an LU-LU session is routed through intermediate nodes, then a different LFSID will be assigned for each TG along the path between the nodes owning the LUs.

3.4 BIND Segmenting and Reassembly

T2.1 nodes optionally support segmenting and reassembly of BIND requests and responses. Path control performs the segmenting, while the address space manager (ASM) performs the reassembly. Like segmenting and reassembly for other basic information units (BIUs), BIND segmenting and reassembly uses the Mapping field in the FID2 transmission header. For details, see *Systems Network Architecture Formats*.

When configuration services (CS) activates a transmission group to an adjacent node, it negotiates with configuration services at the other node the maximum message (BTU) size that can be sent across the transmission group. If the BIND message is larger than the BTU size selected for the transmission group, path control performs BIND segmentation. However, path control can only perform BIND segmentation if ASM at the adjacent node is capable of BIND reassembly. Knowledge of whether or not the receiver is capable of BIND reassembly is exchanged between nodes at transmission group activation time as part of the XID exchange.

If the address space manager does not support BIND reassembly, it will discard any segmented BIND request or response and instruct configuration services to deactivate the transmission group.

3.5 Bind Flow Control

When a node activates a large number of sessions across a transmission group in a short period it may fill up all buffers at the adjacent node. As a consequence, the adjacent node may run into a deadlock situation, as it can no longer obtain free buffers to respond to the activation requests or receive new BIND requests.

To circumvent these types of problems, the address space manager can perform flow control for all BINDs sent and received across a transmission group. The flow control mechanism is called ***adaptive BIND pacing*** and is similar to adaptive session-level pacing; for details, see *Systems Network Architecture Formats* and *SNA LU Version 6 Reference: Peer Protocols.*

Adaptive BIND pacing uses two algorithms, a sender and a receiver algorithm. Both are window based, which means that the sender can send only a limited number, or *window*, of messages per grant of permission-to-send from the receiver. As long as this permission has not been given, the sender must defer sending messages. After receiving permission the sender may send the next window of messages. Because the pacing algorithm allows the window to expand and contract, the term *adaptive* is used.

Chapter 4. Configuration Services

The configuration services (CS) component of the CP in an APPN node manages the node's local resources, such as the links to adjacent nodes. Most of the functions are the same for LEN end nodes, APPN end nodes, and APPN network nodes. Where there are differences, they will be pointed out in this chapter.

Configuration services creates path control instances, which it associates with specific links (TGs) as it activates them, and destroys the path control instances after having deactivated the associated links. It also creates the intranode path control process, which is used for routing messages between LUs that reside in the local node. Configuration services provides information, acquired as a result of its functions, to other components of the node.

The basic functions performed by configuration services are:

- Definition of the node's configuration:
  - Types of data link control (DLC)
  - Ports
  - Adjacent link stations
  - Attached connection networks
  - Adjacent nodes
- Link activation (including XID exchange)
- Nonactivation XID exchange
- Link deactivation.

These functions are described in the following paragraphs.

4.1 Functional Overview

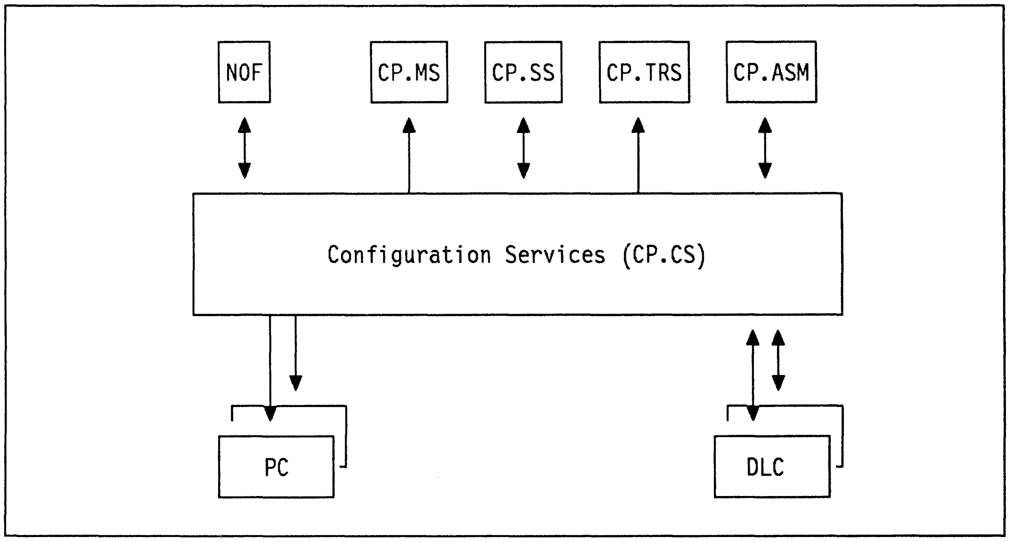

Figure 23. Overview of CS Interaction with Other Components in the Node

The node operator facility (NOF) initializes configuration services. NOF also defines, starts, stops, and queries the components of configuration services.

The following information is passed to configuration services when it is initialized:

- The node's CP name.
- The node's network ID.
- The node's product set ID, containing information such as machine type, machine serial number, software product number, date of link-edit.
- Whether or not negotiable link stations are supported. (Defining a link station as negotiable allows the link station to be either primary or secondary. The actual role is determined during link activation.)
- Whether or not parallel TGs are supported.

4.2 Components

4.2.1 Data Link Control

The DLC layer is responsible for the node-to-node protocols necessary to assure reliable delivery of information between paired stations in nodes attached to a common communication medium. These protocols are provided for sequencing, acknowledgement, error recovery, and the establishment and maintenance of synchronization between the paired stations.

There is one DLC manager and element for each DLC layer instance (a *process* started by the CP). Each DLC layer instance, or process, may manage one or more ports. For details, see "Data Link Control (DLC)" on page 30.

4.2.2 Ports

A port represents a physical connection to the link hardware. The specific component it represents is sometimes referred to as an *adapter*. Each port is associated with a DLC process.

Ports are defined by the node operator facility using the following types of information:

- Associated DLC process.
- Information specific to the port, like link station activation limits and time-out values.
- Information that is common to all link stations associated with the port, for example TG characteristics (modem class, security) and receive buffer size.

  Some of the information is not needed for link activation, but is used for route calculation by route selection services. For details, see "Route Computation: Overview" on page 73.
- Information about any connection network (discussed in "Connection Networks and Virtual Routing Nodes" on page 49) if one or more is defined on the port.

4.2.3 Links

A link represents a connection between a local link station and a link station in an adjacent node. It includes the data link control (DLC), the port, and the link station components. The associated link station in the adjacent node is locally referred to as the ***adjacent link station*** (ALS).

Note: The term link, or physical link, is often used to refer to the physical components which enable two adjacent nodes to communicate. Within APPN a link should be considered as a logical association between two entities in distinct T2.1 nodes.

A link between T2.1 nodes may require that one link station takes the role of *primary* link station and one link station takes the role of *secondary* link station. This role setting does not imply that the link stations maintain a *master slave* relation; see "Link Activation" on page 45.

Link roles are coordinated at link activation time. The link station roles must either be predefined or negotiated during link activation. Predefinition of link station roles requires that the definitions at both ends match. If both nodes define the local link station as primary or both define the local link station as secondary, link activation will fail. Defining a link station as *negotiable* means that the link station role can be either primary or secondary and that the actual role will be determined during link activation. If both ends are defined as negotiable, the final roles are decided on the basis of node identification fields exchanged between the two link stations, during link activation.

Point-to-Point and Multipoint

Links can be either *point-to-point* or *multipoint*. Implementations may provide the ability to add secondary link stations to existing point-to-point connections through dynamic reconfiguration. This type of connection is called a multipoint capable connection.

Point-to-point links are links between two and only two link stations. The link station role, primary or secondary, can be negotiated during link activation. There is no need to define the secondary link station address. If the value is needed, it will be acquired during XID negotiation.

Multipoint, or multipoint capable, links are links between one link station at one end, which is always the primary link station, and one or more adjacent link stations, which are always the secondary link stations. Multipoint links require predefinition of the link station role. Negotiable stations are not usable on multipoint or multipoint-capable link connections because they use the broadcast address to avoid defining the secondary address when they do not know which end will be the secondary station. Any station receiving the broadcast address will respond to it. Multipoint, or multipoint capable, links require explicit definition of the secondary station addresses.

Note: The data link layer protocol on a point-to-point connection can be either a *balanced* or an *unbalanced* protocol. Unbalanced protocols presume a *master slave* relation; balanced protocol presume a peer relation. The DLC layer on a multipoint connection always uses unbalanced link protocols. Examples of balanced DLC protocols are LAPB and LAPD, while an example of an unbalanced DLC protocol is SDLC.

Switched and Nonswitched

Links can be either *switched* or *nonswitched*. Switched links require some kind of "dial" procedure before link activation can take place. Nonswitched links can be activated immediately after a port has become active. Switched link connections are always point-to-point links, but nonswitched links can be either point-to-point or multipoint. Multiple simultaneous switched connections may be supported through a single port.

A switched link may support *auto-activation*, to automatically activate a link when sessions are established using that link station.

A switched link may also be defined as a ***limited resource***, to automatically deactivate a link when no sessions use the link. For example if an X.25 network provider charges its users for the period a switched connection is kept active, network administrators may decide that X.25 links should be deactivated if the link is not used.

Examples of switched link connections are the links between T2.1 nodes attached to an X.25 public switched data network (PSDN) using switched virtual circuits.

Examples of nonswitched link connections are links between adjacent T2.1 nodes connected by a leased line, or T2.1 nodes attached to an X.25 public switched data network (PSDN) using permanent virtual circuits.

4.2.4 Transmission Group

A transmission group (TG) corresponds to a connection with a single adjacent link station. Each TG has a TG number assigned to it during link activation. The APPN architecture does not provide TGs consisting of multiple links. When more than one TG connects two adjacent nodes they are known as ***parallel*** TGs. When a node has TGs connecting to more than one adjacent node, it has ***multiple*** TGs.

The transmission group (TG) number that will be used as part of the link representation will be determined during link activation. The TG number must be unique between a pair of CPs. This allows a TG to be uniquely identified by a pair of (network-qualified) CP names and a TG number.

Table 2. TG Number Space

Parallel TGs supported	Range	Function
No	{0,20}	Predefined
Yes	{1,20}	Predefined
Yes or No	{21,239}	Negotiated
Yes or No	{240,255}	Reserved

Table 2 shows the general rules for determining the TG number. When parallel TGs are not supported between two nodes, any integer between 0 and 255 is permissible as a TG number. When parallel TGs are supported between two nodes, any integer from 1 to 255 is allowed as a TG number. The number 0 is excluded as a valid TG number when parallel TGs are supported, since it has special meaning for the TG negotiation itself.

Any TG less than or equal to 20 is set aside to represent a TG that has been predefined between two nodes. Predefined TGs are used when it is important to match the characteristics of the link stations on both ends of the TGs. TG numbers greater than 239 have a special meaning and must not be used. For example, a subarea network may connect to an APPN network using a VTAM interchange node (ICN). To provide transparency to the T2.1 nodes in the APPN network, all LUs in or accessible through the subarea network are presented as if they reside on an end node which connects to the ICN using TG number 254. See "APPN VTAM" on page 169.

TG numbers from 21 to 239 are selected during TG number negotiation, which is performed during link activation; see "Link Activation" on page 45. The list below summarizes the rules for determining the TG number:

- For connections that are being reactivated, the TG number that was used for the previous activation is reused, if possible.
- If one node sends a TG number of 0, then it is willing to accept the TG number of the other side.
- If both nodes send TG numbers of 0 and parallel TGs between them are not supported, then the TG number is set to 0.
- If both nodes send TG numbers of 0 and parallel TGs between them are supported, then the node with the higher network-qualified CP name picks a valid TG number.
- If neither node sends a TG number of 0, then the number that was sent by the node with the higher network-qualified CP name is used.

4.3 System Definitions

A node is responsible for its own local definition of supported links and their characteristics, node capabilities, and the control point names of the node that can be directly attached; this and other information is maintained by CS in its database.

A link station is a combination of hardware and software which allows a node to control a link. Some characteristics must be defined explicitly through the node operator facility, while others can be either defined explicitly or negotiated with the adjacent link station during link activation.

A node requires the following system definition for a local link station:

- Link station name
- Link station role: primary, secondary, negotiable
- Local link station address for any secondary or negotiable station
- Modem equalization delay value
- Inactivity timer
- Retry limit for mode-setting command (SNRM,SABM)

Certain nodes can act only as primary link stations and require the attaching node to assume the secondary role. This requirement is defined by the network administrator at system definition time for the attached node.

The components of a link (DLC, port, link station) are defined individually. DLC must be defined before its associated ports are defined, and ports must be defined before associated adjacent link stations are defined. More than one adjacent link station may be defined on a port.

An adjacent link station is either defined explicitly, by the node operator facility, or dynamically, using a set of default parameters assigned to them. *Dynamic Link Stations* may be defined because session services (SS) has required the activation of a link, or as a result of an adjacent node activating a link.

A dynamic link station is treated as a *limited resource*, meaning that when no sessions are using the link between the local and the dynamic link station then the link can be deactivated. No CP-CP sessions are supported on connections using dynamic link stations, since CP-CP sessions normally need to be kept up continuously.

Information about the adjacent link station is used when the link station is activated, deactivated or its status queried. To activate a link, the DLC, port, and link station must be activated. Dynamically defined link stations cannot be activated by the operator.

4.3.1 DLC, Port and Link Station Interrelationship

Figure 24 shows how CS maintains information about DLC processes, ports, and link stations. There can be one or more DLC processes per node, one or more ports used by DLC, and one or more link stations per port.

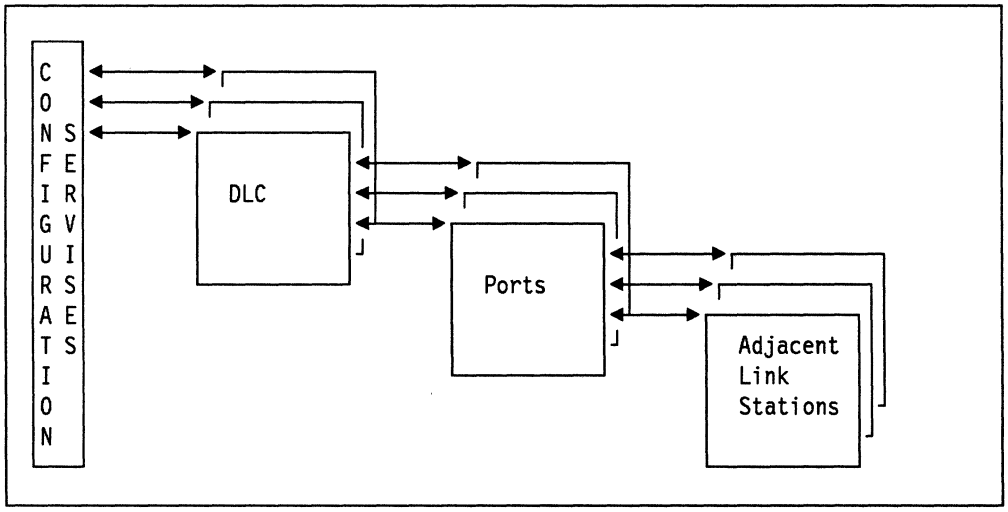

Figure 24. DLC, Port, and Link Station Interrelationship

4.4 Link Activation

Link activation is initiated locally, through an operator command or following a session setup request, or by the adjacent node. DLCs and ports must be defined before they can be activated. Adjacent link stations must be defined before they are activated except in the case of dynamic link stations. A DLC is always activated before its associated ports, and ports are always activated before their associated adjacent link stations.

Link activation encompasses the activation of the physical link connection and the adjacent link station. It consists of at most three phases (see Figure 25):

- Connect Phase
- Prenegotiation XID exchange
- Contact Phase

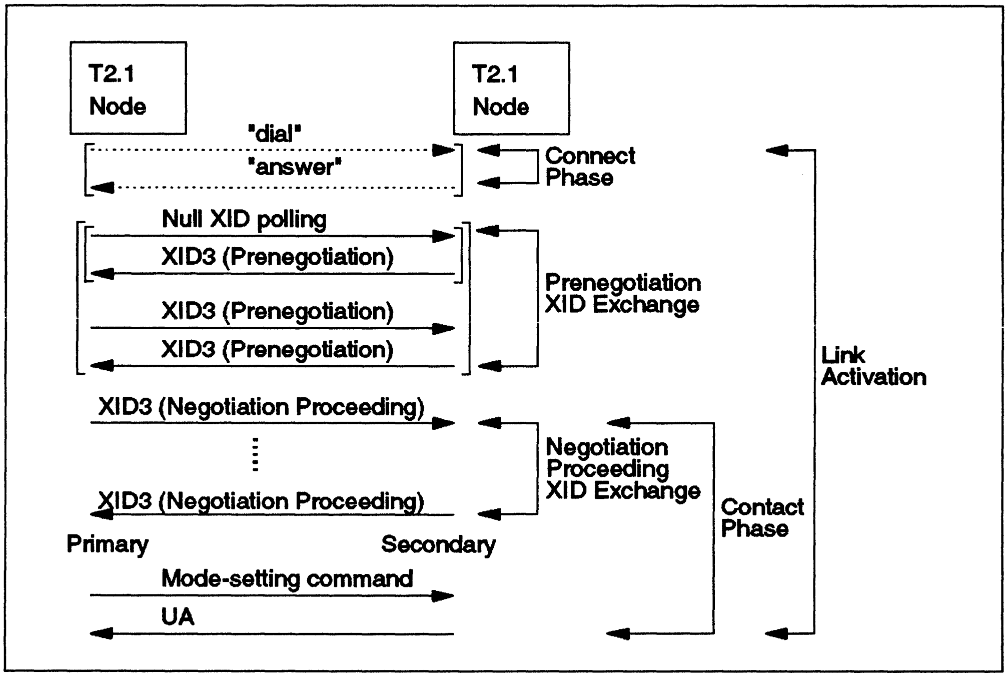

Figure 25. Link Activation. Optional flows are indicated with double brackets.

The connect phase allows initial establishment of communication between nodes. The connect phase is optional and DLC dependent. For switched connections, one may think of "dial" and "answer" procedures, for example the establishment of a virtual circuit if nodes are attached to an X.25 packet switching network. Once the connect phase has completed, the two nodes are able to exchange and establish node characteristics via XID exchanges.

The exchange of "prenegotiation" XIDs is optional as well. It allows a node to determine if the adjacent station is active and to verify the identity of the adjacent node. Node identification fields and, optionally, the CP name will be exchanged. As an example, for switched connections, VTAM will select switched major node definitions based on information obtained during this phase.

APPN nodes use two types of XIDs: A null XID to determine if the adjacent station is active and an XID3 to perform the "prenegotiation" and "negotiation proceeding" process, which is part of the contact phase. For format details, see *Systems Network Architecture Formats*.

As part of the contact phase, the T2.1 nodes will start an XID3 negotiation process to establish primary and secondary roles of the link stations, the TG number, and other characteristics of the link. The result of the primary-secondary role negotiation determines which node will send the mode-setting command (SNRM, SABM) and is also used in the setting of the ODAI field in the LFSID (see "Local-Form Session Identifier (LFSID)" on page 34).

The "negotiation-proceeding" XID3 exchange completes once link station role negotiation and TG number negotiation have completed and when each node has sent and received at least one "negotiation-proceeding" XID3.

After the link activation XID exchange has completed successfully, CS creates a new path control instance and instructs the address space manager (ASM) to activate a new address space. When the address space has been created, CS instructs DLC to perform the DLC mode-setting exchange and notifies topology and routing services (TRS) that a TG has become active. Finally, if during link activation the adjacent node has indicated that CP sessions are either supported and/or requested, CS notifies session services (SS). SS may then activate CP-CP sessions if necessary.

The link is active from the perspective of both nodes when a mode-setting command has been sent and a response returned.

The link can be deactivated from either end, via the the node operator facility, or after failures have been detected on the link station or port. A link defined as a *limited resource* will be deactivated after the number of sessions using the link, falls to zero.

4.4.1 XID3 Negotiation

The contact phase consists of the "negotiation-proceeding" XID3 exchange and the mode-setting sequence.

The XID exchange reduces the requirement for system definition of the adjacent node. During the "negotiation-proceeding" XID exchange, link station roles and the TG number used to represent the link are resolved cooperatively by the two link stations.

The following information, where applicable, is communicated to the adjacent node:

- Adjacent Link Station (ALS) name
- CP capabilities
  - Network node providing services over this link
  - Network node not providing services over this link
  - End node supporting CP-CP sessions over this link
  - End node not supporting CP-CP sessions over this link
  - End node supporting and requesting CP-CP sessions over this link
- CP name
- Link characteristics
- TG number

- Subarea PU name
- Product set ID
- Node capabilities
  - Parallel TG support
  - DLC support

Basic Transmission Unit (BTU) Size

Each link station determines its own maximum send Basic Transmission Unit (BTU) size. It is based on local node definitions and XID information received from the DLC and the adjacent link station. The smallest of the following values will become the actual maximum send BTU size:

- Locally defined send BTU size
- Maximum BTU size as set by DLC
- Maximum receive BTU size of the adjacent link station

When the partner does not support BIND reassembly, the maximum BTU size must be at least 265 if the partner node is an APPN end node or LEN end node, or at least 521 if an APPN network node.

When it supports segmenting, path control will segment outgoing messages that are longer than the maximum BTU size. Figure 26 shows how the maximum BTU impacts this segmenting process.

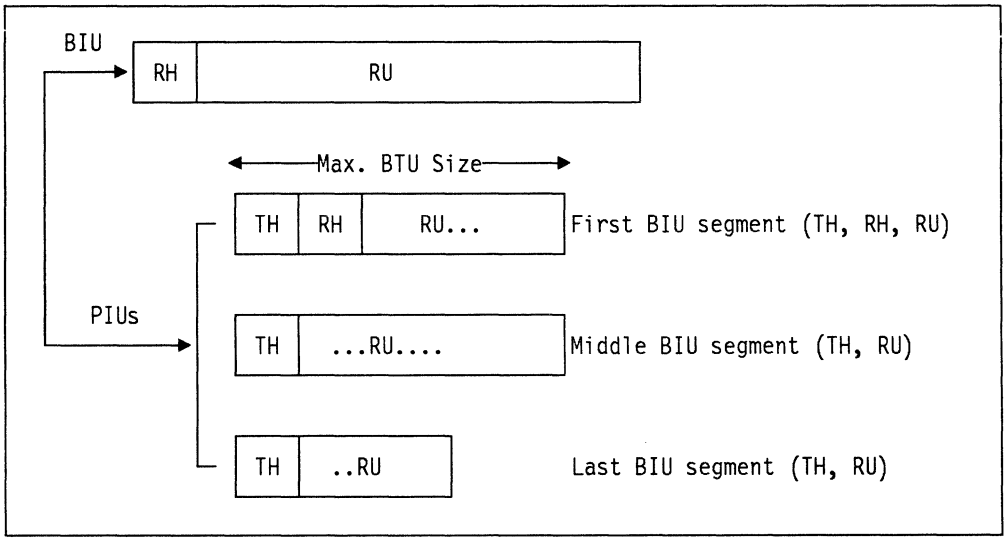
BIU
RH
RU
Max. BTU Size
TH
RH
RU...
First BIU segment (TH, RH, RU)
PIUs
TH
...RU....
Middle BIU segment (TH, RU)
TH
..RU
Last BIU segment (TH, RU)

Figure 26. Length of Segments.

Annotations:

RU	*Request Unit*
RH	*Request Header*
TH	*Transmission Header*
BIU	*Basic Information Unit*
PIU	*Path Information Unit*
BTU	*Basic Transport Unit*

4.5 Non-Activation XID Exchange

Information associated with an already active link may change. To communicate these changes, adjacent link stations use a "nonactivation" XID3 exchange. Possible reasons to start a nonactivation XID3 exchange, are:

- Network node server change
- SSCP takeover
- TG quiescing

All T2.1 nodes support the receipt of nonactivation XID3 when they contain the secondary link station on a connection, but not all implementations support the receipt of nonactivation XID3s when assuming the primary link station role. The capability of a primary link station to receive an XID from the secondary station when no XID command has been issued, is declared in the XID3 during link station activation. Unless both nodes indicate support for this function, secondary-initiated nonactivation XID3 exchanges cannot occur.

When a nonactivation XID3 exchange occurs, the parameters relating to the physical characteristics of the connection and the connecting nodes have already been established. Table 3 shows the XID parameters that, during a nonactivation XID3 exchange, will never change, may change without a CP name change, and may change but only together with a CP name change.

Table 3. XID3 Parameters. Requested link station changes.

Field	Never Changed	Without CP Change	With CP Change
ACTPU Suppression	X		
Link Station Role	X		
CP-CP Session Requested		X	X
CP-CP Session Supported		X	X
TG Number			X
CP Name			X
CP Name Change Requested		X	X
TG Quiescing		X	X

No IBM implementation uses the nonactivation XID3 exchange to change network node server. Instead, this can be achieved using SS-initiated protocols; see "CP-CP Session Activation" on page 113

SSCP takeover is a function of subarea networks which allows one SSCP to gain ownership of NCP boundary function connections which were previously owned by the same or another SSCP, without breaking the connections or disrupting existing LU-LU sessions on the connections. Once the new SSCP has taken over a connection, it processes all session requests that come from or are destined for LUs on that connection. A CP name change and possibly a TG number change take place during a VTAM *SSCP takeover*.

The fundamental role that nonactivation XID3 exchanges play during SSCP takeover requires that the NCP must have the capability of initiating a nonactivation exchange whether it assumes primary or secondary link station role on a connection. As all APPN nodes support the receipt of nonactivation XID3s when they contain the secondary link station, but do not generally support secondary-initiated exchanges when they contain the primary link station on a connection, the NCP providing the boundary function must assume the primary link station role during link activation. NCP only allows role negotiation on SDLC switched connections, and the NCP logic works in a way as to almost always become the primary end of SDLC switched connections. For nonswitched connections, the link station role must be predefined in NCP, so the user can force the NCP to be the primary end. In case of NCP-NCP connections, one of the NCPs must assume the role of the secondary link station. This will not be a problem since the NCP will support secondary-initiated nonactivation XID3 exchanges.

TG quiescing will be done by an NCP when the VR, used by its SSCP-PU session with a VTAM interchange node, is deactivated. Since other APPN nodes have no knowledge of VRs, they will still assume a path to this composite network node is available and continue to send BINDs. To avoid this, NCP will send a nonactivation XID3 with TG quiescing ON to inform adjacent T2.1 nodes. It is up to the adjacent (network) nodes to include the "quiescing" status in the network topology database and send topology database updates (TDUs) accordingly, informing other APPN network nodes in the network. The processing of the TDU for TG quiescing is part of base APPN network node support.

4.6 Connection Networks and Virtual Routing Nodes

A *shared-access transport facility*(SATF), such as a token-ring, allows direct connectivity between any pair of link stations attaching to the facility. Direct connectivity avoids session traffic being routed through intermediate network nodes but requires link definitions at a node for any node to which connectivity is required. See, as an example, (A) in Figure 27 on page 50. ENA and ENB have a direct link, and, although they need the assistance of a network node server to establish a session, the session data is exchanged directly between the two T2.1 nodes. No link has been defined between ENA and ENC and session data will always be routed through at least one intermediate network node.

The number of definitions required is proportional to the square of the number of T2.1 nodes on the SATF, which, as the number of nodes grows, will become very high. See, as an example, (B) in Figure 27 on page 50. Each node requires definitions to all other nodes.

Another drawback of increasing the number of direct links between APPN network nodes is that the number of topology database updates (TDUs) flowing in the network grows rapidly and may degrade the performance of the network. An APPN network node broadcasts TDUs to all adjacent network nodes and after having received a TDU forwards the TDU to all adjacent APPN network nodes, except the APPN network node from which the TDU has been received; for details see "Topology Database Updates" on page 64. As an example, see (C) in Figure 27 on page 50: NN1 will send TDUs to all network nodes which will then forward the TDU to all other network nodes. So, instead of receiving one copy, NN2, NN3 and NN4 receive the TDU three times. Flow reduction mechanisms prevent the network nodes from continuing to forward the TDUs.

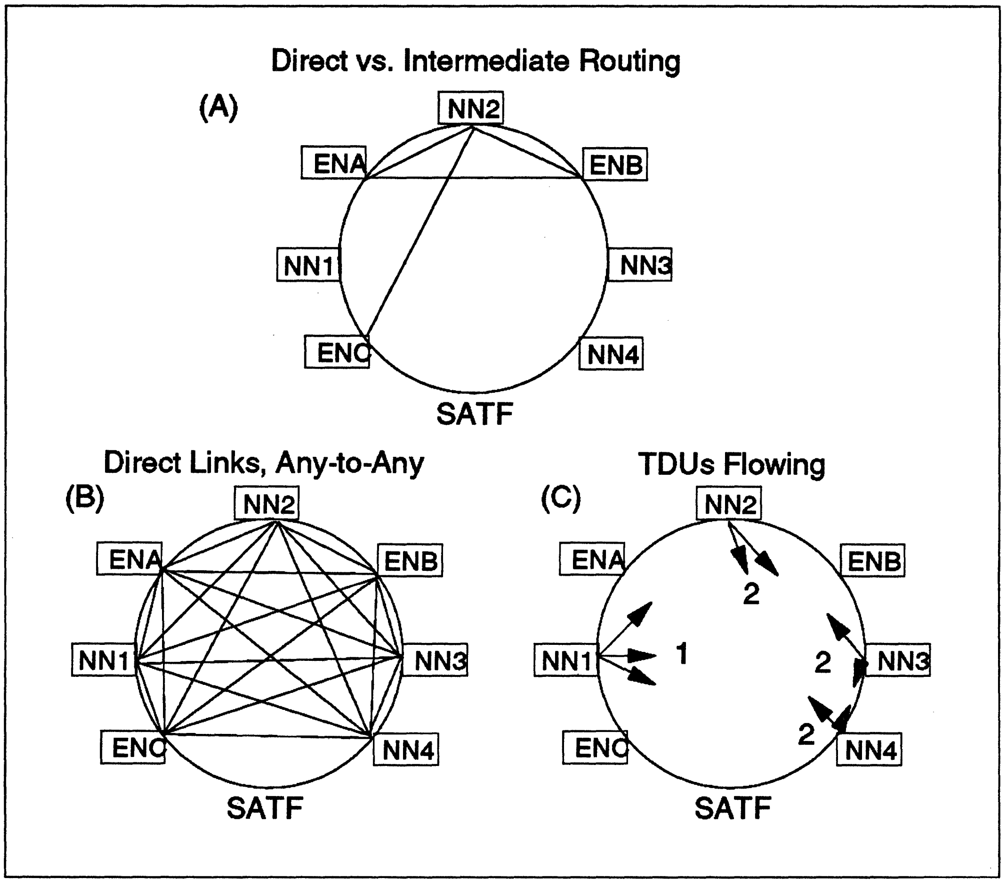

Figure 27. Shared-Access Transport Facility (SATF) without VRN

Thus, defining any-to-any links on a SATF provides optimal session routing but requires a high number of definitions and results in high volumes of TDUs flowing in the network.

To alleviate these problems, APPN allows T2.1 nodes to define a ***virtual routing node*** (VRN) to represent their attachment to a SATF. Session traffic between two T2.1 nodes that have defined the VRN can be routed "through" the VRN, without passing through any real network node. TDUs will never be exchanged with a VRN.

The SATF and the set of all nodes defined as having a connection to a common virtual routing node representing the SATF are said to comprise a ***connection network***. NOF defines a connection network (CN) and specifies a network-qualified name for it. This CN name is used as the CP name of the virtual routing node.

It is important to realize that session setup data and TDUs are routed through an APPN network using CP-CP sessions between adjacent nodes. As a VRN is no real node, T2.1 nodes cannot establish CP-CP sessions *with*, and also cannot establish CP-CP session *through* a VRN. Two T2.1 nodes can establish CP-CP sessions only if a direct link has been defined between the two nodes.

Session establishment between LUs owned by APPN end nodes, provided no direct link has been defined between the two APPN end nodes, requires assistance from a network node server. As the APPN end nodes cannot have CP-CP sessions with a network node server through a VRN, it is required that, besides from defining the connection to the VRN, each APPN end node also defines a link to its network node server. (A) in Figure 28 on page 51 shows the minimal definition requirements for an APPN end node. ENA has defined two connections: one to the VRN and one to its network node server NN2.

Network nodes cannot establish CP-CP sessions through a VRN; therefore, if two APPN end nodes (ENA and ENB) do not share the same network node server, session establishment between LUs on ENA and ENB is possible only if their network node servers have CP-CP "connectivity." See (B) in Figure 28. The latter is also required to allow session establishment between LUs on two network nodes. CP-CP connectivity between two network nodes requires that the two network nodes have defined a link between each other, and CP-CP sessions have been established between the two nodes, or that the two network node servers can exchange data via one or more intermediate network nodes with active CP-CP sessions between each pair of adjacent network nodes.

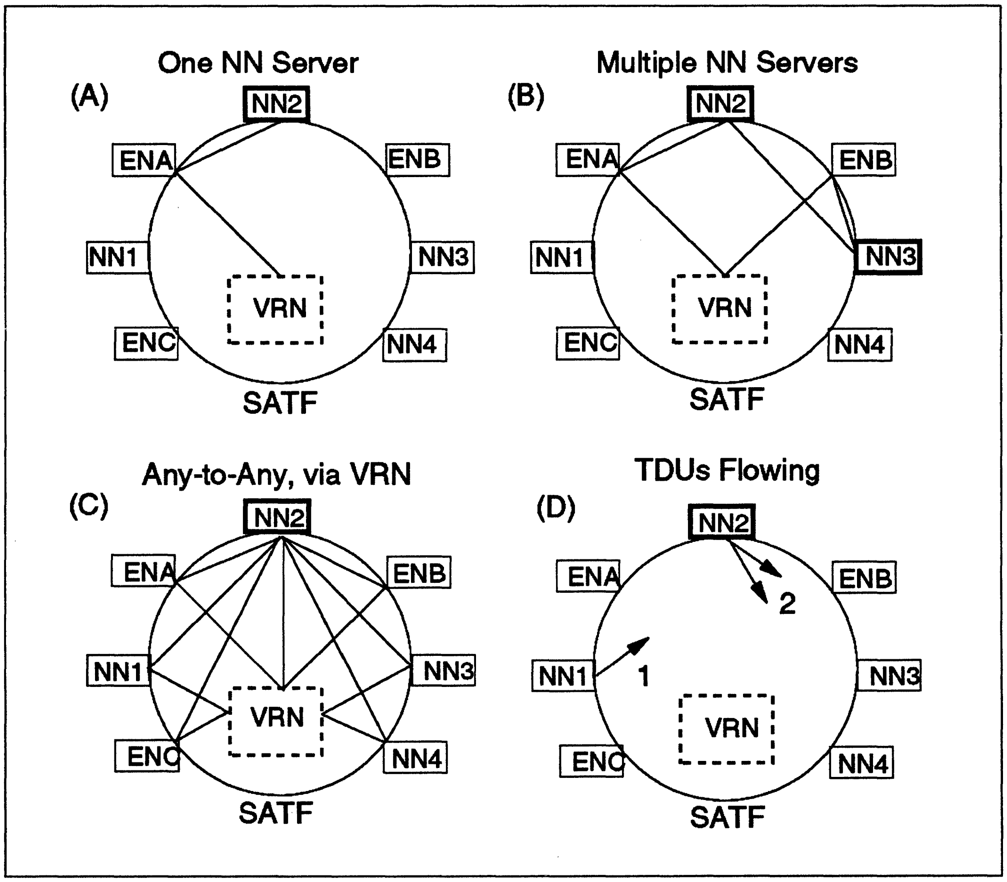

Figure 28. Shared-Access Transport Facility (SATF) with VRN

The benefits of defining a VRN can be seen in (C) in Figure 28. To have any-to-any connectivity without session data being routed through real network nodes, requires only two link definitions in each T2.1 node: one to the VRN and one to a network node server shared by all T2.1 nodes — depicted in the figure as NN2. NN2 is the only node that requires link definitions to all nodes. NN2 only assists in session setup; no session data will be routed through the node. For performance and backup reasons, more than one "common" network node server can be defined.

TDUs will flow only between network nodes that have established CP-CP sessions to each other. Link definitions in a network node can be limited to defining the link to the VRN and to at least one other network node. (D) in Figure 28 depicts a situation such that network nodes NN1, NN3, and NN4 have CP-CP sessions with only one network node, NN2. A TDU from NN1 will be sent to NN2 and, after receipt, forwarded to NN3 and NN4. So instead of receiving multiple copies, each network node receives only one copy of the TDU. When the CP-CP connectivity between network nodes is extended, then the number of TDUs flowing through the network will increase.

4.6.1 The Virtual Routing Node

A virtual routing node (VRN) is not a node, but it is simply a way to define an APPN node's attachment to a shared-access transport facility. It allows LU-LU session data to be routed without intermediate node routing through APPN network nodes. It reduces definition requirements and the number of TDUs flowing through the network.

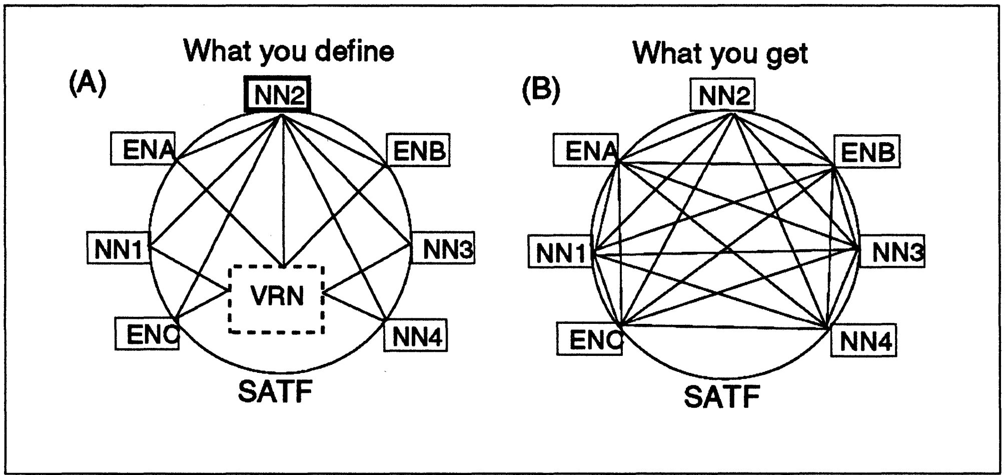

Figure 29. Virtual Routing Node

During LU-LU session establishment, the end nodes report their connection to the VRN, along with local DLC-signaling information such as SAP and MAC addresses on a token-ring, to their network server. The information is carried in the TG vectors, which will be explained in Chapter 5 “Topology and Routing Services” on page 55. The TG vectors describing the “link” to the VRN allows the network node server responsible for route computation, to determine that two nodes can communicate directly. The T2.1 node owning the primary LU receives the DLC-signaling information of the adjacent node, which it can use to activate a dynamic link, if none is already active, to the adjacent node. After the link is activated, a session BIND and RSP(BIND) will flow on this link.

Nodes attaching to a shared-access transport facility (SATF) may define direct connection to other nodes attached to the same SATF, define a *connection network (CN)*, or do both. An APPN end node must define at least a connection to its network node server.

Multiple CNs may be defined per port and a single CN may be defined on multiple ports. All adjacent link stations on a CN are *dynamic* link stations. All dynamic link stations associated with a particular port on the CN share the same characteristics.

The activation of actual connections through a connection network is triggered either by session services (as part of session establishment) or by a remote node. The node operator facility cannot activate connections through a connection network.

Chapter 5. Topology and Routing Services

The topology and routing services function (TRS) resides in every APPN network node and, in a reduced form, in every APPN end node and LEN end node.

In an APPN network node, TRS is responsible for creating and maintaining the class-of-service (COS) database and for creating and maintaining a copy of the ***network topology database***. The network topology database contains information on network node connections to VRNs and other network nodes. In an end node, TRS is responsible for creating and maintaining the class-of-service database (only if the end node supports the class-of-service / transmission-priority-field (COS/TPF) option set), and for maintaining the ***local topology database*** (which TRS in a network node also maintains). The local topology database contains information on connections involving end nodes: EN to EN, EN to VRN, and EN to NN.

During LU-LU session establishment, TRS will be invoked to compute an optimal route through the APPN network between the two nodes on which the LUs reside. TRS in an end node will use the local database to select possible TGs (single hop) from the end node to adjacent nodes. TRS in a network node will use the information provided by the two end nodes, together with the information in the network node's COS and network topology database to select an end-to-end route.

The scope of functions differs among node types. For LEN end nodes, they are very simple, while APPN network nodes can use large databases and sophisticated program logic.

Note: Directory Services is invoked to *locate* a session partner, and Topology and Routing Services is invoked to *compute* an optimal route to the session partner once it has been located.

5.1 Functional Overview

As can be seen from Figure 30 on page 56, TRS consists of three components:

- Topology Database Manager (TDM)

  The TDM is responsible for maintaining the topology and COS databases. The COS database is optional on an end node. On a network node, TDM maintains a network topology database, and on either node, TDM maintains a local topology database.

- Class-of-Service Manager (COSM)

  The COSM provides support for the COS/TPF function. It provides the capability to translate a *mode name* to a COS name and an associated transmission priority. Support of the COS/TPF function is a base function for network nodes but an optional function for end nodes.

- Route Selection Services (RSS)

  RSS is responsible for route computation. A *route* is an ordered sequence of nodes and TGs that represents a path from an origin node to a destination node.

Figure 30 depicts the T2.1 node functions that interface with TRS. The class-of-service manager (COSM) function is optional in end nodes, and the interface between Directory Services (CP.DS) and TRS exists only in APPN network nodes.

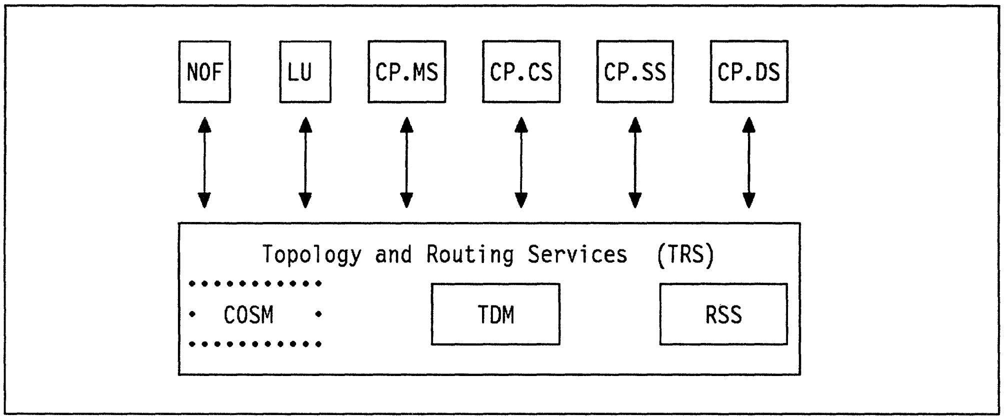

Figure 30. Overview of TRS Components and Protocol Boundaries

Topology and Routing Services (TRS) is initialized by the node operator facility (NOF). NOF passes the following parameters during initialization:

- Type of node
- CP name of this node
- Network ID of this node
- Indication if the COS/TPF function is supported

  The COS/TPF function allows a T2.1 node to translate a mode name to class of service (COS) name and an associated transmission priority (TP).
- The COS database file name
- The topology database file name

The main function of TRS, or actually RSS, is to compute an optimal route between two T2.1 nodes in an APPN network. RSS interacts with the two other TRS components, COSM and TDM, to obtain the necessary information before being able to perform a route computation.

To allow the computation of an optimum route several databases are maintained. Figure 31 on page 57 depicts the various types of databases involved in route calculation and how these are used by the components of topology and routing services.

Route computation is a coordinated activity between TRS components on several APPN nodes. In the following chapters we will give more detail about the information maintained by TRS and explain which components of TRS, and on which nodes, are invoked to make it possible to calculate an optimum route.

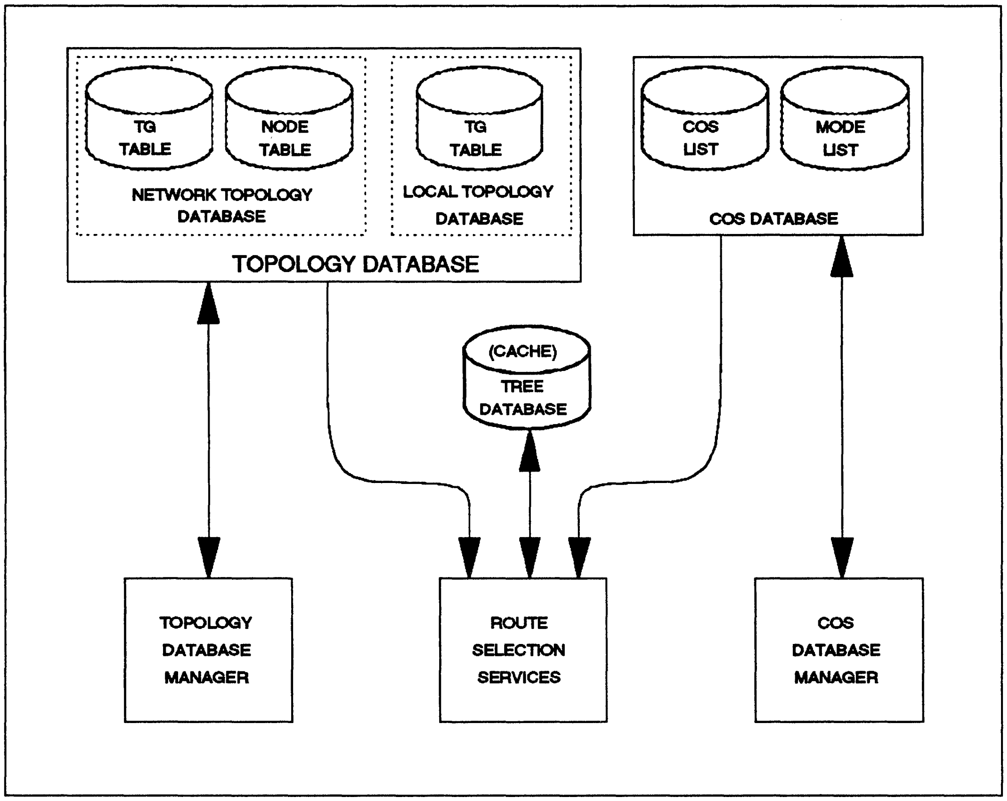

Figure 31. Databases and Subcomponents of Topology and Routing Services

5.2 Resource Characteristics

A *route* in an APPN network is an ordered sequence of nodes and TGs that represents a path from an origin node to a destination node. In order to calculate the optimal route, which means the physical path that best fits the user's requirement for an LU session path, the *actual* node and transmission group (TG) characteristics have to be compared with the *required* route characteristics.

For both TGs and T2.1 nodes, APPN has defined a set of properties that specify their characteristics. APPN also defines the values which can be assigned to each of these properties. To define the resource characteristics, two different data structures are used:

- Binary-valued properties such as operational/non-operational are encoded as property flags (bits).
- Multi-valued properties such as bandwidth are encoded as property indices (bytes). Some indices (such as cost per byte) can have any value within the range allowed, while others (such as security class) have defined values.

Note that some resource properties, for example the TG bandwidth, are static, while others, for example congestion, are dynamic and are periodically updated.

5.2.1 TG Characteristics

Table 4 depicts the TG characteristics. The values are either static (S) or dynamic (D) and can assume binary (B) or multiple (M) values.

Table 4. TG Characteristics

Property	**D(ynamic) or S(tatic)**	**B(inary) or M(ultiple)**
Cost per Byte	S	M
Cost per Connect Time	S	M
Security Level	S	M
Modem Class	S	M
Effective Capacity	S	M
User Defined-1	S	M
User Defined-2	S	M
User Defined-3	S	M
Propagation Delay	S or D	M
Quiescing	D	B
Operational	D	B

The TG characteristics are stored in the topology database and exchanged in topology database updates (TDUs) using control vector (CV) X'47'. For format details, see *Systems Network Architecture Formats*. Some fields are pointed out here.

Cost per Byte

Single byte value in the range 0 to 255 that expresses the relative cost of transmitting a byte over the associated TG. The units for cost-per-byte are user-defined.

Cost per Connect Time

Single byte value in the range 0 to 255 that expresses the relative cost of using a TG. The units for cost-per-connect time are installation defined and are typically based on the applicable tariffs of the transmission facility used by the TG.

Security Level

Indication of the level of security protection provided by the TG. The security values are architecturally defined to provide consistency across all networks. The default is X'01', indicating no security.

Effective Capacity

Highest bit-transmission rate that the TG will be allowed to obtain before being considered overloaded. The effective capacity, defined as a one-byte floating point value, is expressed in units of 300 bps.

User Defined 1,2,3

Up to three user-defined defined values in the range 0 to 255.

Propagation Delay

Time it takes for a signal to travel from one end of the TG to the other. Propagation delay, defined as a one-byte floating point value, is expressed in units of 1 microsecond.

5.2.2 Node Characteristics

Table 5 depicts the node characteristics. The values are either static (S) or dynamic (D) and can assume binary (B) or multiple (M) values.

Table 5. Node Characteristics

Property	**D(ynamic) or S(tatic)**	**B(inary) or M(ultiple)**
Gateway Function Support	S	B
Central Directory Support	S	B
Node Congested	D	B
Intermediate Routing Resources Depleted	D	B
Quiescing	D	B
Node Type	S	M
Route-Addition Resistance	D	M

The node characteristics are stored in the topology database and exchanged in topology database updates (TDUs) using control vectors (CV) X'44' and X'45'. For format details, see *Systems Network Architecture Formats*.

Gateway Function Support

This characteristic indicates that the node supports the gateway function.

Central Directory Support

This characteristic indicates that the node acts as a central directory server; see "Central Resource Registration (CRR)" on page 87.

Node Congested

This characteristic is set and reset by a node based upon one or both of the following congestion measures:

- Cycle utilization of the hardware
- Total buffer utilization (control blocks, message buffers, etc.)

When either of these measures crosses a specified threshold the congestion bit is set. It is not reset until the node is out of the congested state for all of the measures that the node maintains.

The threshold for resetting the bit should be significantly below the threshold for setting the bit. This is necessary to prevent the node from flooding the network with TDUs when the congestion measures are oscillating around the threshold for setting the bit.

Intermediate Routing Resources Depleted

This characteristic indicates whether the node's pool of resources is depleted to the extent that it cannot support additional routes that traverse it but do not terminate at it. The node monitors the set of session connector control blocks, which are required for intermediate routing.

Quiescing

The quiescing bit indicates whether the network operator wants the node to be drained of existing sessions passing the node prior to shutdown. When this bit is set, the node is excluded from subsequent route computations.

Node Type

This characteristic indicates the node type.

Route-Addition Resistance

This characteristic is a binary number between 0 and 255 used as a *node weight* during route calculation. The value is user defined and can be dynamically changed, but implementations may choose to keep it fixed for a node. The lower the value, the more likely it is that this node is used as an intermediate routing node.

This node characteristic could be used, for example, to assign low values to the set of nodes over which the network administrator wants the majority of traffic to flow. This then has the effect of defining a "backbone" network.

5.3 Topology Database

APPN networks consist of a "backbone" structure of one or more network nodes interconnected by TGs, known as ***intermediate routing TGs***, and TGs connecting end nodes to adjacent network nodes, virtual routing nodes, or end nodes, known as ***endpoint TGs***. For an example see Figure 32 on page 61; all TGs attached to either EN1 or EN2 are endpoint TGs, and all TGs between two adjacent network nodes are intermediate routing TGs.

Information about the backbone structure of the APPN network is kept within the *network topology database* which resides on every APPN network node. Information about endpoint TGs is contained within *local network topology databases* which reside on every APPN node or LEN end node.

The primary use of local and network topology databases is to enable route calculation when an LU residing in one APPN node wishes to establish a session with an LU residing in another APPN node. The topology databases enable TRS to determine all possible routes between two T2.1 nodes. The local topology database contributes the end node's TGs, while the network topology database supplies the information on network nodes and the TGs between them.

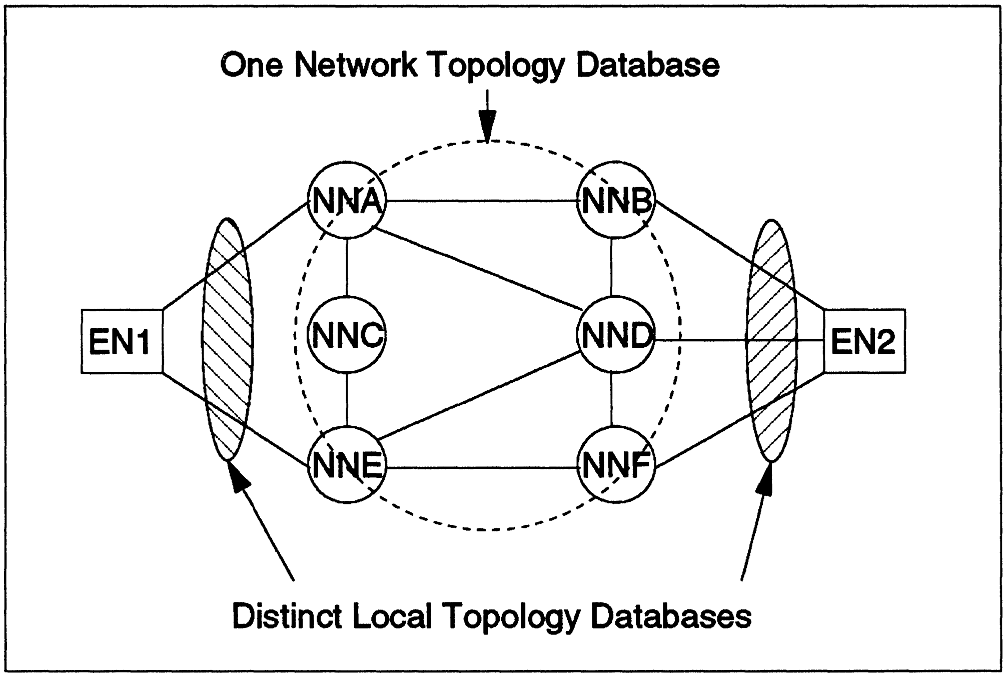
One Network Topology Database
NNA
NNB
NNC
NND
NNE
NNF
EN1
EN2
Distinct Local Topology Databases

Figure 32. Network and Local Topology Database

Figure 32 shows an example of an APPN network and how network topology information is maintained in local network topology databases, and in the network topology database.

5.3.1 Local Topology Database

Each end node maintains information about every endpoint TG attached to the end node itself. The information is kept in a database called the *local topology database*. The local topology database is created and maintained by TDM. It is not saved across IPLs and is rebuilt when the node initializes.

An APPN end node uses its local topology database:

1. When there is no CP-CP session to a network node server, for example, when a CP-CP session is being established.
2. To send information on endpoint TGs to its network node server to complement the network node′s knowledge during the route selection processes.
3. When establishing sessions to pre-defined LUs without the help of a network node server.

The local topology database contains information on *endpoint TGs*. An endpoint TG is not included in the network topology database.

End Node Topology Database Manager

The topology database manager (TDM) creates and maintains the topology database. Entries in the topology database are created automatically, when configuration services informs TDM about newly activated or changed TGs. The operator updates the topology database through configuration services. The topology database is searched by TDM when it receives a query from route selection services or from session services.

5.3.2 Network Topology Database

Each network node maintains information about all network nodes and all intermediate routing TGs within the APPN network in a database called the *network topology database*. The network topology database does not include information on LEN end nodes, APPN end nodes, or the TGs attached to them. It includes information only on network nodes and their connections to virtual routing nodes and other network nodes.

The network topology database is fully replicated on all APPN network nodes. APPN protocols for the distribution of network topology information ensure that every network node is provided with a complete view of the network backbone topology.

In addition, the local copy of the network topology database contains information on the other local TGs of the APPN network node itself. This information is kept locally only and not sent to adjacent network nodes.

The network topology database is created and maintained by TDM and saved across IPLs (***safe-store of network TDB***).

5.3.3 Structure

Node Table

The network topology database contains the following information about the network nodes in the network:

- Network ID
- CP name
- Node characteristics, summarized in "Node Characteristics" on page 59
- Pointer to the node-attached TGs
- Resource sequence number (RSN); see also page 67

TG Table

Both, network and local topology databases, contain information about TGs. TG database entries, consisting of a TG vector and a TG record, are direction dependent, and two entries exist for each TG: one entry describes the TG in one direction, and another entry describes the TG in the opposite direction.

The TG record contains the following information:

- Whether CP-CP sessions are supported
- Pointer to TG vector
- Pointer into weight index structure (see below)
- Status (active or inactive)

The TG vector contains the following information:

- TG number
- Partner-node CP name
- Partner-node type:

  Real or virtual routing node (VRN)
- TG characteristics, as described in "TG Characteristics" on page 58
- Resource sequence number (RSN); see also page 67
- DLC-signaling information

  For TGs to virtual routing nodes (VRNs), DLC-signaling information is maintained. For example, for a token-ring attached node the MAC address of the node is stored.

  The DLC-signaling information is used to allow a station to dynamically establish a connection through a VRN to a remote station when using a shared access transport facility (SATF). For details see "Connection Networks and Virtual Routing Nodes" on page 49.

The Weight Index Structure

TG weights have to be calculated to compute the optimum route between an origin and destination endpoint. This can be a time-consuming process, which has to be repeated for each session setup. For performance reasons, APPN provides an option to cache the TG weights. This option is called the TG weight index structure.

Refer to *SNA Type 2.1 Node Reference* for details about the weight index structure.

5.4 Network Node Topology Database Manager

The network node topology database manager (NNTDM) is a component that resides in every network node and is responsible for maintaining the local copy of the network topology database.

Note: Each T2.1 node is considered the *resource owner* of the node itself and the locally attached TGs. A TG has two resource owners. Nodes on either end of a TG maintain direction-dependent information in the topology database.

Each NNTDM creates and broadcasts topology database updates (TDUs) about the node itself and locally-attached intermediate routing TGs to adjacent network nodes using its CP-CP sessions with adjacent network nodes. NNTDM stores information obtained from TDUs received from adjacent network nodes in its copy of the network topology database, and forwards the TDU to adjacent network nodes. This allows every NNTDM in the network to maintain a consistent copy of the network topology database.

5.4.1 Topology Database Updates

The NNTDMs in two adjacent nodes can start to exchange TDUs after the CP-CP sessions between the two nodes have been established. The TDUs contain:

- Resource identifier
- Resource characteristics
- Resource sequence number (RSN)

When a network node connects to the network for the first time, the network node has no knowledge of remote resources and has only information about local resources. The network node will receive a copy of the current network topology database from the adjacent network node and send TDUs with information about the node itself and locally attached intermediate-routing TGs and connection networks (VRNs). The adjacent network node receiving this information will broadcast these TDUs further into the network.

When two network nodes reconnect after having been temporarily disconnected, only the changed information within the local copies of the network topology database will be exchanged. See the discussion about FRSN on page 67.

Whenever a network node detects a change in its own state, or in the state of locally attached intermediate-routing TGs, it updates the entry for the TG in its local copy of the network topology database, increments the RSN for that resource to the next even value, and informs its adjacent network nodes by broadcasting TDUs.

A timer interval is assigned to every resource entry in the local copy of the network topology database, to allow the resource to be discarded if no information about the resource has been received for 15 days. This process is known as ***garbage collection***. Every NNTDM broadcasts TDUs containing local information every five days, to prevent other network nodes from discarding valid information.

Processing Topology Database Queries

Directory services, session services, and route selection services interface to the NNTDM in order to obtain information from the topology database. Whenever NNTDM updates or deletes a resource, it notifies the route selection services (RSS) component of topology and routing services, to enable RSS to update routing information that has been cached.

Virtual Routing Node (VRN)

Because a VRN is merely a representation of a connection network and does not really exist, it cannot broadcast resource updates. For TGs that connect a network node to a VRN, the network node broadcasts their TG and virtual routing node information over attached intermediate-routing TGs. The node characteristics for a virtual routing node have architecturally defined default values.

Processing Topology Database Updates (TDUs)

When a network node receives a TDU, the TDU may contain resource information for local or remote resources. The processing of the resource information on a network node depends on the resource sequence numbers (RSNs) associated with each resource in both the TDU and network topology database.

Local resource information in TDU: As the owner of the resource the receiving network node has the responsibility of providing the network with valid information on that resource, especially if it detects an inconsistency between information received in a TDU and information stored in its topology database or if some other node indicates an inconsistency for a resource by broadcasting a TDU with an odd RSN.

- If the RSN in the TDU is equal to the RSN in the database and the resource information received is identical to the information in the database, then
  - the network node discards the TDU.
- If the RSN in the TDU is equal to the RSN in the database but the resource information received is not identical to the information in the database, then
  - the network node builds a new TDU with an even RSN that is greater than the RSN received and using the information from its database, which, being the owner of the resource, it knows is valid.
- If the RSN in the TDU is less than the RSN in the database, then
  - the network node builds a new TDU with the RSN from the database and using the information from its database.
- If the RSN in the TDU is greater than the RSN in the database, then
  - the network node builds a new TDU with an even RSN that is greater than the RSN received and using the information from its database.
- If the resource, although specified in a TDU as a resource owned by the receiving network node, is not contained in its local copy of the topology database, then
  - it is stored there and a new TDU is created and broadcast to all adjacent network nodes. In this new TDU, the RSN is incremented to the next even value greater than the originally received RSN and the state of the resource is set to “inoperative.” This will prevent the resource's inclusion in route calculations, and will mark it available for garbage collection done regularly in all nodes.

Whenever a new TDU is built, it is then broadcast to all adjacent network nodes to ensure that all copies of the network topology database are again synchronized.

Remote resource information in TDU: Being not the owner of the resource the receiving network node, in this case, assumes that the TDU carries valid information on the resource which was provided by the owner of the resource. But, nevertheless, it checks the received information and RSN against the information and RSN in its database.

- If the resource is not currently contained in the network node's database, then
  - the network node stores the information from the TDU (including the RSN) in its database and rebroadcasts the TDU to all adjacent network nodes.
- If the RSN in the TDU is greater than the RSN in the database, then
  - the network node stores the information from the TDU (including the RSN) in its database and rebroadcasts the TDU to all adjacent network nodes.
- If the RSN in the TDU is equal to the RSN in the database and the resource information received is identical to the information in the database, then
  - the network node discards the TDU.
- If the RSN in the TDU is even and equal to the RSN in the database but the resource information received is not identical to the information in its database, then
  - the network node builds a new TDU using the information from its database. In the new TDU the RSN from the received TDU is incremented by one, thus forcing the RSN to an odd value. This is used to signal other network nodes that the information about the resource is inconsistent, and that the resource should not be included in route calculations. The owner of the resource then has to resolve the inconsistency and provide the network with valid information about the resource.
- If the RSN in the TDU is odd and equal to the RSN in its database, then
  - the network node discards the TDU.
- If the RSN in the TDU is less than the RSN in the database, then
  - the network node discards the TDU. A new TDU is then built with information from the database (including the RSN) and broadcast to all adjacent network nodes.

5.4.2 Flow Reduction Considerations

The number of topology database updates (TDUs) that flow between network nodes may be a concern in large networks. Several mechanisms have been put in place to reduce the number of TDUs flowing through the network.

1) TDUs for Network Nodes and Intermediate Routing TGs Only

TDUs flowing through the network will never contain node information for end nodes or TG information for endpoint TGs. The information that network nodes maintain in the network topology database about local endpoint TGs is never broadcast.

2) TDUs for Connection Networks

TDUs are not broadcast when a TG to a network node is activated or normally deactivated across a virtual routing node. Only if such a TG fails (abnormal deactivation) are TDUs sent, in order to exclude this TG from route computation.

3) TDUs only "Forwarded"

When a network node receives a topology database update (TDU), it will forward the TDU to all its adjacent network nodes, except the network node from which the TDU has been received.

4) Resource Sequence Number (RSN)

An RSN is associated with the current information about each node and TG in the network topology database; this RSN is assigned by the network node that "owns" the particular resource. A network node owns the node definitions for itself, and the TG characteristics in the direction of adjacent network nodes.

Whenever a network node detects a change in the state of a locally owned resource it increments the RSN to the next even value. It then creates a TDU including the new RSN and broadcasts it to all its adjacent network nodes.

The use of RSNs in TDUs and the network topology database allows a network node to determine whether resource information has been received before. A TDU is discarded and not rebroadcast if the RSN in the TDU is equal to the RSN in the topology database and the information in the TDU is the same as in its database, preventing endless retransmission of resource information. See also "Processing Topology Database Updates (TDUs)" on page 65.

The RSN is an unsigned even integer in a circular number space. The range is 2 to 2^{32} - 1. Odd values, also known as "inconsistent sequence numbers," are used to signal other network nodes that the information about a resource is inconsistent and will trigger error recovery.

5) Flow Reduction Sequence Number (FRSN)

Every network node in the network maintains a separate flow reduction sequence number (FRSN) for each of its adjacent network nodes, and for either direction (receive and send FRSN). FRSNs are incremented each time a TDU is sent. It is included in each TDU and stored with every topology database entry included in a TDU. FRSNs are associated with TDUs, as opposed to RSNs which are associated with resources.

Each network node keeps the FRSN to prevent unnecessary transmission of TDUs. When two network nodes reconnect and establish CP-CP sessions, FRSNs are exchanged to decide what information has not yet been received by the other end. One or more TDUs will be built and sent including all entries of the network topology database that, according to their FRSN for the respective node, have not been previously sent. This insures that only information will be sent that has changed during the time the two nodes were disconnected.

The FRSN is an unsigned integer in a circular number space. The range is 1 to 2^{32} - 1. The value 0 is used to indicate that a network node is requesting a copy of the adjacent node's network topology database.

5.5 Class-of-Service Database

The COS database and the class-of-service manager (COSM) exist in all APPN network nodes and in those APPN end nodes that support the COS/TPF function. The ***COS/TPF function*** is the capability to translate a mode name to a COS name and an associated transmission priority.

The COS database provides the information that allows TRS to select an optimal route between two session endpoints. An optimal route is the physical path that most closely matches the transmission requirements for a specific LU-LU session.

The COS database includes:

- List of mode names

  Each entry contains a mode name and a pointer to the corresponding COS name.

- List of COS names

  Each entry contains a COS definition, which represents one or more sets of acceptable characteristics to which the actual TG and node characteristics are compared, the transmission priority, and the weight index value assigned to the COS.

- Weight index structure

  This structure allows actual TG weights to be computed once and then stored, rather than having to be computed each time a route is requested.

The COS database is maintained independently at each node and can be updated using the node operator facility (NOF).

5.5.1 Mode Name

When an LU starts a session, it uses a mode name to indicate the session characteristics, and the class of service (COS) that it requests for the session. COSM will use the mode name to obtain a COS name from the COS database, allowing route selection services (RSS) to select an appropriate route.

In the COS database, COS and mode entries exist. Each mode entry, referenced by a mode name, contains a pointer to a corresponding COS entry.

The ability to specify a mode name at session establishment time provides a considerable amount of flexibility. IBM provides several pre-defined mode names and corresponding COS names and COS definitions. See, "SNA Defined Modes and Classes of Service" on page 80.

5.5.2 Class of Service (COS)

For each COS, the COS database contains:

- COS name
- Transmission priority:
  - High
  - Medium
  - Low

  APPN distinguishes four transmission priorities. The highest transmission priority, "network" priority, cannot be specified in the COS database and is reserved for network control messages or CP-CP sessions.
- Several rows of COS definitions for TGs, consisting of:
  - Ranges (pairs of high and low values) for the following TG characteristics:
    - Cost per Byte
    - Cost per Connect Time
    - Effective Capacity
    - Propagation Delay
    - Security Level
    - User Defined-1
    - User Defined-2
    - User Defined-3
  - A weight field
- Several rows of COS definitions for nodes, consisting of:
  - Ranges (pairs of high and low values) for the following node characteristics:
    - Route-addition resistance
    - Congestion
  - A weight field

As shown in Figure 33 on page 70, each COS entry in the COS database consists of a transmission priority and one or more rows of TG characteristics. Each row indicates a *range* of acceptable values for each of the TG characteristics. Each row has an associated weight.

During route calculation, RSS uses the TG characteristics to decide which TGs are acceptable and which are not, for this class of service. A TG is considered *acceptable* if all the *actual* TG characteristic values, obtained from the topology database, fall within the range of *required* TG characteristics obtained from the COS database. A TG is considered *unacceptable*, if at least one of the actual TG characteristics falls outside the range of the required TG values.

A COS may define multiple rows of *required* TG characteristics with a weight assigned to each of the rows. The TG weight is a quantitative measure of how well the actual TG characteristics satisfy the session requirements specified by the COS definition. If a TG satisfies the criteria specified by a row of TG characteristics within a COS definition, then the weight of this specific row is used as a TG weight for route computation. If a TG is considered acceptable for more than one row, the lowest weight is assigned to the TG. If a TG does not satisfy the criteria specified by any row of TG characteristics, the TG is assigned an infinite weight.

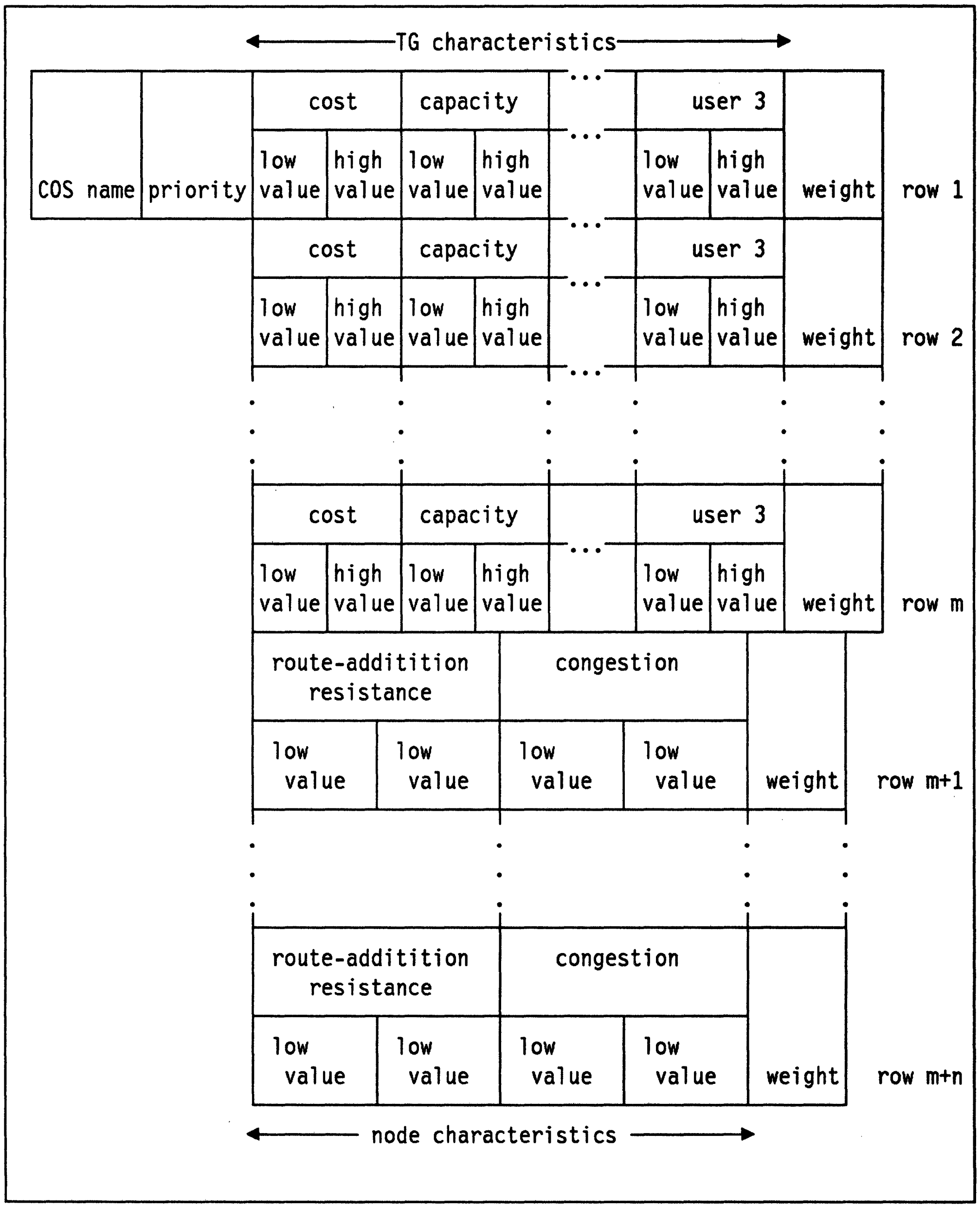

Figure 33. COS Entry with m Rows TG and n Rows Node Characteristics.

The figure shows for a given COS the transmission priority and three rows of required TG characteristics. For each of the rows a weight has been included.

More information about TG weight assignment and route calculation can be obtained from "Route Computation: Overview" on page 73.

Note: Instead of defining fixed TG weight values per row, implementations may allow the invocation of a function that calculates the TG weight value, giving the user more control over the TG weight-assignment process.

5.6 Tree Database

Building and maintaining a tree database is an optional function. It allows an APPN network node to cache *optimal* routes from the APPN network node to all other network nodes (***tree caching***). (The term *tree* or *sink tree* is obtained from graph theory. Graph theory is a mathematical discipline that, among other things, allows optimal route calculation in a network.) For a given node, the optimal routes to all other nodes in a network can be represented in a tree-like (no loops !) structure. See, for example, (C) in Figure 34. Within an APPN network, routing trees are COS dependent, and the tree database contains one tree, per root network node, per class of service.

(C) in Figure 34 shows two tree structures for network node A derived from the APPN network depicted in (A). The network node A, for which the routing tree is valid, is shown at the top.

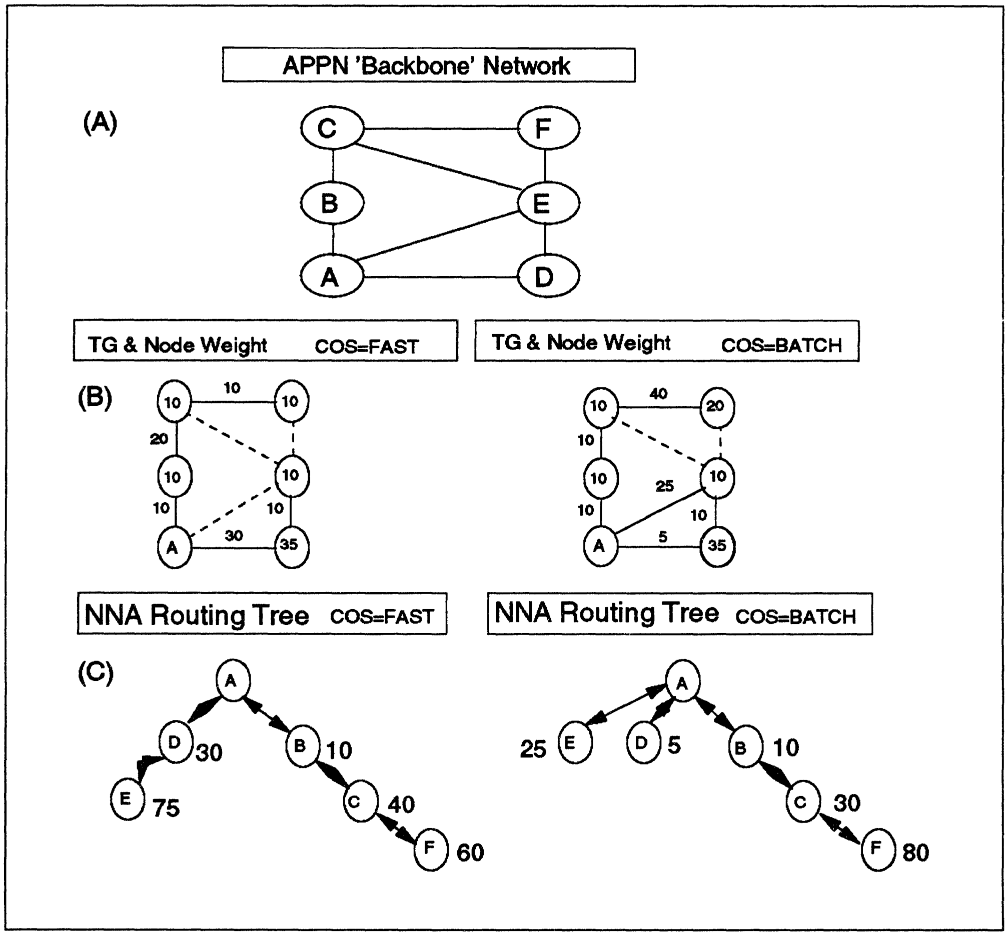

Figure 34. Tree Database Kept at network node A for Different COS

A tree is computed from the perspective of the node; see network node A in Figure 34, drawn at the top. It is uni-directional (the optimum routes are from top to bottom) and includes network nodes and their connecting TGs. For each of the network nodes, a route weight is stored for the route, from the "top" node to the network node itself.

For example, the shortest path from network node A to network node F for COS=FAST has a weight of 60 and uses network node C and network node B as intermediate network nodes using the TGs drawn. Note that, although not shown in the figure, the actual TG numbers between adjacent nodes are included in the tree database.

The route weight is the sum of the weights assigned to each of the components, TGs and nodes, that make up the route. Route weights are COS dependent because TG weights are COS dependent. Route weights are also direction dependent, as for each TG two sets of TG characteristics have been defined, one in either direction. Each node has a node weight, equal to the "route-addition resistance" value assigned to the node; see "Node Characteristics" on page 59.

Note that the tree database contains optimal routes between network nodes. When an optimal route has to be computed between two APPN end nodes, route selection services first checks its tree database, to see if routes already have been computed between network nodes adjacent to both end nodes. If so, route selection services uses this routing information, together with routing information obtained from the end nodes, to compute the optimum end-to-end route. If no information can be obtained for the tree database, route selection services computes new trees and stores them.

The tree database is derived from the network topology database and the COS database. Whereas the network topology database is replicated throughout all network nodes, the tree database is unique for each node. For each COS, a routing tree can be calculated ***from*** the node at the top to each network node within the tree.

The tree database is introduced for performance reasons. It saves the overhead of re-computing the optimal tree for each route request. The tree database can be kept in cache. When no tree database is maintained, trees have to be computed from scratch for each route request.

Trees may be removed when the database is full, after topology changes, after an implementation-defined number of uses, or for load distribution among equally weighted routes. The latter may be done with the expectation that equally weighted routes are randomly selected each time the tree is recalculated (***randomized route calculation***).

5.6.1 Routing Trees

Routing trees represent the *least-weight*, or *shortest path*, from the node at the top to each network node within the tree. Without going into much detail, a few interesting features should be pointed out:

- A routing tree can be computed partially. As soon as the requested destination has become part of the tree, the computation can be stopped.
- The endpoint TG vectors of end nodes can be added to an existing routing tree allowing fast computation of end-to-end routes. See (B) in Figure 35 on page 73.
- A routing tree is expandable (***incremental updates to tree***). Adding a network node to a (partial) tree can be done, in many cases, without having to compute a new tree.
- If intermediate TGs become (in)active, in most cases whole "branches" accessible through another TG can be moved to another part of the routing tree, allowing fast re-computation of the tree. See (C) in Figure 35 on page 73.

- The time to compute a routing tree is proportional to the number of TGs, while the number of network nodes is less important (of course, more network nodes means more TGs).

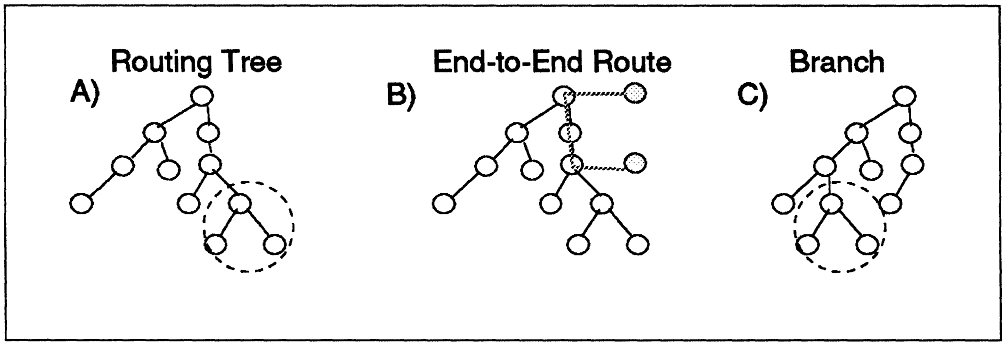

Figure 35. Routing Trees

5.7 Route Computation: Overview

APPN networks consist of a backbone structure, which consists of network nodes interconnected by TGs, known as *intermediate routing TGs*. End nodes are connected to this backbone structure. The TGs connecting the end nodes to adjacent network nodes or end nodes are known as *endpoint TGs*.

A *route* in an APPN network is an ordered sequence of nodes and TGs that represent a path from an origin node to a destination node. To compute the *optimum* route between two nodes in an APPN network requires a number of things:

1. Obtain the *required* route characteristics
2. Obtain all the resources, TGs and network nodes, that make up possible routes
3. Exclude all resources, TGs and network nodes, from route computation if use of the resource is not acceptable
4. Calculate all possible routes and select the optimum, or most preferred, route

1) Obtain Required Route Characteristics

Essential in APPN route calculation is that an optimum route be calculated on a session basis. A route between two APPN nodes that is the optimal route for one session can be far from optimal for a second session.

At session establishment time, an LU indicates, by using a mode name, which type of route is required for the session requested. For example, by using the mode name "FAST" an LU could indicate that a route is required for an interactive application requiring high, and predictable, response times. The mode name "BATCH" could be used for bulk traffic for which response time requirements are less important than throughput requirements.

To enable TRS to select routes on the basis of the mode name given, TRS, or actually COSM, will use this mode name to obtain a class of service (COS) and a transmission priority from its COS database. Each COS contains one or more sets of *required* TG characteristics, expressed in terms of costs, propagation delay, capacity, and so on. For details, see "Class of Service (COS)" on page 69.

To allow route computations, a weight factor has been assigned to each set of required TG characteristics.

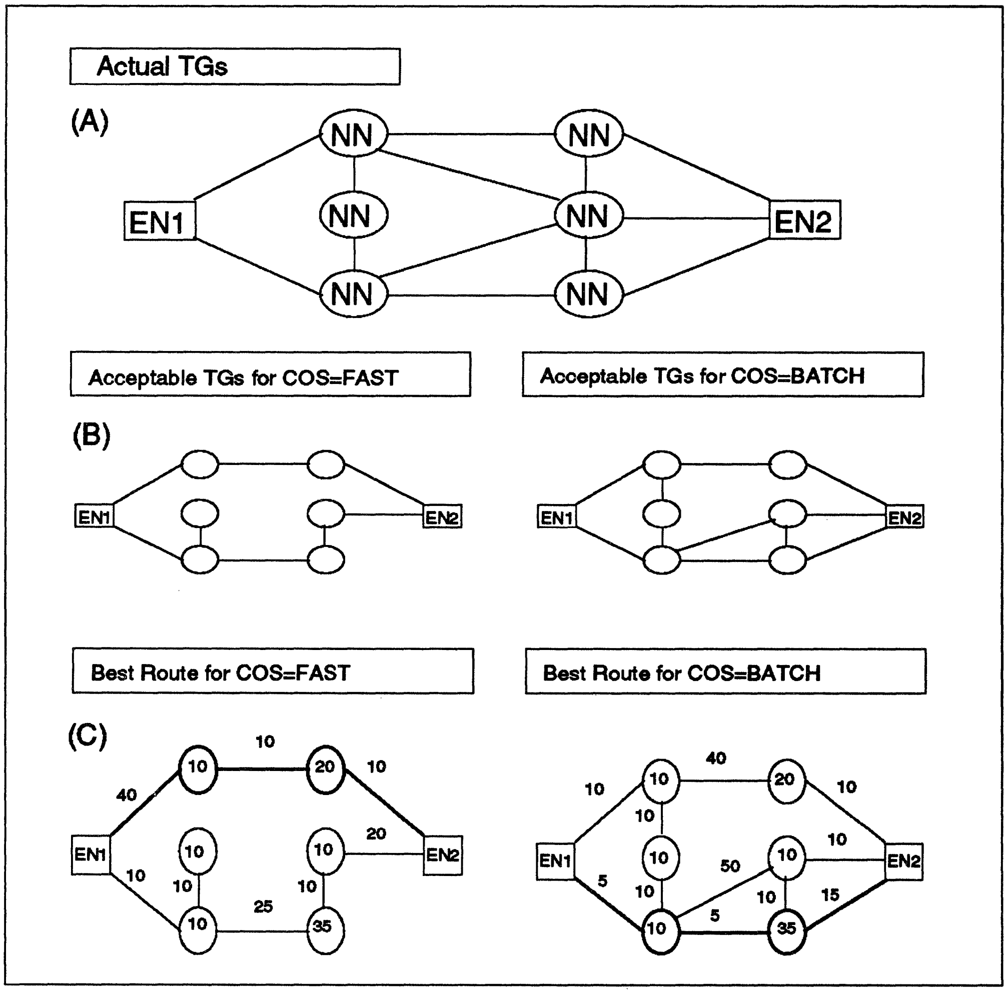

Figure 36. COS-dependent Route Selection

2) Obtain All TGs and Network Nodes That Make Up Possible Routes

To enable TRS to calculate all possible routes between two endpoints, information is required from the network topology database and from the local databases at each of the endpoints. To be specific:

1. Information about network nodes and intermediate routing TGs.

   This information is obtained from the network topology database at the origin network node server.

2. Information about the endpoint TGs of the origin APPN end node to adjacent network nodes and virtual routing nodes.

   This information is obtained from the local topology database at the origin end node. (In the case of a LEN or unauthorized APPN end node, the endpoint TG information is obtained from the origin network node server's local topology database.)

3. Information about the endpoint TGs of the destination APPN end node to adjacent network and virtual routing nodes and to the origin end node.

   This information is obtained from the local topology database at the destination end node. (In the case of a LEN or unauthorized APPN end node, the endpoint TG information is obtained from the destination network node server's local topology database.)

Note: Information about endpoint TGs of the destination node to the origin end node allows TRS to compute direct routes if two end nodes are adjacent.

As an example see (A) in Figure 36. In order to calculate a route from EN1 to EN2, endpoint TGs are obtained from the local topology databases on both EN1 and EN2. Information about intermediate TGs and network nodes is obtained from the network topology database, a copy of which is maintained at every NN.

3) Exclude All Non-Acceptable TGs and Network Nodes

Before starting to compute all possible routes, TRS will exclude the network nodes and TGs that are considered not acceptable for the desired route.

Excluding a node from route calculation depends on the characteristics of the node as maintained in the network topology database. Nodes may, for example, be excluded from route calculation if the network node's intermediate routing resources are depleted.

A TG is excluded from route computation if, for a given COS, TRS has assigned an infinite weight to the TG. The TG weight assignment process is COS dependent. TRS will assign an infinite weight to the TG if the actual TG characteristics do not match the TG characteristics defined in the COS. For details see "Class of Service (COS)" on page 69.

After the exclusion of unacceptable TGs and network nodes, all possible routes can be calculated.

(B) in Figure 36, shows a graph of the network after all unacceptable resources have been removed. Note how this graph depends on the COS used during the route calculation.

4) Compute All and Select the Optimum Route

To compute the optimum route requires a method to quantify the resources, TGs and nodes, that make up a route. APPN architecture allows TRS to assign a weight to each node and TG. By adding up all weights, a route weight can be calculated and the optimum route, the route with the *least-weight*, selected. This route is also known as the *shortest path*.

The weight of a network node is obtained from the network topology database. The weight assigned to every network node is fixed and is not session- or COS-related.

The weight factor assigned to each individual TG is COS related. TRS assigns COS-dependent TG weights using the TG characteristics from topology databases and COS definitions from the COS database.

After TRS has computed a weight for each of the components that make up possible routes, the optimum (*least-weight*) route can be selected. (C) in

Figure 36 on page 74, shows the assigned resource weight and the dependency between *optimum* route and COS.

To compute an optimal route between two end nodes in an APPN network requires the coordinated invocation of TRS, or more precisely the components of TRS — COSM, RSS, and TDM — on several APPN nodes. To understand in what order TRS components are invoked, and on which nodes, requires some insight into how LU-LU sessions are established in an APPN network.

5.7.1 Session Establishment and TRS

This section gives a brief and simplified description of how LU-LU sessions are established in an APPN network, in order to point out which components of TRS are invoked in the various stages of session establishment.

In base APPN, LU-LU session establishment is triggered by the LU known as the *originating LU (OLU)*. The requested session partner is referred to as the *destination LU (DLU)*. Both LUs are controlled by a control point (CP), called the CP of the OLU, CP(OLU), and the CP of the DLU, CP(DLU).

The essential difference between a LEN end node and an APPN end node is the fact that the APPN end node maintains CP-CP sessions with its network node server and is able to request assistance in session establishment from its network node server. Network node servers are referred to as either NNS(OLU) or NNS(DLU). Benefits of the CP-CP session between an APPN end node and its network node server are: locations of DLUs can be learned dynamically and routes are computed that are truly end-to-end. Note that an APPN end node that does not maintain CP-CP sessions with a network node server should be, for the topics discussed in this chapter, considered as a LEN end node.

We limit ourselves to a description of the two cases where both LUs reside on either LEN end nodes or on APPN end nodes.

LUs Residing on LEN End Nodes

Session establishment and invocation of TRS components between two LUs residing on a LEN end node will be explained using Figure 37 on page 77.

The figure shows a case where the CP(OLU), which does not support Locate search requests relies on NNS(OLU) to build a Locate request and to find the DLU. In this example, the route taken by the Locate request is:

```
NNS(OLU) .. -> .. NNS(DLU)
```

If NNS(OLU) and NNS(DLU) are not adjacent, the Locate request will be routed through intermediate network nodes.

The route taken by the BIND request is:

```
CP(OLU) -> NNS(OLU) .. -> ..  NNS(DLU) -> CP(DLU).
```

If NNS(OLU) and NNS(DLU) are not adjacent, the BIND request will be routed through intermediate network nodes. Because of optimum route calculation for this session, the session data (including the BIND) may follow a different route from NNS(OLU) to NNS(DLU) than the Locate search request. Session data is routed along the optimum route, and the Locate search flows along the shortest (minimal hops) path.

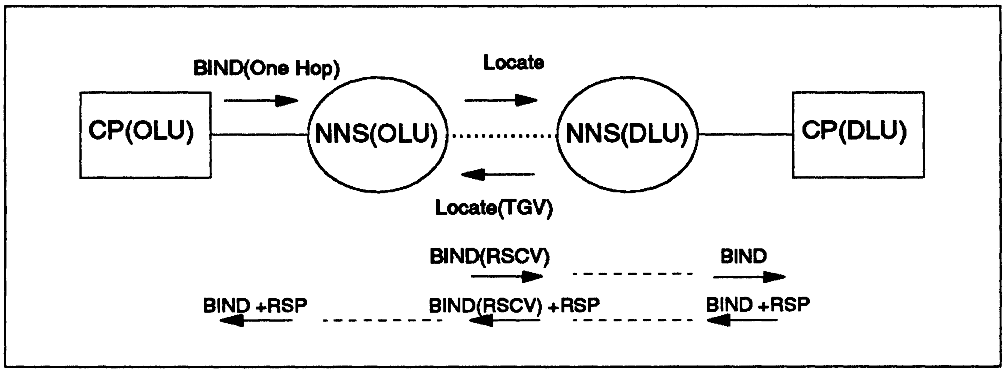

Figure 37. Session Establishment between LUs on LEN End Nodes (Simplified)

On CP(OLU) the DLU must be explicitly defined and is assumed to reside on NNS(OLU). Two components of TRS will be invoked on CP(OLU), namely:

1. COSM, optionally, to perform the mode name to COS name mapping.
2. TDM to select the endpoint TG to NNS(OLU).

   Note: LEN end nodes do not support parallel TGs.

A BIND, optionally containing the COS name, will be sent to NNS(OLU) on the TG selected.

NNS(OLU) will send a *Locate* search request to NNS(DLU). The Locate search request will be a directed search or a broadcast search, depending on NNS(OLU) having knowledge of NNS(DLU) being the network node server of CP(DLU). For details, see "Network Searches" on page 92.

On NNS(DLU) the DLU must be explicitly defined, because NNS(DLU) maintains no CP-CP sessions with CP(DLU) and, therefore, the location of DLU cannot be learned dynamically. NNS(DLU) does not forward the Locate request to CP(DLU); instead, TDM will be invoked to obtain the endpoint TG between NNS(DLU) and CP(DLU). NNS(DLU) returns a positive Locate reply (with a Found) and within the reply, the TG vector describing the connection to CP(DLU).

On NNS(OLU), COSM is invoked to perform the mode-to-COS name translation (if not already done), and to obtain the contents of the COS entry from the COS database. TDM will be invoked to obtain TG and node characteristics from the network topology database. Using this information, the TG on which the BIND has been received from CP(OLU), and the endpoint TG information returned by NNS(DLU), RSS computes an "optimum" route between CP(OLU) and CP(DLU). The computed route will be added, in the form of a route selection control vector (RSCV), to the BIND which is forwarded along the computed route to CP(DLU).

CP(DLU) receives the BIND request and returns a positive response.

Note: The route on which the session data flows is not necessarily an end-to-end optimal route as NNS(OLU) and NNS(DLU) will always be part of the route selected.

LUs Residing on APPN End Nodes

Session establishment and invocation of TRS components between two LUs residing on APPN end nodes will be explained using Figure 38 on page 79:

This figure shows the case where the CP(OLU), since it supports Locate search requests, sends a Locate search request to its network node server, NNS(DLU), when a session is required. In this example, the route taken by the Locate request is:

```
CP(OLU) -> NNS(OLU) .. -> ..  NNS(DLU) -> CP(DLU)
```

If NNS(OLU) and NNS(DLU) are not adjacent, the Locate request will be routed through intermediate network nodes.

The route taken by the BIND request is:

```
CP(OLU) -> NN .. -> .. NN -> CP(DLU)
```

The BIND is routed along an end-to-end optimum route that may be different from the route taken by the Locate request. The network nodes in the session path are optional; if links exist between the end nodes or both are attached to the same connection network (VRN), session data (including BIND) may be routed directly between the two nodes. Note, as part of the session establishment, CP(OLU) forwards all endpoint TGs to adjacent network nodes, and CP(DLU) all endpoint TGs to adjacent network nodes and possible endpoint TGs to CP(OLU). The latter, if available, allows NNS(OLU) to compute a direct route between the end nodes.

Before sending a Locate request to NNS(OLU) two components of TRS will be invoked on CP(OLU), namely:

1. TDM, to obtain endpoint TGs from CP(OLU) to adjacent network nodes and end nodes

   Note: CP(OLU) includes all TGs leading to adjacent network nodes or connection networks that can be used for sessions. It does not include the TGs leading to adjacent APPN end nodes.

2. COSM, optionally, to perform the mode name to COS name mapping

CP(OLU) will send a Locate request for the DLU, using its conwinner CP-CP session, to NNS(OLU). This Locate request contains a (optional) COS name and a TG control vector (TGV1), describing the endpoint TGs from CP(OLU).

NNS(OLU) will forward the *Locate* request to NNS(DLU). The Locate request will be sent directly to NNS(DLU) if NNS(OLU) knows the network node server of the DLU. If not, the Locate request will arrive on NNS(DLU) as a result of a network broadcast search started by NNS(OLU). For details, see "Network Searches" on page 92.

On CP(DLU), TDM will be invoked to obtain the endpoint TGs connecting CP(DLU) to adjacent network nodes and end nodes. This information is added as a TG control vector (TGV2) to the Locate reply returned, via intermediate network nodes, to NNS(OLU).

Note: CP(DLU) includes all TGs leading to adjacent network nodes or connection networks and also includes all TGs to CP(OLU).

On NNS(OLU), COSM is invoked to perform the mode-to-COS name translation (if not already done), and obtain the contents of the COS entry from the COS database. TDM will be invoked to obtain intermediate routing TG and node

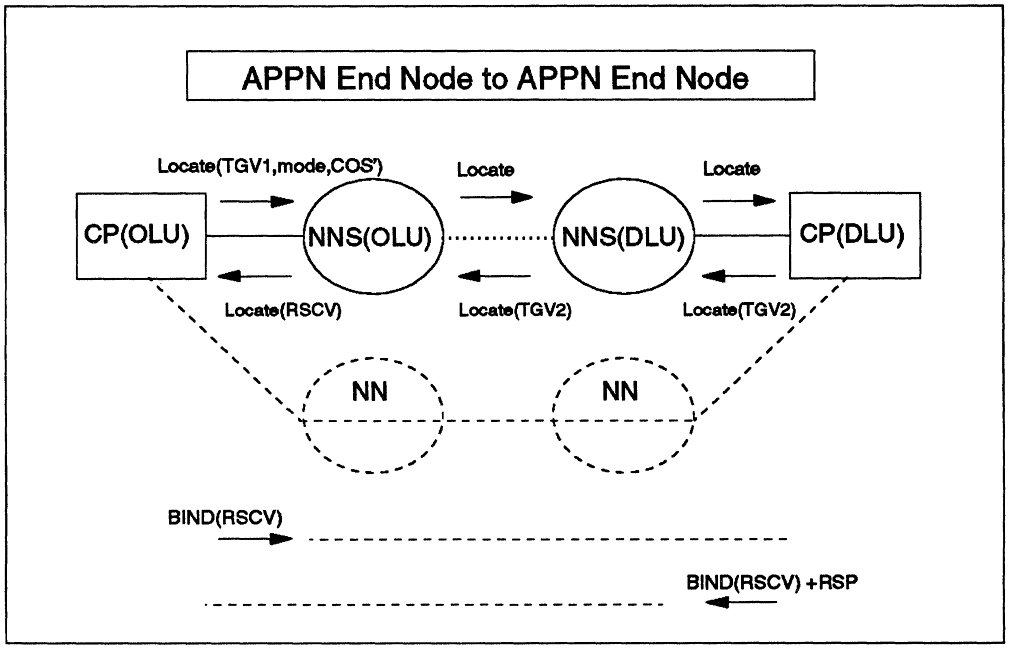

Figure 38. Session Establishment between LUs on APPN End Nodes (Simplified)

characteristics from the network topology database. Using both TGVs (TGV1 and TGV2) and intermediate resource information obtained from its network topology database, RSS on NNS(OLU) will compute an optimum route between CP(OLU) and CP(DLU). It will return this information in a route selection control vector (RSCV), added to the Locate reply, to the CP(OLU).

After receiving the Locate reply, CP(OLU) will construct a BIND to start the session. Among other information, the BIND will contain the RSCV obtained from the NNS(OLU). The BIND will be routed to the CP(DLU) using the routing information within the RSCV. The BIND response will be returned on the reverse path.

The Route Selection Control Vector

A route selection control vector (RSCV) is carried in the BIND and Locate requests and replies, and other RUs to describe a route through the APPN network. A distinction has to be made between the RSCV used for BIND routing, also called *session RSCV*, and the *Locate RSCV*, which is used to route Locate search requests through the network. For details about Locate search requests, see "Network Searches" on page 92.

A Locate RSCV contains a list of CP names from an origin to a destination node, as opposed to a session RSCV, which contains a list of CP names and TGs between each adjacent pair of nodes along a route from an origin to a destination node. The Locate RSCV contains the *shortest* route (minimal hops), whereas the session RSCV yields an *optimum* route. TG information is not required in a Locate RSCV, as the Locate request is forwarded using CP-CP sessions between adjacent network nodes.

The BIND is forwarded to the destination node using a method called *source routing*. Source routing relies on routing information contained in the message itself. Intermediate nodes do not require knowledge about the final destination, they learn from the message itself what the next node is along the route and how to get there. Source routing provides a very fast method of routing messages through a network, as the processing required in intermediate nodes is minimal. Networks offering *connectionless* services often rely on source routing.

Opposed to source routing is the use of *virtual circuits*. Virtual circuits imply a *connection-oriented* network service. A virtual circuit assumes an end-to-end connection that has to be established before messages can be sent. The use of virtual circuits typically depends on routing tables maintained in intermediate nodes to route messages from an origin to a destination node. Each message contains a logical channel identifier, which is used by intermediate nodes to index their routing tables and find what the next node is along the route and how to get there. The logical channels do not have end-to-end meaning and may vary (be "swapped") between pairs of adjacent nodes

As mentioned, an example of source routing is the method used to forward a BIND in an APPN network. The session RSCV is used by each intermediate node to obtain the next node and a TG to the node, along the route. Session data, however, is routed over a virtual circuit. During BIND flow, intermediate nodes will initialize their routing tables and assign logical channels, or local-form session identifiers (LFSIDs), for each TG along the route. Session data routed between two LUs contains an LFSID, which is then used for intermediate routing. LFSIDs have only local significance and vary between each pair of nodes along the route. Intermediate network nodes will swap LFSIDs within the header of the message, on each TG, as the session data is routed towards the destination node. For details, see "Local-Form Session Identifier (LFSID)" on page 34.

Route Selection Using Virtual Routing Nodes

APPN end nodes that have defined a connection network, will include the endpoint TG to the virtual routing node in the TG control vectors (TGVs) added to the Locate request or reply. The TGVs also contain DLC-signaling information such as the end node's MAC address on a token-ring.

RSS on NNS(OLU) will detect if both APPN end nodes have defined a TG to the same VRN and optionally, if no lower weight routes exist, select the route "through" the VRN as the optimal route.

Being connected to the same connection network indicates that both APPN end nodes attach to the same *shared-access transport facility* (SATF), and session data can be sent directly, without intermediate node routing, between the APPN end nodes.

5.8 SNA Defined Modes and Classes of Service

Generally speaking, each installation is free to choose its mode names, COS names, and COS definitions; however, because definitions on one node may imply definitions on a second node, synchronization of definitions is sometimes required. For example, when an end node performs mode name to COS name mapping, it assumes the COS definitions are present in its network node server.

To simplify table maintenance, SNA has defined default mode names, related COS names and COS definitions for the various classes of service.

Below is a list of the SNA-defined names. The contents of the COS tables are described in *SNA Type 2.1 Node Reference*. The contents of the modes are described in *SNA LU Version 6 Reference: Peer Protocols*.

Mode Name	Corresponding COS Name
Default[1]	#CONNECT
#BATCH	#BATCH
#INTER	#INTER
#BATCHSC	#BATCHSC
#INTERSC	#INTERSC
CPSVCMG	CPSVCMG
SNASVCMG	SNASVCMG

Notes:

1. If no mode name is specified during a session establishment request, implementations use COS name #CONNECT.
2. The character "#" represents the hexadecimal value X'7B'.

In most cases, the default values in the IBM supplied table will be adequate. In particular, small networks will not realize much benefit from modifying the standard tables. In larger networks, modifications may be required in order to achieve the desired amount of load distribution, if the nodes do not support randomization during route selection.

Chapter 6. Directory Services

The directory services component of the control point is responsible for the management of the directory database and the search for network resources throughout an APPN network.

6.1 Functional Overview

The major components of DS are:

- Directory Database Function (DDB)

  The Directory Database Function is responsible for the database lookup and database maintenance logic of DS. Part of the database maintenance logic is the Resource Registration (RR) Function. RR is responsible for sending and receiving requests for resource registration.

  The DDB component is not available on LEN end nodes.

- Maintain CP Status Function (MCPS)

  The Maintain CP Status Function is responsible for keeping DS aware of other control points that it wishes to communicate with. In the case of an APPN end node, the MCPS function maintains awareness of the sessions with the node's network node server. In the case of an APPN network node, the MCPS maintains awareness of APPN end nodes and LEN end nodes within the APPN network node's domain, and additionally, other APPN network nodes to which it has CP-CP sessions.

- Network Search Function (NS)

  The Network Search Function is responsible for sending and receiving resource search requests to and from other APPN nodes in the network.

  This DS component is not available on LEN end nodes.

Figure 39 depicts the node functions that interface with DS.

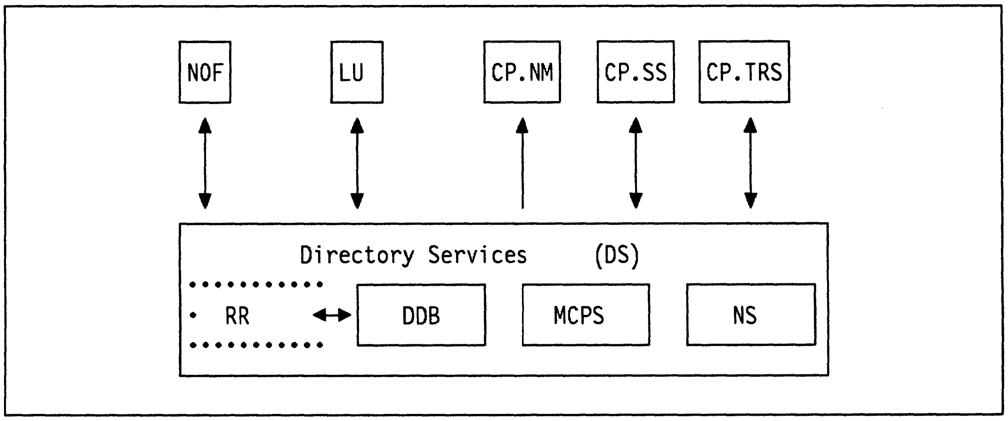

Figure 39. Overview of DS Components and Protocol Boundaries

Directory Services (DS) is created and initialized by the node operator facility (NOF) at node initialization time. NOF passes DS the following parameters at initialization time:

- Node type (LEN end node, APPN end node, or APPN network node)
- The network ID of the node
- The control point name of the node
- Whether or not resources should be registered

  APPN end nodes may register resources at their network node server, and APPN network nodes may register resources at a central directory server.

6.2 Directory Database Function

Each node uses a *local directory database* to maintain awareness of network resources and their location in the APPN network. The directory database function on each node is responsible for maintaining the local directory database and locating resources in either the local directory database or directory databases residing on other nodes. Resources can be *local*, *same-domain* or *other-domain* to a specific node.

When trying to locate a resource, DS does not restrict itself to resource information maintained locally, but also tries to use information contained in remote directory databases. The term *network directory database* or *distributed directory database*, is used to refer to a "virtual" database containing all resource information contained within the network. When using the phrase "DS is responsible for locating a resource in the network database" this refers to a coordinated effort of DS on one or more nodes to obtain the resource information from any local directory database on each of the nodes.

An APPN or LEN end node is responsible for maintaining database entries for:

- Local resources
- Local resources on adjacent nodes with which it wishes to have peer-to-peer sessions, that is, establish sessions without the support of an APPN network node

An APPN end node may choose to inform its network node server of some or all the resources located on itself, a process known as *end node resource registration*. APPN architecture does not allow an APPN end node to register resources on its network node server that are owned by other end nodes. Directory database entries on an end node are entered via the node operator facility (NOF).

An APPN network node is responsible for maintaining database entries for:

- Local resources (that means, LUs)
- End node resources within the APPN network node's domain
- End node or network node resources outside the APPN network node's domain

An APPN network node may choose to inform other network node servers of some or all the resources located on itself or within its domain, a process known as *central resource registration*. Directory database entries on an APPN network node are entered via the node operator facility (NOF), by resource registration, or by caching information obtained via network searches.

Each DS is responsible for maintaining directory information about network resources. This information includes:

- Network-qualified resource name

  For a resource of type LU, the LU name is given. For a resource of type control point, the CP name is given.

- The resource type, either:
  - LU
  - APPN or LEN end node CP
  - APPN network node CP

- Indicator specifying whether resource registration is required and the registration status, which is either:
  - Not registered
  - Registration in progress
  - Registered

Resources are either unique or non-unique within a network. SNA requires that any network accessible unit (NAU) can be distinguished from other resources in the network. NAUs are resources of the type LU, APPN network node CP, APPN end node CP, and LEN end node CP. It is strongly recommended to use a consistent naming convention for NAUs, to prevent duplicated names. Duplicate names will lead to errors and degrade the performance of the network.

In the directory database, a resource “hierarchy” is maintained. For example for a resource of type LU, the database contains pointers to the CP name of the node owning the LU and the CP name of the network node server.

Directory database entries are entered by system definition, by resource registration, or by caching information obtained from network searches.

6.2.1 System Defined Resources

Although APPN directory services (DS) is very dynamic and resource knowledge can be obtained dynamically, each resource must at least be defined at the node where it resides.

LEN end nodes lack the support for CP-CP sessions and resources on other nodes cannot be learned dynamically; therefore, LEN end nodes require that all network resources that will be accessed by local resources be defined in the local directory database of the LEN end node. Alternatively, if a LEN end node is connected to an APPN network node, then all LEN end node resources that need to be accessed from or through the APPN network node, must be defined at the network node. To simplify this registration, directory services provides a facility using *generic names* and *wildcards*:

- Wildcards

  Wildcards are represented with an asterisk (*). An asterisk results in a match for each network resource that is searched for by directory services.

- Generic names

  Partially specified names are represented by one or more start characters of the resource name followed by an asterisk. For example, if all network resources on a LEN end node start with the characters ***ITSC***, then the partially specified name could look like ***ITSC****.

For details on how the use of wildcards impacts DS search logic, see "Wildcards" on page 97.

Network accessible unit resources owned by an APPN end node must always be defined at its network node server if the APPN end node is defined as an *unauthorized* end node. A network node server will never query unauthorized APPN end nodes to locate resources, and does not allow session establishment initiated by LUs on the end node if the LU is not explicitly defined.

In addition, APPN network nodes may define same-domain APPN end node's resources and other-domain resources to improve network search performance.

LU Name Equal CP Name

Installations may choose to select the same name for an LU as the CP name of the owning node. This reduces the system definition at the local node. Another advantage of using LU names equal to the CP name is when a directory search is required.

Before a network node server of the node owning the OLU searches its local directory database for the destination LU, it checks with topology and routing services to see whether the DLU name is equal to one of the control point names known in the network topology database. If the LU name is equal to a CP name of an active network node, then directory services does not need to perform a directory database or network search. The topology database is only queried by the network node starting the resource search procedure. Since the network topology database contains only network nodes, defining an LU name equal to a CP name will limit resource search time only if the target LU is contained in a network node.

6.2.2 Resource Registration

APPN distinguishes two types of resource registration:

1. End Node Resource Registration

   APPN end nodes register local resources at their network nodes.

2. Central Resource Registration

   APPN network nodes register local and same-domain resources at a network node known as the *central directory server*.

The reason for resource registration is to improve network search performance. Details about end node and central resource registration will be given in the following sections.

End Node Resource Registration

End node resource registration is an optional facility on APPN end nodes, which allows an APPN end node (registration requester) to register network accessible resources at its network node server (registration server). Supporting the receipt of the registration request is a base function for APPN network nodes. APPN network nodes allow registration requests only from APPN end nodes that have been defined as *authorized* end nodes.

Note that resources on end nodes that do not perform resource registration must be system defined at the network node server.

Following CP-CP session establishment, when CP capabilities are exchanged between the APPN end node and its serving network node, an authorized APPN end node may then register resources that it wishes to make available to the network.

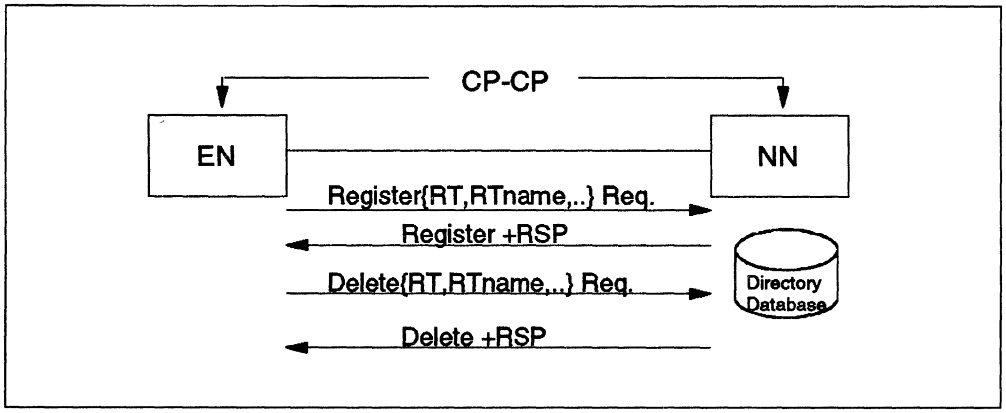

Figure 40. Resource Type (RT) and Name Registration

The register request may contain control vectors describing multiple resources; however, the total length of the Locate/Register must not exceed 1024 bytes. The end node will not initiate an additional registration request until it has received a reply from the previous registration request. The network node server may use this function to control the rate at which it receives registration requests. The network node will send a reply to show the success (or failure) of the resource registration operation; see Figure 40.

When DS sends a resource registration request, DS changes the status of the resource to *registration pending*. The status becomes *registered* or *not registered*, when the APPN end node receives respectively a positive, or a negative reply to the registration request.

An APPN end node may delete resources from its network node server's directory database using an explicit deletion request; see Figure 40. Directory services will also remove resources registered by an APPN end node when the CP-CP sessions between the APPN end node and the network node server are terminated.

To change a resource entry, the APPN end node must first delete the old entry and then completely register the resource again.

Central Resource Registration (CRR)

The APPN *central resource registration* architecture allows one or more network nodes in a network to act as central ***directory servers***. Instead of trying to locate a resource themselves, network nodes query their closest central directory server. The central directory server then takes responsibility for locating the resource, either by querying its own cache, querying other central directory servers, or by initiating a network search. The central directory server concept maximizes the sharing of cached directory entries and, therefore, minimizes the number of network searches.

Central resource registration (CRR) depends on two functions, which are:

1. Registration server
2. Registration requester

CRR allows a network node (registration requester) to register resources at a central directory server (registration server). Both the registration requester and server function are optional functions in an APPN network.

For performance and reliability reasons, more than one central directory server may be present. Central directory servers defined as having equal capabilities are referred to as *alternate* directory servers.

The APPN topology database (TDB) is used to allow identification of central directory servers and their capabilities to every network node. Directory servers identify themselves with an indicator in the topology database update (TDU) messages when connecting to the network, thereby informing all network nodes of their presence. Only APPN network nodes that support the resource registration requester function recognize the server capabilities data included in the TDU. APPN network nodes that do not support the resource registration requester function do not recognize the server capabilities data within the TDU and just store and forward the TDU as they receive it.

Central resource registration allows a network node to register its resources at a directory server, eliminating the need for broadcast searches to locate registered resources. Once the resource is registered, all network nodes may find the resource by sending a directed Locate search request to the central directory server. Resource registration reduces the number of network broadcasts considerably.

Note: The difference between a broadcast search and a directed search will be explained in "Network Searches" on page 92.

When registering its resources with its network node server, an APPN end node indicates which of those should be centrally registered, as specified in its local definitions. Since it is optional for end nodes to register resources with their network node servers, any unregistered resources may still require a broadcast search to locate the resource, preventing total elimination of broadcast searches.

Since no direct sessions exist from the directory services of a network node to the directory services of a central directory server, the existing CP-CP sessions and network search service transaction programs provide the means for transporting registration data. By adding register variables to a directed Locate search request sent to the central directory server, a network node is able to register its resources at the directory server. Intermediate network nodes will only look at the routing information within the Locate request and ignore the appended register variables.

The register request may contain control vectors describing multiple resources; however, the total length of the Register must not exceed 1024 bytes.

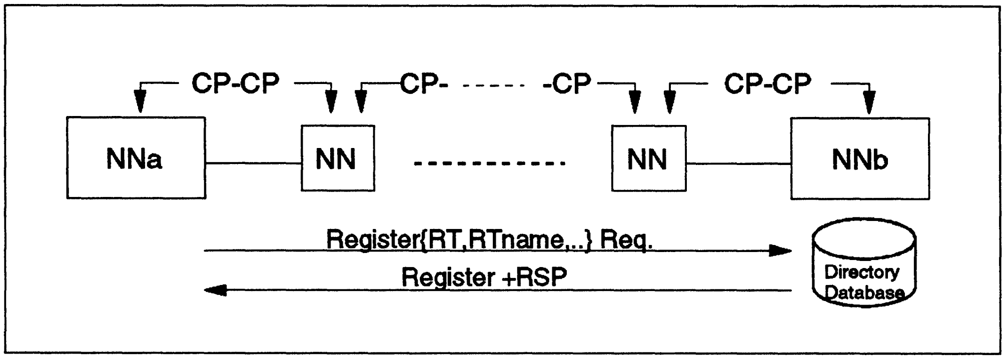

Figure 41. Central Resource Registration.

Registration requester (NNa) registering Resource Type (RT) and RT name at a registration server (NNb)

Registration of resources by the CRR function will result in those resources being cached at the central directory server, and will be handled appropriately, for example replaced by least-recently-used (LRU) algorithms or overlaid by new information, without requiring explicit registration information. The central directory server will send a reply to show the success (or failure) of the resource registration operation; see Figure 41. A negative response, indicating registration failure, will contain error information.

The origin network node will not register any additional information until it has received a reply from the previous registration request. The central directory server may use this function to control the rate at which it receives registration requests from a particular node.

CRR is handled differently from end node registration. It does not require an explicit delete with a subsequent register operation to change resource information registered at the central directory server. Information can be updated simply by submitting a register request that will overlay the existing information. Deletion of information will happen as a consequence of NN search requests.

Optionally, implementations save registered directory entries across IPLs.

6.2.3 Cached Directory Entry

APPN network nodes dynamically increase the information in the directory database by caching the results of directory searches. Figure 42 on page 90 depicts the concept of resource caching.

Session services in CP(OLU) invokes DS to locate the DLU. Assuming CP(OLU) has no system-defined entry for the DLU, DS sends a one-hop Locate request for the DLU to DS on its network node server, NNS(OLU). The Locate search request also contains information about the OLU, enabling NNS(OLU) to cache a directory entry for the OLU before starting a network search for the DLU. The search request may either be a directed (DLU location known in NNS(OLU) directory) or a broadcast search (DLU location unknown). When the network node server of CP(DLU), NNS(DLU), receives the Locate request and NNS(DLU) is able to locate the DLU, NNS(DLU) will cache a directory entry for both OLU and DLU, and return a positive response to NNS(OLU). NNS(OLU) will cache a directory entry for the DLU and return a positive response to CP(OLU).

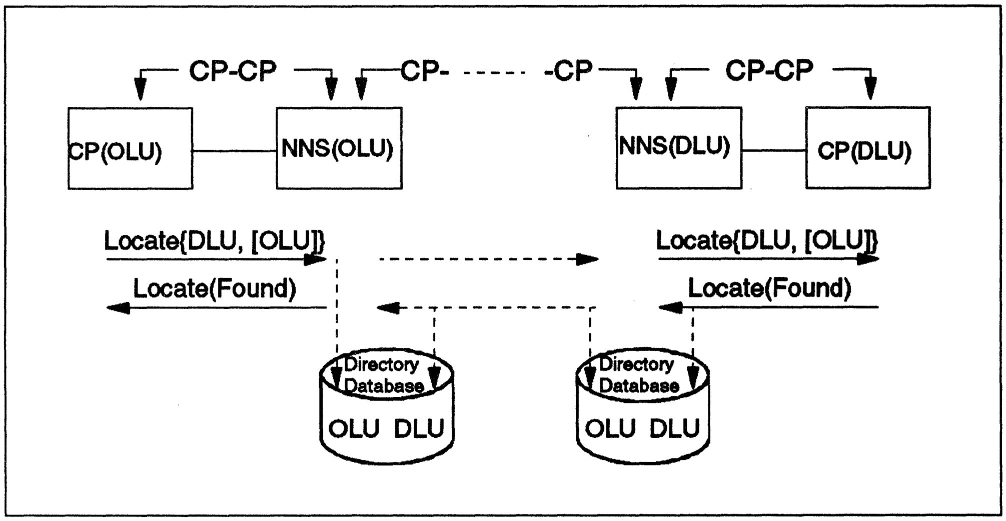

Figure 42. Resource Caching

The information retrieved through caching may ultimately result in huge local directory databases and even include resource entries that are no longer in use or up-to-date. It is up to the implementation of the APPN network node function to decide how the cache entries are maintained, when they are deleted or replaced, and whether entries are saved across IPLs (safe-store of DS cache).

For example, the Network Services/2 product (APPN for OS/2*) saves its cache directory to disk every 20 updates. In addition, it allows for a total of 255 cached directory entries. If all 255 cache entries are in use, new entries to be cached will replace the oldest cache entries first.

An APPN network node that caches resource entries that are owned by end nodes for which it provides network node services, deletes these entries when the CP-CP sessions with the end node are deactivated.

6.3 Maintain CP Status Function (MCSF)

All APPN nodes that exchange information on CP-CP sessions are interested in the enabled status of partner CPs. Among other things, CP-CP sessions are required for resource registration and to locate resources. The maintain CP status function (MCSF) in DS is present in APPN end nodes and network nodes; it maintains a list of CPs on adjacent APPN nodes and, on network nodes only, a list of active central directory servers.

MCSF learns about the central directory server from the network topology database; for details, see "Central Resource Registration (CRR)" on page 87. Status changes of a directory server CP in the network topology database are reflected in the information MCSF maintains about this directory server.

Entries for APPN (network node or end node) CPs are maintained dynamically as CP-CP sessions are established and terminated. Session services informs DS whenever a CP-CP session is established or deactivated.

End nodes are either *authorized* or *unauthorized* end nodes. APPN network nodes accept register requests only from authorized end nodes.

In order to locate a resource, an APPN network node will query only authorized end nodes. APPN end nodes do not always support the receipt of Locate requests for resources that have not been registered or cached at the network node server. An APPN network node will not query adjacent APPN end nodes that have indicated they are not willing to accept, and handle, Locate requests for such resources. Note that all APPN end nodes support the receipt of Locate search requests for registered or cached resources, for example, to verify that a resource is active and to return the end node TG vectors.

Entries of adjacent LEN end nodes cannot be learned on the basis of CP-CP sessions, but can be optionally cached when BINDs are received from the adjacent LEN end node with an undefined LU as PLU. They have to be defined by system definitions if they shall act as SLUs before they themselves request a session through the network node server. Because of the lack of CP-CP sessions, LEN end nodes cannot support resource registration and Locate search requests; therefore, DS on an adjacent APPN network node is not interested in the authorization status or Locate support of LEN end nodes.

6.4 Network Search Function (NS)

The primary function of the network search (NS) function in DS is to locate network resources and to control the flow of search requests and replies, through the network.

When handling a directory search request, the NS function invokes the directory database function to determine the knowledge that the CP has about the resource in question. Depending on that knowledge, the NS function may choose to reply to the request or to forward the request to another node. When the NS function chooses to send a request to another node, its exercises its transport logic. This logic controls the sending of directory messages carrying search requests and replies. These messages are called *Locate searches.*

Additionally, the Locate searches are capable of carrying non-DS data and can be used by other CP components for the transport of their control data. Such other components are termed DS users or DS applications. For example, session services (SS) acts as a DS user when requesting a directory search, for example to locate an LU, and delivery of SS variables. Examples of SS variables which may be included, when session services requests DS to locate an LU, are:

- The fully qualified procedure correlation identifier (FQPCID)
- The destination LU
- The origin LU
- Mode name
- COS name
- Endpoint TG vectors

Endpoint TG vectors are included in a Locate search by SS(OLU), but not forwarded beyond session services of OLU's network node server; SS(DLU) then sends endpoint TG vectors in the Locate reply back to NNS(OLU).

6.4.1 Search Terminology

The DS user, or DS application, refers to the process or CP component that asks DS to find a target resource.

Locate search refers to the signals that DS components in one node send to DS components in other nodes when looking for resources.

Historically, the originator of a search request is referred to as the originating CP(OLU). We will refer to the destination node as CP(DLU), to the network node server of CP(OLU) as NNS(OLU) and to the network node server of CP(DLU) as NNS(DLU). Note that the "network node server" of a network node is the network node itself and CP(OLU) and CP(DLU) may have the same network node server.

6.4.2 Network Searches

Directory services (DS) will be invoked to obtain the location of a resource. If the local directory database function indicates that the CP has no knowledge of the resource, the request may be forwarded to another node. The messages used by DS on different nodes are *Locate searches.* Locate search requests are always sent on the *conwinner* CP-CP session to an adjacent CP.

There are three types of Locate search requests:

1. One-hop search
2. Directed search
3. Broadcast search

After describing each type of Locate search request in the following sections, we will continue with a description of how the various types of Locate searches are used on LEN end nodes, APPN end nodes and APPN network nodes.

One-Hop Search

A one-hop search is a Locate search request that is exchanged between an APPN end node and its network node server. The one-hop search is sent on the CP-CP session between the control points; see (A) in Figure 43 on page 93. The one-hop is conceptually simpler than the two other types of searches (broadcast and directed) because no routing information is needed.

Directed Search

A directed Locate search request is a request that is sent along a predefined path from one network node to another network node. The origin network node calculates a path of CP-CP session hops to the target network node and appends the routing information to the search. Each network node along the path relies on that routing information for choosing the next hop and ensuring that the search travels directly to the destination network node.

The routing information for the directed Locate search request is contained within a *Locate RSCV.* A Locate RSCV (route selection control vector) defines the nodes on the search path including a series of network node names. A locate RSCV describes the shortest path, that is, the path with the least number of hops, to a destination node.

A directed search is used, by:

1. NNS(OLU), when NNS(OLU) has a directory entry indicating that the destination resource is an other-domain resource.
2. NNS(OLU), when invoking a central directory server.
3. A central directory server, when the central directory server has a directory entry indicating the destination location is an other-domain resource.
4. A central directory server, to query alternate directory servers.

The reason for sending a directed Locate search in 2 and 4 is to obtain resource information from DS on the remote node. The reason for sending a directed Locate search in 1 and 3 is to verify the accuracy of directory information and to obtain the endpoint TGs of the end node that owns the destination resource. This procedure is known as *resource verification*. Because of the optional *nonverify function*, a network node server (point 1) may decide not to perform resource verification, whereas a central directory server (point 3) always performs resource verification. For details about the nonverify function, see "Nonverify Function" on page 102.

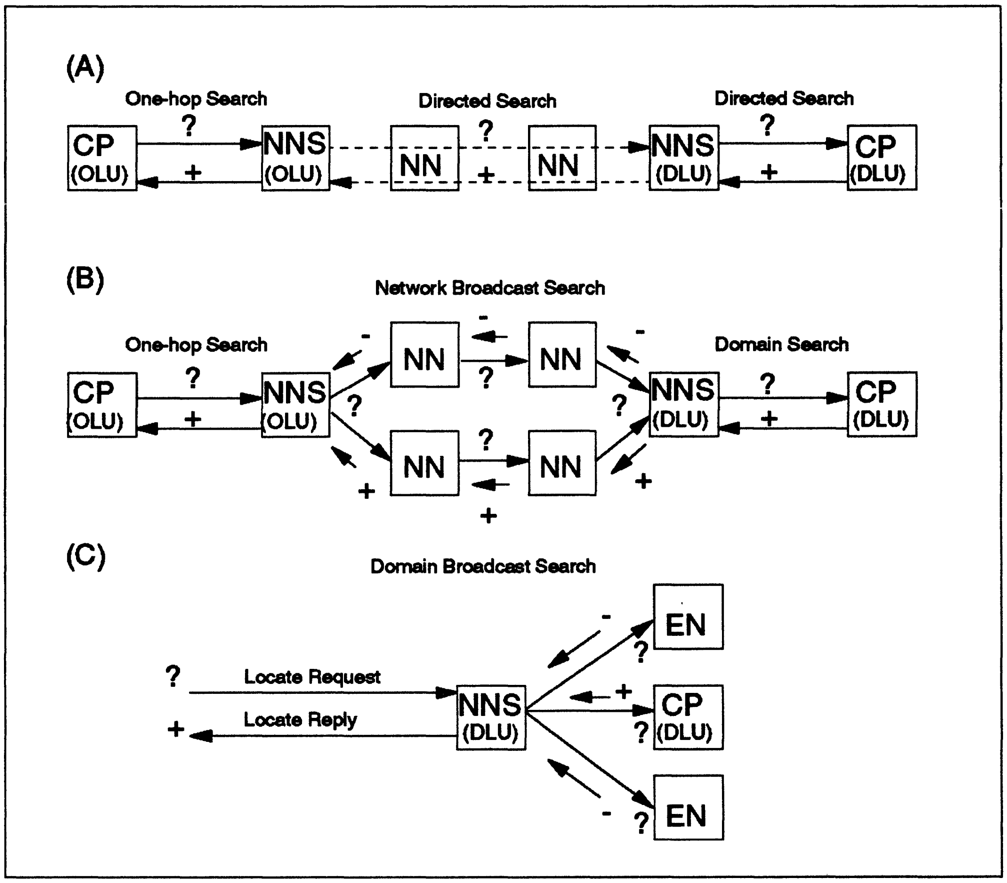

Figure 43. Locate Search Requests

Example (A) in Figure 43 depicts a directed Locate search request. NNS(OLU) determines from its directory database the network node server of the DLU, calculates a route to NNS(DLU), adds the routing information to the request and forwards the Locate search to the next network node on the calculated route. Intermediate network nodes forward the Locate search along the route, using routing information within the Locate RSCV, to NNS(DLU).

Broadcast Search

When the DLU is unknown, broadcast searches are used by network nodes to send Locate search requests to multiple CPs. Two types of broadcast searches can be distinguished:

- Domain broadcast: to query APPN end nodes in the network node's domain
- Network broadcast: to query all network nodes in the network

Broadcast searches are always done in parallel, which allows DS to locate a resource quickly.

Domain Broadcast Search: A network node that starts a domain broadcast search will send a Locate search for the destination resource to adjacent APPN end nodes. See (C) in Figure 43 on page 93.

APPN end nodes will be included in an APPN network node's domain search only if the APPN end nodes have been defined as authorized end nodes, and during the exchange of CP capabilities the end nodes have indicated they are willing to be included in a domain search of their network node server.

A domain broadcast cannot be distinguished at an end node from a directed search. An end node receiving the search has no awareness of which other nodes are being searched; it simply searches its database and returns a reply. The coordination of the domain broadcast is the responsibility of the network node. If more than one positive reply is returned, DS uses the ***first*** positive reply. Note that receiving more than one positive response on a domain search indicates a definition error.

Network Broadcast Search: By performing a network broadcast search, a network node is able to query all APPN network nodes in the network. As Locate search requests always flow on CP-CP sessions, which can only be established between adjacent nodes, a network node starts a network broadcast search by sending a Locate request to all adjacent network nodes; the adjacent network nodes forward the request to their adjacent network nodes and so on until every network node has received a copy of the Locate request. See, as an example, Figure 44 on page 95.

A network broadcast search is used:

- When the NNS(OLU) has no directory entry for a destination resource, and the resource location cannot be found differently, for example via a domain search or by querying a central directory server.
- When a central directory server has no directory entry for a destination resource and the resource location cannot be found differently, for example via a domain search or by searching alternate directory servers.

A network broadcast search, sending a Locate search to each adjacent network node, is the search of last resort, because the broadcast search floods the network with requests for the location of the target resource and therefore has negative performance implications for the network. A network node will perform a network broadcast only if it is not able to locate a resource differently. Network nodes capable of querying a central directory server, which optionally performs a network broadcast search itself, will never perform a network broadcast search.

As each network node forwards the request to all its neighbor network nodes, except to the node from which the request has been received, network nodes

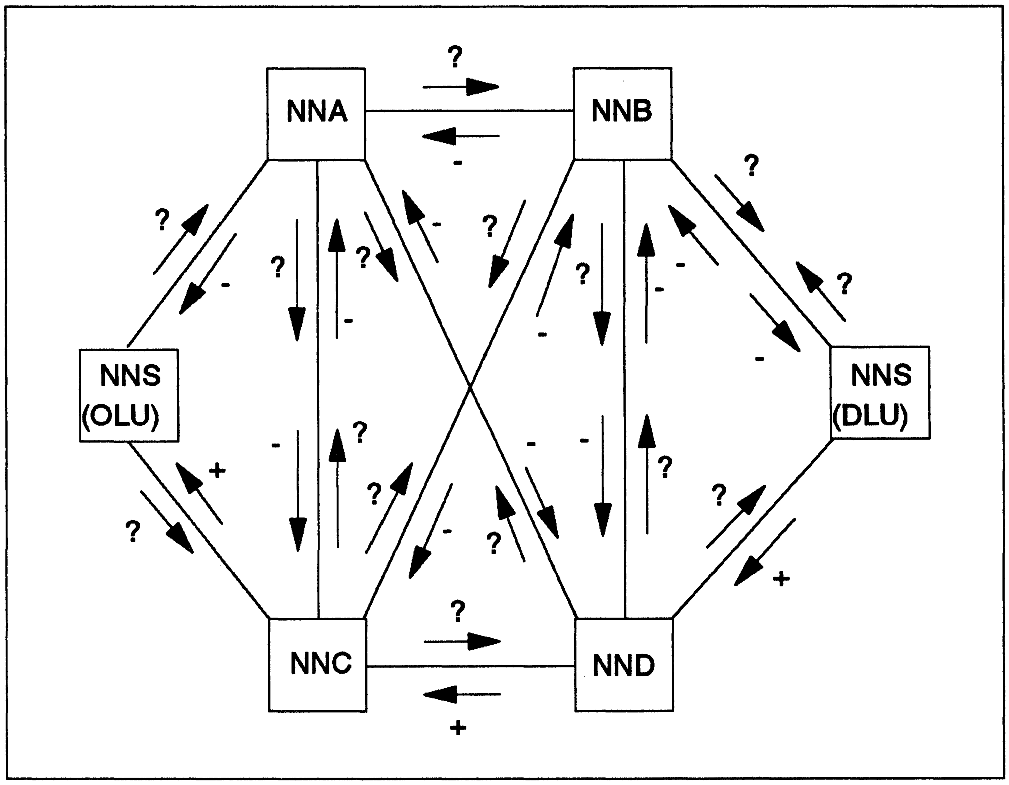

Figure 44. Network Broadcast. NNS(OLU) starts a broadcast search.

can receive multiple copies of the same request. A simple mechanism prevents unnecessary forwarding of the broadcast request. All Locate requests are uniquely identified by an FQPCID (fully qualified procedure correlation ID). By temporarily storing FQPCIDs, comparing stored values with the FQPCID within broadcast requests received, and returning a negative reply to duplicate requests, each network node makes sure that only one copy of the Locate request is forwarded.

As depicted in Figure 44, NNS(OLU) starts the network broadcast search by sending a Locate search request to its adjacent network nodes. Each receiving network node should propagate the Locate search to its neighbor network nodes before checking local resources to allow the network search to progress rapidly, but an implementation may decide to check local resources first.

Each network node maintains a status of all broadcast search requests sent to adjacent network nodes. Normally, the replies from the adjacent nodes are consolidated and as soon as all replies are received, a reply is sent to the originator of the broadcast search. However, the broadcast algorithm requires that positive replies be returned immediately (in APPN good news travels fast). Therefore, if the target resource is a local resource, the resource has been found within the node's domain, or the node receives a positive reply from a neighbor node, then the node returns a positive reply immediately, regardless of whether all nodes have replied.

A Locate search reply can be "complete" or "incomplete." A complete reply indicates that this is the last reply to be returned, whereas an incomplete reply is sent by a node that has information to be sent immediately but has not received a reply to all requests forwarded. Each node will send a final reply, preceded optionally by one or more (positive) incomplete replies.

Each network node will consider a broadcast search to be completed when all adjacent network nodes have returned a complete reply. The broadcast originating node may receive more than one positive reply after a broadcast search because of the target resource being defined on multiple nodes. Duplicate definitions are not necessarily erroneous, for example a LEN end node's resources may be defined, either explicitly or using wildcard definitions, on all network nodes to which the LEN end node is connected. The broadcast originating node will use the ***first*** positive reply that results from an explicit definition, or if none is received, the first reply indicating a wildcard definition. See also "Wildcards" on page 97.

Figure 45 depicts the DS search logic on a network node during a network broadcast search.

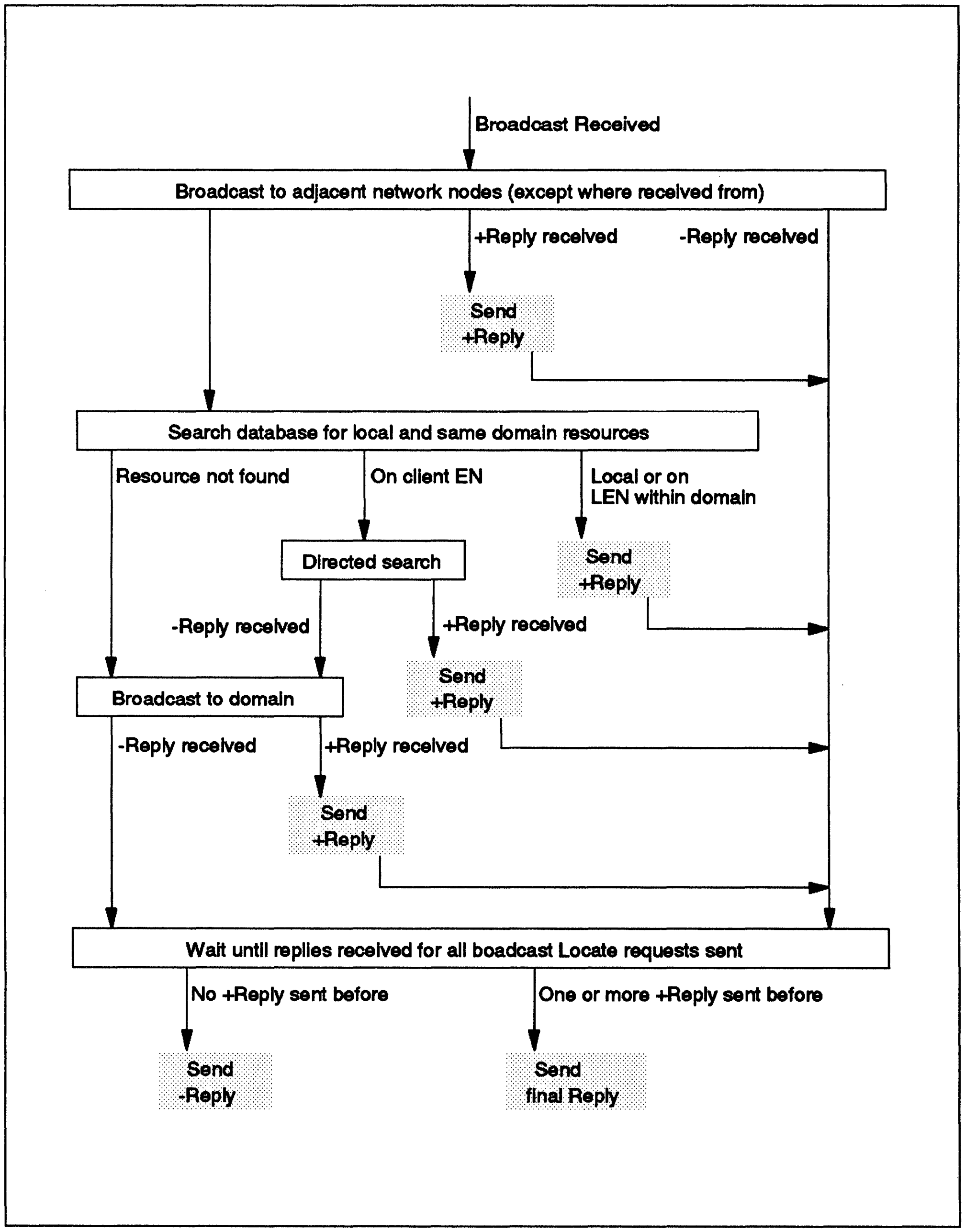

Figure 45. DS Search Logic during Broadcast Search

6.4.3 LEN End Nodes

Locate search requests originating on a LEN end node are restricted to a search of the local directory database only. If the LEN end node cannot locate the resource in its local directory database, directory services at the LEN end node returns a "Locate failure" to the initiator of the request. Entries within the directory database of a LEN end node for resources not locate on the LEN end node itself can only be the result of system definition.

Any resource not located on the LEN end node itself has to be defined as being located on an adjacent node although the actual location may be anywhere in the network. A BIND is then sent to the adjacent node that will locate the destination node using its normal search and session setup logic.

Wildcards

All the LEN end node's resources to be accessed as DLUs must be defined on the LEN end node's serving network node. In the case of a LEN end node that supports a large number of resources, for example a subarea network attached as a LEN end node to an APPN network, a large number of definitions will be required.

To alleviate this definition problem on such a network node, directory services provides generic and wildcard routing. For example, in the network node directory, there could be entries for "RAL*" and "*." For details on how to define resources using generic (partially specified) names and wildcards, see page 85.

When a network node receives a Locate search request for a resource, the network node checks the directory entries that have fully specified resource names. If the resource cannot be found, the resource name is compared with the partial entries. Any LU name beginning with "RAL" would, for example match "RAL*." Finally, if no match can be found, the directory is checked to see if a wildcard entry "*," which matches all resources, has been defined. Once a match has been found, a positive Locate search response is returned.

Problems may arise when a network broadcast search is sent and more than one network node, using either explicit, partially specified, or wildcard resource definitions, returns positive Locate search replies. A solution to this problem is that the network node returning the positive reply will indicate if the resource was found using a wildcard definition. The network node from which the broadcast search was originated, differentiates between the replies. DS will return to the DS user, for example session services, the first positive reply based on an "explicit" definition (which could be a partial definition) or, if none was received, the first positive reply as a result of a wildcard definition.

Care should be taken and, most important, a consistent naming convention is required when using wildcards. Although it reduces the number of definitions it can easily lead to errors. Only one network node in a network should ever define a wildcard entry.

6.4.4 APPN End Nodes

Directory services at an APPN end node, in conjunction with directory services at its network node server, offers distributed search facilities throughout the APPN network. If a search request fails at the APPN end node, the APPN end node automatically sends a one-hop search request to its network node server.

Directory services at the APPN network node is responsible for a Locate search through the network.

Locate search requests will be received by APPN end nodes in the following two cases:

1. Its network node server is handling a search request and has information, either system-defined, registered, or cached, that the end node owns the destination resource.
2. Its network node server is handling a search request, has no information, either system-defined, registered, or cached and has, therefore, started a domain search.

   When CP-CP sessions are established between an end node and its network node server, authorized APPN end nodes may request to participate in domain search requests that originate from the network node server.

When receiving a Locate search request, the APPN end node will check its directory for the target resource. A positive reply will include the TG vectors of the end node.

6.4.5 APPN Network Nodes

For an overview of the DS search logic on APPN network nodes see Figure 46 on page 99 and Figure 47 on page 100. The first figure depicts the search logic on the network node server of the node from which the resource search originates, and the second figure depicts the search logic on the network node server of the node owning the destination resource. An APPN network node may receive Locate search requests, from:

1. DS users within the node itself or from served APPN end nodes, using a one-hop search request

   The network node will check its directory database and return a positive reply if the destination resource is a local resource or resides on an adjacent LEN end node. A same-domain resource on an adjacent APPN end node will be verified by sending a Locate search request. When supporting the (optional) nonverify function, a network node server will not perform the resource verification if the request indicates this. For details, see "Nonverify Function" on page 102.

   The network node will send a directed Locate search request to the network node server of the destination resource if the directory database search indicates the resource was other-domain.

   If the resource cannot be found in the directory database, resource verification is not successful, or the other-domain directed Locate search fails, the network node starts a domain broadcast search.

   If the domain broadcast is not successful, the network node will either start a network broadcast search or send a directed request to the closest (minimal-weight route) directory server. The latter will be done only if central directory servers exist in the network and the network node supports querying a directory server. A network node will never start a network broadcast after querying a central directory server.

2. Network nodes performing a network broadcast search

   The network broadcast algorithm has been described in "Network Broadcast Search" on page 94. As part of the broadcast search the network node will perform the local activities mentioned in 1, that is, checking the directory database for local or same-domain resources, (optionally) verifying the resource, and if necessary starting a domain search.

3. Network nodes performing a directed search

   An APPN network node will receive a directed search, if the sending network node assumes the resource is within the domain of the receiving network node or if the sending network node has no directory entry for the resource and the receiving network node performs the function of central directory server. A network node performing the function of central directory server is described in "Central Resource Registration (CRR)" on page 87.

   When the network node receives a directed Locate search request it will perform the local activities mentioned in 1 on page 98, that is, checking the directory database for local or same-domain resources, (optionally) verifying the resource, and if necessary starting a domain search. Implementations may decide to start the domain search before checking the directory database.

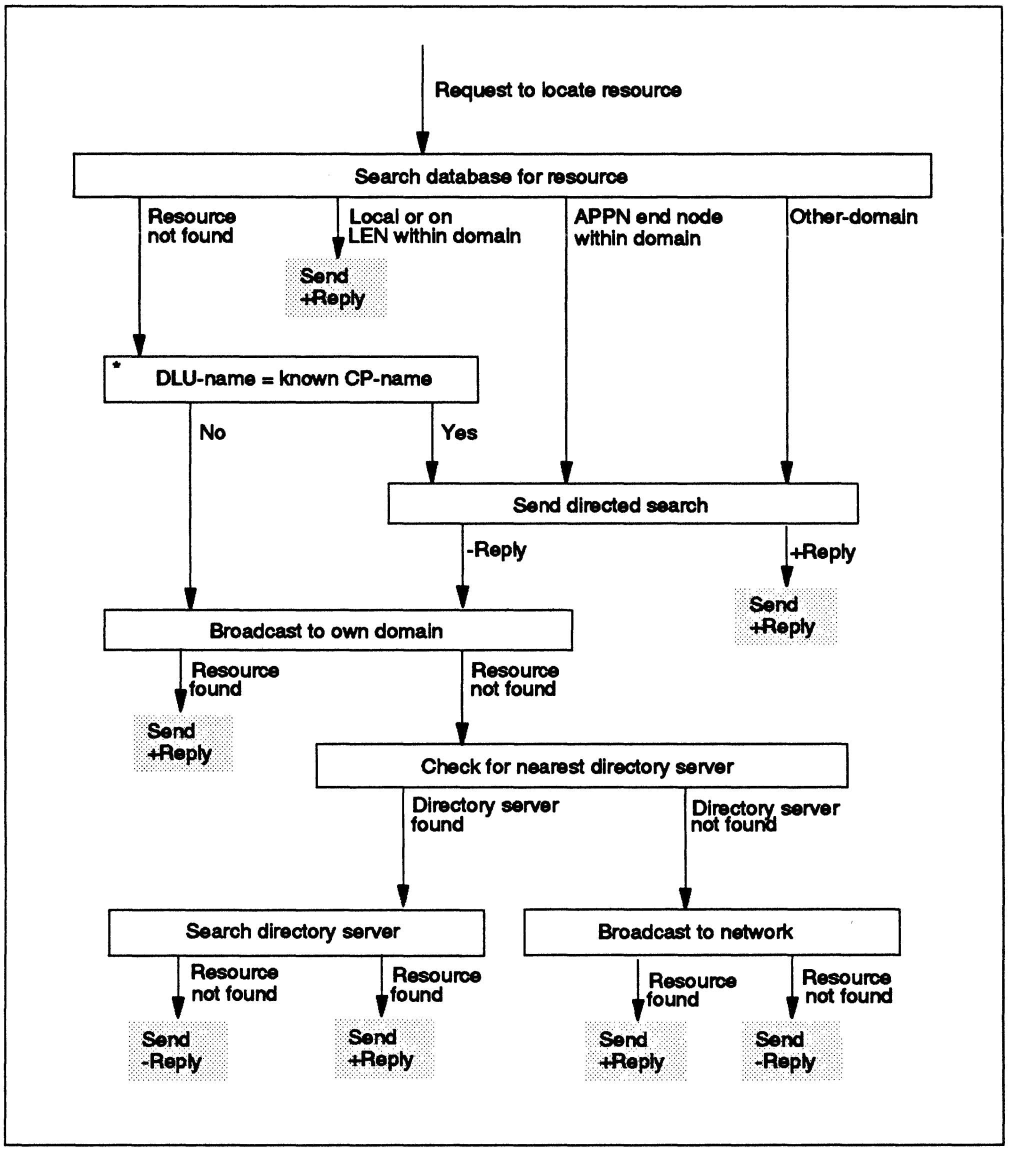

Figure 46. NNS(OLU) Search Logic

Figure 46 depicts the DS search activities on the network node serving the node that owns the OLU.

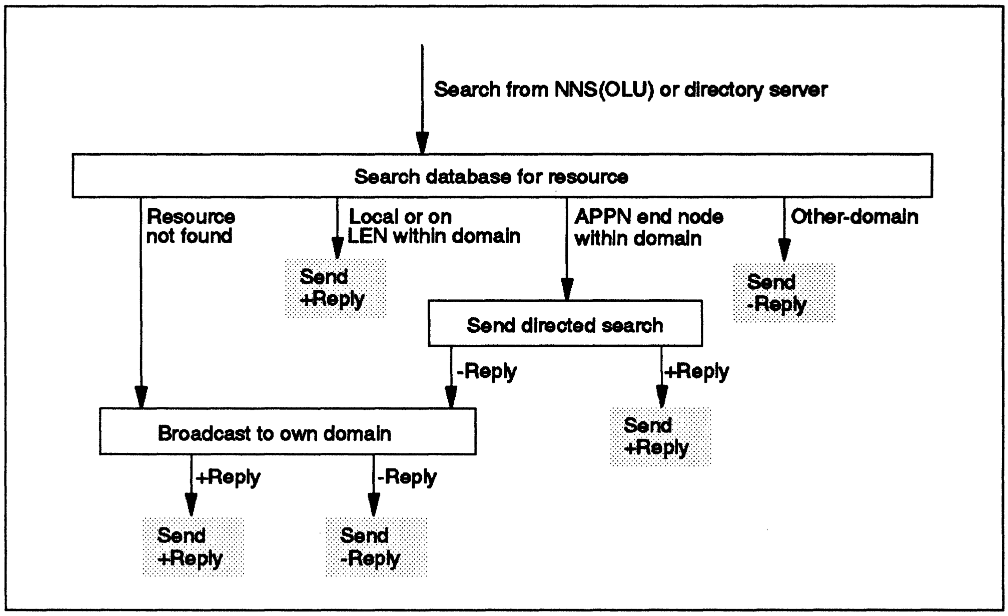

Figure 47. NNS(DLU) Search Logic

Figure 47 depicts the DS search activities on the network node serving the node that owns the DLU.

6.4.6 Search at a Directory Server

The search procedure, see also Figure 48 on page 101, at a network node acting as a central directory server, is as follows:

The central directory server's directory is searched for an entry that matches the query. If the search is successful, the resource will be verified, by sending a directed Locate search request to the NNS(DLU), to obtain the endpoint TG vectors and verify the accuracy of the directory entry. A central directory server will always perform resource verification, even if the nonverify function is supported. For details, see "Nonverify Function" on page 102.

If the resource has not been found or resource verification indicates an erroneous directory entry, the central directory server checks the network topology database for "alternate" directory servers. The central directory server will send a directed Locate search to all alternate directory servers in parallel.

The directory server will use the *first* positive reply to verify the resource and obtain the endpoint TG vectors. Verification is done by sending a directed Locate to the network node server of the destination resource.

During verification it is possible that other replies will be returned as a result of the multiple alternate directory servers being queried. These replies will be discarded if they indicate the same resource location as the one currently being verified. The replies are stored if they indicate a different resource location. Verification will be retried until the stored replies are exhausted or a successful verify occurs.

If after the previously described actions the resource has not been located and/or resource verification was not successful, then the central directory server will start a network broadcast search.

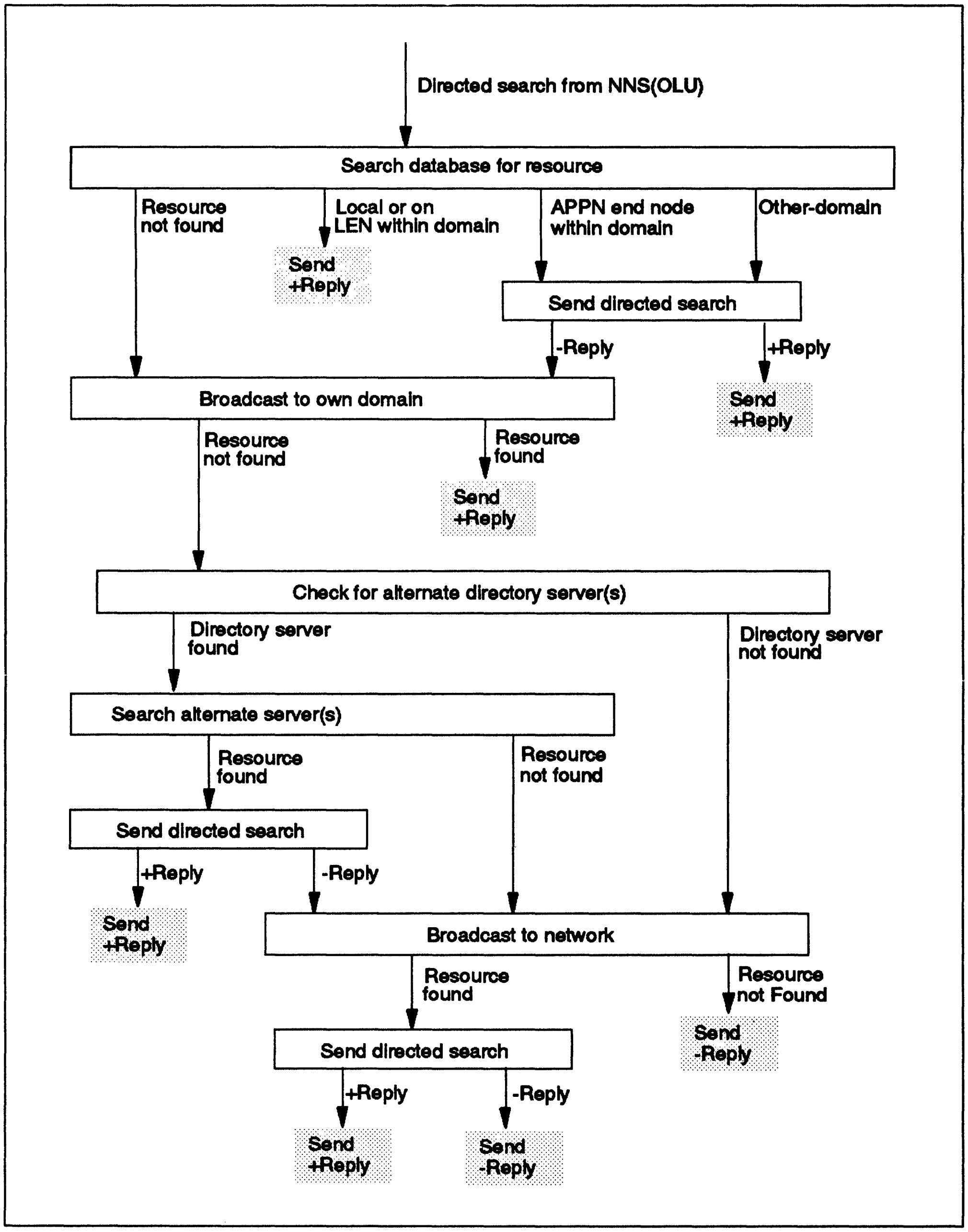

Figure 48. Central Directory Server Search Logic: Overview

6.4.7 Alternate Directory Server

Directory servers with equivalent capabilities will be queried in parallel by the directory server, which is referred to as the "origin" directory server.

When an alternate directory server is queried, it performs the following steps: the local directory is searched and, optionally, a domain search is started. The domain search is optional, since it is the intention of this search to treat the directories of alternate directory servers as an ***extended cache***.

If the alternate directory server locates the resource in its local directory as being in its domain, it may optionally verify that resource. In this case, the origin directory server is informed that verification was successful. An alternate directory server will not perform a network broadcast search for a resource it cannot find locally; the network broadcast search will be done by the central directory server that was queried first by the NNS(OLU).

6.5 Nonverify Function

In the base APPN architecture, a network node that receives a session initiation request will send a Locate search request to the node containing the destination LU. This is done for two reasons. The first reason is to assure the accuracy of directory cache entries; if the information is incorrect, the search will fail and the network node will send a directed search to a central directory server or start a broadcast search. The second reason is to obtain the endpoint TG vectors of the node containing the destination LU.

The following sections describe an optional function, called the ***nonverify function*** that provides an alternate method to obtain endpoint TG vectors. The nonverify function reduces the network traffic and the time required to establish a session.

Support for the (optional) nonverify function is in two parts, one at an end node and one at a network node. The support at a network node is further divided into an NNS(OLU) and an NNS(DLU) part.

- ***End Node part:***

  Assuming both the end node and its network node server support the nonverify function, the end node will send topology database updates (TDUs) describing its endpoint TGs and notify the network node server of BIND failures caused by incorrect directory cache entries.

- ***NNS(OLU) part:***

  The network node server of the originating LU expands the existing APPN support for caching results of directory searches. Besides resource location information, directory services (DS) on a network node will also cache the endpoint TG vectors of the end node owning the DLU. If an NNS(OLU) has this information when it receives a Locate request indicating that the use of the nonverify function is allowed, it will return a reply containing the Route Selection Control Vector (RSCV) for the session, without sending a Locate search request to the node containing the destination LU.

- ***NNS(DLU) part:***

  The network node server of the destination LU part of the nonverify function expands the existing APPN support of end node registered resources. It will use the information it has from the end nodes to respond to nonverify Locate requests.

Each network node will receive TDUs from end nodes in its domain that support the nonverify function. It will store them in its own topology database, but not broadcast them to any of the adjacent network nodes.

6.5.1 Nonverify Function at the End Node

When an end node sends a Locate request to its network node server, it indicates whether the use of the nonverify function is allowed, using a ***verify not required*** indicator. This indicator should be set ON if the Locate request is for a PLU-initiated session that can tolerate a BIND failure, and the last BIND sent to that target LU has not failed because of using the nonverify function.

When an end node receives the reply from its network node server, the ***verify not performed*** indicator will be set ON if the nonverify function was actually used by the network node that replied.

If an end node receives the reply from its network node server with the "verify not performed" indicator ON, and the following BIND fails due to the nonverify function, the end node should send another Locate request. The "verify not required" should be set OFF, forcing the network node server to find the target LU and update its information about the target LU. BINDs may fail because information that DS has cached is not correct: for example, an endpoint TG, being used for route computation, is not active anymore.

Note: Setting the "verification not required" indicator appropriately on Locate requests and repeating a search that failed because of the nonverify function are functions that are also performed by a network node that contains an OLU.

6.5.2 Nonverify Function Aspects Common to both of an NN's Roles

An end node that supports the nonverify function will send topology database updates (TDUs) representing its ***entire*** local topology database to its network node server, when, during exchange of CP capabilities, the network node server has indicated that it supports the nonverify function. This is done to make the network node server aware of direct connections between end nodes and connections from the end node to connection networks and to other network nodes.

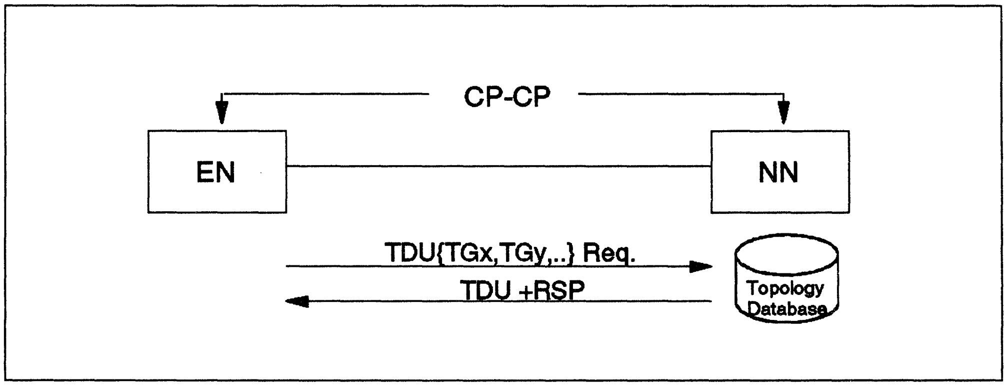

Figure 49. TDUs from APPN End Nodes

A network node implementing the nonverify function *also* caches endpoint TGs in its directory database, as described in the following section.

Network Node Information Caching

Base APPN architecture allows a network node to create or update cache entries in its directory database when it is either the network node serving the search origin or the network node serving the search destination. Cache entries made by an OLU server describe the search origin and the search destination; those made by a DLU server describe the search origin. Such entries are made both on directed and broadcast searches.

To improve cache maintenance for the nonverify function, network nodes use instellation-specific default values for a ***reference count*** and/or a ***validity timer value*** (see "Network Node Cache Maintenance" on page 105 how these values are used). Use of the timer value is obligatory, but use of the reference count is optional. Whenever a cache entry is created, a timer value and a reference count are associated with that particular entry. The reference count will always be initialized with the installation's default value, and the timer value will be the installation's default value or the value received from another node supporting the nonverify function, whichever value is smaller.

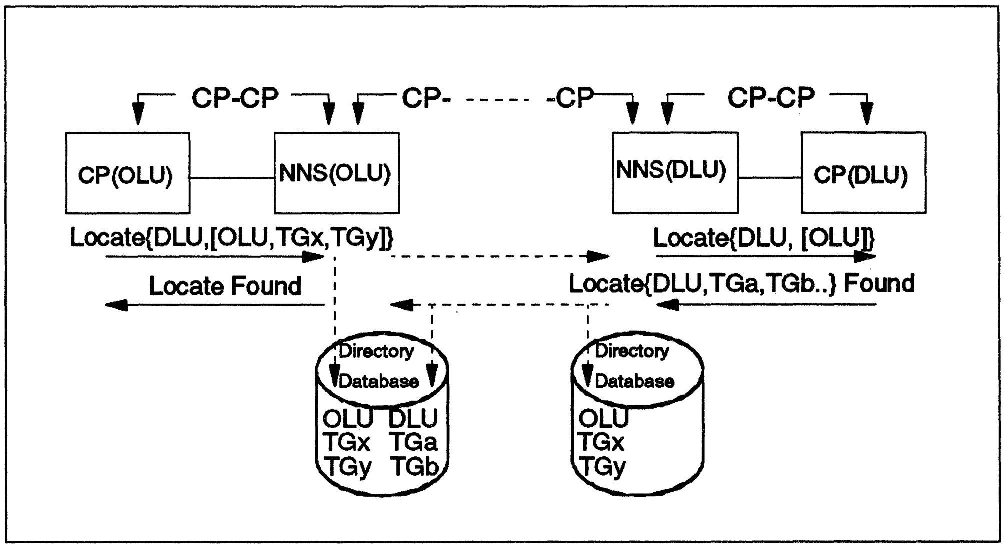

Figure 50. Resource and TG Caching

Cache entries will show whether a resource is ***available***, ***unavailable***, or ***unknown***. For a resource of type LU, "available" means that it is able to accept or send a BIND. "Unavailable" resources are present within the network but currently not ready for use. A cache entry indicating that a resource is unavailable prevents the network from being flooded with session requests for a major subsystem (like IMS for example) during recovery after a failure, delaying recovery or even bringing about conditions similar to those which have caused the initial subsystem failure. Cache entries for "unknown" resources indicate that a previous broadcast search was not successful. The status "unknown" prevents repetition of broadcast searches for resources that have not been defined in the network.

Whenever a network node caches, in its directory database, the resource location present in a Locate search request (search origin) or a Locate search reply (search destination), it will also cache the endpoint TG vectors (if present) of the node containing the resource. Availability information is cached from Locate search replies. The intention is to always cache the most recent information, so that it can be used to satisfy other nonverify Locate requests.

If a search destination is not found after a broadcast search, the originating network node will add a cache entry indicating that the destination resource is "unknown" and initialize the associated timer and reference count to their default values. Until this entry is deleted, because the timer is expired or the reference count exhausted, it will respond to the next search requests for the "unknown" resource by sending a negative Locate reply. No domain search is performed. This concept is also known as ***negative caching***.

Network Node Cache Maintenance

As in base APPN, a network node may delete its cache entries that are least recently used, once the cache is full. Another way for the cache entry to be deleted is to wait until the associated timer or reference count expires.

Whenever a node references a cache entry, it decrements the reference count associated with that entry. If, when an entry is referenced, the reference count reaches zero or the timer value has been exceeded (the entry "expires"), the node's subsequent action depends on whether the entry shows the resource to be "available," "unavailable," or "unknown."

When an entry for an unknown resource has expired, the node simply deletes it. A subsequent search for the resource will result in the processing of the normal search logic with the result that the resource is either still unknown or available or known but unavailable.

When an entry for an available or unavailable resource has expired, the node uses the location information contained in the expired entry to attempt to refresh this entry. This is done when an expired entry is referenced. Refreshing the cache means sending a directed search and, if that search fails, performing normal network search logic. The result of the search will be an "available" or "unavailable" resource with up-to-date location and endpoint TG vectors, or an "unknown" resource entry. The associated timer and optional reference count will be reinitialized.

A cache entry can also alter its indication of whether a resource is "available" or "unavailable" through the normal action of the caching mechanism. This can happen, for example, if the network node receives a Locate request for a cached resource and it learns that the status of this resource has changed when following the procedures described in "Nonverify Function at Origin Network Node Server" on page 106 and "Nonverify Function at Destination Network Node Server" on page 107.

Network Node Nonverify RSCV Calculation

The NNS of the PLU will compute the Route Selection Control Vector (RSCV) for a session using the endpoint TG vectors of the CP(SLU) and CP(PLU). CP(PLU) TG vectors can be obtained from either the NNS topology database or from the Locate request. End nodes supporting the nonverify function may optionally include PLU TG vectors in a Locate request. If they are present, the NNS(PLU) will use them to compute the route for the particular session.

The DLU TG vectors that are used by the NNS(OLU) in computing an RSCV to reply to a nonverify Locate request come from the directory cache of the NNS(OLU). If the network node server of the OLU has not cached the necessary information, a Locate request will be forwarded, as either a directed or a broadcast search to the NNS(DLU).

When both the PLU and SLU are in the same domain and the network node's role is both NNS(OLU) and NNS(DLU), it may use the TG vectors in the directory cache. However, in computing the RSCV, the NNS(OLU) will first check the CP(OLU) TG vectors in its topology database for direct connections between CP(OLU) and CP(DLU). This is because TG vectors contained within the NNS directory cache will be specific to a previous CP(DLU), CP(OLU) Locate request. CP(DLU)s return TG vectors reflecting all connections to network nodes, connection networks, and any direct connection to the CP(OLU), but this may not be the same CP(OLU) that is currently requesting a session to the CP(DLU).

In base APPN the only way to establish a session is that an LU requests a session and this LU then becomes the primary LU for this session. As in subarea SNA sessions are often initiated by dependent LUs, which can act only as secondary LUs, APPN VTAM supports also SLU-initiated sessions (see "Session Services Extensions" on page 122). For these SLU-initiated sessions, an optimization is possible because the network node serving the PLU will receive the Locate request before the PLU node does. If it knows the PLU location and has PLU TGs, it can compute the RSCV for the session and include it in the Locate request that is sent to the PLU node. The PLU node is then able to send the session BIND immediately after returning a positive reply to the Locate search request.

6.5.3 Nonverify Function at Origin Network Node Server

When a network node performing the role of NNS(OLU) receives a Locate request, its actions are based on the setting of the "verify not required" indicator in the Locate request and the contents of its cache. When the cache entry shows:

1. ***Resource Available***

   And:

   a. ***Verify not required*** is specified in the Locate

      The NNS will attempt to compute an RSCV for the session using the DLU TG vectors from its directory cache. If the RSCV computation is successful, a Locate reply containing the RSCV will be returned; this reply will have the ***verify not performed*** indicator set. If it is unable to compute an RSCV for the session, it will perform its normal network search logic.

   b. ***Verify required*** is specified in the Locate

      A directed search will be sent to the node containing the destination resource. If the search fails, it will perform its normal network search logic.

2. ***Resource Unavailable***

   If the cache entry shows the destination resource is "unavailable" then a directed Locate will be sent to the NNS(DLU), regardless of the "verify not required" indicator. The "verify not required" indicator will be set equal to the setting on the incoming request. If the NNS(DLU) replies with resource "available," the new information will be cached and session establishment will proceed. If the NNS(DLU) replies with resource "unavailable," the new information will be cached and session establishment will not proceed. If the NNS(DLU) replies with resource "unknown," the NNS(OLU) will perform its normal network search logic.

3. ***Resource Unknown***

   A negative Locate reply will be returned, regardless of the "verify not required" indicator. A resource is considered "unknown," when it is not found during a previous broadcast search. An aging mechanism ensures that after a specified time or number of references this cache entry is deleted and a resource that gets available can be found (see "Network Node Cache Maintenance" on page 105).

4. ***Resource Not Cached***

   The normal network search logic will be performed.

Whenever a network node does its normal search logic, the "verify not required" indicator will be set as it was in the received session request. Whenever an NNS(OLU) returns a search reply to the OLU node, the "verify performed" indicator will be set as it was when the search reply was received from another network node.

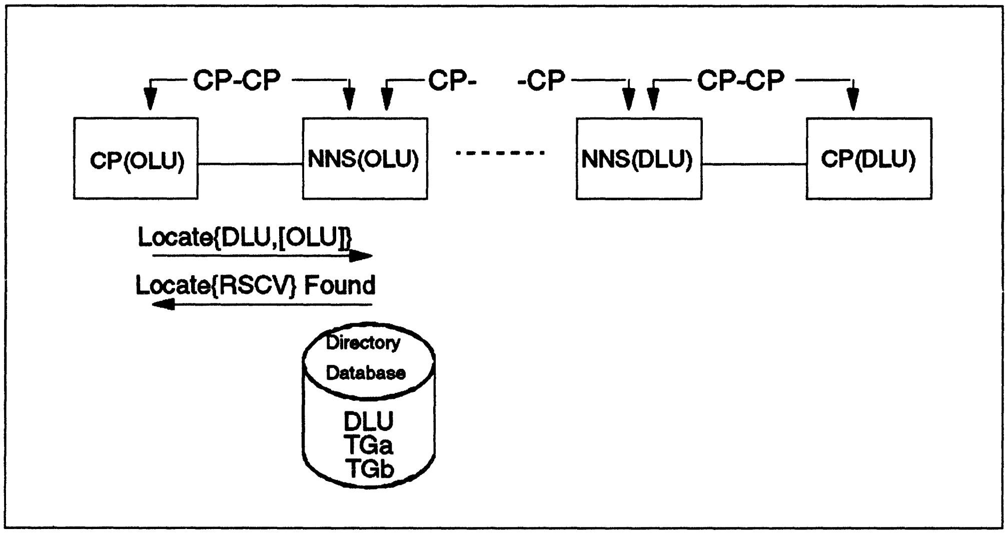

Figure 51. Nonverify Function at Origin Network Node Server

Verified Replies Are Preferred: In base APPN, a network node computes a session route using the first explicit (non-wildcard) Locate reply it receives after issuing a broadcast search. This changes for network nodes when implementing the nonverify function. If the NNS(OLU) has performed a broadcast search, it will no longer automatically accept the first positive reply. Rather, it will accept the ***first*** verified explicit positive reply, or if none is received, the ***last*** nonverified acceptable reply. This design allows NNS(OLU) to hold replies in only one buffer; arriving nonverified replies are stored overwriting previous nonverified replies until either the first verified reply arrives, which is then used, or the search is completed, in which case the stored last nonverified reply is used.

6.5.4 Nonverify Function at Destination Network Node Server

A network node server may receive a Locate request for a DLU within its domain, with the "verify not required" indicator set to ON if the originating network node lacks the information needed to satisfy a search request that shows that verification is not required. If the originating network node does not know the DLU's location, its Locate request will take the form of a broadcast search. If it knows the DLU's location, but was unable to compute a route or refresh the cache entry for a known (available or unavailable) resource, its

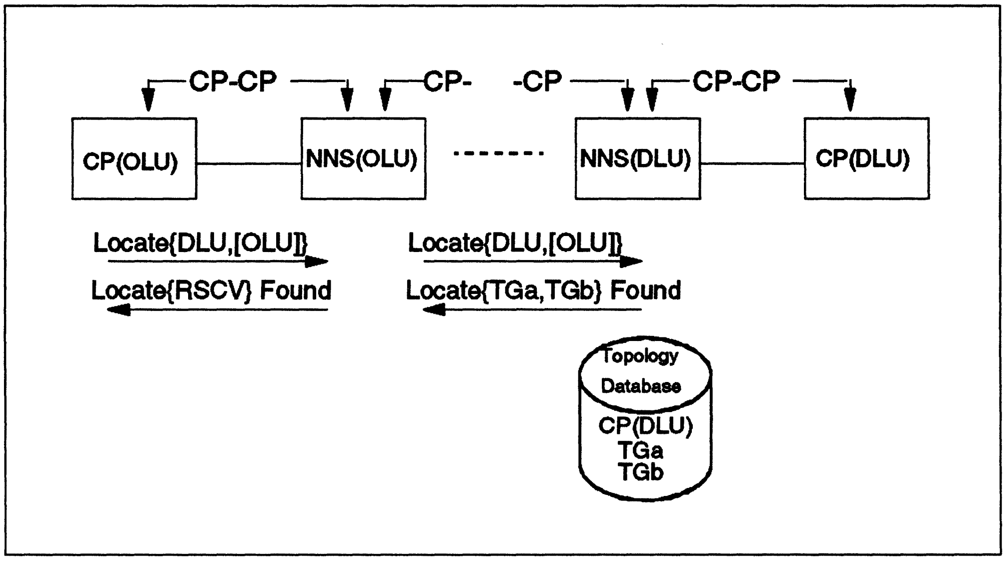

Figure 52. Nonverify Function at Destination Network Node Server

Locate request will take the form of a directed search (to obtain up-to-date endpoint TG vectors and availability information).

When the DLU is located on the receiving network node itself or on a client LEN end node, the network node **always** sends a positive Locate reply with the "verification not performed" indicator set to OFF (meaning verification performed).

When the DLU is located on a client APPN end node, the network node sends a positive Locate reply with the "verification not performed" indicator set to ON (meaning verification not performed) if the the Locate request indicates "verification not required." If verification is required, the network node server will forward the Locate request to the adjacent end node

6.5.5 Nonverify Function at Central Directory Server

A central directory server is a network node that supports the central registration server function, and locate functions performed are, to a large extent, the same as described in "Nonverify Function at Origin Network Node Server" on page 106; however, some differences exist.

An origin network node server may decide to query a central directory server, if its directory database does not contain information about the destination resource; for details, see "Central Resource Registration (CRR)" on page 87. The central directory server will search its directory database and may eventually query alternate directory servers. Alternate directory servers will be invoked only if no directory information can be found, or the resource status is *unknown*.

Information obtained from either the central directory server's directory database or from alternate directory servers indicating the resource is *available* or *unavailable*, must always be verified if the the resource is an other-domain resource, because a directory server never caches TG information for other-domain resources.

A positive reply, together with the resource location and TG vectors, will be returned for *available* resources. A negative reply will be returned for *unavailable* or *unknown* resources. This forces the network node to forward any future searches for the resource to the central directory server, which is more likely to have accurate resource information.

6.5.6 Last-Hop BIND COS Validation and TG Selection

Base APPN architecture leaves open a possibility that the endpoint TG vectors that were used during the route computation are no longer valid when the BIND is ready to traverse the last hop of the session route. The penultimate node on the session path (the next-to-last node specified in the RSCV) may be the DLU's network node server or it may be some other network node to which the end node once reported that it had connectivity.

To make matters worse, no mechanism exists to notify a nonadjacent network node of a change in cached endpoint TG vectors that it has previously obtained using the nonverify function. With the nonverify function, a greater time (on average) will have elapsed between a network node learning and caching the CP(DLU) TG vectors, as compared to base APPN which always uses a directed Locate to obtain a fresh set of end node TG vectors before route calculation takes place.

A fix for this is implemented on all nodes that support the nonverify function:

Important

Note: A key assumption underlying this design is that Class of Service (COS) definitions have to be uniform throughout an APPN network.

Before a node, located one hop away from an end node destination, sends a BIND to that SLU node, it will check the RSCV to determine whether the characteristics of the last-hop TG are acceptable for the class of service specified in the BIND. If they are, the BIND can be sent in the normal manner. (The penultimate node uses its usual techniques to map COS to TG characteristics.)

- If no such TG number currently exists but a TG is available with acceptable characteristics, then the BIND is sent on the best available active TG, as if it were the TG initially specified by the PLU.
- If no TG yielding the desired class of service is currently active between the two nodes, the penultimate node will try to activate the best available automatically activatable TG to the end node that can provide the required class of service.
- If none of these actions is possible, then the penultimate node will reject the BIND with an extended sense data control vector indicating the reason for the session failure.

In both cases where a different TG is used from the one originally specified, the penultimate node modifies the BIND RSCV before sending it to the SLU node to accurately reflect the TG number actually selected. This is done for network management purposes. The BIND RSCV is stored at the SLU node, and the RSCV for the BIND response is generated from the BIND RSCV and returned to the PLU node, where it is also stored, in case it is needed for session problem determination.

Chapter 7. Session Services

The session services component of the control point generates unique session identifiers, activates and deactivates CP-CP sessions, and assists LUs in initiating and activating LU-LU sessions.

7.1 Functional Overview

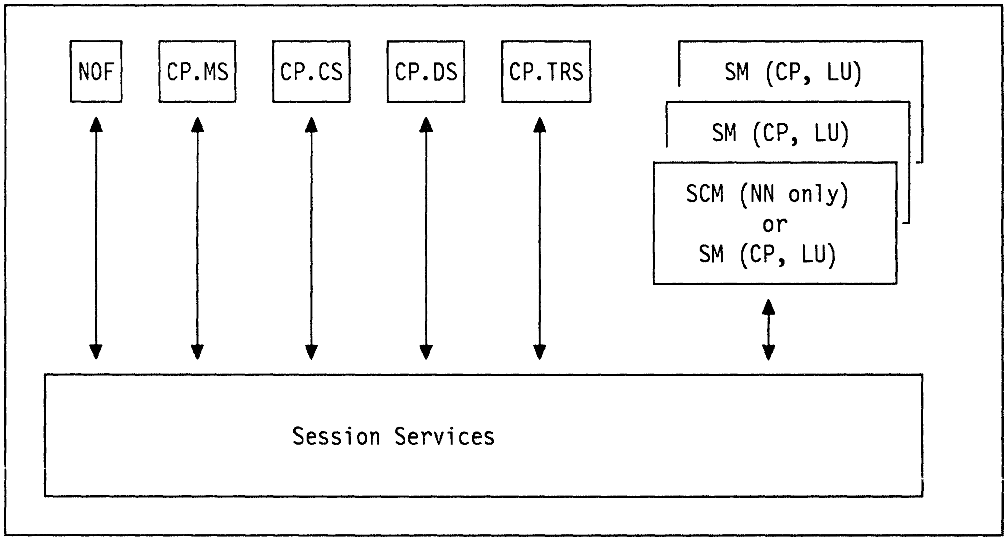

Figure 53. Overview of SS Interaction with Other Components in the Node

The following information is passed when session services is initialized by the node operator facility (NOF):

- Type of node
- CP name of this node
- Network ID of this node
- Indication if the COS/TPF function is supported

  The COS/TPF function allows a node to translate a mode name to class-of-service (COS) name and an associated transmission priority (TP). For more information see “Class of Service (COS)” on page 69.

- Indication if the node is to be included in a network node domain search

  If a network node server is not able to locate a resource it may decide to query all authorized APPN end nodes within its domain. Only the APPN end nodes that have explicitly indicated they want to be searched, will be included in the domain search. For more information see “Domain Broadcast Search” on page 94.

Session services (SS) generates unique session identifiers, activates and deactivates CP-CP sessions, and provides LU-LU session initiation assistance and information to the session managers (SM) representing the LUs at the endpoints of a session. SS invokes directory services (DS) to locate a partner LU, invokes topology and routing services (TRS) to calculate an optimum route between an origin and destination node, informs management services (MS) about newly activated or inactivated CP-CP sessions, and may invoke configuration services (CS) to activate TGs. Each of these functions will be described in the following sections.

7.2 Fully Qualified Procedure Correlation Identifier (FQPCID)

Session services assigns network-unique session identifiers, also called the "fully qualified procedure correlation identifier" (FQPCID), for the following reasons:

- To correlate requests and replies sent between APPN nodes.

  Examples are resource registration requests, topology database updates (TDUs), and Locate requests exchanged during session initiation. Note, that the term "session identifier" is somewhat confusing, as the FQPCID is also used to identify non-session type data.

- To identify a session during cleanup or recovery procedures.
- To identify a session for problem determination.
- To identify a session for accounting, auditing, and performance-monitoring purposes.

The FQPCID is assigned at the node from which a session establishment or non-session request originates. A session related FQPCID identifies a particular session as long as this session remains active and all requests and replies that relate to this particular session (Locate, BIND, UNBIND) include its FQPCID.

To ensure uniqueness throughout the network, the FQPCID consists of a fixed-length (8-byte) PCID field concatenated with the length and the qualified network name of the control point that generated the FQPCID. The PCID contains a four-byte value derived (using a hashing function) from the qualified network name of the CP and a four-byte sequence number. The sequence number is incremented by 1 each time session services assigns an FQPCID. The initial value of the sequence number is either derived (also using a hashing function) from the time-of-day (TOD) clock, or, for implementations that do not have a suitable clock, a monotonically increasing number (until wrapping occurs) with the last value safely stored across IPLs. A detailed description of the FQPCID generation process can be found in *SNA Type 2.1 Node Reference*.

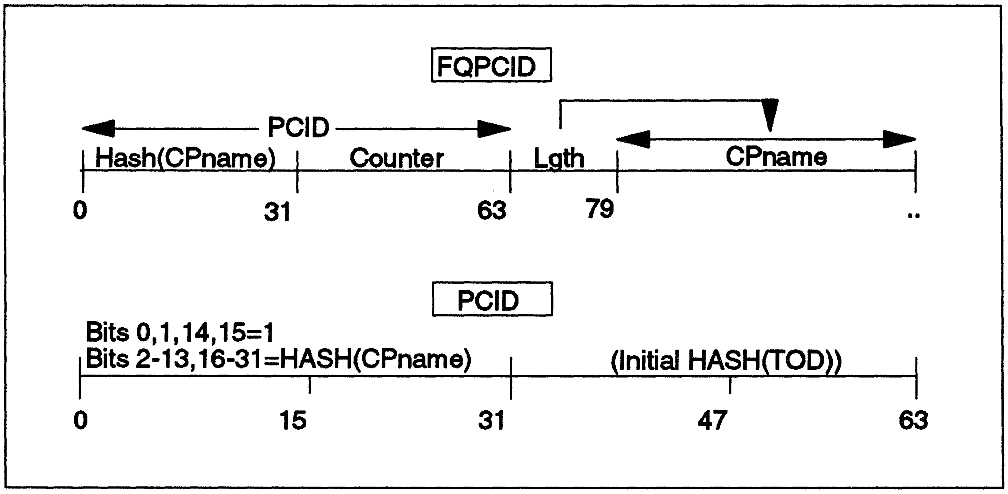

Figure 54. Fully Qualified Procedure Correlation Identifier (FQPCID)

Although the FQPCID is intended to be unique, collisions may occur because of the nature of hashing. When a collision is detected, a negative response is returned to the originating node. The node generating the colliding FQPCID is responsible for resolving the collision. When a collision occurs, a new PCID is generated by adding a random number in the range 1-255 (8 bits) to bits 0-31 of the PCID and forcing the format bits (0, 1, 14, and 15) to 1. This new value is then retained as a seed value in future PCID generation.

Note: Collisions of FQPCIDs may occur only when duplicate CP names have been assigned.

7.3 CP-CP Session Activation

CP components, specifically, topology and routing services (TRS), directory services (DS), session services (SS), and management services (MS) use CP-CP sessions between adjacent APPN nodes to exchange information with their counterparts in other APPN nodes. Examples of CP-CP session usage are the exchange of network topology updates (TDUs), distribution of Locate search requests, and the exchange of CP capabilities. Note that CP-CP sessions are established between adjacent nodes only. If a CP component communicates with a peer component on a nonadjacent node, the information to be exchanged travels via multiple CP-CP sessions.

CP-CP sessions are always logical unit type 6.2 (LU 6.2) sessions. Using this session type, a contention situation could arise if both session partners attempted to allocate a conversation at the same time. This situation is resolved by defining one of the sessions the *contention-winner*, also often called *conwinner* session and the other the *contention-loser*, or *conloser* session. The primary session partner refers to its session as the contention-winner session, and the secondary session partner refers to that same session as the contention-loser session. For more information on LU 6.2 protocols see *SNA LU Version 6 Reference: Peer Protocols*.

CP-CP sessions are always established in parallel, such that each partner maintains a conwinner and a conloser session. Each node will use its conwinner session to transmit requests and to send replies.

Basic APPN architecture designates exchanges of XID3s as the means of requesting CP-CP sessions with adjacent nodes. When a link between two adjacent APPN nodes is activated, each node indicates, as part of an XID exchange, if CP-CP sessions are required and supported. Configuration services, at each node, then signals session services that CP-CP sessions are required, causing SS to initiate the conwinner CP-CP session between the two nodes. This occurs asynchronously at both ends, resulting in the activation of the parallel CP-CP sessions between the two nodes.

End nodes and network nodes will now determine, independently from the information given in the XID3 exchange that CP-CP sessions are requested, whether to send a BIND to the adjacent node for CP-CP sessions. As before, an end node is responsible for determining which network node it will select to be its network node server. It indicates its choice of server by sending a BIND for its conwinner CP-CP session to an adjacent network node, which then accepts its role as a network node server by sending a BIND for its conwinner session. This design allows for a simple recovery from failed CP-CP sessions between an end node and its network node server. The end node selects the next network node capable of being its network node server and simply sends a BIND for the conwinner CP-CP session.

When SS initiates a CP-CP session, it notifies DS that the session is pending-active so that DS may queue any network operations involving the CP-CP session partner, for example directory searches, until the contention-winner session becomes active. SS notifies the resource manager to activate the CP-CP contention-winner session with the adjacent node. The session manager of the CP invokes SS to do its normal session initiation (for example, assign FQPCID) with a mode name *CPSVCMG*, which indicates a CP-CP session.

In order for one node to consider an adjacent node enabled, for example to send it Locate search requests, both CP-CP sessions with it must be enabled. The contention-winner CP-CP session is considered to be enabled when SS receives its partner node's CP capabilities on that session. The contention-loser CP-CP session is considered to be enabled when SS has sent its CP capabilities on that session.

7.3.1 Control Point Capabilities

Immediately following the activation of the CP-CP sessions between the CPs in the two nodes, a CP capabilities exchange occurs on the CP-CP sessions. This exchange determines the extent of network services that each node supports, and provides the basis for future CP-CP communication between the nodes.

Each node requests the CP capabilities of its partner node over the CP-CP session it has initiated (contention-winner session), and it includes its own CP capabilities in the request. Each node also sends its own CP capabilities when it receives a request for them over its contention-loser CP-CP session. The exchange of control point capabilities is done using service transaction programs.

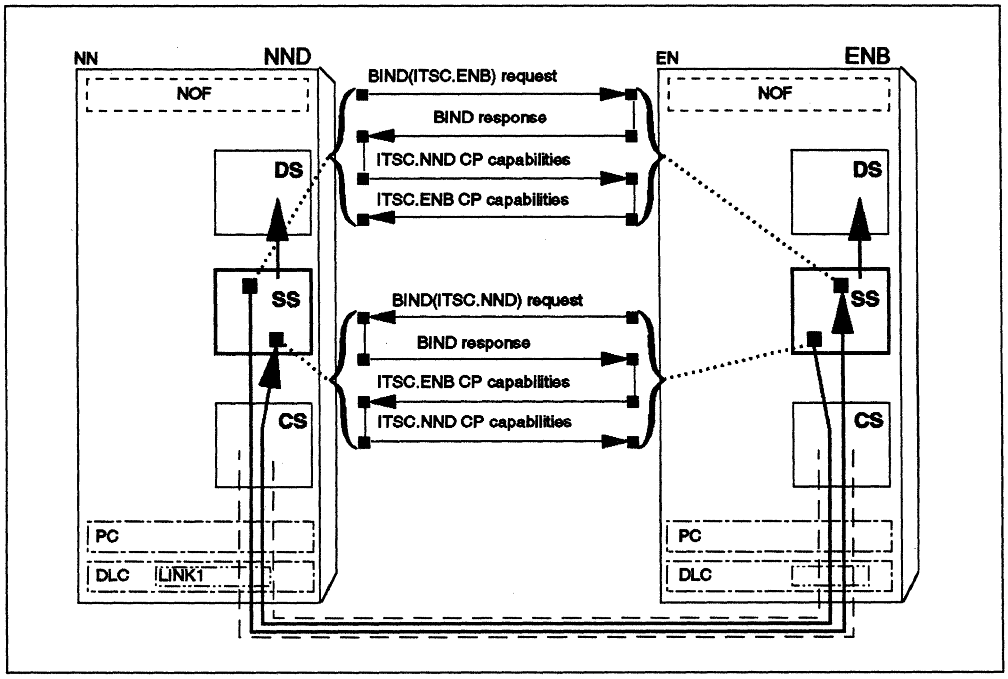

Figure 55. Session Services Activates CP-CP Session

Figure 55 shows how session services in each node activates its contention-winner session with the other node. Session services activates a session by sending a BIND command to the partner node of the session partner; the session partner accepts the session by returning a BIND response. On receipt of the BIND response, the nodes will exchange control point capabilities with each other.

After the CP-CP sessions have been established, the nodes exchange their control point capabilities. Among other things, the following information is exchanged:

- Topology database update (TDU) receipt supported

  The sending CP supports receipt of TDUs. This indication is set by network nodes.

- Flow reduction sequence number (FRSN)

  The flow reduction sequence number identifies the latest CP capabilities or topology database update GDS variable received by the sender of this CP capabilities GDS variable. It is included only when TDU receipt is supported. For details see "Flow Reduction Considerations" on page 66.

- Request/reply management data support

  The sending CP supports receipt of request for management services data and sending replies to the requests.

- CP-CP session activation support

  The sender indicates that he supports the activation of CP-CP sessions independently from link activation.

- Resource search capability

  This parameter is set by APPN end nodes that support a domain search from their network node server. It specifies the resource types for which the end node may be searched for by its network node server. Currently only resource type "LU" is supported. For more information see "Domain Broadcast Search" on page 94.

When a network node server receives an end node's capabilities, they are retained only if the end node is authorized to provide its own CP capabilities. Otherwise the CP capabilities defined locally (by NOF) at the network node for the end node are retained. End node authorization at a network node is specified at system-definition time when NOF defines the end node to the network node.

7.4 CP-CP Session Deactivation

A session services component with active CP-CP sessions may receive requests to deactivate CP-CP sessions with an adjacent node. The main reasons to deactivate CP-CP sessions may be:

- Normal CP-CP session deactivation

  Normal deactivation may be the case if the node itself or the partner node no longer requires the CP-CP sessions. For example, an APPN end node may decide to switch to another network node server or one of the session partners may be taken out-of-service.

- Abnormal CP-CP session deactivation

  Abnormal CP-CP session deactivation is needed when a link failure or serious protocol violations occur on the CP-CP sessions.

In both cases, the CP-CP session will be deactivated. However a link failure will be regarded as a recoverable error and session services will immediately activate the CP-CP session again. If the link failure persists, session services will retry the CP-CP session activation until the retry limit is exceeded. The setting of the retry limit is implementation dependent.

7.5 LU-LU Session Initiation

As described in "Logical Unit (LU)" on page 20, APPN and LEN nodes support LUs that can both initiate sessions and respond to session activation requests. The BIND sender is referred to as the primary LU (PLU); the BIND receiver is referred to as the secondary LU (SLU). A session starts when the PLU sends a BIND and the SLU responds with RSP(BIND) and stops when UNBIND and RSP(UNBIND) are exchanged. The UNBIND may be sent by either LU. The PLU specifies in the BIND request information like:

- The network-qualified name of the PLU
- The network-qualified name of the SLU
- Session characteristics such as maximum RU size and pacing windows
- The route through the network towards the SLU
- The unique session identifier (FQPCID).

Besides using the terms *primary* LU (PLU) and *secondary* LU (SLU), the terms *origin* LU (OLU) and *destination* LU (DLU) are often used as well to indicate which LU (actually the node owning the LU) is responsible for locating the partner LU. The request to locate a session partner flows from OLU to DLU, and the reply in the opposite direction.

In base APPN, the OLU is always the PLU and the DLU is always the SLU. By using the terms OLU and DLU, session setup procedures can be adequately described. However, with the introduction of APPN VTAM, sessions are no longer limited to PLU-initiated sessions. APPN VTAM allows session initiation from the PLU, the SLU, or from a third-party LU. The term, *initiating LU* (ILU), is used to indicate the LU that initiates an LU-LU session. Although a BIND request always flows from PLU to SLU, the Locate search request to find a session partner may originate from either side, depending on which LU has initiated the session.

As only APPN VTAM has implemented *session services extensions*, which among other things allows SLU and third-party session initiation in an APPN network, we have chosen to describe base APPN session services separately from the functions contained within session services extensions. For a description of session services extensions, see "Session Services Extensions" on page 122. Note, base APPN does cover the situation when a VTAM system connects to an APPN network as a LEN end node.

7.5.1 CD-Initiate Processing

Session services (SS) provides LU-LU session initiation assistance and information to the session managers (SM) representing the LUs at the endpoints of a session. Among other things, SS will invoke directory services (DS) to locate a destination LU. DS uses Locate search requests to find a resource. DS allows DS users, such as session services, to exchange data using the Locate search requests.

The primary means of communicating session initiation information between session endpoints in separate nodes is the Cross-Domain Initiate (CD-Initiate) GDS variable. SS will add the CD-Initiate GDS variable to the Locate search request. A Locate search request containing the CD-Initiate GDS variable is sometimes also called a *Locate/CD-Initiate*.

Listed below are the CD-Initiate fields that are modified or referenced by SS at the session endpoints. For a detailed description, see *Systems Network Architecture Formats*.

- Session polarity

  Indicates which LU is expected to be the BIND sender (PLU), either OLU or DLU. In base APPN, the PLU is always identical with the OLU.

- Mode name

  The mode name, allowing a COS to be selected for the LU-LU session requested.

- Additional control vectors

  - COS/TPF control vector

    The COS/TPF control vector is included in the CP-Initiate. Session establishment requests from an end node contain this information only if the end node supports the COS/TPF function. For details, see "Session Establishment and TRS" on page 76.

  - Route selection control vector, RSCV

    The route to the DLU is calculated, and a route selection control vector (RSCV) is built by the network node server of the node containing the originating LU. The RSCV will be used to route the BIND request from the PLU to the SLU.

  - TG descriptor control vector

    An APPN end node includes endpoint TG information in the Locate search request, CP(OLU), and reply, CP(DLU). CP(OLU) includes TG information from endpoint TGs between the end node and adjacent network nodes and connection networks. CP(DLU) includes TG information from the end node to adjacent network nodes and connection networks but also TG information for TGs to CP(OLU). For details, see "Session Establishment and TRS" on page 76.

7.5.2 Directory Search and Route Computation

SS of the origin node initiates the route computation by invoking DS to search its directory for the DLU. DS will search its local directory, which contains entries for LUs residing in the end node or in an adjacent end node. If the local search is successful, and if the LU resides in an adjacent end node, TRS is invoked by SS to generate an RSCV containing the single TG (hop) to the peer node.

If the local search is not successful, DS of an APPN end node passes the Locate/CD-Initiate to its network node server, which initiates a distributed search of the network for the DLU. If the distributed search is successful, TRS of the network node server computes the route and provides an RSCV in the Locate/CD-Initiate reply to the APPN end node; see "Examples."

Note: On LEN end nodes all destination LUs have to be defined as residing on adjacent nodes. LUs that do not actually reside on adjacent nodes need to be defined as if they reside on an adjacent network node. A LEN end node starts a session by passing a BIND to the adjacent node on which it assumes the destination LU resides. If a network node receives a BIND, the network node then takes the necessary steps to locate the DLU and forward the BIND.

7.5.3 Route Activation

After the route has been computed and the RSCV provided to SS of the origin node, the TG on which the BIND has to be sent may not be active. SS, of the origin end node, inspects the RSCV for the first TG (hop) and, optionally, invokes CS to activate that TG. After the TG is successfully activated, SS sends the session information to the session managers (SM) representing the LUs, so that the BIND can be sent to the SLU.

7.5.4 Examples

To establish a session between two LUs requires the invocation of directory services (DS), topology and routing services (TRS), and session services (SS) components on several T2.1 nodes, namely:

- Node owning the OLU: CP(OLU)
- Network node server of the OLU: NNS(OLU)
- Network node server of the DLU: NNS(DLU)
- Node owning the DLU: CP(DLU)

In the following two sections we describe the session establishment between two LUs residing on a LEN end node, and between two LUs residing on APPN end nodes.

LUs Residing on LEN End Nodes

Figure 56 on page 119 depicts the internode sequences involved in session establishment between two LUs residing on LEN end nodes.

Note: The "NNS(OLU)" of an origin LEN end node is the network node that receives the BIND by which a session from a LEN end node is started. The "NNS(DLU)" of a destination LEN end node is the network node to which the LEN end node connects and on which the DLU has been defined.

The figure shows a case where the CP(OLU), which cannot support Locate search requests, relies on NNS(OLU) to build a Locate request, to add the

CD-Initiate variable, and to find the DLU. In this example, the route taken by the Locate request is:

```
NNS(OLU) .. -> .. NNS(DLU)
```

If NNS(OLU) and NNS(DLU) are not adjacent, the Locate request will be routed through intermediate network nodes.

The route taken by the BIND request is:

```
CP(OLU) -> NNS(OLU) .. -> ..  NNS(DLU) -> CP(DLU).
```

If NNS(OLU) and NNS(DLU) are not adjacent, the BIND request will be routed through intermediate network nodes. Because of optimum route calculation for this session, the session data (including the BIND) may follow a different route from NNS(OLU) to NNS(DLU) from the Locate request.

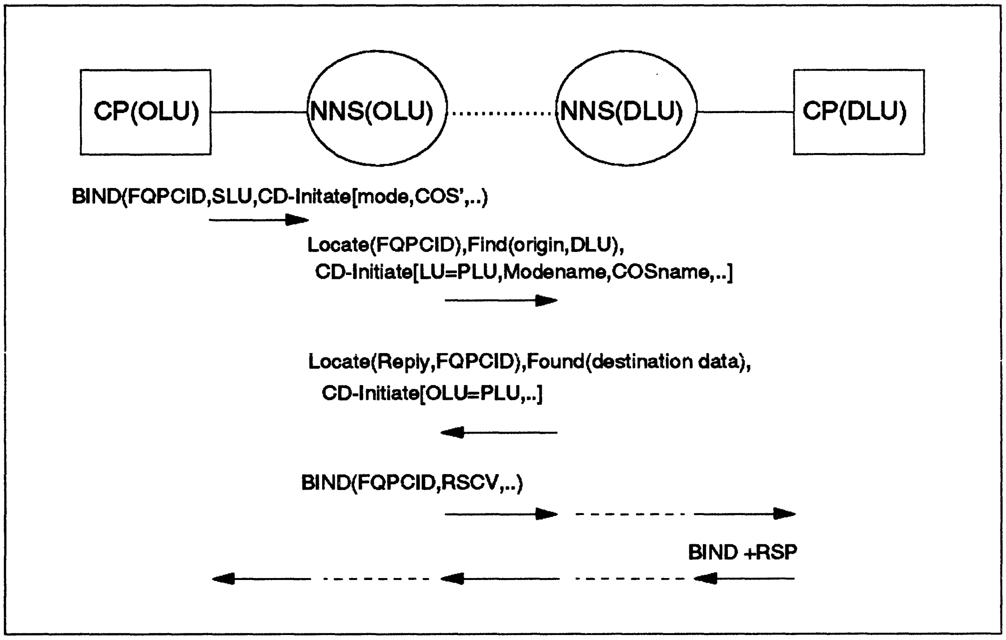
CP(OLU)
NNS(OLU)
NNS(DLU)
CP(DLU)
BIND(FQPCID,SLU,CD-Initate[mode,COS',..)
Locate(FQPCID),Find(origin,DLU),
CD-Initiate[LU=PLU,Modename,COSname,..]
Locate(Reply,FQPCID),Found(destination data),
CD-Initiate[OLU=PLU,..]
BIND(FQPCID,RSCV,..)
BIND +RSP

Figure 56. Session Establishment between LUs on LEN End Nodes

Annotations:

1. To CP(OLU) all destination LUs appear to be adjacent, so the LEN end node sends the BIND to its network node server. The BIND contains the session parameters requested by the OLU, as well as the DLU name, the FQPCID, the mode name and, optionally, the COS name. The COS name will be included only if CP(OLU) supports the COS/TPF function.

2. NNS(OLU) examines the BIND and extracts the DLU name. Since the DLU is not in this node, NNS(OLU) searches its directory database. NNS(OLU) will send a directed Locate request to verify the DLU when the DLU location can be obtained. NNS(OLU) will perform a broadcast search to locate the DLU if no information can be found. See also "Directory Services" on page 83.

   Note: A *cached* directory entry is an entry stored by the network node as a result of a previous search operation revealing the DLU's location; its presence in the directory allows NNS(OLU) to perform a *directed Locate.*

   Before sending a directed Locate request, NNS(OLU) computes a route to NNS(DLU) and provides an appropriate *Locate RSCV.* A Locate RSCV contains a list of CP names from an origin to a destination network node server, as opposed to a *BIND*, or *session RSCV*, which contains a list of CP names and TGs between each adjacent pair of nodes along a route from an origin to a destination endpoint node. The Locate RSCV contains the *shortest* route (minimal hops) between NNS(OLU) and NNS(DLU), whereas the session RSCV contains an *optimum* route between PLU and SLU. TG information is not required in a Locate RSCV as the Locate request is forwarded using CP-CP sessions between adjacent nodes.

NNS(OLU) builds all variables that have to be added to the Locate search request. It includes the FQPCID obtained from the BIND, the OLU and DLU name from the BIND, the CP name of CP(OLU) and NNS(OLU), and the mode and COS names. The COS name is obtained from the BIND or, if not present, obtained from the COS database on NNS(OLU).

3. NNS(DLU) searches its directory database and finds an entry that indicates that DLU is located on a LEN end node within its domain. NNS(DLU) returns a positive reply to the Locate request including endpoint TG information of the TG between NNS(DLU) and CP(DLU).
4. NNS(OLU) receives the Locate reply and uses its network topology database, the TG on which the BIND has been received from CP(OLU), and the endpoint TG vectors received from NNS(DLU) to compute the optimum session route using either the COS obtained from the BIND, or, using the mode name, the COS obtained from its local COS database. Among other things, the route selection control vector (RSCV) indicating the session route is appended to the BIND and sent along the route to CP(DLU).
5. CP(SLU), which in base APPN is always CP(DLU), returns a positive response to the BIND which is sent along the session path to CP(PLU).

LUs Residing on APPN End Nodes

Figure 57 on page 121 depicts the internode sequences involved in session establishment between two LUs residing on APPN end nodes.

This figure shows the case where the CP(OLU), since it supports Locate search requests, sends a Locate search request to its network node server, NNS(DLU), when a session is required. In this example, the route taken by the Locate request is:

```
CP(OLU) -> NNS(OLU) .. -> ..  NNS(DLU) -> CP(DLU)
```

If NNS(OLU) and NNS(DLU) are not adjacent, the Locate request will be routed through intermediate network nodes.

The route taken by the BIND request is:

```
CP(OLU) -> NN .. -> .. NN -> CP(DLU)
```

The BIND is routed along an end-to-end optimum route, which may be different from the route taken by the Locate request. The network nodes in the session path are optional; if links exist between the end nodes, session data (including BIND) may routed directly between the two nodes. Note that, as part of the session establishment, CP(OLU) forwards all endpoint TG vectors to connection networks and adjacent network nodes, and CP(DLU) forwards all endpoint TG vectors to connection networks and adjacent network nodes and possible endpoint TG vectors to CP(OLU). The latter, if available, allows NNS(OLU) to compute a direct route between the end nodes.

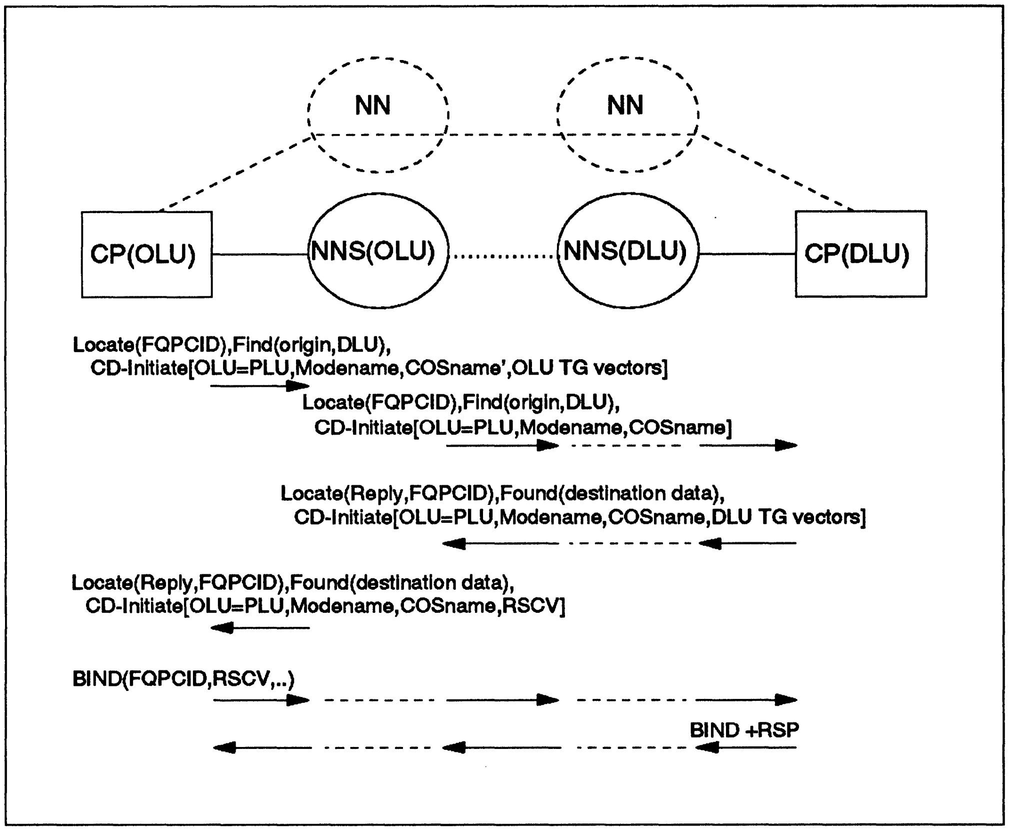
NN
NN
CP(OLU)
NNS(OLU)
NNS(DLU)
CP(DLU)
Locate(FQPCID),Find(origin,DLU),
CD-Initiate[OLU=PLU,Modename,COSname',OLU TG vectors]
Locate(FQPCID),Find(origin,DLU),
CD-Initiate[OLU=PLU,Modename,COSname]
Locate(Reply,FQPCID),Found(destination data),
CD-Initiate[OLU=PLU,Modename,COSname,DLU TG vectors]
Locate(Reply,FQPCID),Found(destination data),
CD-Initiate[OLU=PLU,Modename,COSname,RSCV]
BIND(FQPCID,RSCV,..)
BIND +RSP

Figure 57. Session Establishment between LUs on APPN End Nodes

Annotations:

1. In this configuration, CP(OLU) sends a Locate request to its network node server. The Locate/CD-Initiate contains all of the end node's endpoint TG vectors, as well as the DLU name, the FQPCID, the mode name and, optionally, the COS name. The COS name is included only if CP(OLU) supports the COS/TPF function.

2. NNS(OLU) searches its directory database as described in the previous example and sends a Locate/CD-Initiate request, and either a directed or a broadcast search request, to NNS(DLU). For details see 2 on page 119.

   NNS(DLU) forwards the Locate/CD-Initiate to CP(DLU).

3. CP(DLU) returns a Locate/CD-Initiate reply including endpoint TG vectors, confirming the location of the DLU.

4. NNS(OLU) receives the reply with the destination node's endpoint TG vectors. It uses the TG vectors in conjunction with its topology database to compute the optimum route using the COS, which is either obtained from the BIND, or, using the mode name, obtained from its local COS database. The resulting RSCV is appended to the Locate/CD-Initiate reply, which is returned to CP(OLU).

5. CP(PLU), which in base APPN is always CP(OLU), constructs a BIND and copies the RSCV from the Locate/CD-Initiate reply into the BIND. CP(PLU), and, subsequently, each intermediate network node along the session route, examines the RSCV to determine how to route the BIND request and initialize the session path.

6. CP(SLU), which in base APPN is always CP(DLU), returns a positive reply to the BIND which is sent along the session path to CP(PLU).

7.6 Session Services Extensions

Base APPN architecture provides functions that support PLU-initiated sessions only. There are no facilities to provide functions like SLU-initiated sessions, session queuing, or third-party initiation, functions well known and widely used in subarea SNA.

If all application programs and devices became LU 6.2 capable, these functions would be less important because LU 6.2 capable LUs can send a BIND, instead of a "request BIND," and establish parallel sessions instead of session queuing because of session limits. However, a possible migration from other LU session types to LU 6.2 would take a long time, and facilities such as third-party initiation and queuing for availability continue to be required for menu, security, and other applications. Also, LU 6.2 capable LUs such as printers have limited resources, and thus session limits may need to be imposed.

Enabling session services extensions for LU-LU sessions requires the implementation of this function on each of the nodes on which session endpoints reside, and on possible network node servers if any of these nodes is an APPN end node.

Note: Only VTAM supports session services extensions; therefore, in Figure 58, ENA, NNB, and NNC can be only VTAM or composite (VTAM/NCP) nodes. For details about the APPN implementation in VTAM, see "APPN VTAM" on page 169.

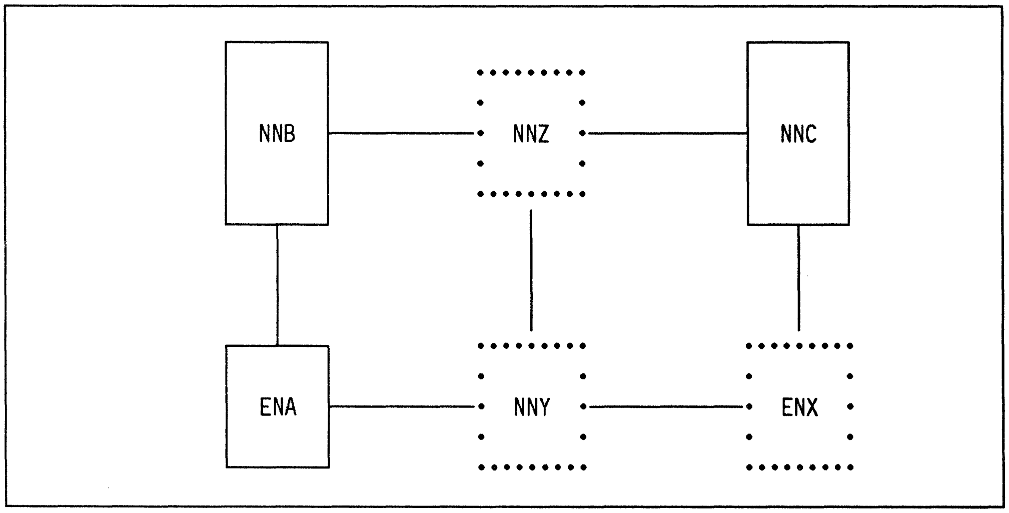

Figure 58. Session Services Extensions.

Notes:

1. The nodes (ENX, NNY, NNZ) in the dotted boxes do not have the session services extensions functions installed. LUs on these nodes can participate only in PLU-initiated sessions.
2. The other nodes (ENA, NNB, NNC) support the session services extensions. LU-LU session establishment can take place in many ways.
3. It is essential for the LUs on an end node (ENA) that the network node providing the server function (NNB) has implemented the additional functions as well.

Details about session services extensions are given in the next section.

Implementing the session services extensions:

- Enhances the support provided for LU 6.2 sessions.
- Is required to fully support the attachment of dependent LUs to APPN networks.

  For a description of dependent LUs and their support within APPN networks, see "Dependent and Independent LUs" on page 20.
- Is necessary on the interchange node (see the discussion of interchange node in "APPN VTAM" on page 169) connecting APPN and subarea networks in order to provide transparency to the LUs in subarea networks.

7.6.1 Additional Functions

Session services extensions defines additional information in the cross-domain initiate GDS variable in order to support functions currently available and widely used in subarea networks:

1. ***Additional Types of Session Initiation***
   1. ***SLU Initiate*** - An SLU must be able to initiate sessions, and a node that attaches dependent LUs must be able to provide mode names, COS names, BIND images, and device characteristics when required.
   2. ***Queuing*** - Session queuing is the process of suspending the establishment of an LU-LU session until a needed resource (either an LU or a session with that LU) is available. There are two basic reasons for the queuing of a session initiation request, and it may be queued for either or both reasons. The reasons are:
      - ***Queue for Enabled:*** A session initiation request may be queued because the PLU or SLU is not enabled for a session (for example, a printer is powered off or an application program is not initialized). Queuing for enabled is performed once the destination LU has been found.
      - ***Queue for Session Limit:*** When an LU has the capability to have only a limited number of sessions and this number is reached, subsequent session initiation requests may be queued. Once a current session is terminated, the node containing that LU will dequeue the first session initiation request that was queued for session limit.

      Session initiation requests that are queued require that the Locate chain be maintained between the nodes of the session partners as long as the request is queued. Once the required resource becomes available, the node managing that LU dequeues pending session initiation requests for that LU and resumes the network flows needed to establish the sessions.

      Session requests indicate the queuing position for the request, should it become queued. Normally, requests are queued FIFO (first in, first out). This allows session requests to be dequeued in the order they are received. However, to support VTAM's version of third-party initiate (CLSDST PASS), LIFO (last in, first out) is used to ensure that the SLU is directly passed from the current PLU to the next PLU indicated in the request.

c. ***Third-Party Initiation*** - A function, limited to PLUs, that allows the LU to establish a session between an LU it is currently having a session with, and a third LU. The LU being passed must be the SLU in the new session. The LU initiating the session setup request may be a menu server, a help function, or some other application program that might have reasons to end its session with the SLU and, in its place, initiate a session between the SLU and some other application program.

d. ***Automatic Logon*** - Sessions provided via automatic logon are useful for a device like an automatic teller machine which should be kept in session with a controlling application, or for assuring that terminals are connected to a menu or security application program when powered on. Automatic logon provides a method for automatically establishing a session between an SLU and a designated controlling PLU whenever the SLU is enabled and below its session limit.

   If the SLU is single-session capable a determination of whether to reestablish the SLU's automatic logon session must be made whenever the SLU's current session terminates. The automatic logon session will be reestablished unless the SLU has a request queued for session limit, which will then be dequeued.

   If the SLU is multi-session capable automatic logon establishes a session between the SLU and the controlling PLU whenever such a session does not already exist.

e. ***Session Release Request*** - A PLU may initiate a session with an SLU and indicate in the request that, if the SLU is at its session limit, the current PLU should be notified that another PLU would like a session with the SLU. The PLU that sends the new session initiation request must indicate that the request may be queued. If the SLU is enabled and not at its session limit the session will be initiated. If the SLU is at its session limit the session request will be queued and the current PLU will be notified. The current PLU may terminate its session with the SLU or ignore the request.

   This function is normally used to improve the availability of printers shared by different application programs. The PLU receiving the release request will terminate its session, for example, if no output is queued for the (printer) SLU, or once the current listing is finished.

2. ***Request LU Status*** - This function allows an OLU (which has to be the PLU) node to request LU status information, in a session initiation request of type "search only," by setting the "LU status requested" indicator in the Cdinit GDS variable that it sends. The DLU node, if it supports this function, will include an "LU status control list (X'01')" in the Cdinit GDS variable that it sends in reply to provide status information of the DLU.

7.6.2 Initiating LUs and Initiate Types

With the extensions described, the number and kinds of session initiation procedures have grown significantly. The initiating LU (ILU) can be either the PLU, the SLU, or a third-party LU currently in session with the SLU. Furthermore, inquiries may be sent by an LU to retrieve information regardless of intended session role.

In order to support the different session initiation requests described in the previous section, additional initiate types have to be specified in Locate-Cdinit requests that are not necessary in base APPN architecture where only the PLU can request a session, and this session request failes when the SLU is not available. The initiate types that may be requested are:

- ***Search Only (S)***: The request origin is attempting to locate an LU but there is no implication that a session will be established. The origin is not requesting a session, but rather requesting information (for example, DLU available or not available, at its session limit or not at its session limit) that may allow it to establish a session.
- ***Initiate Only (I)***: The request origin is attempting to establish a session. If the DLU does session-limit management resource reservation (that means, reservation of an available session) is requested. If the resources are currently not available, then the session request should fail.

  This is the normal session initation type in base APPN.
- ***Initiate or Notify (I/N)***: The request origin is attempting to establish a session. If the DLU does session-limit management resource reservation is requested. If the requested LU is not enabled (that means, currently not willing to accept or send a BIND), then the request origin requests notification when it is enabled. Once the DLU becomes enabled, CP(DLU) remembers that CP(OLU) requested notification and sends a Locate-Notify(Resource Enabled) to CP(OLU), which then restarts the session initiation procedure.

  This type is used when an automatic logon is attempted.
- ***Initiate or Queue (I/Q)***: The request origin is attempting to establish a session. If the DLU does session-limit management, resource reservation is requested. If the resources are currently not available, then the request origin is willing to have the session initiation queued. Queueing conditions are queue for enabled, queue for session limit, or both.

  This type is used, for example, when a VTAM application program requests a session with an SLU that is a printer or display currently being powered off. The control unit IBM 3174 notifies the SSCP that the device is enabled for sessions, when it is powered on. Thus the SSCP can resume the session setup procedure at this time and dequeue the session request. Using this type of session initiation procedure the PLU has to request the session only once instead of periodically having to request a session.
- ***Queue Only (Q)***: The request origin is attempting to establish a session but the session initiation request is to be queued, since the request origin is not yet ready. The queueing conditions available to the request origin are queue for enabled, queue for session limit, or both.

The permitted combinations are shown in Table 6 below.

Table 6. Session Services Requests

	S	I	I/N	I/Q	Q
PLU Initiated	X	X (1)		X	X
SLU Initiated		X	X (2)	X	X
Third-Party Initiated				X	X

Note:

1. The only type of session initiation request supported in base APPN is a PLU originated Locate-Cdinit with "Initiate Only"
2. Used for automatic logon support.

7.6.3 Session Characteristics

In order to support SLU-initiated sessions, some of the information that is maintained by the SLU must be transferred in a Locate request or reply to the PLU node, so the PLU can properly establish a session with the SLU without having to predefine this information at the PLU. The information that needs to be transferred includes the BIND image, which contains session parameters, and device characteristics. This data will be copied into the BIND flowing from the PLU to the SLU.

BIND Image

In adding support to APPN for LU types other than 6.2, BIND images need to be passed from the SLU node to the PLU node. For a non-LU 6.2 session, the node owning the SLU will always provide a BIND image on either a Locate request (SLU-initiated session) or on a Locate reply (PLU-initiated session).

For an SLU-initiated session, the BIND image will always be provided by the SLU node regardless of the LU type. For a PLU-initiated session, the BIND image is always provided by the SLU node for session types other then LU 6.2.

In the subarea architecture, the SSCP of the SLU always sends the BIND image (via the SSCP of the PLU) to the PLU regardless of the session type, even for an LU 6.2 session.

The BIND image is carried in the "BIND Image (X'31')" control vector that is included in the Locate-Cdinit request or reply flowing from the CP(SLU) to the CP(PLU).

Device Characteristics

Device characteristics for non-SNA devices (such as BSC 3270 terminals) are required for application programs to interact properly with the device. Some fields in the device characteristics such as the terminal model number are used to derive a default display screen size, buffer size for the terminal, and so on, even though the preferable way is to use the information that can be supplied in the session parameters.

Other fields in the device characteristics, such as terminal type and the device addresses needed for the 3270 copy function, do not have appropriate counterparts in the BIND image.

The device characteristics will be included in the Locate request or reply from the SLU node to the PLU *when and only when* the BIND image is included.

7.6.4 Search Procedure Identification

The Fully Qualified Procedure Correlation ID (FQPCID) is used as a universal identifier for session related procedures. That is, an FQPCID is assigned at initiation and used on all flows (for example, Locate, BINDs, and UNBINDs) to identify the referenced session. It is also used for network management as the unique identifier for a session.

Depending on the type of session initiation, a number of subprocedures need to be done, details of which are beyond the scope of this document. To distinquish between the various subprocedures, the PCID Modifier control vector is used in the Locate request/reply. It contains the Procedure Resubmit Number (PRN), and the PCID Modifier List. This information is used by the PCID caching logic in APPN VTAM or subarea VTAM nodes (see page 155) to avoid duplicate searches for resources in, or accessible through, the subarea network.

Chapter 8. Network Management

Network management is the process of planning, organizing, monitoring, and controlling an APPN network. The architecture provided to assist in network management of SNA systems is called *management services* and is implemented as a set of functions and services designed to capture and use the information needed for effective management. For details about SNA management services see *Systems Network Architecture Management Services: Reference.*

8.1 Network Management Categories

Network management can be divided into the following categories:

- Configuration management
- Problem management
- Change management
- Performance and accounting management
- Operations management

Configuration Management

Configuration management is the control of information necessary to identify network resources. This identification includes information such as machine type and serial number (hardware), program number, release, and maintenance level (software or microcode), vendor and service organization, etc.

The configuration information may assist other network management categories, for example:

- Problem management may use the configuration data to determine the physical identity and location of a network resource, and the organization responsible for service.
- Change management may use the configuration data to schedule changes and analyze the effects of these changes.

Problem Management

Problem management is the process of managing a problem or potential problem from its detection through its final resolution. The term problem denotes an error condition resulting in an actual or potential loss of availability of a system resource that is visible to the end user. Problems may originate in hardware, software, or as a result of external causes such as user procedures.

The elements of problem management are:

- *Problem determination* is the element of problem management that detects the problem or impending problem and isolates the problem to the failing component.
- *Problem diagnosis* is the element of problem management that determines the exact cause of the problem and identifies the action required to resolve the problem.
- *Problem bypass and recovery* is the element of problem management that implements a partial or complete circumvention of the problem, while the

original problem is being diagnosed and a permanent solution is being worked on. For.example, when a leased telephone line fails, the bypass could be to use a switched connection until the leased line has been repaired.

- *Problem resolution* is the element of problem management that schedules and tests the repair action, and reports the problem as closed and the resource back in service.
- *Problem tracking and control* is the element of problem management that tracks the problem from problem determination until final resolution.

Change Management

Change management is the process of planning and controlling changes in a network. A change is defined as an addition, modification, or deletion of a network component. The component being either hardware (including microcode) or software. The software could be either system or application (vendor supplied or user written).

The elements of change management are:

- *Change planning* is the element of change management that encompasses all the activities required to take place before changes can be distributed and installed.
- *Change control* is the element of change management that distributes change files to entry points and installs them there. These changes may be either installed on a trial basis or in production.
- *Node activation* is the element of change management that reactivates altered entry points according to the change management plan.

Performance and Accounting Management

Performance and accounting management is the process of quantifying, measuring, reporting, and controlling the responsiveness, availability, utilization, and costs of network components.

Operations Management

Operations management provides the capability to control distributed network resources. Activating and deactivating resources, as well as setting resource clocks are all functions that are included in this category. In addition, a cancelation function has been defined that enables previously sent commands (including those executing at the target) to be terminated.

As an implementation option, operations management commands may be initiated as a result of system notification forwarding.

8.2 Management Services Roles

In terms of management services, SNA nodes fall into two basic categories, which help explain the node's role in the network. An ***entry point*** is an SNA node that provides distributed network management support. It may be a T2, T2.1, T4, or T5 node. It sends SNA-formatted network management data about itself and the resources it controls to the second major type of node, known as ***focal point***, for centralized processing. The entry point also receives and executes focal-point-initiated requests to manage and control its resources. The network management data, or management services (MS) data, can be *solicited* (requested by the focal point) or *unsolicited* MS information on events occurring within the entry point. An example of unsolicited information would be an *alert* sent by an entry point as notification of a link failure.

The concept of a focal point permits centralized management of a distributed network. A focal point is an entry point that provides centralized management and control for other entry points for one or more network management categories.

Focal points and entry points have relationships with each other for one or more categories of network management. Relationships between a focal point and entry points for problem management may or may not be the same as those established for change management, for example. A single communications systems or network may have multiple focal points.

The manner in which the focal points and entry points interact to accomplish the goal of network management is introduced in the following sections.

8.2.1 Focal Point Concepts

When a focal point to entry point relationship needs to be established to enable the sending of unsolicited data from the entry point, *MS capabilities* major vectors are exchanged between the focal and entry point. These exchanges establish the relationship between between the focal and entry point for a particular category of management services. The set of nodes having this relationship with a focal point is known as a ***sphere of control (SOC)***, of the focal point, and each of the individual nodes directly in the sphere of control is known as an ***SOC node***.

Note: For the change management category, the relationship between an entry and a focal point is not established by the exchange of MS capabilities.

APPN network nodes are SOC nodes both for themselves and for their served end nodes. The network node provides focal point notification messages to its served end nodes. This simplifies network administration and reduces network startup overhead, since focal points need be aware of only the network nodes in their SOC. However, end nodes may optionally provide the same level of support as network nodes, and be SOC nodes themselves, for example, when the services of a network server are not available.

It is possible for a focal point to have no SOC nodes, in which case it is said to have a *null sphere of control*.

A network may have multiple focal points. These focal points may have responsibility for the same or different categories of management service data. However, the spheres of control for multiple focal points may not overlap.

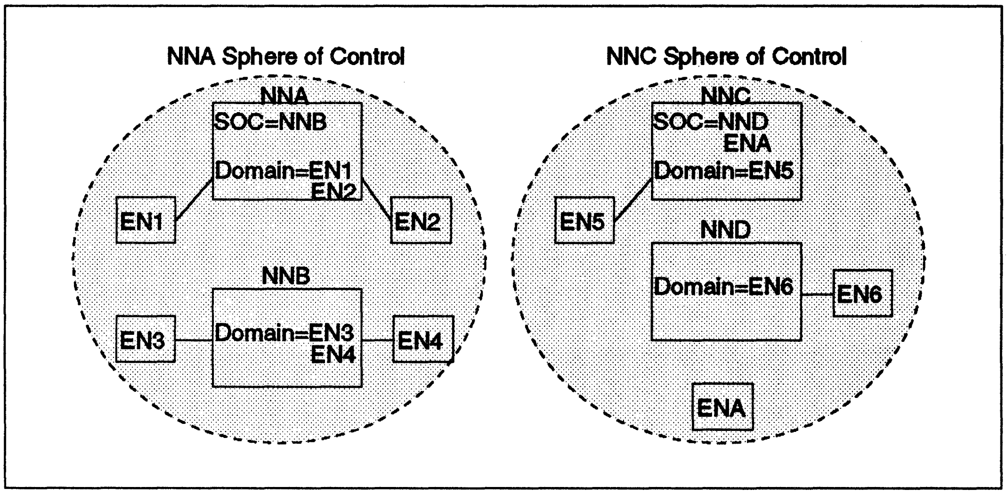

Figure 59. Sphere of Control (SOC)

Nodes can be assigned to a focal point's sphere of control, or can be acquired independently of network operator definition (for example, from the topology of the network). In the first case, the focal point is an *assigned* focal point; in the second case, the focal point is referred to as a *default* focal point.

Assigned Focal Point: An *explicitly* defined sphere of control is one that is defined at the focal point. The focal point is responsible for initiating and establishing this focal point (FP) to entry point (EP) relationship. The FP type for an explicit FP-EP relationship is called ***explicit primary***.

An *implicitly* defined sphere of control is one that is defined at the various entry points. It is not explicitly defined at the focal point. The entry points are responsible for initiating and establishing this focal point to entry point relationship. The FP type for an implicit FP-EP relationship is called ***implicit primary***.

Default Focal Point: A *default* focal point does not have a sphere of control explicitly assigned. Instead, it learns of the identity of APPN network nodes by examining the network topology. Network nodes will only accept the services from a default focal point, if no other focal point has been assigned. The default FP-EP relationship applies only to EP nodes that are network nodes. The FP type for a default FP-EP relationship is called ***default primary***.

Backup Focal Point: A *primary* focal point is the preferred destination for unsolicited data for a particular management services category. A *backup* focal point is one that provides management services for a node in the event that the services of the primary focal point are unavailable. The FP type for an implicit (backup) FP-EP relationship is called ***backup***.

Domain Focal Point: A network node will exchange MS capabilities messages with APPN end nodes it's serving to establish the domain FP-EP relationship. This relationship applies only to EP nodes that are APPN end nodes. The serving network node can have an explicit, implicit, default or host FP-EP relationship. The FP type for a domain FP-EP relationship is called ***domain***.

Host Focal Point: A **host** FP-EP relationship may be established if the EP node can establish an SSCP-PU session to a host node. No MS capabilities are exchanged.

Nested Focal Point: The relationship of focal-entry points may be nested; that means a focal point can be an SOC node in another focal point's sphere of control. The relationship between a *nesting* focal point (see for example NNE in Figure 60 on page 133) and a *nested* focal point (NNA or NNC) is established the same way as the relationship between any focal point and the nodes in its sphere of control. Notice, that since a focal point is the focal point for itself, it would never accept a request from a default focal point. Nesting focal points must always be assigned.

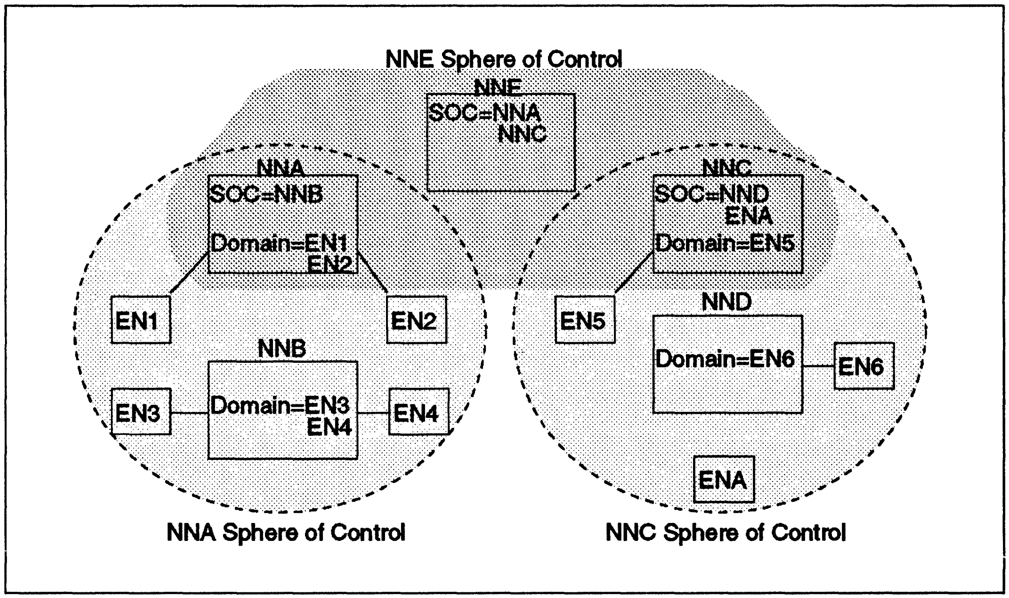

Figure 60. Nested Focal Points

The sphere of control relationship between focal points allows the "lower-level" focal point to forward network management information that it does not filter to its "higher-level" focal point. Nesting focal points cannot have overlapping spheres of control. This means that nodes in the sphere of control of the nested focal point are not in the sphere of control of the nesting level focal point.

8.3 Management Services Components

Management services distinguishes three components. These are:

- Local management services, hereafter referred to as LMS
- Control point management services, hereafter referred to as CPMS
- Physical unit management services, hereafter referred to as PUMS

  Note: The functions of PUMS as described in this chapter cover the management services of an SNA Type 2.0 node. Similar functions, although performed by a CP, are performed in a T2.1 node attached to a VTAM or NCP boundary function.

8.3.1 Local Management Services

LMS is the network management portion that is implemented in components and layers of a T2.1 node; see Figure 61 below and Figure 62 on page 135. The LMS function is implemented in control point components such as topology and routing services, directory services, and session services, but also in the SNA layers such as data link control and path control. The LMS in each component or layer gathers information and forwards this information to its CPMS. The interface used between the CPMS and LMS is implementation dependent. The LMS also receives and executes network management requests from the CPMS. The results of the network management requests are returned to the CPMS for further processing.

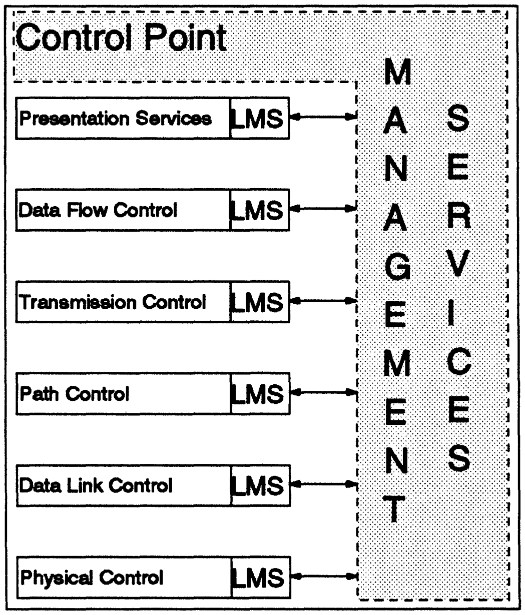

Figure 61. Local Management Services (LMS)

8.3.2 Introduction to Control Point Management Services (CPMS)

Control point management services (CPMS) is implemented in every T2.1 node. CPMS is a CP component of a T2.1 node that assists a network operator in the management and control of the node. The CPMS receives commands from the network operator or other CPMS instances, converts these commands in installation unique formats, and routes these to the appropriate LMS function for further processing. Information received from LMS, either solicited or unsolicited, is converted to standardized management services formats and routed to either the network operator or other CPMS instances.

In an APPN network, every APPN node contains CPMS. In an APPN end node, CPMS acts only as an entry point; in an APPN network node, CPMS can act as an entry point or a focal point.

CPMS: Overview

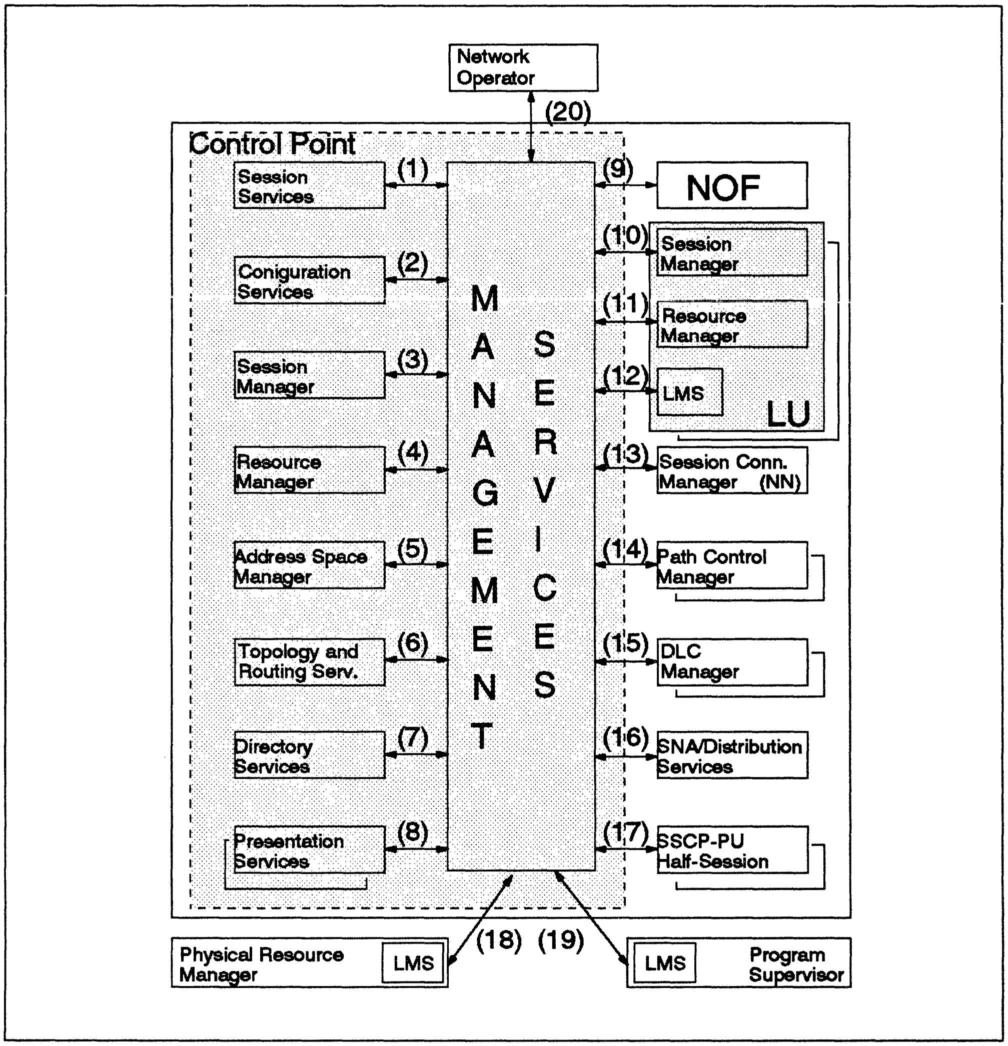

Figure 62. CP Management Services Boundaries with Other Components

Figure 62 illustrates the components with which CPMS in an APPN network node exchanges data. These are:

1. Session Services (SS)

- Upon request, SS provides information about the currently active sessions.
- Upon request, SS assigns FQPCIDs to CPMS. The FQPCID is required on the interface to directory services.
- SS in a network node provides unsolicited notification to CPMS when CP-CP sessions to served end nodes change status.
- SS in an end node provides unsolicited notification to CPMS when CP-CP sessions to the network node server become active or inactive.
- SS provides unsolicited notification of problems detected by the component.

2. Configuration Services (CS)

   - Upon request, CS provides configuration information about its domain.
   - CS provides unsolicited notification of problems detected by the component.

3. CP Session Manager

   - Upon request, the CP session manager provides information about the currently active LU 6.2 sessions for which the CP is a session endpoint.

4. CP Resource Manager

   - Upon request, the CP resource manager provides information about conversations on sessions for which the CP is a session endpoint.

5. Address Space Manager (ASM)

   - Upon request, ASM provides the names of all active LUs at this CP.
   - ASM provides unsolicited notification of problems detected by the component.

6. Topology and Routing Services (TRS)

   - TRS provides unsolicited notification of all nodes for which connectivity has just been required.
   - TRS provides unsolicited notification of problems detected by the component.

7. Directory Services (DS)

   - Upon request, DS provides the names of active LUs.
   - Upon request, DS locates network resources for CPMS.
   - DS provides unsolicited notification of problems detected by the component.

8. CP Presentation Services

   - CP presentation services provides the LU 6.2 protocol boundary used by MS service transaction programs that communicate over CP-CP sessions.

9. Node Operator Facility (NOF)

   - The NOF is the component that provides operator control of the local node, such as initialization of other CP components (including management services).

10. LU Session Manager

    - Upon request the LU session manager (LU 6.2 only) provides information about the currently active sessions for which the LU is a session endpoint.

11. LU Resource Manager

    - Upon request the LU resource manager (LU 6.2 only) provides information about conversations on sessions for which the LU is a session endpoint.

12. Logical Unit Local Management Services

    - The LU LMS provides unsolicited notification of problems detected by the LU.

13. Session Connector Manager (SCM)

    - Upon request SCM provides LU session data for sessions passing through the node.
    - SCM provides unsolicited notification of problems detected by the component.

14. Path Control Manager Local Management Services

    - The path control manager provides unsolicited notification of problems detected by the component.

15. Data Link Control Local Management Services

    - Upon request data link control (DLC) tests resources, for example links and modems, sets or retrieves management services parameters, and provides traces.
    - DLC provides unsolicited notification of errors and traffic statistics when problems associated with links and link stations are encountered, or when a counter threshold is exceeded.

16. SNA/Distribution Services

    - SNA/DS provides the capability to send and receive CP-MSU's, SNA/File Services (SNA/FS) agent objects, and SNA/FS files (bulk data) over an LU-LU session using LU 6.2 protocols.

    The change management category uses SNA/FS and SNA/DS for distribution of potentially large files, and issues the commands to manipulate them. For more details, see "SNA Distribution Services" on page 143.

17. SSCP-PU Half-Session

    - The SSCP-PU half-session provides communication with PUs within the CP's domain (only if the T2.1 node has implemented SSCP functions).

18. Physical Resource Manager Local Management Services

    - The physical resource manager LMS provides unsolicited notification of problems with the node physical resources, for example, tapes, disks, storage and microcode.

19. Program Supervisor Local Management Services (LMS)

    - On request, the program supervisor LMS alters software and microcode components.

20. Network Operator

    - The network operator requests management services from CPMS.
    - The network operator receives management services from PUMS in a Type 4 or Type 2.0 node, either unsolicited or upon request. This data may have been received directly from PUMS on an SSCP-PU session, or

received indirectly from PUMS via a controlling CPMS on a CP-CP session.

- The network operator receives management services data from CPMS in an APPN network node either unsolicited or upon request. This data may have been received from CPMS in an SSCP-PU or CP-CP session, or received indirectly via a controlling CPMS or a serving CPMS on a CP-CP session.

Note: The term network operator actually refers to the programming which supports an operator, either human or programmed.

8.3.3 Introduction to Physical Unit Management Services (PUMS)

Physical unit management services (PUMS) is the component of an SNA *physical unit* (PU) responsible for providing general management services to the node and its associated resources. The functions of PUMS as described in this section cover the management services of an SNA Type 2.0 node. Similar functions, although performed by the CP, are present in T2.1 nodes that attach to a VTAM or NCP boundary function. In a T2.1 node the CP acts as a PU for the purpose of management services.

PUMS requires an SSCP-PU session with its controlling *System Services Control Point* (SSCP) to forward network management data from the SSCP or receive network management requests from the SSCP. The management services commands received from the SSCP are converted to installation unique formats, and forwarded to the LMS for further processing. Information received from the LMS, solicited or unsolicited, is converted to a network management vector transport (NMVT) and sent across the SSCP-PU session to the SSCP.

PUMS: Overview

Figure 63 gives an overview of the PU management services boundaries with other components within an SNA node.

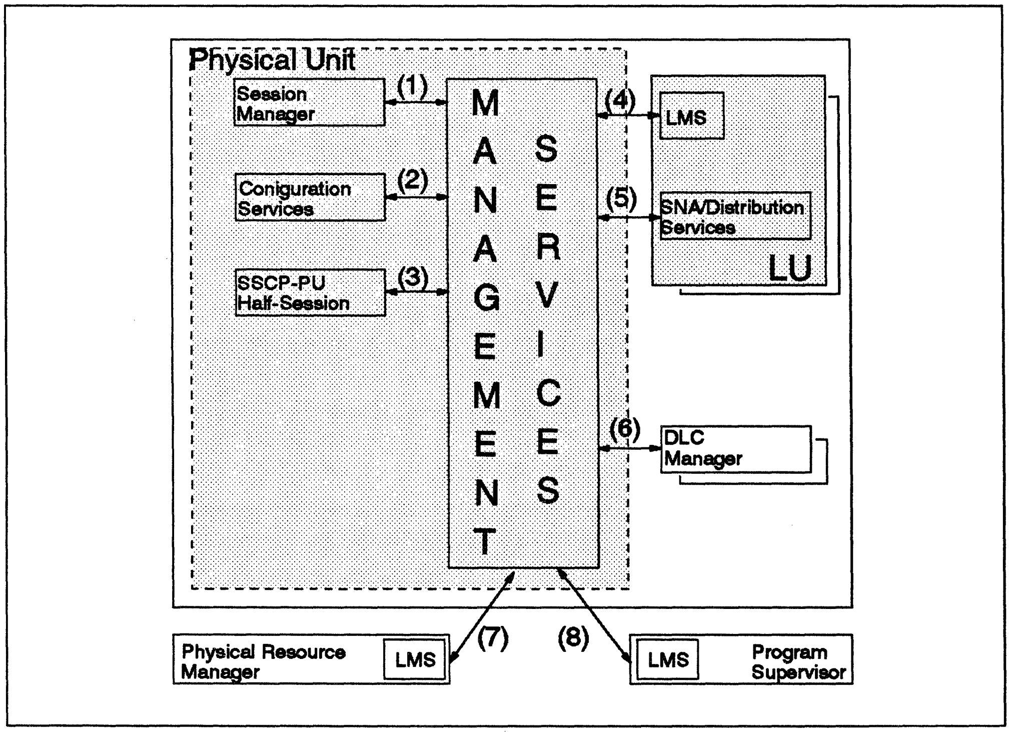

Figure 63. PU Management Services Boundaries with Other Components

1. PU Session Manager

    - Upon request, the PU session manager provides information about the currently active sessions managed by the PU.

2. PU Configuration Services

    - Upon request, PU configuration services provides information that uniquely identifies the hardware and the software of the node and provides a list of active LUs.
    - PU configuration services provides unsolicited information when the SSCP-PU session becomes active.

3. SSCP-PU Half-Session

    - The SSCP-PU half-session provides communication (over SSCP-PU sessions) with a resource's controlling CPMS.

4. LU Local Management Services

    - Upon request, the LU LMS sets response-time measurement parameters, and provides response-time data.
    - The LU LMS provides unsolicited notification of problems within the LU and unsolicited response-time data.

5. SNA/Distribution Services

    - SNA/DS provides the capability to send and receive CP-MSUs, SNA/File Services (SNA/FS) agent objects, and SNA/FS files (bulk data) over an LU-LU session using the LU 6.2 protocol.

      The change management category uses SNA/FS and SNA/DS for distribution of potentially large files, and issues the commands to manipulate them. For more details, see "SNA Distribution Services" on page 143.

6. Data Link Control Local Management Services

    - The DLC manager LMS provides unsolicited notification of problems with links.

7. Physical Resource Manager Local Management Services

    - The physical resource manager LMS provides unsolicited notification of problems with the node physical resources, for example, tapes, disks, storage and microcode.

8. Program Supervisor Local Management Services

    - Upon request, the program supervisor LMS alters software and microcode components.

8.4 Transport of Management Services Data

The SNA network management services function is provided by the combined functions of the control point management services, physical unit management services, and local management services components. The following sections describe the options available to management services application programs for transporting management services data. A distinction can be made between:

- Transport of Management Services Data on the SSCP-PU Session

  T2.1 node may optionally have established an SSCP-PU session. The SSCP-PU session is used for transferring management services data between a control point and a physical unit. Note that CPMS in a T2.1 node that uses the SSCP-PU session for its management services communication with the SSCP performs the same function as PUMS in a Type 2.0 node.

- SNA Distribution Services

  The change management category uses SNA/File Services and SNA/Distribution Services for distribution of potentially large files, requests to manipulate them, and reports to track the distribution and installation. These employ LU-LU sessions.

- Multiple Domain Support

  The third type of management services transport, defined for the transfer of MS data between control points, is called ***MULTIPLE_DOMAIN_SUPPORT (MDS)***. This transport provides the transaction routing between management services application programs via CP-CP or LU-LU sessions.

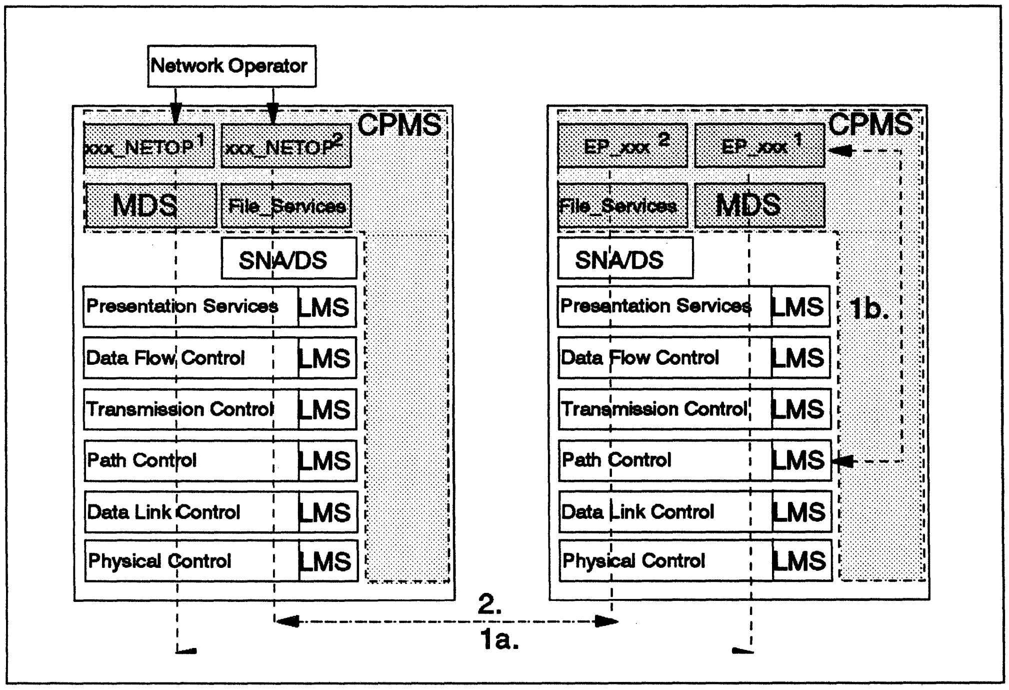

Figure 64. Communication between CPMS Instances

Figure 64 gives an example of how two CPMS instances may communicate. Focal point MS functions xxx_NETOP1 and xxx_NETOP2 communicate with entry point MS functions EP_xxx1 and EP_xxx2, respectively. Communication is possible via MDS (see 1a.) or SNA/DS (see 2.). Within the entry point an LMS instance (for example path control, see 1b.) communicates with EP_xxx1.

Communication of focal point function xxx_NETOP with an entry point LMS instance is always via EP_xxx.

8.4.1 Management Services Formats

Two ways exist to encode management services formats. A management services unit (MSU) is a management services encoding that is formatted according to a major vector, subvector, subfield scheme (see Figure 65). The other way of encoding a management services format uses a non-MSU scheme, and therefore does not use the major vector scheme.

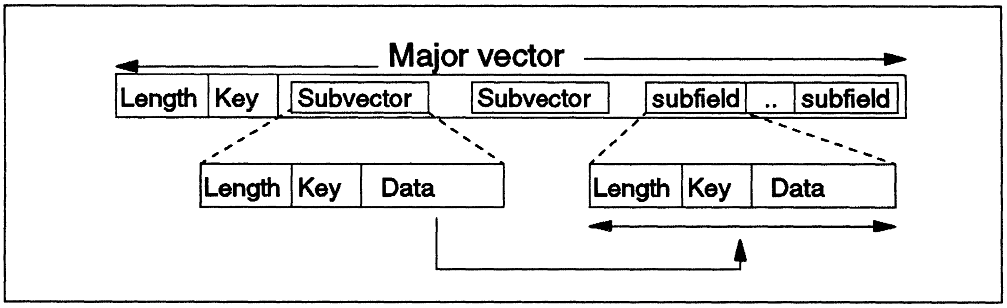

Figure 65. Overview of a Management Services Major Vector

The MSU that flows on an SSCP-PU session is called a ***network management vector transport (NMVT)***. In addition some management services request units may flow on an SSCP-PU session that does not exhibit the MSU encoding scheme. They are termed non-MSU management services request units. For details see *Systems Network Architecture Management Services: Reference.*

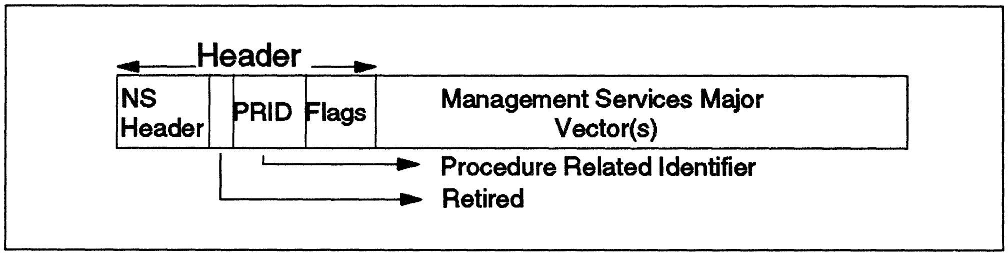
Header
NS Header
PRID
Flags
Management Services Major Vector(s)
Procedure Related Identifier
Retired

Figure 66. The NMVT Message Unit Format

The MSUs transported on CP-CP and LU-LU sessions between CP instances are general data stream (GDS) variables, which adhere to the encoding rules for GDS variables (for details see *Systems Network Architecture Formats*).

The *multiple-domain support message unit* (MDS-MU) GDS variable is used for the transport of non-bulk MS data in APPN networks. The MDS-MU has two components: the MDS header and the MS application program data; see Figure 67 on page 142. The MDS header consists of MDS routing information (origin and destination names) and a correlation variable. The latter allows MDS and MS application programs to correctly correlate MDS-MUs.

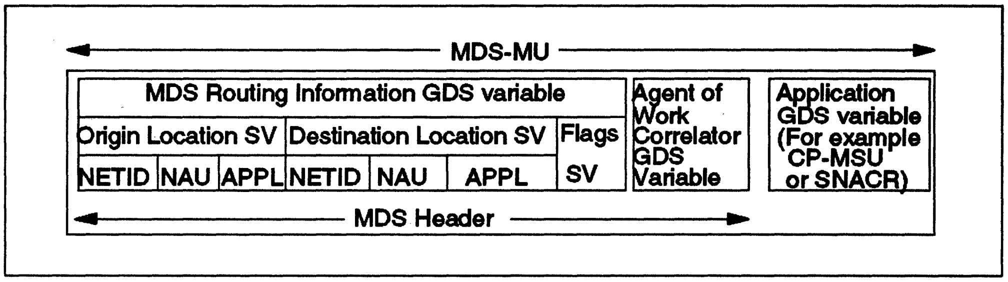
MDS-MU
MDS Routing Information GDS variable
Origin Location SV
Destination Location SV
Flags SV
NETID
NAU
APPL
NETID
NAU
APPL
Agent of Work Correlator GDS Variable
Application GDS variable (For example CP-MSU or SNACR)
MDS Header

Figure 67. The MDS Message Unit Format.

Legend

APPL	=	Application Program Name
NAU	=	Network Accessible Unit (Name)
SV	=	Subvector
GDS	=	General Data Stream
CP-MSU	=	Control Point Management Services Unit
SNACR	=	SNA Condition Report

The *control point management services unit* (CP-MSU) is a GDS variable that can be used for transmitting either non-bulk data in the MDS-MU variable or transmitting bulk data using SNA/DS.

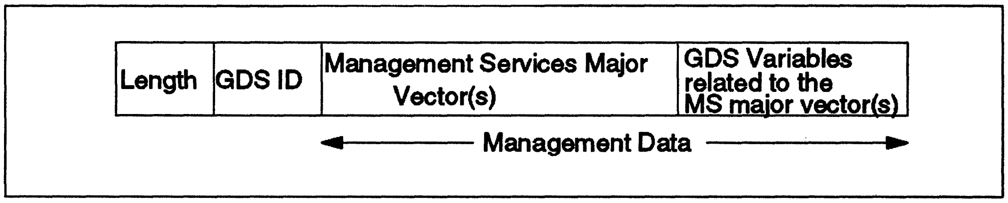
Length
GDS ID
Management Services Major Vector(s)
GDS Variables related to the MS major vector(s)
Management Data

Figure 68. The CP-MSU GDS Variable Format

The following list shows when management services information is exchanged:

- A network operator communicating with an instance of CPMS.
- Communication between instances of CPMS in different nodes via MDS on CP-CP or LU-LU sessions using LU 6.2 protocols.
- CPMS communicating with PUMS on a control point to physical unit (SSCP-PU) session.
- Both CPMS and PUMS communicating directly with an instance of LMS.
- Bulk data being transported between CPMS and PUMS, or between instances of CPMS, using SNA/DS protocols on LU-LU sessions.

8.4.2 Transport of Management Services Data on the SSCP-PU Session

The primary path for transport of SNA management services (SNA/MS) data between CPMS and PUMS is the SSCP-PU session. SNA/MS plays no role in the establishment of this session. Since the session is established when a PU is activated, it is already present when PUMS comes up. From the point of view of PUMS, the SSCP-PU session is simply a pipe through which management services requests and data can be exchanged with the PU's controlling SSCP.

8.4.3 MULTIPLE_DOMAIN_SUPPORT

The service provided by CPMS that provides for the routing of data between MS application programs over CP-CP and LU-LU sessions is called multiple-domain support (MDS). MDS consists of a router and multiple service transaction programs. The *MDS router* routes message units between MS application programs residing in the same node and uses the MDS service transaction programs (STPs) to route message units between MS application programs residing in different nodes. Example A in Figure 69 depicts the sessions used by MDS for default routing in an APPN network. MDS default routing uses LU-LU sessions (mode SNASVCMG) between network nodes (including focal points) and CP-CP sessions (mode CPSVCMG) between network node servers and their client end nodes. An LU-LU session directly from a focal point to an end node may also be used as shown in example B below.

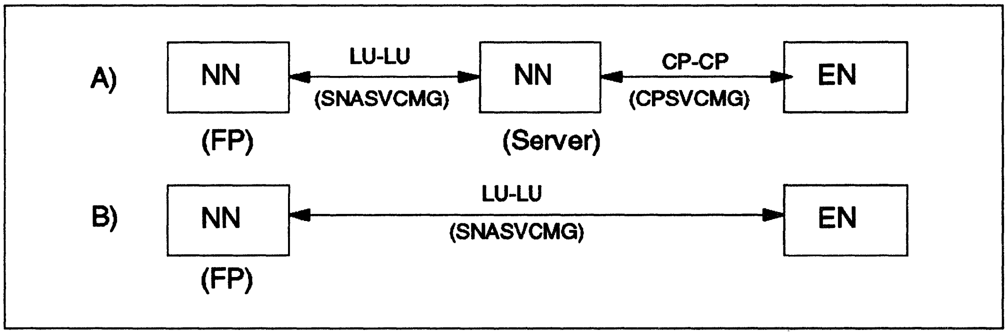

Figure 69. Sessions Used by MDS in an APPN Network

Notice that to exchange messages between management service transaction programs in the focal point and the end node, the data may flow through the network node server as shown in example A.

8.4.4 SNA Distribution Services

SNA management services uses SNA/DS for the transport of requests, reports, and bulk data. An LU-LU session directly from a focal point to an end node is used to exchange data as shown in Figure 70.

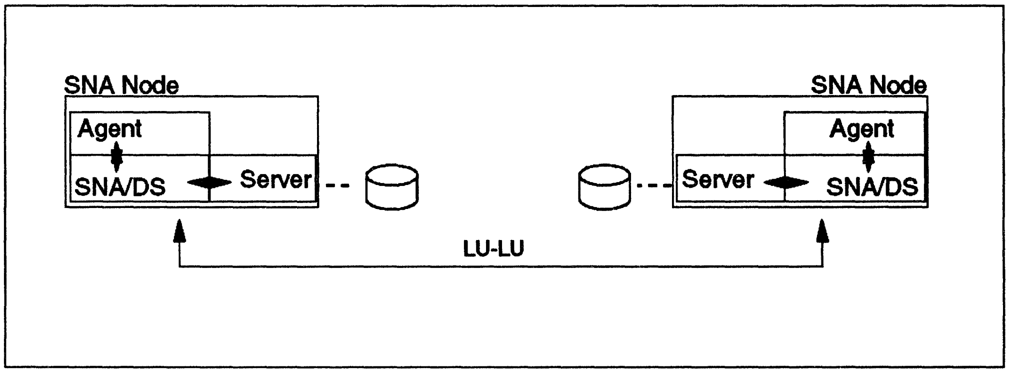

Figure 70. SNA Distribution Services

This figure shows an application transaction program, or *agent* in SNA/DS terminology, which uses SNA/DS as a transport mechanism. For a detailed description of SNA/Distribution Services refer to *SNA Distribution Services Reference*. The *server*, also provided by SNA/DS, is invoked to handle staging and destaging of (typically large) files to the system storage facilities.

The building and parsing of the object handled by the server (the server object) for network management is not different from that for other SNA/DS agents. For this reason, architecture has been developed for the server, called *SNA/File Services* (SNA/FS). For a detailed description of SNA/File Services, refer to *SNA File Services Reference*.

8.5 Network Management Functions

Network management architecture addresses the management services for different SNA nodes. The differences in SNA nodes are not only the SNA node types, for example T2.1, T4 (subarea NCP), T5 (subarea VTAM), but also the difference in functions and capabilities implemented for each SNA node type. For example, a T2.1 SNA node may be either a LEN end node, end node, or a network node. Therefore, the network management architecture has split the management services into *function sets*. A management services function set is a collection of services that together perform an overall management services function. Each MS function has a mandatory or *base subset* that all implementations of that function set must support. The rest of the function set is composed of *optional subsets*. Implementations of that function set can choose to support some or all of the optional subset, depending on their *role* requirements. Defined MS roles are:

- CPMS in an APPN end node
- CPMS in an APPN network node
- PUMS in a Type 2.0 node
- CPMS in a node implementing an SSCP (for example AS/400)
- PUMS in a Type 4 node
- PUMS in a Type 5 node (for example VTAM)

This document discusses only the first three MS roles.

Note: In order to clearly distinguish names of MS function sets when they appear in this publication, they are generally given descriptive multiple-word names, capitalized, and connected with underscore characters. For example, the function set that describes how PUMS sends data over the SSCP-PU session is denoted by the name "SEND_DATA_SSCP_PU."

8.5.1 Electives

Certain functions can be implemented in more than one way. If the effect can be observed at the MS protocol boundary, then that choice is called an *elective*. Electives are not optional functions, but are choices that regulate how or when a function is provided. If another component can observe the effect of an elective choice, then that component must also be able to support all of the possible effects of the elective choices. Product implementations make elective choices for performance or development-cost reasons.

8.5.2 Function Sets for CPMS and PUMS

Figure 71 and Figure 72 depict the base and the optional MS function sets for PUMS in a Type 2.0 node and CPMS in an APPN node, respectively. The figures also show how the various function sets relate, such that each function set requires the function sets in its lower layers.

Note: The notation "EP_xxx" is used to represent entry point function sets (for example, EP_ALERT).

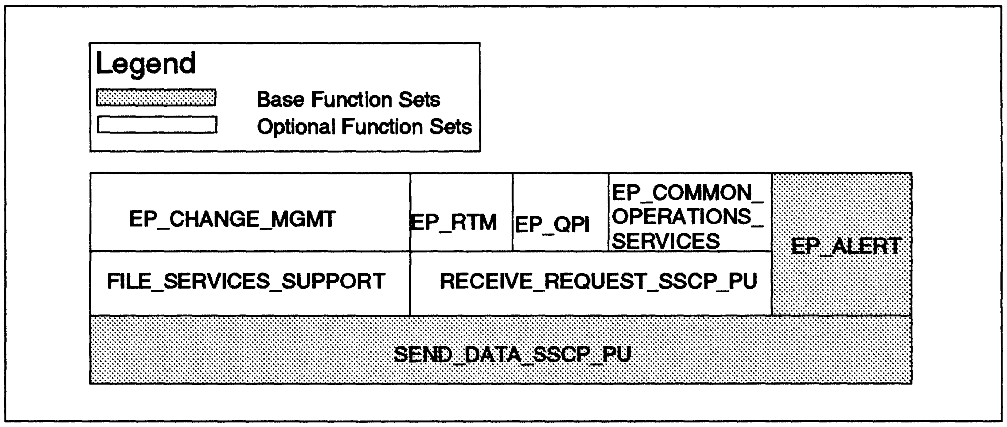

Figure 71. PUMS Function Sets in a Type 2.0 Node.

Note: *CPMS in a T2.1 node attached to NCP's or VTAM's boundary function performs the same function as PUMS in a Type 2.0 node.*

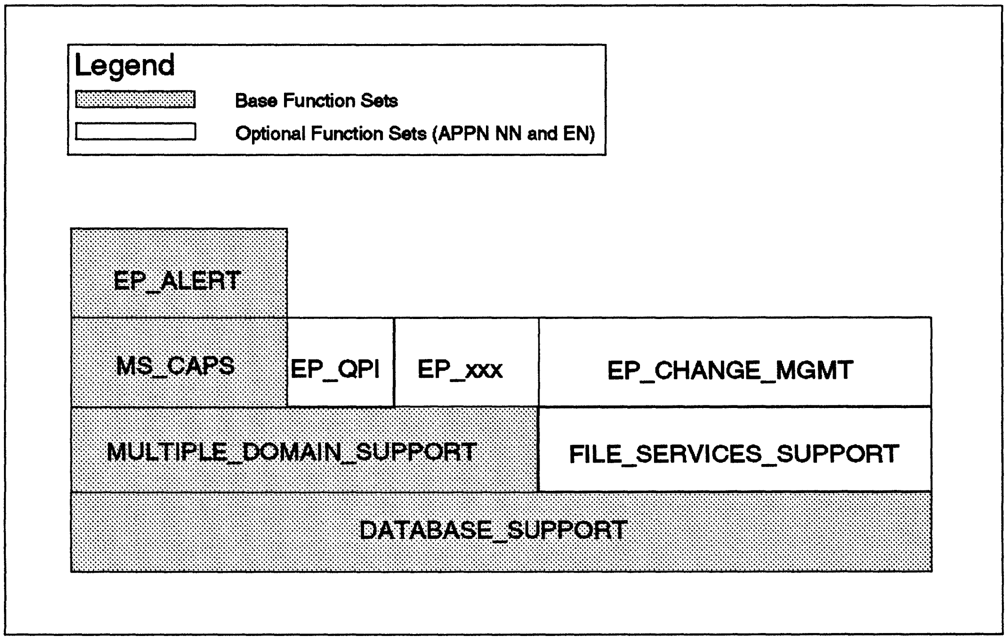

Figure 72. CPMS Function Sets in APPN Network and End Nodes.

Note: *EP_xxx represents any of the following specialized entry point function sets:*

- *EP_OPERATIONS_MGMT*
- *EP_COMMON_OPERATIONS_SERVICES*

8.5.3 Function Sets: Description

Table 7 gives an overview of the MS function sets relevant for CPMS on APPN nodes or PUMS on a Type 2.0 node. It mentions the general management function sets and the specialized functions sets for entry points.

Table 7. General and Specialized Function Sets

General Function Set	Specialized Function Set for Entry Points
RECEIVE_REQUEST_SSCP_PU	EP_ALERT
SEND_REQUEST_SSCP_PU	EP_RTM
MULTIPLE_DOMAIN_SUPPORT	EP_QPI
RECEIVE_DATA_SSCP_PU	EP_CHANGE_MANAGEMENT
SEND_DATA_SSCP_PU	EP_COMMON_OPERATIONS_SERVICES
FILE_SERVICES_SUPPORT	EP_OPERATIONS_MGMT
DATABASE_SUPPORT	
MS_CAPS	

The following section gives a short description of each of the generalized function sets and specialized function sets for entry points mentioned in Table 7.

RECEIVE_REQUEST_SSCP_PU: It provides the support to receive network management vector transport RUs and pass the vector to the appropriate function group set.

MULTIPLE_DOMAIN_SUPPORT: It provides the capability to send management services requests and data between management functions in the same or different nodes. The optional subsets are:

- End Node Support (Optional Subset 1)

  The end node support is applicable to network nodes only. It consists of the MDS router functions for the entry point.

- Network Node Support (Optional Subset 2)

  The network node support is applicable to network nodes only. It consists of the MDS router functions for network nodes.

- High Performance Option (Optional Subset 3)

  The high performance option is applicable to network nodes only. It provides the ability for management services applications to use persistent conversations over dedicated sessions, thus improving the performance for management services applications with higher transaction rates. The base set uses short conversations over shared sessions to transport the management services units. In addition, it uses LU 6.2 confirmations for reliable delivery of the data. The overhead introduced this way is containable if the transaction rate remains low.

- Transport Confirmation Option (Optional Subset 4)

  The transport confirmation option is applicable to network nodes only. It provides the ability for management services application programs to omit the LU 6.2 confirmations for each management services unit, thus increasing the session throughput.

SEND_DATA_SSCP_PU: It provides the support for sending network management vector transport RUs across an SSCP-PU session to a subarea CPMS.

FILE_SERVICES: It provides the support to route management services requests and bulk data between nodes using SNA distribution services.

- Network Operator Support (Optional Subset 1)

  Network operator support is applicable to both end nodes and network nodes. It provides the support to interact with the node operator at the node, to receive request verbs, and return reply verbs.

- File Deletion Support (Optional Subset 2)

  File deletion support is only applicable to network nodes. It provides the support to interact with the network operator to delete files.

DATABASE_SUPPORT: It provides the support to manage the management services historical database for the node. The database contains data passed to it in network management vector transport format.

MS_CAPS (MS capabilities): It provides the support for getting information from a focal point and to route this information to local application programs on a node. An APPN end node can either communicate directly with its focal point, using an LU-LU session, or, indirectly through its network node server.

- Have a Backup or Implicit FP (Optional Subset 1)

  Support for backup or implicit focal point is applicable to end nodes and network nodes. It provides the support for a node to have a backup focal point or an implicit focal point.

- Be a Sphere_of_Control End Node (Optional Subset 2)

  Support for being a sphere of control node applicable to end nodes. It provides the support for an entry point to directly communicate with its focal point. Normally, an entry point communicates indirectly with its focal point through its network node server.

- Base Network Node Support (Optional Subset 3)

  Support for base network node support is required for network nodes. It provides the support necessary for a network node to be an SOC node, and enables the node to send and receive MS capabilities from the entry point side of the relationship.

- Have a Subarea Focal Point (Optional Subset 4)

  Support for subarea focal point is applicable to network nodes only. It provides the ability for the network node to act as a pseudo focal point for its domain on behalf of a subarea focal point. It will forward the data it receives on an SSCP-PU session to a subarea focal point.

EP_ALERT: It is responsible for:

Detecting an alert condition for any resource controlled by its node

Building the alert major vector

Passing the vector to the Multiple Domain Support for further processing by a focal point.

The following optional subsets are available for EP_ALERT:

- Problem Diagnosis Data (Optional Subset 1)

  Support for problem diagnosis data means that the alert vector contains a problem diagnosis section. The problem diagnosis section may contain, for example, a malfunction code.

- Delayed Alert (Optional Subset 2)

  This function is not supported for T2.1 nodes. Support for delayed alert means that an entry point can delay the alerts when the session with its focal point is lost. As soon as the session with the focal point is reestablished the alerts held will be forwarded to the focal point.

- Held Alert for PUMS (Optional Subset 3)

  Support for held alert for PUMS means that the entry point is capable of holding alerts until the session with the PUMS is reestablished.

- Operator-Initiated Alert (Optional Subset 4)

  Operator-initiated alerts provide a mechanism for the network operator to initiate the reporting of an alert condition. Normally, these are conditions that cannot be detected by the control point.

- Qualified Message Data (Optional Subset 5)

  Support for the qualified message data provides the ability to generate alerts using indexed text messages and qualifier data. The receiver of the alert creates the alert message by using the index and qualifier data to reconstruct the message from its local message table. For example, if the national language differs between focal point and entry point, this subset allows the focal point and entry point to generate the alert message in their own national language.

- Text Message (Optional Subset 6)

  Support for text message provides the capability to include, in the alert, a character string of 236 characters.

- LAN Alert (Optional Subset 7)

  Support for LAN alert provides the capability to send alerts for errors detected at the MAC layer of a token-ring, Ethernet, or bridged LAN.

- SDLC/LAN LLC Alert (Optional Subset 8)

  Support for SDLC/LAN LLC alerts provides the capability to send alerts for problems detected on SDLC and LAN logical link level control.

- X.21 Alert (Optional Subset 9)

  Support for X.21 alerts provides the capability to send alerts for problems detected on X.21 link connections. This will also include the alerts for X.21 short hold mode.

- Hybrid Alert (Optional Subset 10)

  Support for hybrid alert is not available for T2.1 nodes. It provides support for nodes to send alerts in a form that can be both processed by the current version of CPMS as well as a backlevel version.

- X.25 Alert (Optional Subset 11)

  Support for X.25 alerts provides the capability to send alerts for problems detected on X.25 connections.

- Held Alert for CPMS (Optional Subset 11)

  Support for held alerts for CPMS provides the capability to hold alerts when the focal point is not available, and to send the alerts, with an indication that the alert was held, when the focal point is available again.

EP_RTM: It provides the capability to measure and monitor end user response times.

- Local Display (Optional Subset 1)

  Support for local display provides the capability to display the measurements at the node implementing this function set. The focal point can send commands to enable or disable the local display.

EP_QPI: It provides the capability to physically identify the SNA node and attached devices upon request.

EP_CHANGE_MGMT: It provides the capability to respond to change control and activation requests from a change management focal point or local operator interface.

- Production-Only Activation (Optional Subset 1)

  Support for production-only activation provides the capability to respond to requests from the focal point for activation of only those versions of components marked in-production.

EP_COMMON_OPERATIONS_SERVICES: It provides the capability to support communication between network operators and served network management applications.

EP_OPERATIONS_MGMT: It provides the capability to receive operations management commands from network operators and replies or reports from second-level application programs. Unsolicited messages may also be received from application programs served by EP_OPERATIONS_MGMT and sent to the operations management focal point.

Chapter 9. Border Node

This chapter describes an optional APPN function supported on network nodes, known as the ***border node (BN)*** function, which introduces the concept of *subnetworking*. An APPN subnetwork can be seen as a collection of interconnected nodes and communication links with some logical association. Two or more subnets form a *composite* network.

According to the base APPN architecture, network nodes within an APPN network must share the same NETID (as opposed to end nodes, which may have a NETID different from their adjacent node's NETID)

The main reason for the introduction of the border node concept is that it permits APPN networks having different NETIDs to connect, thus allowing session setup across subnetwork boundaries. But the concept of subnetworking is not limited to networks with different NETIDs; it also allows partitioning of an APPN network with the same NETID. This process is also known as *clustering*, and the subnetworks are called *clusters*.

Subnetworks (also called *subnets*) within the context of the border node function are *topology subnetworks*. Topology database updates (TDUs), indicating topology changes and flowing through subnets, are prevented from crossing subnet boundaries. The subnetwork in which a resource resides is its *native subnetwork* (or native subnet) and the subnets in its composite network in which it does not reside as its *nonnative subnetworks*.

One immediate benefit from subnetting is that the number of TDUs flowing through topology subnetworks will be lower, thus reducing the storage requirements for the network topology database in network nodes in each of the subnets. This reduces network flows and allows network nodes with limited resources to participate in APPN networking.

The following chapters describe the two stages in which the border node concept is being introduced. The first stage, Border Node R1, is already supported on AS/400. The second stage, Border Node R2, has been announced to be implemented in a future release of VTAM.

9.1 Border Node R1 (BN R1)

This section describes the first stage of the APPN border node function. As explained, a border node is an APPN network node that includes the APPN BN function. To establish distinct subnets, a BN disables the passing of topology information between native and nonnative network nodes. Figure 73 shows the basic form of two subnets being interconnected by a Release 1 border node. Topology separation is a result of the fact that the border node portrays itself as a network node to native partner network nodes and as an EN to nonnative network nodes. Therefore, the BN receives TDU messages only from network nodes within the native subnet.

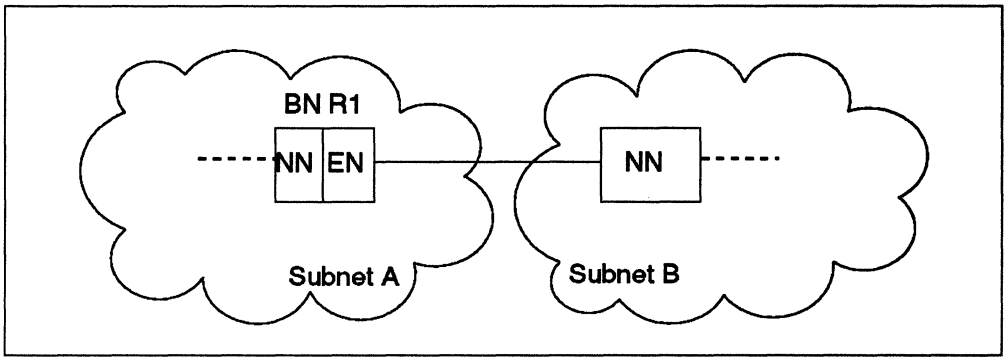

Figure 73. Border Node Connection

In the nonnative subnetwork, the BN must link to either an NN or a BN. When two BNs interconnect (see Figure 74), because of the asymmetrical nature of a BN, it will be decided during XID exchange, which border node presents the NN and which the EN image. The BN with the lower (network-qualified) CP name elects to present an EN image, and the one with the higher CP name an NN image.

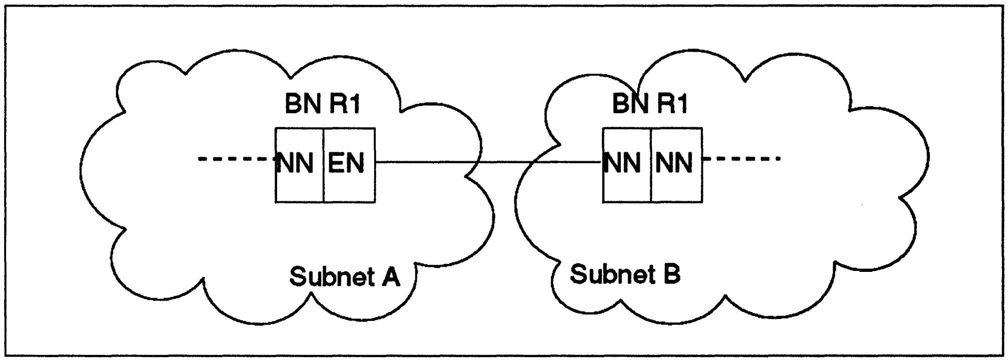

Figure 74. BN to BN Connection

Figure 75 on page 153 depicts two subnets connected by multiple BNs in parallel. This configuration provides enhanced internetwork availability and bandwidth. The figure illustrates parallel connections where multiple BNs reside in in the same subnet, or where one BN exists in each subnet. Although a BN can be connected to both network nodes and end nodes in nonnative subnets, only connections to network nodes will provide a connection between the two networks through which sessions can be established between LUs residing on any nodes in either subnet.

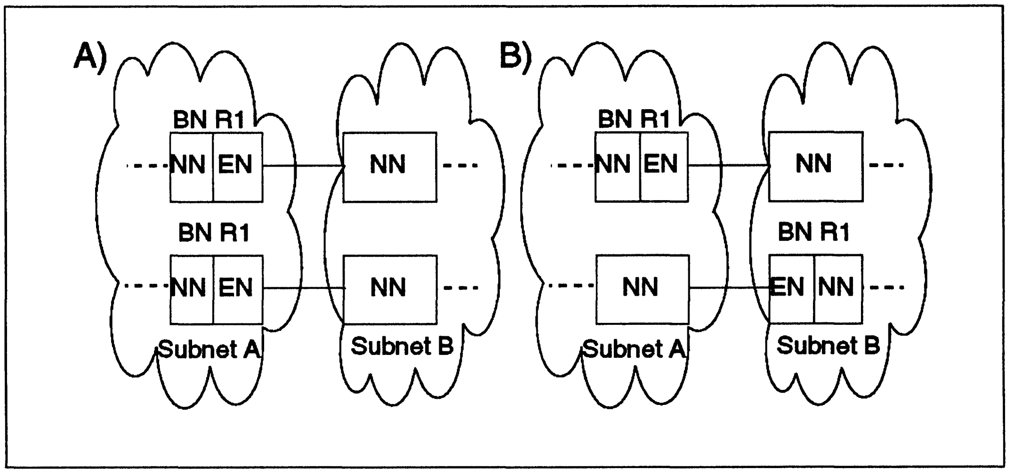

Figure 75. Parallel BN Connections

Figure 76 illustrates two examples of how a BN connects several subnets. BN in subnet A connects to NNs in subnet E and subnet C. The NNs in both subnets will serve as the network node server of the BN, NNS(BN), for their respective subnets. BN in subnet D depicts the sitiation where the BN portrays itself as an APPN end node to an NN in one subnet, and as an APPN network node to a BN (acting as an EN) in another subnet. Because of a limitation of BN R1 described below, no sessions can be established between LUs residing in subnets A and D and between LUs residing in subnets C and E.

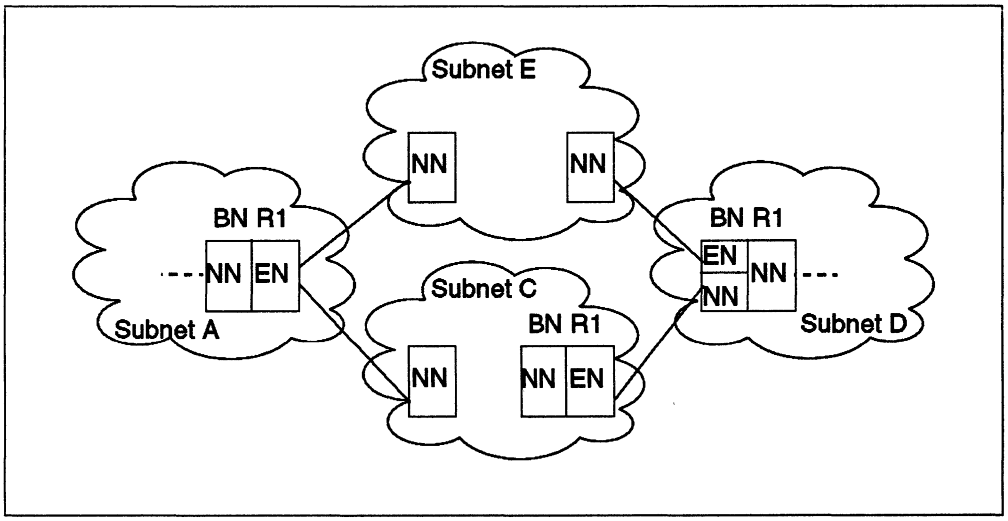

Figure 76. BN to Multiple Subnet Connections

9.1.1 Subnet Searches

The border node concept allows session setup between LUs residing in different subnetworks. When implementing BN R1, cross-network sessions are only possible between session partners that reside in adjacent subnetworks.

According to the APPN architecture, the destination LU (DLU) must be located first, with a Locate search request, before a session BIND can flow between the session partners. Locate search requests will be sent over an intersubnetwork link only if the NETID of the destination resource matches that of the node receiving the Locate request over the link.

Note: Because of this limitation, DLUs residing in the native subnet must have the same NETID as the BN, and DLUs residing in the nonnative subnet must have the NETID of the NN connected to the BN.

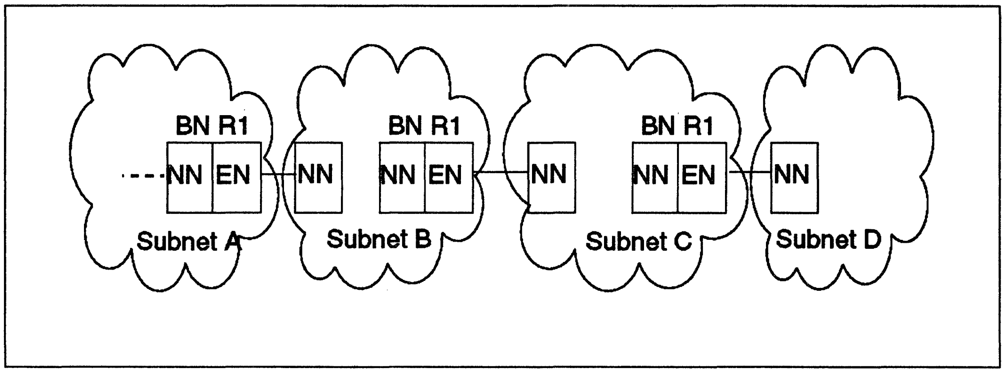

Figure 77. Cascaded Network Support (BN R1).

LUs in subnets C and D are not able to establish sessions with LUs in subnet A.

To allow a border node to control searches across subnet boundaries, border node architecture defines two functions during the Locate search flows in addition to base APPN architecture:

- The initiator of a search procedure can indicate in the Locate request that the search must not be propagated across a subnetwork border. A border node will then not accept a Locate request received over an intersubnetwork link and does itself not forward a Locate request across an intersubnetwork link, if the Locate request indicates that the search should be limited to the native subnetwork of the search origin.
- A border node, to be specific its "EN side," will add the information to a Locate request, before forwarding it across an intersubnetwork link, that this request has already crossed a subnetwork border. If a BN (again, its "EN side") finds this information in a Locate request received across an intersubnetwork link, it sends back a negative reply and does not propagate this search request into its native subnetwork. This has the effect that sessions across subnetwork boundaries are (with border node architecture R1) limited to sessions between LUs residing in adjacent subnetworks.

9.1.2 Parallel BN Connections and Duplicate Search Collisions

Because border nodes route search requests into adjacent subnets, it is possible that multiple Locate search requests enter the destination subnet, for example when a broadcast search occurs in the origin subnet and multiple BNs provide connections between the origin and destination subnet.

The impact is deemed to be minimal, because once a resource has been located, the network node server of the originating LU will cache information associated with a specific BN and subsequent searches will not collide.

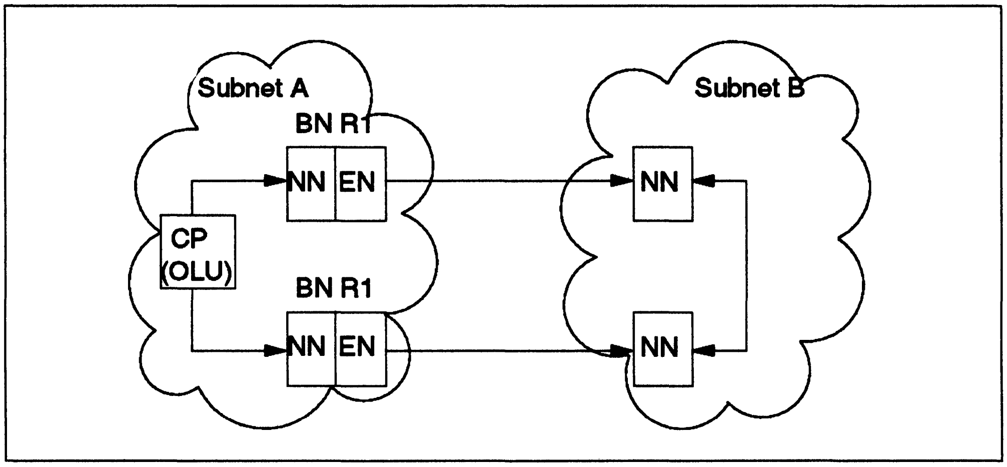

Figure 78. Multiple Broadcasts in the Nonnative Subnet

To minimize the impact of this collision problem, the concept of **PCID caching** has been developed. A BN that implements PCID caching does two things:

1. ***PCID Caching***

   A border node caches the FQPCIDs of all those searches received from adjacent network nodes in its native subnetwork that have already crossed a subnetwork boundary (according to the information in the Locate request).

2. ***PCID Checking***

   A border node receiving a Locate request across an intersubnetwork connection rejects this request if the FQPCID is contained within its PCID cache, indicating that this request has been forwarded into its native subnetwork already through an alternate border node.

The benefit of PCID caching is shown in Figure 78. A broadcast search is started in subnet A from CP(OLU). The search request will be sent to each of the border nodes and forwarded to the nonnative subnet. The border nodes may, therefore, receive the search twice, as a broadcast search in their native subnetwork and as a domain search from their NNS(BN) in the nonnative subnetwork. If the nonnative domain request reaches a border node first and the BN caches the PCID, the second request can be prevented from entering the nonnative network.

9.1.3 Route Selection

Internetwork routing through BNs is accomplished by calculating two piecewise optimal routes:

1. ***OLU in the same subnet as the BN***. See A) in Figure 79 on page 156.

   The first calculation will take place at the network node server of CP(OLU), NNS(OLU), and result in an optimal route from CP(OLU) to BN. The second route computation will take place at the nonnative network node server of BN and result in an optimal route from BN to CP(DLU).

2. ***OLU in a different subnet from the BN***. See B) in Figure 79 on page 156.

   The first calculation will take place at the network node server of CP(OLU), NNS(OLU), and result in an optimal route from CP(OLU) to BN. The second route computation will take place at BN and result in an optimal route from BN to CP(DLU).

Together these routes form a composite end-to-end route, which is piecewise optimal but not necessarily end-to-end optimal. Session routing will always be through the border node through which the DLU has been located first. During route calculation, other subnet connections will not be considered.

Each of the optimal routes is calculated as a specific function of the node and the TG characteristics within a given subnet. COS names being used for session setup must be present in both subnets and should "map" to physical routes with equivalent characteristics. It is recommended, therefore, to use the SNA-defined default names.

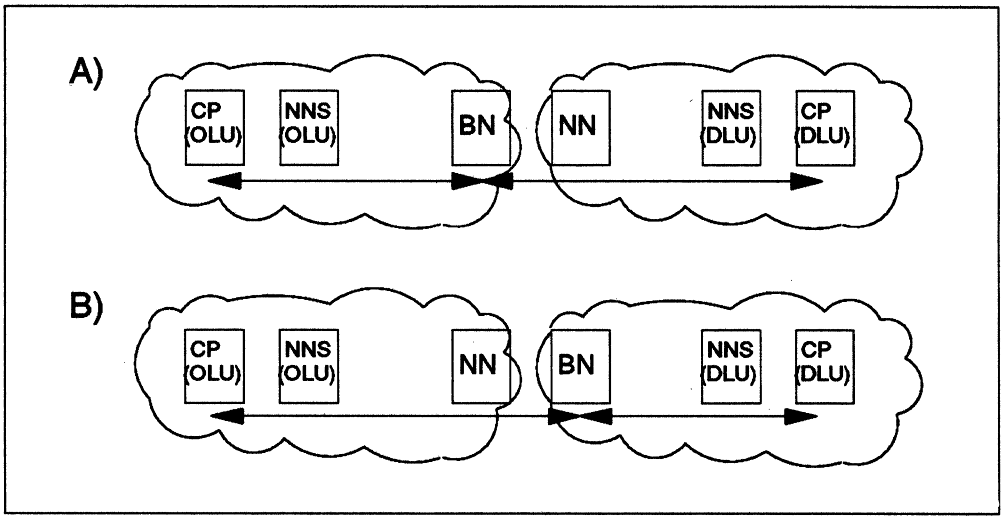

Figure 79. Piecewise Route Calculation

9.2 Border Node R2 (BN R2)

Border Node R2 describes the second stage of the APPN border node function. The announced boundary node support for APPN VTAM will be based on Border Node R2.

Major enhancements in comparison with BN R1 are:

- BN R2 allows session establishment between LUs residing in nonadjacent subnets.
- PCID caching, see page 155, becomes a mandatory function.

With BN R2, the terms ***intermediate*** and ***peripheral*** subnet have been introduced. The term intermediate subnet is self-explanatory; for example, see subnets B and C in Figure 80 on page 157. Peripheral subnets are subnets that may contain an endpoint of an intersubnet session, but never act as an intermediate subnetwork to connect two different subnetworks, for example subnet A and subnet D in Figure 80 on page 157.

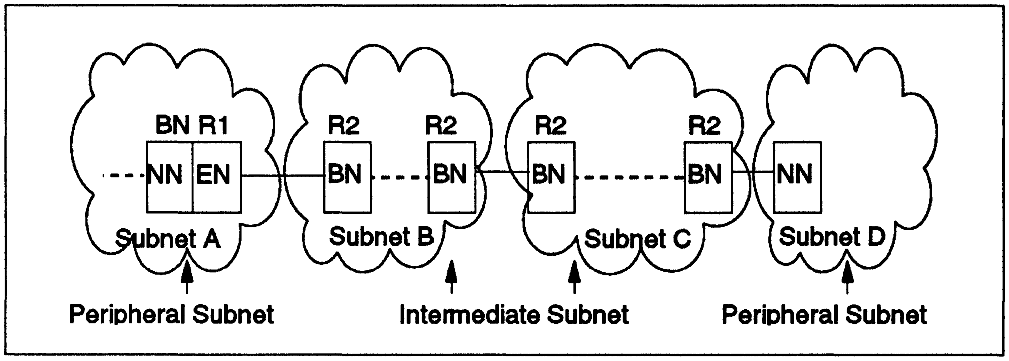

Figure 80. Cascaded Network Support (BN R2)

Boundary nodes of intermediate subnets must be BN R2 nodes. At the boundary of a peripheral subnet is either an “ordinary” NN or a BN R1. Recall, that a border node is a network node containing the border node function.

Chapter 10. APPN Implementations

Several products have implemented APPN and LEN functions, either as APPN nodes or as LEN end nodes. Among them are VTAM and NCP, AS/400, PS/2, and 3174. The characteristics of the implementations in VTAM/NCP, in the AS/400, in the 3174 and in the PS/2 as APPN and LEN nodes are summarized in this chapter. For an extensive description of APPN VTAM you should read "APPN VTAM" on page 169.

Some other IBM products with APPN implementations are not described in this document (such as the S/36 and DPPX/370 Release 3). APPN implementations in other manufacturer's equipment are not covered here either.

The size of an APPN network may be limited by restrictions caused by implementations having limited storage and processor capacity. In the following section, we mention some limitations imposed by several APPN implementations. The values specified are maximum values, but for performance reasons use of lower values may be recommended.

APPN architecture contains several optional functions. Table 9 on page 168 gives an overview of which functions have been implemented on the platforms mentioned before. For each of the functions, a reference has been made to the page where more information about the function can be found.

The evolution of SNA will continue and enhancements to APPN and APPN implementations will continue to be introduced. You should be aware that this chapter describes current hardware and software.

10.1 AS/400

APPN functions were available when the first version of the AS/400 was announced in 1988. The core functions had already been implemented in the S/36, the AS/400's predecessor. The AS/400 can be configured as a network node, as an end node, or as a LEN node. Further information can be found in *AS/400 Advanced Peer-to-Peer Networking* or in the *AS/400 Peer-to-Peer Networking Guide*.

APPN support is part of OS/400, the operating system of the AS/400.

10.1.1 Terminology

The term "location" is used for LU (logical unit).

A remote node is also called a "controller" or "control unit."

A "device" is the representation of a remote location (LU) in the local node.

Wildcard routing is also called "*ANY" routing.

Multi-Network Connectivity
The AS/400 is the first APPN system to implement border node functions with OS/400 V2.1. This capability is referred to as multi-network connectivity.

Congestion
The maximum number of intermediate routing sessions supported by a network node can be defined by the network administrator. Network nodes are said to be

congested if 90% of that number is reached. The node becomes "uncongested," when the actual number of intermediate routing sessions becomes less than 80% of the defined maximum.

10.1.2 System Definitions

The AS/400 can be defined to have multiple local LU names. Local resources have to be defined. Remote LUs need be defined only for:

- LUs in adjacent LEN end nodes
- LUs in adjacent APPN end nodes without CP-CP sessions, if the LU name is different from the CP name
- LUs in adjacent APPN end nodes that do not register and do not allow domain broadcast
- LUs in adjacent unauthorized APPN end nodes
- LUs, for which session security is defined
- Single-session LUs

Controller descriptions for LAN devices can be created automatically; however, their use is limited to independent LUs. Therefore, if an AS/400 connects to VTAM or NCP and dependent LU support is required, the controller and device descriptions need to be entered manually.

Limitations

The maximum number of conversations between local and remote transaction programs is 512 per mode. (A mode name is used when an LU starts a session to indicate the required session characteristics.)

The maximum number of sessions routed through an AS/400 network node is 9999.

The maximum number of modes simultaneously in use between local and remote LUs is 14.

The maximum number of "devices" that can be associated with a controller is 254.

The maximum RU length is 16 384 bytes.

10.2 3174 Establishment Controller

The Advanced Peer-to-Peer Networking Licensed Internal Code (LIC) adds APPN network node capabilities to the 3174 establishment controller. The APPN LIC feature is a separately orderable, no-charge feature that requires Configuration Support-C R1 LIC, as announced in January, 1991.

The APPN implementation allows the 3174 to be customized as an APPN network node only. For detailed information, refer to *3174 Planning Guide Configuration Support C*.

The 3174 network node supports links to other APPN network nodes, APPN end nodes, and LEN end nodes. Links supported are:

- Connections to APPN network nodes via token-ring, SDLC, and channel links.
- Connections to APPN end nodes and LEN end nodes via token-ring, coax attachments, SDLC, and channel links.

Notes:

1. To connect LEN end nodes and APPN end nodes via coax attachment to the 3174 network node requires the Peer Communication LIC feature.
2. APPN network node and end node connections via SDLC and channel links are supported only if the adjacent node is a VTAM or a composite VTAM/NCP node. (APPN support for VTAM is introduced with VTAM V4R1, also called APPN VTAM.)
3. LEN connections via SDLC and channel links are supported only if the adjacent node is a VTAM or a composite VTAM/NCP node. (LEN support for VTAM is introduced with VTAM V3R2, also called LEN VTAM.)

10.2.1 Terminology and Implementation Specifics

In the 3174 context, the term "gateway" applies to the 3174 token-ring gateway feature.

Dependent and independent LU traffic is supported on the same link to an adjacent VTAM APPN or VTAM LEN node. If dependent LUs attach to the 3174, an SSCP-PU session is requested when exchanging XIDs during link activation.

The 3174 considers "dynamic links" as limited resources. When the number of sessions using a specific link goes to zero, the link is taken down.

The 3174 assumes that all end nodes are authorized, meaning:

- Resource registration requests will be accepted from all end nodes within the domain of the 3174 network node.
- End node′s requests to be included in a domain search for resources not known to the 3174 network node, will be granted.

Network Node Characteristic: The 3174 network node uses two indicators within the node characteristics to regulate the number of sessions being routed through the network node. When the nummer of sessions concurrently being routed through the 3174 network node reaches the maximum number configured, the 3174 network node broadcasts TDUs indicating "intermediate routing resources depleted"; when the number of free buffers falls below a critical level, it broadcasts TDUs indicating "congestion." Other APPN network nodes will use this information to avoid additional sessions from being routed through the 3174 network node.

10.2.2 System Definitions

The 3174 does not allow users to enter system definitions through the node operator facility when the 3174 is online; all definitions must be entered during offline customization.

Limitations

The safe/store cache function is supported only if the 3174 has a hard file.

When connecting to a (VTAM APPN or VTAM LEN) T2.1 node on SDLC and channel links, the 3174 is always the secondary station (not negotiable).

The node's "route-addition resistance" is fixed at 128.

The maximum RU size is 8KB.

The maximum number of sessions routed through the 3174 network node is 1000.

The maximum number of links supported by the 3174 is 255. If a 4Mbps token-ring adapter is used, the limit is 140; when an 8KB frame size is used, then the maximum number of links supported drops to 100.

The maximum number of adjacent network nodes is eight.

10.3 PS/2

The PS/2 support for LEN end nodes was announced in 1988 and the support for APPN network nodes and APPN end nodes was announced in March, 1991. Originally (OS/2* V1.1, V1.2 and V1.3) LEN end node support was part of OS/2 Extended Edition (EE), which among other things offered Communications Manager support. APPN support, for APPN network nodes and APPN end nodes, has been introduced with a separate product called Networking Services/2 (NS/2). NS/2 is an extension of the OS/2 V1.3 Communications Manager support. With the introduction of OS/2 V2.0, LEN and APPN support both have become part of Communications Manager support within OS/2 Extended Services (ES). For detailed information, refer to *Extended Services for OS/2 Information and Planning Guide*.

Networking Services/DOS V1.0 announced in March, 1992, allows DOS workstations to participate, as LEN end nodes, in APPN as well as in SNA subarea environments.

Note: The following sections and Table 9 on page 168 cover the APPN support in OS/2 V2 ES V1.0.

10.3.1 Terminology and Implementation Specifics

During session initiation, an APPN or LEN end node tries to locate the partner LU within its local directory database and if no information can be found queries its network node server (APPN end node only). Session initiation will fail if the partner LU cannot be located. However, PS/2 end nodes can define, using a "local wildcard" definition, a ***substitute network node server***. The substitute network node server is an adjacent APPN network node with which no CP-CP sessions have been established. If, during session initiation, the partner LU cannot be located locally, the PS/2 end node sends a BIND to the substitute network node server. The substitute network node server then becomes responsible for locating the partner LUs, route calculation, and forwarding the BIND to the destination node. The concept of substitute network node server can be used to define a backup network node server for PS/2 APPN end nodes and provides a means of using wildcard definitions on a PS/2 LEN end node for its local use only.

Session Flow Control and Congestion

To manage the flow of data over a network, Communications Manager uses adaptive session-level pacing. The pacing occurs between each pair of adjacent nodes participating in the session route. When the APPC component of the Communications Manager is started, it determines the amount of memory that can be locked, that is, made non-swappable and non-moveable. Communications Manager then computes 30% of this amount as *available memory*, that is, the memory that can be used to transmit and receive user data. The amount of available memory dictates how the adaptive pacing algorithm is used, as well as determining when the T2.1 node becomes *congested*.

Table 8 on page 164 shows the memory allocation algorithm for congestion tolerance.

Table 8. Memory Allocation Algorithm

Memory Status	**Available Memory**	**Adaptive Pacing Action**	**Congested Status**
OK	50%-100%	Allow pacing window size to increase	No
Limited	37%-50%	Keep current pacing window size	No
Low	128KB-37%	Reduce pacing window size by 50%	No change
Serious	64KB-128KB	Set current pacing window size to 1	Yes
Serious	0-64KB	Hold pacing messages	Yes

The congestion status change is broadcast by a network node in topology database updates (TDUs) to other network nodes within the network.

10.3.2 System Definitions

OS/2 can be defined as a primary, secondary, or negotiable link station.

10.3.3 Limitations

Only one network node can be specified as server. But, another server can be designated as a substitute server (by using the end node's local wildcard function).

Route-addition resistance is fixed at 128.

The cache directory can hold up to 255 LUs. When more are learned, the oldest ones are discarded. The cache directory is saved to disk after every 20 updates.

10.4 VTAM and NCP

VTAM and NCP announced LEN support in 1987. The bulletin *Enterprise Networking with SNA Type 2.1 Nodes* gives a technical overview of this implementation. VTAM V4R1, announced in March, 1992, will allow VTAM and the then current NCP, to portray itself as an APPN network node or APPN end node.

When referring to the APPN support introduced with VTAM V4R1, we will use the term APPN VTAM.

Information in this chapter is based on APPN VTAM. LEN functions have been introduced with VTAM V3.2 and NCP V5R2.1.

Note: The following section gives a very condensed description of APPN VTAM; a more extensive discussion can be found in "APPN VTAM" on page 169.

10.4.1 Terminology and Implementation Specifics

APPN VTAM allows host systems to attach to APPN networks as APPN network nodes, APPN end nodes, or LEN end nodes. The function within VTAM and NCP which allows nonsubarea (or peripheral) nodes to connect is called the, VTAM or NCP, boundary function.

APPN VTAM offers extended connectivity in a transparent manner for both APPN and subarea LUs, without loss of functionality. All LU-LU session capabilities present in a pure subarea or a pure APPN environment, are also supported in a combined APPN/subarea networking environment.

To get full APPN connectivity requires full CP-CP connectivity. APPN VTAM allows CP-CP sessions to be established between an APPN VTAM node and any adjacent APPN node.

For other T2.1 nodes in an APPN network, the VTAM/NCP complex is an APPN network node, APPN end node, or LEN end node; however, internally, subarea protocols are used. VTAM and NCP configured as an APPN network node or LEN end node allow intermediate session routing within the VTAM/NCP complex.

Dynamic Cross-Domain Resource

In VTAM all independent LUs owned by attached APPN nodes can be either explicitly or dynamically defined. Resources not explicitly defined will, during session establishment, be defined as dynamic cross-domain resources. Dynamic definition of independent LUs owned by adjacent APPN nodes has been introduced with VTAM V3R4 and NCP V5.4.

APPN versus Subarea Flows

Within a subarea network, LUs are located using CDINIT or DSRLST requests. The VTAM host (see for example NN1 or NN2 in Figure 81 on page 166) that transforms APPN requests into subarea requests and vice versa, is called an interchange node (ICN). CDINIT and DSRLST routing in a subarea network can be seen as a sequence of directed searches.

Surrogate Network Node Server

A subarea network may enable session establishment between two disjoint APPN networks by giving an APPN network node appearance to either APPN network; see for example Figure 81 on page 166. If the ICNs providing the boundary function, NN1 and NN2, are two separate VTAMs between which no

CP-CP, only SSCP-SSCP, connectivity exists, then each of the T2.1 nodes in the "other" APPN network appears to its T2.1 endpoint partner as an APPN end node that connects to a *surrogate network server.* For example, NNA "sees" NNB as an ENB connected to the surrogate network node server NN1, and NNB "sees" NNA as an ENA connected to surrogate network node server NN2.

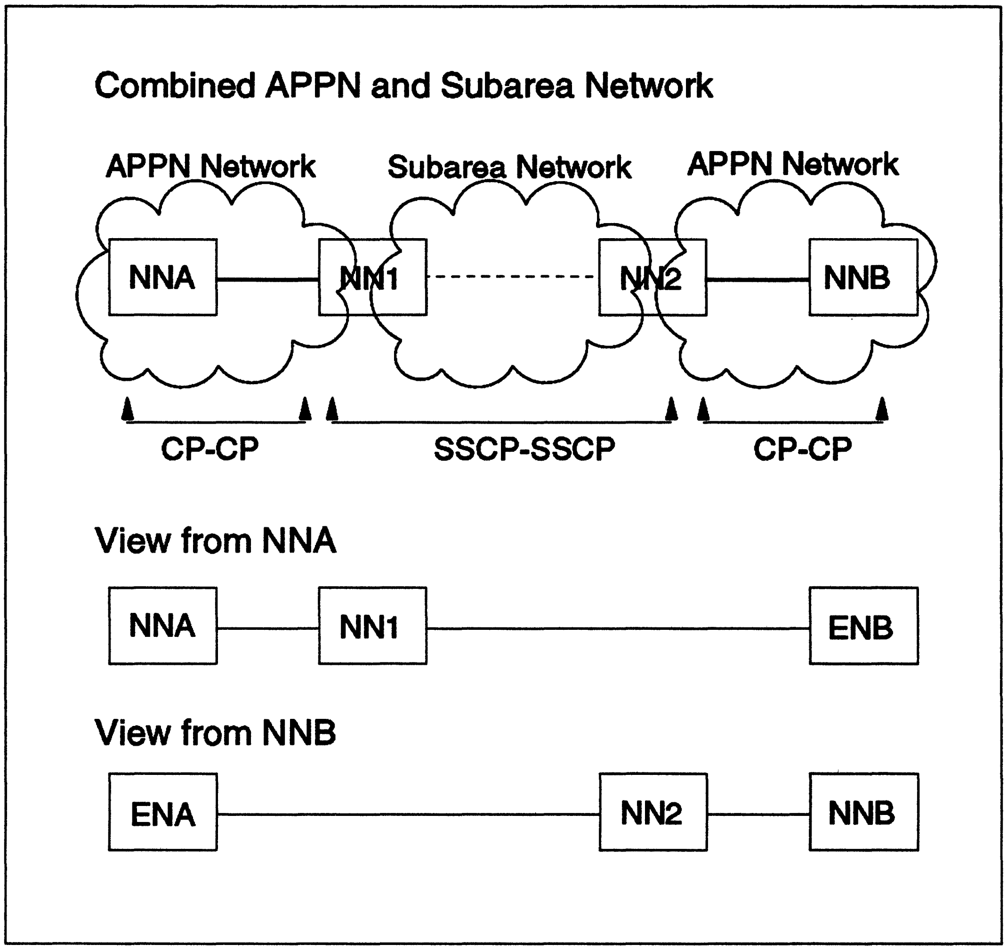

Figure 81. Surrogate Network Node Servers (NN1 and NN2)

Note, that the LUs within the subarea that are owned by different VTAMs from the one providing the boundary function, also appear to reside on an APPN end node. For more details about the concept of "surrogate network server" see page 176.

Transmission Priority and Class of Service

The COS name is obtained from a (VTAM) mode table (MODETAB). The class of service is used to select an operational route and a transmission priority from a list of predefined routes within a COS table (COSTAB). The transmission priority is used within the subarea part of the network (between VTAMs and NCPs) and optionally on the boundary links to attached T2.1 nodes. The latter part of a route within a subarea network is often called the *route extension.*

Casual Connection

VTAM/NCP may portray itself as an APPN network node, an APPN end node, or a LEN end node. A LEN connection between two (VTAM/NCP) nodes is called a *casual connection*; each side sees the adjacent side as one LEN end node.

Connection Networks

Currently, APPN VTAM does not allow the definition of its own attachment to connection network; however, APPN VTAM, as a network node server, is able to recognize a connection network and calculate a route through a virtual routing node.

Adjacent Link Station (ALS) Selection Function

The ALS selection function in the VTAM session management exit can be programmed to select a route to a destination LU when multiple LEN connections exist to an adjacent APPN network. This is equivalent to selective wildcard routing.

10.4.2 System Definitions

Generally, LUs are defined locally, that is, only once in a subarea network. For LEN connections, LUs have to be defined on both sides. Yet, there are some functions that provide dynamic network access and eliminate the need for multiple definitions.

Self-Defining Independent LUs

When a BIND from a LEN end node enters the subarea network, the OLU can automatically be defined (dynamic CDRSC). This function complements the wildcard search function in APPN networks. The DLU can be predefined, a dynamic CDRSC, or automatically defined by a VTAM exit.

Dynamic Switched Definition Support

Dynamic switched definition support simplifies adding switched devices to the network, including token-ring attached devices, which are treated as switched devices when connected through NCP's or VTAM's boundary function (not through a 3174 gateway). This support is for dependent or independent logical units. For dial-in support, reusable model definitions together with an installation exit routine are used.

Limitations

Route selection between APPN networks and subarea networks is not seamless, as independent algorithms apply.

Multiple LEN connections from the APPN network to the subarea network require at least VTAM V3R4 and NCP V5R4.

10.5 Summary of Implemented Functions

Table 9. Optional Functions

		APPN VTAM			AS/400			PS/2			3174
Function	Page	NN	EN	LEN	NN	EN	LEN	NN	EN	LEN	NN
Session Services											
Limited Resource	42	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes
Bind Segmenting	36	no	no	no	yes	yes	yes	yes	yes	yes	yes
Bind Reassembly	36	no	no	no	yes	yes	yes	yes	yes	yes	yes
Session RU Segmenting	27	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes
Session RU Reassembly	28	no	no	no	yes	yes	yes	yes	yes	yes	yes
Dependent LU Server	184	yes[1]	no	no	no	no	no	no	no	no	no
Dependent LU Requester	184	no	no	no	no	no	no	no	no	no	no[1]
Session Services Extensions	122	yes	yes	yes	no	no	no	no	no	no	no
User LU Name same as CP Name	86	no	no	no	yes	yes	yes	yes	yes	yes	yes
Border Node	151	yes[2]	n.a.	n.a.	yes[3]	n.a.	n.a.	no	n.a.	n.a.	no
Directory Services											
Central Directory Server (CDS)	87	yes	n.a.	n.a.	no	n.a.	n.a.	no	n.a.	n.a.	no
CDS User	87	yes	n.a.	n.a.	yes	n.a.	n.a.	yes	n.a.	n.a.	yes
End Node Resource Registration Server	86	yes	n.a.	n.a.	yes	n.a.	n.a.	yes	n.a.	n.a.	yes
End Node Resource Registration Requester	86	n.a.	yes	n.a.	n.a.	yes	n.a.	n.a.	yes	n.a.	n.a.
Safe-Store of DS Cache	90	yes	n.a.	n.a.	yes	n.a.	n.a.	yes	n.a.	n.a.	yes
Nonverify Function	102	yes	yes	n.a.	no	no	n.a.	no	no	n.a.	no
Negative Caching	105	yes	n.a.	n.a.	no	n.a.	n.a.	no	n.a.	n.a.	no
Topology and Routing Services											
Mode to COS Name Mapping	68	yes	yes	yes	yes	yes.	yes	yes	yes	no	yes
Safe-Store of Network TDB	62	yes	n.a.	n.a.	yes	n.a.	n.a.	no	n.a.	n.a.	no
Garbage Collection of Network TDB	64	yes	n.a.	n.a.	yes	n.a.	n.a.	yes	n.a.	n.a.	yes
Randomized Route Computation	72	yes	n.a.	n.a.	yes	n.a.	n.a.	yes	n.a.	n.a.	yes
Tree Caching	71	yes	n.a.	n.a.	yes	n.a.	n.a.	yes[5]	n.a.	n.a.	no
Incremental Updates to Trees	72	yes	n.a.	n.a.	yes	n.a.	n.a.	yes	n.a.	n.a.	no
Management Services											
Multiple-Domain Support	146	yes[7]	yes[7]	yes[7]	yes	yes	yes	yes	yes	no	yes
Domain FP-EP Relation	132	no	no	n.a.	yes	yes	n.a.	yes	yes	n.a.	yes
Explicit Primary FP-EP Relation	132	yes[7]	yes[7]	yes[7]	yes	yes	yes	yes	yes	no	no
Implicit Primary FP-EP Relation	132	yes[7]	yes[7]	yes[7]	yes	yes	yes	no	no	no	no
Held Alerts	149	n.a.	n.a.	n.a.	yes	yes	yes	yes	yes	no	yes
Receive NMVT on SSCP-PU Session	146	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes
Send NMVT on SSCP-PU Session	146	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes
Connectivity											
Connection Network	50	no[4]	no	n.a.	yes	yes	n.a.	yes	yes	n.a.	yes
Multiple TGs	42	yes	yes	yes	yes	yes	yes	yes	yes	yes	no
Parallel TGs	42	yes	yes	n.a.	yes	yes	no	yes	yes	yes[6]	no

n.a. = Not Applicable

Notes:

1. Statement of direction
2. Border Node R2: Statement of direction
3. Border Node R1
4. APPN VTAM is not able to define its own attachment to connection networks, but APPN VTAM acting as a network node server will recognize end node's connections to a virtual routing node and calculate a route accordingly.
5. Tree computation is partial, until destination node found.
6. Although not part of APPN architecture.
7. Multiple-domain support introduced with Netview V2R2. No CP-CP sessions, only LU-LU sessions.

Appendix A. APPN VTAM

VTAM V4R1 will be the first implementation of APPN on an IBM mainframe. As its name implies, APPN architecture reverses the hierarchical nature of SNA. By using a peer-to-peer approach, APPN offers advantages over subarea SNA such as:

- Better performance during session initiation - APPN uses fewer line flows per LU-LU session during initiation.
- Improved performance during network activation - APPN can eliminate control sessions, such as SSCP-PU and SSCP-LU, thereby eliminating many control flows during network activation.
- Reduced system definitions - APPN does not use PATH decks as it learns about network topology dynamically.
- Increased availability - as the topology is learned dynamically, there is no need to shut down parts of the network in order to add a single node.

Note: In the following sections we will use the term APPN VTAM when referring to VTAM V4R1, unless there are specific reasons to mention VTAM V4R1.

A.1 Overview

Starting from VTAM V3R2, a VTAM or composite node (VTAM and NCP) can portray itself as a LEN node to APPN nodes. With the introduction of VTAM V4R1, VTAM can now present an APPN image, either end node or network node. In addition to this, VTAM will be able to maintain subarea connections.

APPN VTAM is able:

- To support APPN nodes attached to the boundary function supplied by VTAM itself or supplied by NCP. A FID2 connection between a VTAM node and any node that operates as an APPN node is referred to as a boundary function transmission group (BF-TG).
- To support multiple connections to the same APPN node, known as parallel TGs.
- To allow the exchange of levels of CP support (including the CP-CP session over the connection), during establishment of an APPN connection.

By supporting an APPN appearance to the APPN network and a subarea appearance to the subarea network, APPN VTAM (see Figure 82 on page 170):

- Enables subarea LU-LU sessions ***through*** and ***into*** an APPN network.

  LUs within any of the VTAM domains shown in Figure 82, can have sessions with either LUs on NNA or NNB, or any other LU in any VTAM domain.

- Enables APPN LU-LU sessions ***through*** and ***into*** the subarea network.

  LUs on APPN network nodes NNA and NNB can have sessions with any LU, on either NNA or NNB, or with LUs in any VTAM domain.

- Creates a migration path from subarea to APPN networking.

  Only the VTAMs providing the APPN boundary function need to be on the current software levels. LUs controlled by back-level VTAMs, for example

VTAM6, can also establish LU-LU sessions with LUs in APPN or nonadjacent subarea networks.

APPN VTAM offers extended connectivity in a transparent manner for both APPN and subarea LUs, without loss of function. All LU-LU session capabilities present in a pure subarea or a pure APPN environment, are also supported in a combined APPN/subarea networking environment. For details and limitations see "LU-LU Sessions" on page 182.

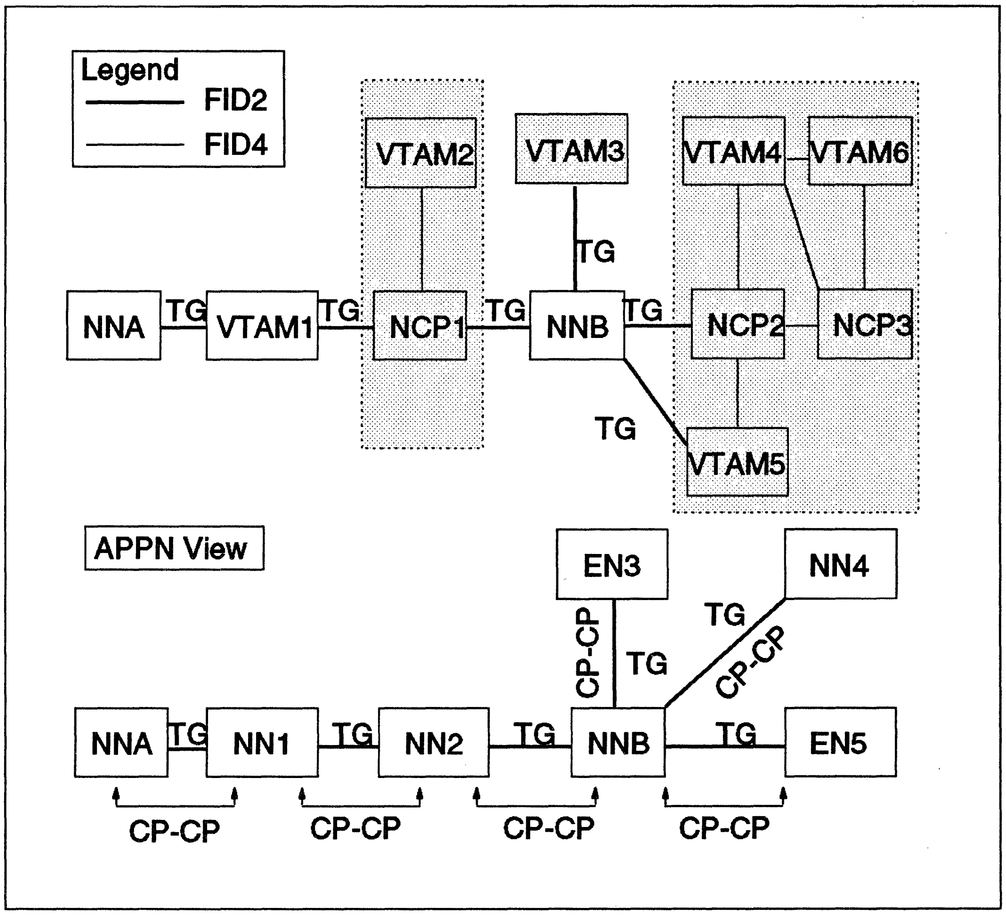

Figure 82. VTAM APPN Support.

In the APPN view the VTAM and composite network nodes (CNNs) are represented by NNx (x=1, 2, 4) and ENy (y=3, 5).

To get full APPN connectivity requires full CP-CP connectivity. APPN VTAM allows CP-CP sessions to be established between an APPN VTAM node and any adjacent APPN node. The CP-CP sessions:

- Traverse an APPN VTAM or NCP boundary function to an adjacent node
- Give APPN network connectivity into and across the subarea network
- Move APPN functions into the subarea network with full directory and session services functions

It is a user's choice to define existing subarea (FID4) links to T2.1 (FID2) links and use CP-CP sessions. A mixture of SSCP-SSCP and CP-CP sessions may be used.

A.2 VTAM Node Types

Possible node configurations and their functional abilities are determined by the VTAM start options. An APPN VTAM host can be configured as:

1. ***Subarea VTAM***

   The default parameter setting is such that VTAM V4R1 continues to operate as a "pure" subarea node. APPN VTAM, when started as a subarea node, supports SSCP-SSCP sessions but cannot have CP-CP sessions. It supports LEN connections with the same support provided by VTAM V3R4.1.

   As an example, see VTAM6 in Figure 82.

2. ***Interchange Node (ICN)***

   A VTAM host configured as an interchange node (ICN) is comprised of:

   - A single APPN VTAM node.
   - Optionally, one or more NCPs. An APPN VTAM node and one or more NCPs owned by VTAM is called a ***composite network node (CNN)***.

   An ICN is intended to replace the subarea CMC host. It may own NCPs and is the repository of all the functions provided by the CMC host. It provides ownership of dependent LUs, allowing these LUs to operate unchanged.

   The ICN routes sessions from APPN nodes into and through the subarea network using subarea routing, without exposing the subarea implementation to the APPN part of the network. This is accomplished by making the APPN VTAM node, plus all its owned resources, appear to other nodes as a single APPN network node with multiple connections. At the same time the ICN, and the NCPs it owns, will maintain their subarea appearance to other subarea nodes.

   The ICN supports SSCP-SSCP sessions with other VTAM nodes as well as CP-CP sessions with adjacent APPN network nodes and end nodes. This support allows the ICN to use both APPN and subarea data flows to locate LUs and to provide the best route between nodes. APPN session setup protocols, which flow on CP-CP sessions, are converted to the corresponding subarea protocols that flow on SSCP-SSCP sessions, and vice versa.

   As an example, see VTAM2 and VTAM4 in Figure 82.

3. ***Migration Data Host (MDH)***

   A migration data host (MDH) is a VTAM host that acts as an APPN end node and maintains FID2 connections to adjacent APPN network nodes. An MDH is able to maintain FID4 connections to directly attached NCPs but is not able to own NCPs.

   When maintaining FID2 connections to adjacent APPN network nodes, the MDH must have one of the following connections to these network nodes:

   - A channel connection, when connecting to an NCP that is part of an adjacent composite network node
   - A token-ring connection using a 3172

   An MDH is able to maintain both SSCP-SSCP and CP-CP sessions.

   MDHs do not provide intermediate session routing, and they do not transform the APPN session setup protocols to subarea session setup protocols or vice versa.

As an example, see VTAM5 in Figure 82.

4. ***APPN End Node Only***

   APPN VTAM configured as just an APPN end node has no subarea number assigned, is not able to maintain SSCP-SSCP sessions, does not support FID4 connections, and is not able to own NCPs. The VTAM end node is able to maintain CP-CP sessions and supports FID2 connections. It is added to the APPN network without requiring subarea network routing definitions.

   As an example, see VTAM3 in Figure 82.

5. ***APPN Network Node Only***

   APPN VTAM configured as just an APPN network node has no subarea number assigned, is not able to maintain SSCP-SSCP sessions, does not support FID4 connections, and is not able to own NCPs. The VTAM network node is able to maintain CP-CP sessions and supports FID2 connections. It is added to the APPN network without requiring subarea network routing definitions.

   As an example, see VTAM1 in Figure 82.

A.2.1 APPN/VTAM Network Node

APPN VTAM configured as a network node, either as a network node only or as an interchange node:

- Is able to perform APPN intermediate session routing.
- Maintains CP-CP sessions with adjacent network nodes and, optionally, with adjacent end nodes.
- Providing it has active CP-CP sessions with adjacent end nodes, allows these ENs to register the EN′s resources at the NN.
- Is able to dynamically inform a directory server of NN′s local resources and resources on served ENs.
- Can be configured as a central directory server to receive dynamic resource information from NNs.
- May own dependent LUs residing on the VTAM node itself or on nodes adjacent to the VTAM or NCP boundary function.

A.2.2 APPN/VTAM End Node

APPN VTAM configured as an end node, either as an end node only or as a migration data host:

- Is not able to perform APPN intermediate session routing.
- Can have only CP-CP sessions with the adjacent network node acting as its network node server.
- Registers its resources at its network node server. Resources explicitly excluded from being registered can be found by domain searches only if they are cached at their network node server because of an earlier session request originating in their EN.
- Requests not to be searched by its NN server when performing an APPN domain search.
- May own dependent LUs residing on the VTAM node itself or on nodes adjacent to the VTAM boundary function.

Possible node configurations and their functional abilities are summarized in the table below.

Table 10. Node Type Functional Summary

Node Type	APPN Nodetype	HOSTSA Number	CP-CP Sessions	SSCP-SSCP Sessions	NCP Ownership	Interchange Function[1]
Subarea Node Only	n/a	yes	no	yes	yes	no
Interchange Node	NN	yes	yes	yes	yes	yes
Migration Data Host	EN	yes	yes	yes	no	no
APPN EN Only	EN	no	yes	no	no	no
APPN NN Only	NN	no	yes	no	no	no

Note:

1. Interchange function allows APPN session setup protocols, which flow on CP-CP sessions, to be converted to the corresponding subarea protocols, which flow on SSCP-SSCP sessions, and vice versa.

A.3 CP-CP Sessions

VTAM V4.1 uses the same name for both the SSCP and the CP. The CP functions are similar to the subarea SSCP. APPN VTAM establishes CP sessions to increase connectivity into a network and to assist in LU-LU session initiation and termination.

CP-CP sessions, using APPC/VTAM support for the LU 6.2 sessions, help to create a contiguous APPN network. APPN directory services, topology services, and network management are dependent on CP-CP sessions.

All CP-CP sessions are currently supported over FID2 links only. The FID2 links (APPN TGs) are provided by either the boundary function of VTAM or the boundary function of NCP. Two different types of CP-CP sessions exist:

1. SNA Services Manager Session
2. CP Services Manager Session

SNA Services Manager Sessions provide transport for network management data, such as ALERTS. The entry point CP and focal point CP can have an SNA Service Manager session even if they are not adjacent to one another. The sessions use mode name SNASVCMG.

CP Services Manager Sessions provide a transport facility for directory (resource search and registry) and topology data. The sessions always exist in pairs, each CP being *contention-winner* in one session and *contention-loser* in the other. The two nodes must be adjacent, the sessions may follow different BF-TGs but both CP-CP sessions must use the same VR as the SSCP-PU session with the NCP that provides the boundary function. The sessions use mode name CPSVCMG.

APPN architecture allows CP-CP sessions between network nodes having different NETIDs, if at least one of the network nodes supports the border node function; current APPN VTAM does not support the border node function, although it may, as an NN, attach to a border node. IBM has announced that future releases of VTAM will support the border node function.

An end node can establish CP-CP sessions with an adjacent NN that has a different NETID.

An end node can have CP-CP sessions with only one network node at a time, which then is called the end node's network node server. An APPN VTAM end node (either a "pure" APPN EN or a migration data host can define a sequence of possible NN servers. VTAM starts to establish CP-CP sessions with the NN node listed first. If CP-CP sessions with its network node server fail, VTAM will try to establish sessions with either the first NN in the list or the next NN in sequence, depending on a user-defined service order.

The VTAM CP utilizes logic that is already implemented in subarea VTAM, to provide the full CP function as defined by APPN architecture.. A VTAM CP performs the following functions:

- Management Services Transport (MST)
- Directory Services (DS)
- Topology and Routing Services (TRS)
- Session Services for CP (SSC)
- Session Services for LU (SSL)

The APPN control point (CP) is treated in VTAM as an LU and internally represented as a VTAM application program. The VTAM CP functions are performed by different transaction programs.

A.3.1 Topology and Routing Services

The main purpose of topology and routing services is to maintain information about nodes, transmission groups (TGs), and classes of service (COS) so that appropriate routes through the network can be calculated. Topology and routing services is a function present in every network node and, with reduced functions, in every end node. There are two kinds of topology databases in an APPN network:

1. Local Topology Database

   In APPN architecture, this data set exists on APPN LEN nodes as well as on APPN end nodes.

   Note: In VTAM's current LEN implementation, local topology information about T2.1 nodes and TGs is kept with the rest of the subarea configuration. Because it is not a separate database, VTAM acting as a LEN node cannot strictly be said to have a local topology database.

   In an APPN end node, TRS uses the local database to supply the endpoint transmission goup vectors (TGVs) to the network node server during a search procedure.

   APPN VTAM does not save the local topology database but rebuilds it when VTAM reinitializes.

2. Network Topology Database

   After directory services has located a resource, topology and routing services in a network node will use the network topology database when calculating a route to that resource. The database contains information about NNs and TGs, and is identical on every NN in an APPN network. As the network topology changes, topology database updates are exchanged between adjacent NNs over the CP-CP sessions. To ensure that unnecessary topology updates are not propagated through the network, APPN VTAM has implemented APPN flow reduction mechanisms such as the flow reduction sequence number (***FRSN***) and resource sequence number (***RSN***).

   APPN VTAM keeps routes it has calculated between nodes and reuses the routes if applicable.

   In the network topology database, information is also kept about endpoint TGVs. They are received from local ENs that register their endpoint TGVs, as part of the ***nonverify function***. This information is not sent to other NNs in topology database updates (TDUs).

   In APPN VTAM, the network topology database can be saved to disk via an operator command. VTAM will use the information on disk to rebuild its topology database at initialization time.

   Note: The content of the network topology database is similar to other APPN implementations. One difference is the fact that VTAM has chosen to implement an architectural option that allows it to store the weight of TGs in the topology database, which reduces computing time when calculating routes.

Class-of-Service Functions

The COS database is an optional database as defined in the APPN end node architecture. If an EN does not support the COS/TPF (class of service / transmission priority field), then it relies on its NN server to provide a COS mapping. This mapping is done when the EN sends a request to set up a session with a resource using a certain mode name.

APPN VTAM provides a similar COS database on both its EN and NN implementations. VTAM allows COS definitions to be added or modified dynamically.

Mapping between the mode names and (APPN) COS names is done using definitions from the IBM-supplied default Logon Mode table: ISTINCLM. For this purpose a new parameter, APPNCOS, has been added to entries defined in the table. The existing COS keyword will be used, to select routes through the subarea network.

Mode to Class of Service Mapping.

1. APPN COS Selection

   When an interchange node is calculating the route to be used for a session that passes from a subarea network to an APPN network, an APPN class of service (APPNCOS) will be selected.

   The mode name to subarea COS mapping is done on the APPN side of the interchange node.

2. Subarea COS Selection

When an interchange node is calculating the route to be used for a session that passes from an APPN network to a subarea network, a subarea class of service (COS) will be selected.

The mode name to APPN COS mapping is done on the subarea side of the interchange node.

Route Selection Services

Route selection services is responsible for calculating the optimum route through the APPN network. The mode name specified in a session initiation request is mapped to a COS name. The COS selected indicates the ***required*** characteristics of the session. The information contained within the topology database contains the ***actual*** characteristics of the resources (NNs and TGs) in the APPN network. Together with TGVs obtained from the end nodes, an optimal route will be computed.

The route description is contained within a route selection control vector (RSCV). The RSCV contains a series of TG vectors from the node on which the PLU resides to the node which contains the SLU.

If there are multiple APPN networks separated by subarea networks, then in each APPN network a separate RSCV is calculated, to describe the route through the network. For the connection to an LU that resides ***in*** or is accessible ***through*** the subarea network, TG number 254 will be used. See Figure 83 on page 177.

Note: TG number 254 is a reserved TG number, which cannot be defined by customers. To provide transparancy to the T2.1 nodes in the APPN network, all LUs in or accessible through the subarea network, except the LUs owned by the ICN itself, are presented as if they reside on an end node that connects to the ICN using TG number 254. This ICN is also known as a ***surrogate*** network node server.

For example, assume an LU on NNA establishes a session with an LU on NNB. No end-to-end CP "connectivity" exists as VTAM1 and VTAM2 are connected via subarea (SSCP-SSCP) protocols.

Note: APPN topology database updates (TDUs) flow on CP-CP sessions and because there is no end-to-end CP connectivity between the two APPN nodes, NNA and NNB, are topologically isolated. Also, APPN session setup messages flow on CP-CP sessions, but because the VTAM interchange nodes VTAM1 and VTAM2 convert the APPN message flows into subarea flows, and vice versa, LU-LU session establishment is possible between LUs owned by NNA and NNB.

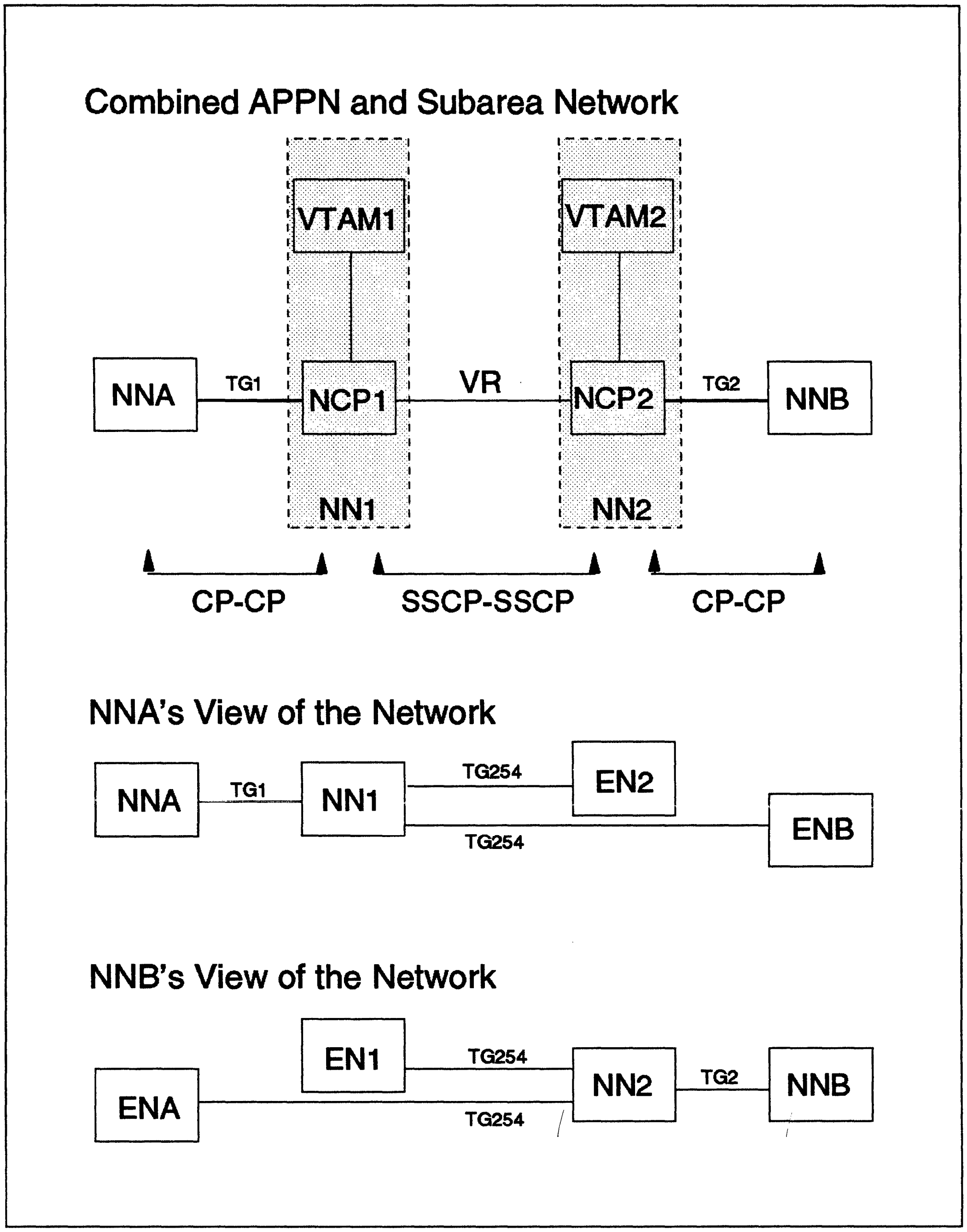

Figure 83. VTAM Interchange Node with Limited APPN Connectivity.

In the APPN view, the composite network nodes (CNNs) are represented by NN1 and NN2. Because the CNNs are connected using subarea (VR) protocols, two (topologically) disjoint APPN networks result. LU-LU session establishment is possible between any LUs.

Composite network node NN1, which is the APPN representation of VTAM1 and NCP1, will function as a surrogate network node server for node NNB. From the perspective of NNA, the DLU resides on an APPN "end" node connected via TG254 to the composite network node NN1. The BIND received by interchange node VTAM1 contains in its RSCV two TG descriptions:

```
(TG to NN1) and (TG254 to CP(SLU))
```

From the perspective of NNB, the OLU resides on a APPN "end" node connected via TG254 to the composite (surrogate) network node server NN2. The BIND sent by interchange node VTAM2, contains in its RSCV the following two TG descriptions:

```
(TG254 to NN2) and (TG to NNB)
```

Route calculation in a base APPN network is the responsibility of the network node server of the PLU. In a combined APPN/subarea environment, route selection may become the responsibility of the network node server of the OLU (which is not necessarily the PLU). For details, see page /CALC/.

A.4 Directory Services

The VTAM directory services component is responsible for managing the local directory database and controlling the search for network resources. One, or multiple, VTAM NNs may act as ***central directory server***. All directory servers are considered to have equal capabilities. If multiple directory servers exist, then a VTAM network node will query its closest directory server first. If the resource has not been located, then alternate directory servers will be searched in *sequence*.

VTAM has implemented the ***nonverify function***; see page 102.

The cache entries within VTAM's directory services show whether the resource is *available*, *unavailable*, or *unknown*. Knowledge of unknown resources will be maintained for a user-defined period. During this time the node will send a negative reply to Locate searches. This function is also known as ***negative caching***.

A.4.1 Directory Services Database

APPN VTAM, if configured as a network node, maintains information about resources in its directory database. If configured as an end node, the resource information is kept in the VTAM resource definition table (RDT).

The database contains location and availability information about network resources. Each APPN network node (NN) contains a directory services database. The database learns about resources through pre-definition, resource registration and network searches. As a node learns new information about resources in the network, it will update its database to reflect the new information. The database serves as a dynamic means of keeping track of network resources. The database kept in storage will be written to disk via an operator command or during an orderly VTAM shutdown. When VTAM reinitializes, VTAM will use this information to rebuild its working database.

The resources kept in storage have an entry type associated with it as follows:

- ***Register***

  These entries are written into the database as a result of end node registration. Resources are deleted on request of the EN or after the CP-CP session to the EN becomes inactive. Entries will be updated on request of the end node that did the registration, after an operator command, or after a directed Locate to the owning node returned a "resource unknown."

  These entries are ***not*** written to disk.

- ***Defined***

  These entries are written into the database after activation of CDRSC major nodes. The resources are deleted after deactivation of the CDRSC major node. Entries will be updated after a directed Locate to the owning node returns a "resource unknown," after a broadcast search discovers that the resource has been moved, or after an operator command.

  These entries are written to disk. When VTAM is restarted, priority is given to the information stored, over predefinition, when both have an entry for a resource.

- ***Dynamic***

  These entries are written into the database as a result of network searches, or as result of central resource registration. Entries are deleted:

  - After a search fails and this node is the network node server of both OLU and DLU
  - After a search fails and this node is a directory server doing a search for the DLU
  - If a DLU entry has not been used for eight days

  Entries are updated after a search discovers that resources have been moved.

  Dynamic entries are written to disk, with the exception of information obtained from OLU resource caching.

A.4.2 Resource Registry

APPN VTAM implements the following APPN resource registration functions, both as a requester and as a server:

- Registration of end node's resources at its network node server.
- Registration of end node's and network node's resources at a directory server (also known as ***central resource registration***).

  Note: An end node can only request its network node server to register resources at the directory server; it cannot register the resources itself.

Registering resources, most notably VTAM application programs (as LUs), as they are the most likely targets of LU-LU session setup traffic, will cut down the network traffic for session setup. VTAM allows resource registration to be user controlled. A resource can be:

- Not registered
- Registered at the network node server
- Both registered at the network node server and the central directory server

Registering CDRSC resources can be done to "preload" the DS database to avoid broadcast search processing for known LUs.

Dependent LUs owned by ENs need to be registered because a VTAM end node does not allow itself to be searched for resources. There is no need, except in order to reduce the number of setup flows, to register dependent LUs owned by an NN.

A.4.3 Network Searches

APPN VTAM gives the user extensive control over the order in which the network is searched for resources, and has implemented a number of mechanisms to optimize search procedures and to avoid duplicate searches.

The APPN and subarea search forwarding algorithms are modified to allow the propagation of the search request into the APPN or subarea network. The VTAM interchange node (ICN) transforms resource search procedures and session setup protocols from APPN Locate formats to the corresponding CDINIT and DSRLST formats, and vice versa.

Although a VTAM interchange node (ICN) gives the external appearance of a single node, internally there are two logical nodes, an "APPN side" and a "subarea side." As a result, when searching for a resource, special consideration is given to the way each side determines whether the other side owns a resource or knows about it. The APPN and subarea search algorithms are modified to interrogate the local database of the other side before forwarding a search request to other nodes.

Equivalent to the APPN local database at the APPN side is the resource definition table (RDT) at the subarea side. The VTAM RDT contains resources, same or cross-domain that are either defined explicitly or learned dynamically. A cache search of the subarea side includes checking the RDT and resolving possible USERVARs. VTAM distinguishes two types of local subarea cache searches:

1. The Limited Subarea Cache Search:

   Only DLUs present in the ICN's domain are considered. These include application programs, dependent LUs, and independent LUs that have an active LEN connection from that ICN toward the DLU.

2. The Extended Subarea Cache Search:

   Both, same-domain and cross-domain definitions cached in the ICN will be considered. VTAM will perform ALIAS translation, call its adjacent link station (ALS) exit, and so on. No search will go into the subarea if the resource is not found. If an entry is found, VTAM will do a CDINIT/DSRLST type search to verify that the resource is available.

Besides a cache search, the ICN may perform a full search of the subarea network as well.

When an ICN receives a search request for a resource, then VTAM will always check its local directory services database and perform a local subarea search. A VTAM network node server will query topology and routing services for the resource in case an LU has the same name as its CP. If the resource has been found locally, then the request is immediately forwarded to the resource.

If VTAM has no knowledge of the resource, then VTAM will start querying the authorized end nodes that have not registered their resources before starting an APPN or subarea network search.

APPN and Subarea Search Order

For search requests originating from its subarea side, the ICN will use existing logic to scan through its adjacent SSCP tables. To direct VTAM to start an APPN network search, a special entry is used. When requested to start a network search from either its APPN or subarea side, VTAM will, at a user-controlled point, start to perform the APPN network search.

Subarea Search

When an ICN receives a search request from its APPN side during APPN searches, then it is a user's choice:

- To include the subarea side of the ICN in the network search
- To exclude the subarea side of the ICN from the network search
- To limit the subarea search to the cached entries the ICN has on its subarea side

Disjoint Subarea Networks

Two subarea networks are said to be ***disjoint***, if they have the same NETID and are connected only by an APPN network; that means, no SSCP-SSCP sessions exist between VTAMs residing in different parts of the network. When an ICN receives a search request from the APPN network, it normally will not forward it into the subarea network if the request was sent into the APPN network by an ICN with the same NETID as the receiving ICN. This is done to prevent search requests looping between a subarea and APPN network connected by multiple interchange nodes. But, the user has the option to explicitly specify a remote ICN as being disjoint. Search requests from the APPN network originating from this ICN will then be forwarded into the subarea network.

Serial Interchange Node Search

An ICN will never forward a search request into the subarea network when it has received a locate request with the ***suppress subarea search*** bit ON. The ICN will perform a local subarea search only.

This bit will be set by ICNs when starting an APPN broadcast search. If the APPN broadcast search is unsuccessful, direct searches are sent to ICNs to which APPN connectivity exists. This process is called ***serial interchange node search***. The method described effectively splits the broadcast search into two parts:

1. An APPN broadcast, done in parallel
2. A subarea search, done sequentially

A.4.4 Avoiding Duplicate Searches

A VTAM interchange node may receive a search request at its subarea side from either a directory server or from another interchange node. If the request is received from another interchange node and APPN routes can be calculated to the origin interchange node, then the receiving interchange node will never forward the request into the APPN network through which connectivity exists. Topology and routing services provides a list of the interchange nodes in the APPN network from its topology database.

VTAM interchange nodes will use an *SSCP visit count* field in APPN Locate requests and subarea CDINIT/DSRLST to limit the number of SSCPs that are tried on a specific search path. Although APPN nodes do not use the SSCP visit count, they will pass its value unchanged. Each gateway SSCP performing SNI rerouting, or ICN performing an APPN/subarea (or vice versa) transformation of the search request, will decrease the count by one. If the count falls to zero, then VTAM will not propagate the request but return a negative reply or response.

To avoid duplicate searches in the subarea parts of the network, an ICN performs caching of searches whenever it transforms APPN to subarea searches, or vice versa. The node performs caching on the basis of:

- Fully Qualified Procedure-Correlation Identifier (FQPCID)

  The FQPCID is used to correlate a Locate search with its replies.

- Procedure Resubmit Number (PRN)

  PRN is used by subarea and interchange nodes to distinguish related search procedures. The use of PRN is part of base APPN architecture, and only the NNS(OLU) and NNS(DLU) will modify the PRN.

- PCID Modifier

  The PCID modifier in the Locate request is used to distinguish subprocedures for a Locate procedure. Besides the origin and destination node, intermediate nodes may also start subprocedures.

- SSCP visit count

It is a user's option to specify the period that the search request is cached (default 8 seconds). The ICN node will delay successive search requests for the same DLU in order to prevent multiple broadcast searches across the APPN and subarea part of the network.

A.5 LU-LU Sessions

Base APPN architecture limits sessions to PLU-initiated, LU 6.2 sessions. APPN VTAM extends existing subarea functions (such as SLU-initiated and third-party initiated sessions, autologon, session release request, etc.) for all LU types, to APPN networks. This has been accomplished by mapping APPN and existing subarea session setup protocols.

APPN VTAM will be able:

1. To work with other APPN nodes to establish LU-LU sessions through an APPN network
2. To work with prior releases of VTAM to establish LU-LU sessions through subarea networks
3. To work jointly with other APPN products and prior releases of VTAM to establish LU-LU sessions through combined APPN and subarea networks

APPN VTAM has implemented the functions described in "Session Services Extensions" on page 122, to make sure that all LU-LU session capabilities present in either pure subarea or pure APPN networks are also supported in a combined APPN and subarea networking environment, independent of the APPN and subarea components connecting both session partners.

Only the VTAM interchange node providing the connection between the subarea and APPN network needs to be on the current software level; SSCPs having subarea connectivity to these boundary VTAMs may be back-level VTAMs.

If one, or both, session partners are independent LUs, then LU-LU sessions are limited to PLU-initiated LU 6.2 sessions. If both LUs are controlled by a VTAM SSCP, then any session type known to subarea SNA is supported.

A.6 Dependent LU Support

As mentioned in "Dependent and Independent LUs" on page 20, all LUs depend on the services of a control point. The essence of a dependent LU is the fact that it is always dependent on the services of a control point outside of the node on which the LU resides. In this section we will focus on the case of an LU dependent on services offered by a VTAM SSCP, and residing on either a VTAM node or on nodes adjacent to the VTAM or NCP boundary function.

Among other things, SSCP support includes SLU, PLU, and third-party initiation, autologon support, and session queuing/notification. It also includes interpret functions and unformatted and formatted session services support, for example to allow a human operator to request an SLU-initiated LU2 session.

Currently, for dependent LUs residing on a node adjacent to the VTAM or NCP boundary functions, the LU sessions will always traverse the VTAM or NCP boundary function.

Note: APPN VTAM nodes configured as end nodes are not able to perform APPN intermediate session routing, but do allow nodes to attach using the VTAM boundary function. Dependent LU sessions may traverse the VTAM node via its boundary function.

A.6.1 Dependent LU Server

As mentioned in the previous chapter, current support for dependent LUs requires that the LUs reside on either a VTAM node, or on a node adjacent to the VTAM or NCP boundary function.

To allow the session capabilities currently provided by VTAM SSCPs to all LUs in an APPN network, would require either that the SSCP functions be distributed to remote APPN nodes, or that the SSCP function be enhanced allowing VTAM to serve nonadjacent nodes.

IBM has announced extended APPN support for dependent LUs, based on enhanced SSCP support by VTAM. The enhanced support allows traditional SSCP-PU and SSCP-LU data flows to be multiplexed in LU 6.2 CP-CP sessions to nonadjacent nodes. Benefits of this approach are:

- VTAM extends its SSCP support to LUs residing on nodes that are nonadjacent to the VTAM or NCP boundary function.
- Dependent LU sessions do not necessarily need to traverse a VTAM or NCP boundary function. APPN search logic is used to find the best route.

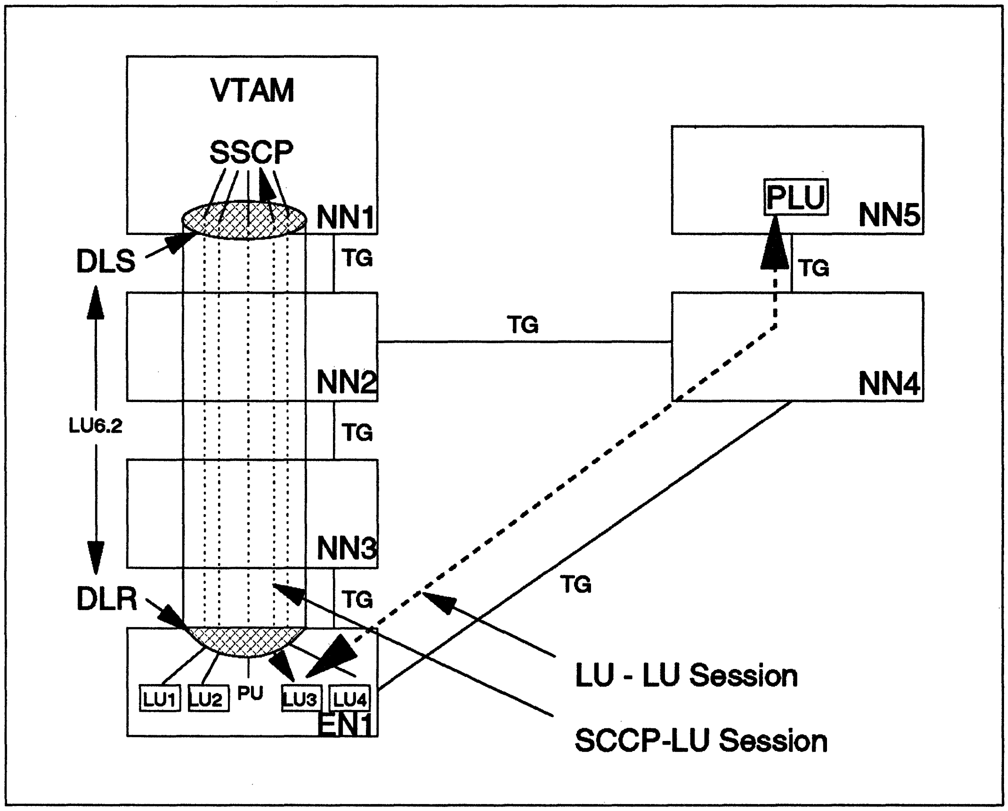

Figure 84. Enhanced Dependent LU Support. Data flowing on the SSCP-LU session between VTAM and LU3 on EN1 is encapsulated in the LU6.2 session between the DLS function on NN1 and the DLR function on EN1. The LU-LU session can use the optimal path between EN1 and NN5.

As shown in Figure 84, the extended support for SSCP dependent LUs requires two functions:

1. Dependent LU Server (DLS)
2. Dependent LU Requester (DLR)

VTAM will enhance its support for dependent LUs by implementing dependent LU server (DLS) support. The SSCP-PU and SSCP-LU session data flows are encapsulated in two parallel LU 6.2 sessions between the dependent LU server (DLS) and the dependent LU requester (DLR). T2.1 nodes (EN or NN) that want to support dependent secondary LUs have to implement the dependent LU requester (DLR) support to invoke SSCP services.

Note: The DLS/DLR concept requires that the PLU and its CP support sessions with dependent LUs. Currently, only VTAM and VTAM application programs provide this support.

Abbreviations

ABM	asynchronous balanced mode
ALS	adjacent link station
APPC	advanced program-to-program communication
APPN	advanced peer-to-peer networking
ASM	address space manager
BN	border node
BTU	basic transmission unit
CN	connection network
COS	class of service
COSM	class-of-service manager
CP	control point
CPMS	control point management services
CS	configuration services
DAF	destination address field
DD	directory database
DLU	destination logical unit
DLC	data link control
DLR	dependent LU requester
DLS	dependent LU server
DS	directory services
EN	end node
ENCP	end node control point
EP	entry point
FID	format identifier
FQPCID	fully qualified procedure correlation identifier
FRSN	flow-reduction sequence number
GDS	general data stream
ILU	initiating logical unit
ISO	international standards organization
ISR	intermediate session routing
LEN	low-entry networking
LFSID	local-form session identifier
LU	logical unit
LMS	local management services
LN	low-entry networking node
MAC	medium-access control
MS	management services
MSU	management services unit
NAU	network accessible unit
NCP	network control program
NN	network node
NNS	network node server
NNCP	network node control point
NNTDM	network node topology database manager
NOF	node operator facility
NRM	normal-response mode
OAF	origin address field
ODAI	OAF′/DAF′ assignor indicator
OLU	origin logical unit
PC	path control
PU	physical unit
PUMS	physical unit management services
RSCV	route selection control vector
RSN	resource sequence number
RSS	route selection services
RU	request unit
SABM	set asynchronous balanced mode
SAP	service access point
SATF	shared-access transport facility
SCM	session connector manager
SDLC	synchronous data link control
SNA	system network architecture
SNRM	set normal response mode
SOC	sphere of control
SS	session services
SSCP	system service control point
TDB	topology database
TDM	topology database manager
TDU	topology database update
TG	transmission group
TPF	transmission priority field
TRS	topology and routing services
TSO	time-sharing option

VRN	virtual routing node
VTAM	virtual telecommunications access method
XID	exchange identification

Index

GG24-3669-01

APPN Architecture and Product Implementations
Tutorial

GG24-3669-01

PRINTED IN THE U.S.A.

GG24-3669-01