IBM

International Technical Support Centers

APPN ARCHITECTURE AND PRODUCT IMPLEMENTATIONS TUTORIAL

GG24-3669-00

APPN Architecture and Product Implementations Tutorial

Document Number GG24-3669

May 1991

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.

First Edition (May 1991)

This edition applies to IBM Advanced Peer-to-Peer Networking Architecture and Product Family Overview that was announced on March 5th, 1991.

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

A form for reader's comments appears at the back of this publication. If the form has been removed, address your comments to:

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

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.

CSYS (173 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.

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

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:

- The first two chapters "APPN Overview" and "Node Structure" provide an overview of Advanced Peer-to-Peer Networking concepts and terminology.
- The following five chapters describe the architected functions of the control point in more detail. Each chapter is self-contained, but there are references to previous and subsequent chapters. This document is not a reference manual, but each chapter should be read from the beginning, to get a full understanding.
- The two chapters on "APPN Implementations" and "Sample Configurations" go beyond the architecture and consider some APPN products.
- The last chapter addresses some selected functions related to APPN for large networks.
- Every attempt was made to shun abbreviations. Some could not be avoided, in particular in the artwork. They are explained in "Abbreviations" on page 169.

Related Publications

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

- 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, *Systems Network Architecture Technical Overview*
- GC30-3084, *Systems Network Architecture Transaction Programmers Reference Manual for LU Type 6.2*
- GG24-3433, *Enterprise Networking with SNA Type 2.1 Nodes*
- SC30-3346, *Systems Network Architecture Management Services Reference*
- SC30-3383, *LAN Technical Reference*
- SC30-3422, *Systems Network Architecture Type 2.1 Node Reference*
- SC31-6808, *Systems Network Architecture LU 6.2 Reference: Peer Protocols*
- SC52-1110, *NS/2 Installation and Network Administrator's Guide*
- ISBN 0-201-00029-6, *The Design and Analysis of Computer Algorithms* by Aho, Hopcroft, Ullman.

The following publications are the first ones in a group of ITSC documents about APPN implementations.

- GG24-3287, *AS/400 Advanced Peer-to-Peer Networking*
- GG24-3662, *Networking Services/2 Installation, Customization, and Operation*

Acknowledgements

The advisor for this project was:

Paul Berdowski
International Technical Support Center, Raleigh

The authors of this document are:

Bernd Kampmann
IBM Germany

Martin Numan
IBM The Netherlands

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

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

Marilyn Beumeler
IBM Raleigh

Larry Plank
IBM Rochester

Gary Schultz
IBM Raleigh

John Q. Walker II
IBM Raleigh

Thanks to the following people for the invaluable advice and guidance provided in the production of this document:

Shawn Walsh
IBM Raleigh

Chapter 1. APPN Overview

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

1.1 LEN and APPN

A network can be very simple. For example, two PS/2s* could be connected by a telephone line to form a Low Entry Network (LEN) connection, as shown in the figure below.

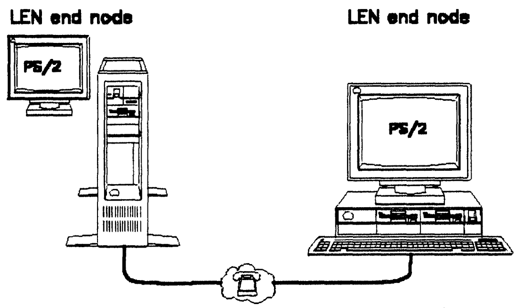

Figure 1. Two PS/2s Forming a LEN connection

The purpose of connecting these two systems is to exchange data between two end users. An end user could 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). For program-to-program communication, this LU would typically be an LU 6.2. When two LUs join in order to exchange data, this is called an LU-LU session.

In the case above, the two systems (PS/2s) and their corresponding LUs are known as LEN end nodes. This does not mean that all PS/2s are LEN end nodes, nor does it mean that all LEN end nodes are PS/2s. In fact, there are other systems that can act as LEN end nodes. VTAM* and NCP are among them. (VTAM and NCP are the software products for large systems that enable data exchange between end users. Data exchange between large systems is commonly known as subarea networking.) Using the architectural terms, the configuration above could be drawn as shown in Figure 2.

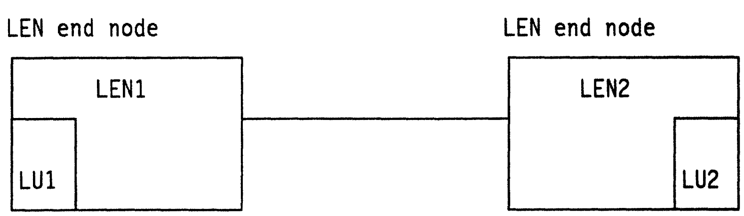

Figure 2. The Basic LEN Connection

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, from the point of view of 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 for building 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.

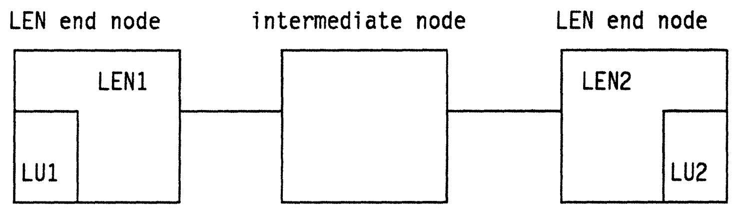

Figure 3. LEN End Nodes Connected to an Intermediate Node

It must be noted that the intermediate routing function is not defined by the LEN architecture. 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. According to the LEN architecture, the relation between LEN end nodes is always a "*two*-node peer relationship".

The intermediate node could be, for example, VTAM and NCP. These products have had the intermediate routing function for some time and implemented the LEN end node function later. In this way, VTAM and NCP cannot only be used with subarea networks, but they can also provide intermediate routing between LEN end nodes. The following figure gives an example of this configuration with VTAM on an ES/3090* as intermediate node.

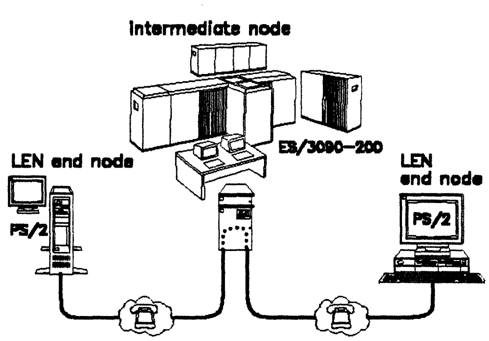

Figure 4. VTAM/NCP Providing the Intermediate Routing Function for LEN End Nodes

For the next step, we take the AS/400* as an example. In Figure 5 on page 3 you see another three-node network. The system in the middle has intermediate routing functions, of course. But there is more to this network. The nodes implement the Advanced Peer-to-Peer Networking (APPN) architecture, which has been announced and published as the latest extension to SNA (System Network Architecture). That means they have all the capabilities of a LEN end node (T2.1 node), plus some additional ones.

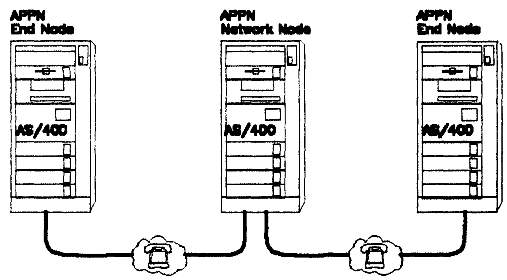

Figure 5. AS/400 as APPN End Node and APPN Network Node

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.

The APPN network node has intermediate routing function and provides services to the end node, which the end node cannot or need not provide itself. Figure 6 gives you an example of the services provided by the network node. When LU1 requests a session to a partner LU3 somewhere in the network, the network node will locate the partner LU and assist in establishing the session.

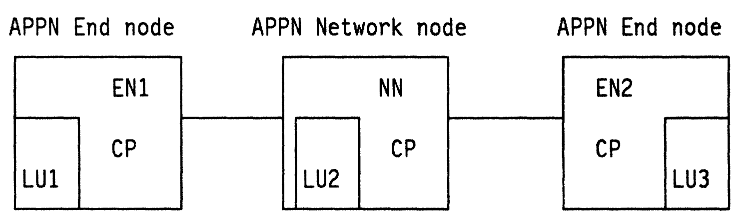

Figure 6. Advanced Peer-to-Peer Networking with Three Nodes

The previous pictures are only examples for different configurations of peer nodes. They do not imply that networks can only consist of nodes of the same product type, such as only AS/400 or only PS/2. In fact, the contrary is true. The following figures give examples for that. Figure 7 on page 4 shows an APPN network node that connects to two APPN end nodes and one LEN end node. The APPN nodes communicate through CP-CP sessions. Sessions can be established from any LU to any LU.

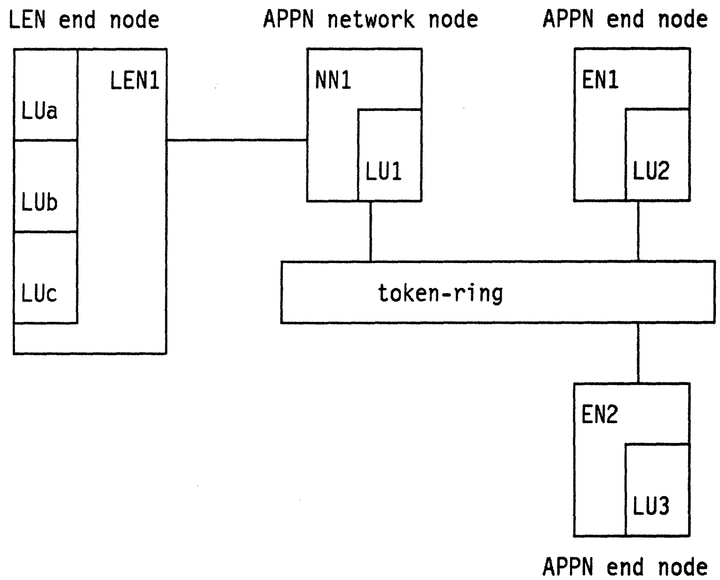

Figure 7. APPN Network with Different Node Types

While the previous figure explained the architectural node types used in the network, the next picture (Figure 8) shows a variety of products connecting through different link protocols.

The APPN network node is implemented in a 3174 workstation controller, which provides services for two PS/2s, which act as APPN end nodes. They are connected to the 3174 through a token-ring.

The 3174 is also the gateway to the subarea network. The subarea network is composed of large host systems with VTAM (T5 node), communication controllers with NCP (T4 nodes), and their directly attached resources. In this example, the connection to the subarea network is a LEN connection. As we said before, LEN means "two-node peer relationship". So, from the 3174, the whole subarea network is viewed as one LEN end node with many LUs. On the other hand, the NCP connected to the 3174 and its owning VTAM sees the LUs on the token-ring as residing in the 3174.

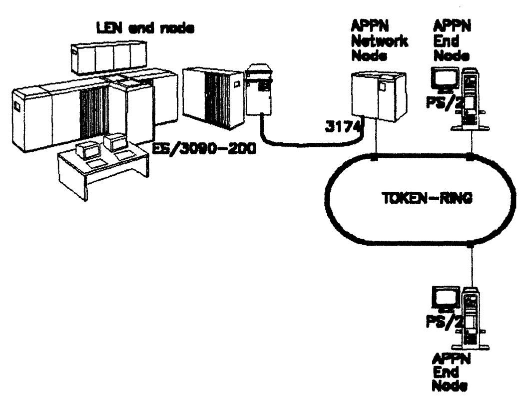

Figure 8. APPN Network with LEN Connection to Subarea Network

APPN networks can consist of any number of nodes. Different transport protocols may be used. Figure 9 on page 5 gives an example of a larger network. Here, the subarea network connects to two APPN networks[1]. It would also be possible for an APPN network to connect to two subarea networks[1].

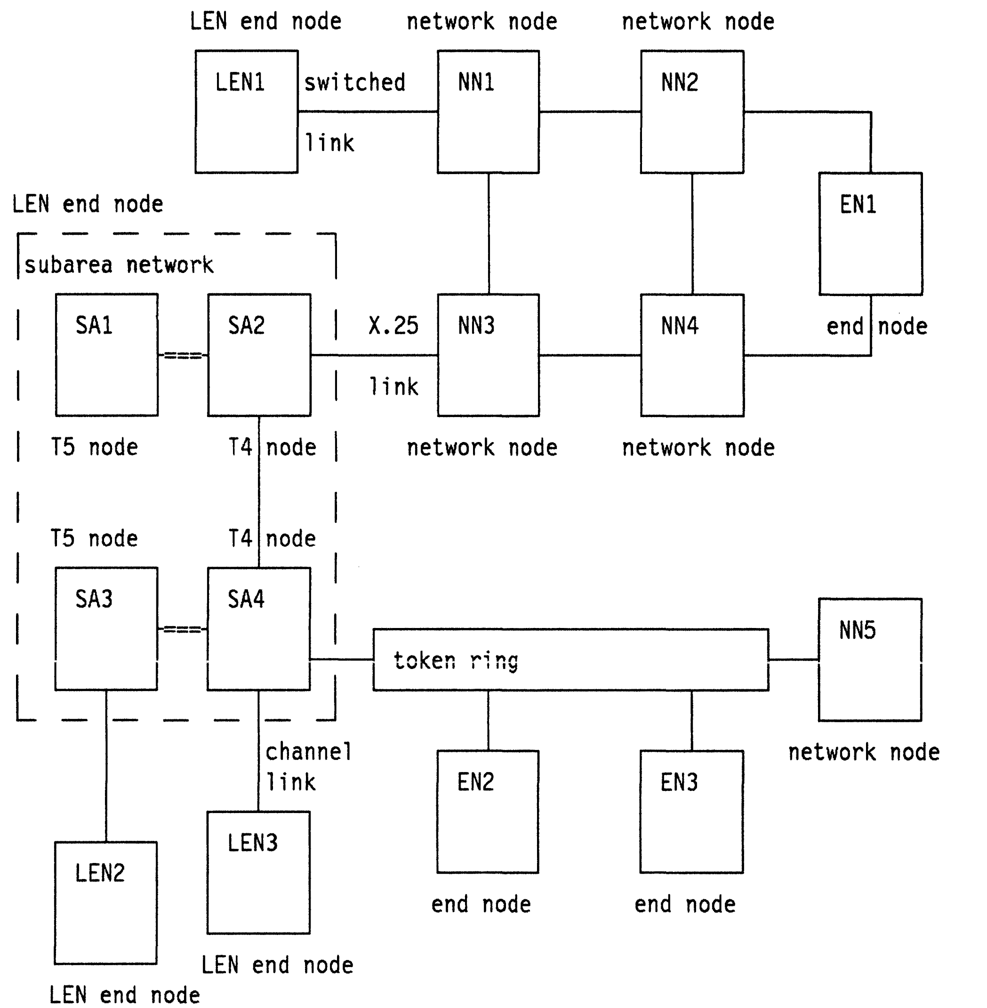

Figure 9. APPN Network with Various Node Types and Link Protocols

You have seen that the architecture defines several types of nodes and that the CPs of these nodes have a different scope of functions. The node types are defined more precisely later in this chapter. The CP functions are covered in several chapters from page 17 to page 135. The implementation of the functions defined by the architecture may be different in different products. Chapter 8, "APPN Implementations" on page 135, will provide details.

1 Two APPN networks that connect through a subarea network, or two subarea networks that connect through an APPN network, can only establish LU-LU sessions across the dissimilar network. They cannot establish control sessions, that is, CP-CP sessions or SSCP-SSCP sessions across the dissimilar network.

1.2 Names

Names are important. Network names eliminate the need for the end user to know the location of different resources within the network.

The Network Accessible Unit

In a network, all components that can establish sessions with one another are called network accessible units. Examples are CPs and LUs. NAU was previously used as an abbreviation for "network *addressable* units". In the APPN context, NAUs are represented by their name rather than by their address. Therefore, the terminology has been refined.

Within one network, the names of the network accessible units must be unique. An administrative process must be put into place to ensure the uniqueness of the names. In order to make the administration of names easier, "the network" can be divided into partitions.

Network Identifiers

Each partition of "the network" is given a unique network ID. The network ID is used throughout SNA - in the subarea portion of the network as well as in the APPN portion. Because the names of LUs and CPs need to be unique only within the scope of a network ID, they can be assigned independently across distinct network IDs.

Registered network IDs can help network administrators to ensure the uniqueness of the network ID they use. For this reason, IBM* provides a worldwide registry for network IDs; information on the registration process can be obtained through IBM branch offices.

A registered network ID is 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).

Unregistered network IDs can be 1 to 8 bytes long.

Network Names

A network name is an identifier of a network resource. Each CP, LU, link, and linkstation 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).

Comparison to subarea network: In VTAM and NCP, definitions are made offline in VTAMLST.

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.

In an APPN network, the names NETA.LUA and NETB.LUA, for example, are always treated as distinct entities. An obvious benefit of this independent name administration is that an end node can dial into separate APPN networks without being limited by name conflicts.

1.3 Addresses

The addresses used in SNA are either network addresses or local addresses. Network addresses uniquely identify a resource throughout the 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 *temporarily* assigned. They are assigned at session initiation, and released when the session ends. The information stored in the node is contained in a session connector. It is also kept during the life of the session.

The session identifier is associated with

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

Figure 10 shows a session between LU1 in the originating end node and LU2 in the destination end node. There are two intermediate nodes in between, with a total of three "hops" or session stages. This session can be thought of as a sequence of three session stages. On each session stage, each pair of adjacent nodes assigns a distinct session identifier.

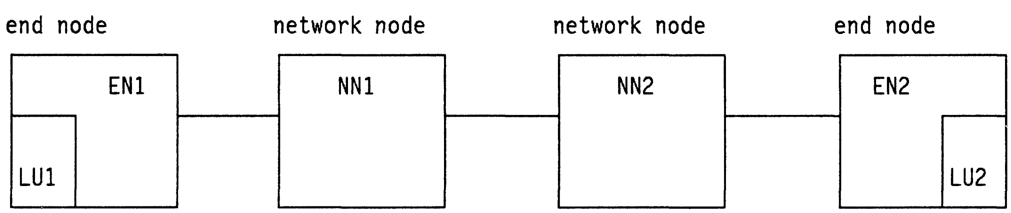

Figure 10. Session with Several Session Stages

As you can see, the session identifiers will be different on different session stages. They are definitely not identical throughout the network. That 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. Details may be found in "Address Space Manager" on page 21.

When it is necessary to reference a session by a network-wide identifier, the fully qualified procedure correlation ID(FQPCID) is used. It is described in "Generate FQPCID" on page 107.

Comparison to subarea network: The subarea network uses local addresses *and* network addresses. The local addresses are used between the peripheral node and the boundary functions of VTAM and NCP. The network addresses are used between subareas. For NCP, the addresses (both network and local) are assigned during NCP generation. For VTAM, the addresses are assigned during major node activation. In the subarea network, the addresses are assigned "forever"; that means they survive a session termination.

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 that node. Consequently, all APPN networks are multi-domain networks.

Though all APPN nodes are peers and do not rely on other nodes to control their resources, 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 this node's local resources *and* the resources of those nodes that use the network node's services. Thus, the domains of the end nodes and LEN end nodes are included in the domains of their respective node servers.

Comparison to subarea network: Within the subarea network, a domain is the portion of the network managed by the system services control point (SSCP) in a T5 node. Apart from providing services for the resources in its domain, the SSCP has control over its own resources *and* all its subordinate nodes and their resources. For example, the SSCP in VTAM activates and deactivates the lines and PUs of an NCP.

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. Throughout this document we will use the term **LEN end node**. With the APPN extensions to SNA, two other types of T2.1 nodes are introduced. They are the APPN end node and the APPN network 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 destination LUs that are known on one of the network nodes in the network.

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. As soon as 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. For example, an APPN network node that is taken out of service will be 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 sessions through its node from one adjacent node to another adjacent node. This function is referred to as intermediate session routing.

1.5.2 APPN End Node

An APPN end node provides limited directory and routing services for LUs local to this node. The APPN end node can select an APPN network node and request this network node to be its network node server. If accepted by the network node, the APPN end node automatically registers its local resources with the network node server. This enables the network node server to intercept search requests for resources that are located on the APPN end node and pass the request on to the APPN end node.

The APPN end node automatically forwards all session initiation requests for resources that are unknown to the APPN end node. The APPN network node will use its distributed directory and routing facilities to locate the LU and calculate the route starting at the APPN end node towards the destination LU.

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

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 LUs, either controlled by the LEN end node or on an adjacent node, are registered on the LEN end node. LUs on adjacent nodes need to be manually registered with the control point name of that adjacent node. If an LU on a LEN end node initiates a session with an LU on an adjacent node, then the LEN end node obtains the registered control point name of the adjacent node and sends a session activation (BIND) request to that adjacent node.

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

The resources in a domain of the SNA subarea network are controlled through an 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 the S/370 type of mainframe that implemented VTAM and its attached 3745 that implemented 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 referred to as 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 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.

However, the subarea terminology used for the T2.1 nodes differs from the APPN terminology. The T2.1 node is always referred to as the *peripheral node* that has implemented the LEN end node functions. The subarea node to which the T2.1 node connects is referred to as the *boundary node*.

From the T2.1 node perspective, the names for these nodes are in accordance with the APPN naming conventions; that is, the T2.1 node could be either a LEN end node, APPN end node, or APPN network node, while the subarea node to which it connects is a LEN end node. Please note that subarea nodes do not support CP-CP sessions, nor is there a subarea implementation of APPN end nodes or APPN network nodes.

For example, in Figure 11 on page 11, an APPN configuration is depicted that consists of three nodes. The configuration includes the subarea complex

(ES/3090-200 with a 3745) that implemented the LEN end node function, an AS/400 that implemented the network node function, and a PS/2 that implemented the APPN end node function.

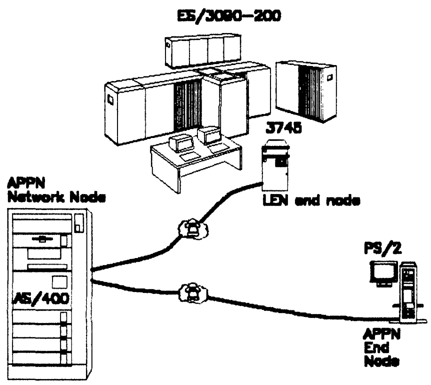

Figure 11. APPN View of Node Types

Figure 12 depicts the same configuration, but now from the perspective of the subarea network. The 3745 that was previously referred to as a LEN end node is now called the boundary node. The AS/400 that was previously referred to as an APPN network node is now called the peripheral node. Please note further that from a subarea perspective the PS/2 is no longer part of the configuration. The subarea assumes that all LUs, located on the PS/2, are really located on the AS/400. The AS/400 in this case may act as the intermediate node and route the session request and its session data to the PS/2.

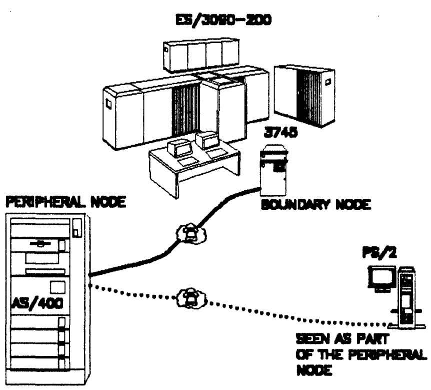

Figure 12. Subarea View of Node Types

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 a VTAM host and a NCP represent two nodes, but when connected to an APPN node, it will appear as **one** logical T2.1 node.

Virtual Routing Node

The virtual routing node is a representation of a node's attachment to a shared access transport facility (SATF) such as a token-ring. For more information see "Connection Networks Using a Shared-Access Transport Facility" on page 40.

Border Node

The border node represents a network node that provides multi-network capabilities. For more information see "Border Nodes" on page 167.

1.6 APPN Network Configuration used in this Document

The examples that are given in this manual will use the configuration as depicted in Figure 13 on page 13.
The configuration consists of two APPN end nodes and four APPN network nodes. The links between the nodes are called TG(1). The configuration for each node is:

- APPN end node "ENA"

  "ENA" connects to APPN network nodes "NNA" and "NNC" but will use APPN network node "NNA" as its network node server. The local LUs controlled by "ENA" are LU1, LU2, and LU3.

- APPN network node "NNA"

  "NNA" connects to network node "NNB" and provides network node services for APPN end node "ENA". The local LUs controlled by "NNA" are LUA and LUB.

- APPN network node "NNB"

  "NNB" connects to network nodes "NNA", "NNC", and "NND". The local LUs controlled by "NNB" are LU4 and LU5.

- APPN network node "NNC"

  "NNC" connects to network node "NNB" and to APPN end nodes "ENA" and "ENB". The local LUs controlled by "NNC" are LU6 and LUE.

- APPN network node "NND"

  "NND" connects to network node "NNB" and provides network node services for APPN end node "ENB". The local LUs controlled by "NND" are LUC and LUD.

- APPN end node "ENB"

  "ENB" connects to APPN network nodes "NNC" and "NND" but will use APPN network node "NND" as its network node server. The local LUs controlled by "ENB" are LU7, LU8, and LU9.

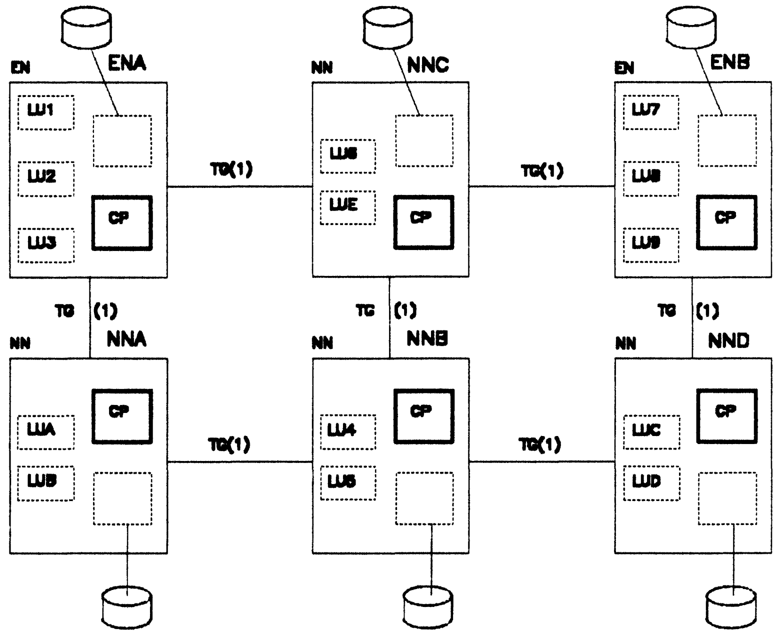

Figure 13. APPN Network Configuration Used in This Document

Chapter 2. Node Structure

This chapter describes the node structure for T2.1 nodes. A T2.1 node allows peer-to-peer communication between adjacent T2.1 nodes, and provides the physical and session-level connectivity for support of LU 6.2.

The APPN architecture has been designed to work very closely with IBM's LU 6.2 (also known as Advanced-Program-to-Program Communication or APPC). APPC provides a standard set of functions useable for all kinds of connectivity requirements, including program-to-program, program-to-device, and device-to-device communication. APPN both uses and enables APPC.

APPN's system of network control and configuration uses LU 6.2 sessions to exchange messages and network information between nodes. APPN enables APPC because APPN protocols accommodate more fully the notion of communication between peer devices from which APPC is derived, particularly in their application to distributed processing environments.

The T2.1 node distinguishes three major components:

- Path Control Network
- The Network Accessible Unit
- The Node Operator Facility.

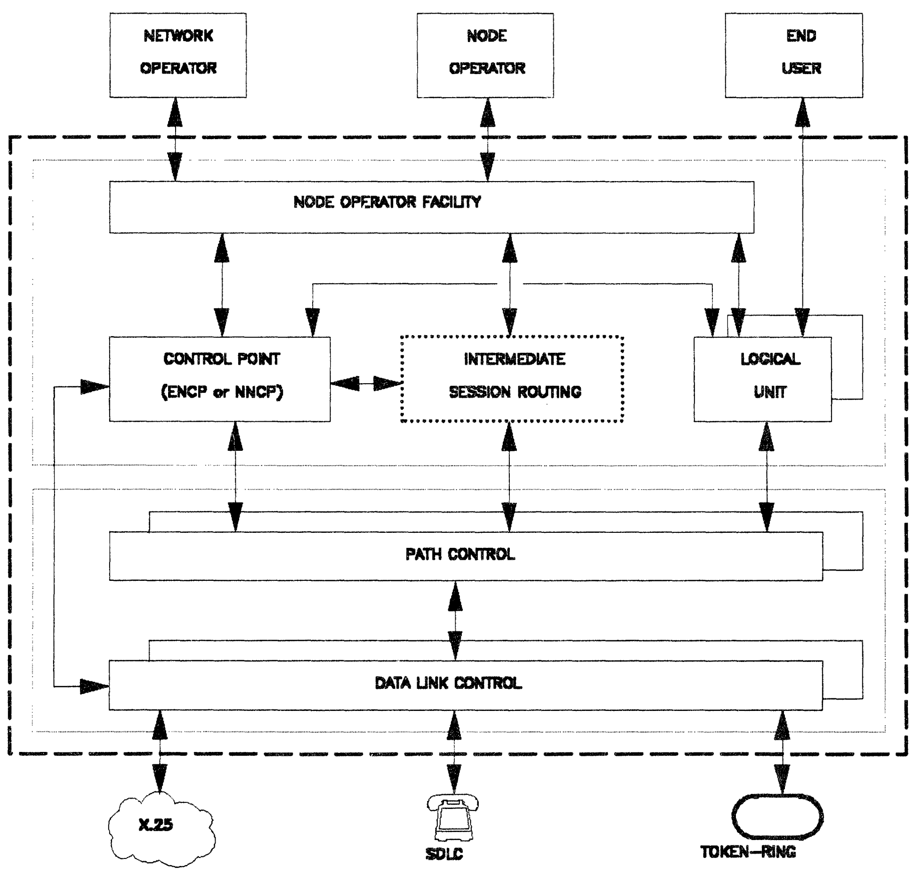

Figure 14. Components of an APPN Node

2.1 Path Control Network

The path control network addresses the three lower layers of the SNA architecture, that is:

- Physical layer

  The physical layer is responsible for the physical and electrical interface to the hardware components that connect to private lines or public networks.

- Data link control layer

  DLC is responsible for reliable delivery of message units between adjacent link stations. DLC assists in connection establishment with an adjacent link station and in primary/secondary role negotiation. DLC provides protocols for SDLC, X.25, token-ring, and S/370 channel connections.

- Path control layer

  Path control is responsible for delivering message units between session layer components in the same or different nodes. Session layer components consist of half sessions in logical units and control points, as well as in session connectors residing in intermediate network nodes or boundary nodes, that are on the path between the endpoints of a session.

  Each session is assigned one of the four transmission priorities, that is, network, high, medium, or low transmission priority. This transmission priority is reflected in the outgoing message units of the session.

  Path control maintains four queues for outgoing messages, one for each transmission priority. It delivers these message units from the queues in a priority order that is from high to low.

2.2 Network Accessible Unit

The network accessible unit addresses the upper four layers of the SNA architecture, that is:

- Transaction services layer
- Presentation services layer
- Data flow control layer
- Transmission control layer.

For more information on SNA layered structure see *Systems Network Architecture Technical Overview*. The network accessible unit for T2.1 nodes distinguishes three types of network accessible units:

- Control point
- Intermediate session routing
- Logical unit.

2.2.1 Control Point

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 assists in finding the partner 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

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

- Topology and routing services

  In LEN end nodes and APPN end nodes, topology and routing services collect information on links and adjacent nodes. In APPN network nodes, topology and routing services additionally collect and exchange 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

  The directory services component is responsible for locating network resources throughout the APPN network. On LEN end nodes, directory services only searches network resources in its local database. On APPN end nodes, directory services searches network resources in its local database first, but, if the network resource can not be located, it uses the distributed search facilities provided by the APPN network node with which it has established CP-CP sessions.

  In order to locate these 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 APPN end node for which it provides network node services, directory services at the APPN network node temporarily registers APPN end node's resources in its local directory database.

- Session services

  The session services component of CP 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

  The address space manager manages the session addresses that are used by path control to identify each individual session on a particular link. The optional features of address space manager are BIND reassembly and adaptive BIND pacing.

- Management services

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

2.2.2 Intermediate Session Routing

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. Most of these sessions pass through network nodes that are in between these two endpoints. As these intermediate nodes do not control one of the endpoints, LU services at these nodes cannot be invoked. In these intermediate nodes it is the responsibility of the intermediate session routing (ISR), to route the session data through the node.

The ISR component distinguishes two functions. They are:

- Session connector manager

  The session connector manager processes the BIND and UNBIND requests and responses for the intermediate node. It creates the session connector when the BIND is received and destroys the session connector when the UNBIND is received.

- Session connector

  The session connector is built by the session connector manager on the intermediate node when the session is established. Its function is to route the session traffic through the intermediate node.

Session Connector Manager

When the address space manager receives a BIND destined for an LU on another node, it will pass the BIND to the session connector manager, hereafter referred to as SCM. The SCM performs the following functions:

- Interfaces with session services to obtain the path control element address of the transmission group(TG) in the direction of the destination LU. In our example in Figure 15 on page 19, it will be the TG towards end node "ENB".
- Interfaces with address space manager to obtain an LFSID for the TG in the direction of the destination LU.
- Negotiates the maximum RU size allowed for the intermediate node. The BIND contains the maximum send RU size for both the primary as well as the secondary LU. However, the intermediate node may have defined a maximum RU size allowed for that node. If the maximum RU size allowed for the intermediate node is less than the RU size defined in the BIND, then the SCM sets the exceeding RU size in the BIND to the maximum allowed for the node. For example the 3174 network node allows for a maximum RU size of 8KB. If the BIND defines an RU size (either primary or secondary) above 8KB, then the 3174 network node will change the BIND and set the exceeding RU size to 8KB.
- Negotiates the type of session pacing and window sizes used. The SCM always sets the adaptive pacing indicator in the BIND. The window sizes set by the primary logical unit in the BIND will not be changed by the SCM unless an installation has defined specific window sizes for that intermediate node. In that case the SCM will set window sizes in the BIND accordingly.
- Builds the session connector instance that will contain, amongst others, the fully qualified procedure correlation ID(FQPCID) of the session, the local-form session identifier(LFSID) used by the session on the incoming TG, and the LFSID that will be used on the outgoing TG.

For example, in Figure 15 on page 19, LU1 at "ENA" sends a BIND request to LU7 at "ENB". As node "ENB" is not adjacent to "ENA", information about the route of the BIND through the network is attached to the BIND, and the BIND is sent to adjacent node "NNC". The address space manager at "NNC" receives the BIND and passes the BIND request to the ISR component as the destination LU (LU7) is not located on "NNC".

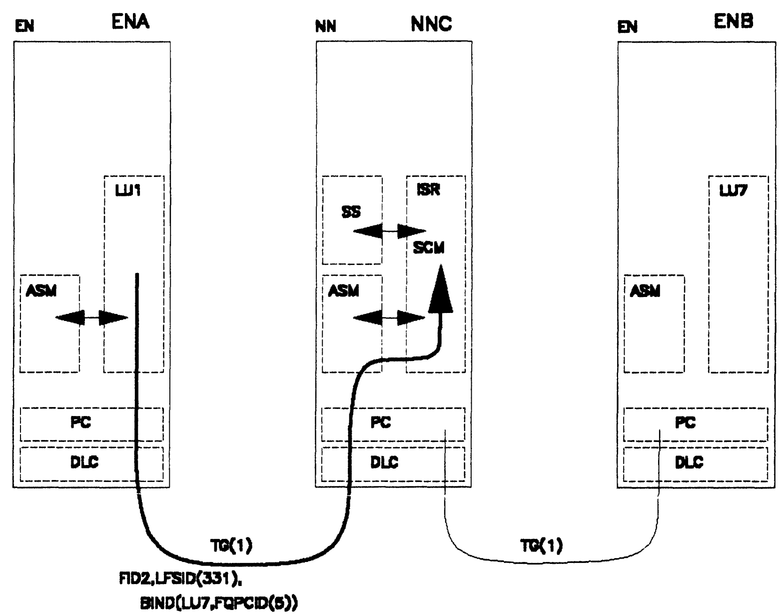

Figure 15. BIND Request on Intermediate Node

After the SCM has changed the BIND according to its installation-defined parameters, it sends the BIND to path control, which will forward the BIND to the next node.

Session Connector

The SCM builds a session connector instance for each session passing through it. The session connector is responsible for:

- Reassembling the RU if it was segmented by the sending node. The sending node segments the RU if it does not fit into the maximum BTU size allowed for the TG.
- Maintaining the session pacing counts for the session passing through it, for both the receiving as well as sending TG. It uses pacing messages to inform the sender to send the next set of messages or to temporarily stop sending messages. There are two kinds of pacing techniques, these are the fixed and the adaptive session-level pacing. With fixed session-level pacing the number of messages sent in one window are preset at BIND time for the duration of the session. With adaptive session-level pacing the receiver can dynamically adapt the number of messages sent in one window depending

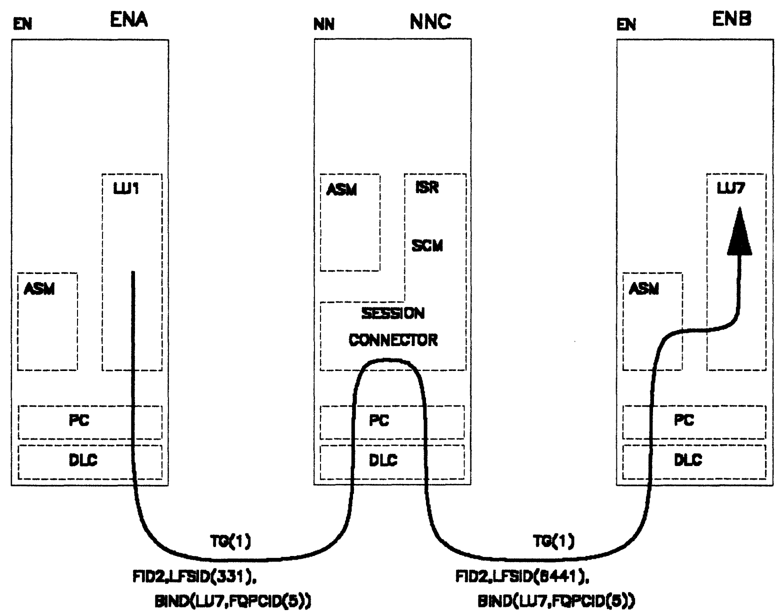

Figure 16. BIND Request on Destination Node

on the level of congestion at its node. Thus, each time the receiver responds to a pacing request from the sender, it informs the sender about the window size to be used for the next set of messages. This window size could be either lower, equal to, or higher than the previous window size. For more information on pacing see *Systems Network Architecture Technical Overview*.

- Routing the session data for the intermediate session to the associated TG. The session connector derives the TG and corresponding LFSID from the session connector instance created by SCM.

2.2.3 Logical Unit

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.

An LU can be classified as SSCP-dependent or SSCP-independent, depending on the LU-LU session initiation protocols used. The SSCP-dependent LU, hereafter referred to as the dependent LU, requires a session with its controlling SSCP. Across this session the dependent LU sends a session initiation request to the SSCP. The SSCP assists in session initiation for the LU by locating the destination LU (DLU) and requesting the PLU to activate (BIND) the session. Dependent LUs reside in T2.0 as well as T2.1 nodes.

An SSCP-independent LU, hereafter referred to as independent LU, sends a session activation (BIND) request directly to its partner LU. The independent LU does not request the assistance of the SSCP, thus the SSCP-LU session does not exist. The independent LU only resides in T2.1 nodes.

2.3 Address Space Manager

The functions of the address space manager are:

- Address space manager initialization
- Address space generation
- Local-form session identifier (LFSID) assignment
- BIND reassembly
- Adaptive BIND pacing.

Address Space Manager Initialization

The address space manager is created by the node operator facility at node initialization time. The node operator facility initializes the node when the "START_NODE" command [2] is issued by the node operator. As part of the initialization process the node operator facility creates the address space manager and 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

In Figure 17 on page 22, the node operator issues the "START_NODE" command. On receipt of the command the node operator facility initializes node "NNB". One of the control point components that is initialized by the node operator facility is the address space manager. The node operator facility derives the following parameters from the "START_NODE" command and passes these parameters to address space manager:

- The control point name is "NNB"
- The network ID is "ITSC"
- Support for BIND reassembly

Address Space Generation

Configuration services is responsible for activating transmission groups with adjacent nodes. As part of the transmission group activation process configuration services on both nodes negotiates the role (primary or secondary link station) for each node on the transmission group. For more information see "Link Station Role Negotiation" on page 35.

When the transmission group (TG) is activated and the link station roles are set, configuration services on both nodes notify the address space manager about the TG and the link station role for that node. The address space manager generates an address space for that TG, which consists of 131 072 local-form session identifiers, hereafter referred to as LFSID. The address space manager

[2] The commands referenced in this document are the commands as defined by the architecture, thus the command names and structures used in an implementation may differ.

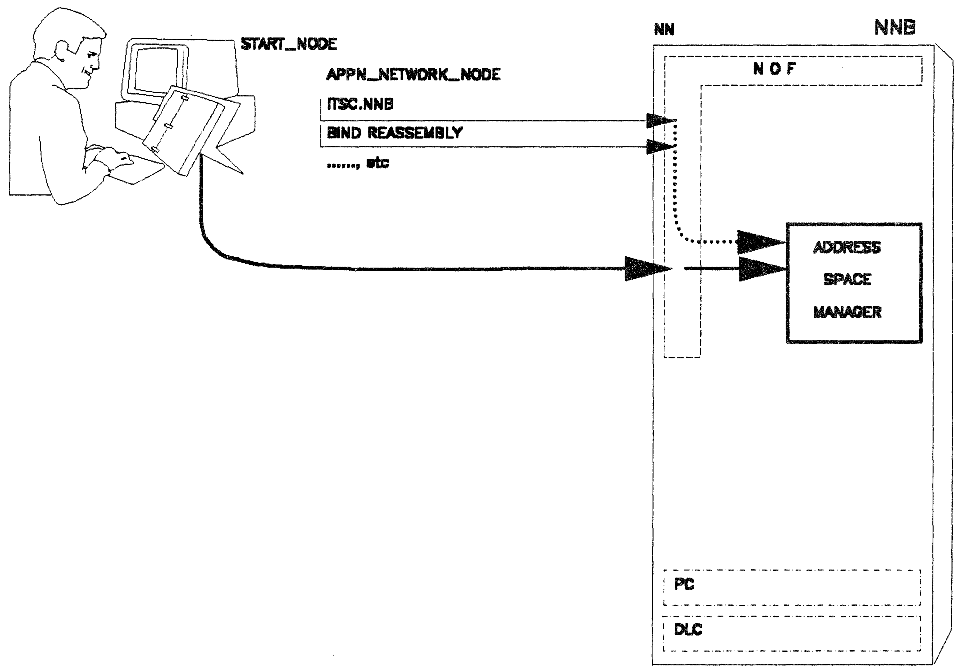

Figure 17. Address Space Manager Initialization

in the node that sends the BIND request over the TG assigns the LFSID for the session on that TG.

To prevent the address space manager in the adjacent node from selecting the same LFSID for another session on the same TG, the address space is divided into two partitions providing 65 536 LFSIDs per address space manager. The address space manager located on the node that provides the primary link station role on the TG uses addresses 0 - 65 536 and the address space manager on the node that provides the secondary link station role on the TG uses addresses 65 537 - 131 072.

The example in Figure 18 has two TGs between nodes A and B. Node A is the primary link station for TG(1) and the secondary link station for TG(2).

Local-Form Session Identifier Assignment

The transmission group between adjacent nodes can be used by multiple sessions. In order to distinguish the messages for a particular session, path control requires a unique session identifier in the messages.

The messages are enveloped in an SNA header that is known as the transmission header (TH) format identifier 2, hereafter referred to as FID2. In subarea networking FID2 is used between boundary and peripheral nodes. The FID2 contains two address fields, each one byte in length. For sessions involving the SSCP or dependent LUs, one field (SIDH) is used to identify the sender at the

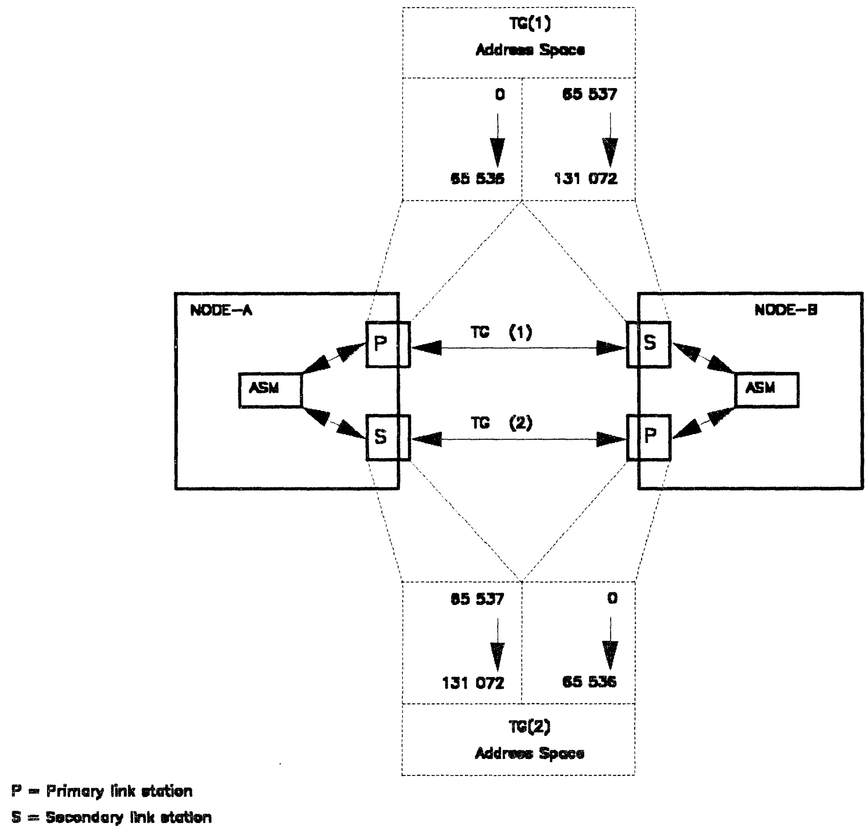

Figure 18. TG Address Space

boundary node and the other field (SIDL) is used to identify the receiver at the peripheral node. The maximum number of senders in the boundary node is two, that is, the SSCP (System Services Control Point) for the SSCP-LU and SSCP-PU session, and the PLU for the LU-LU session. The maximum number of receivers at the peripheral node is 256.

T2.1 nodes support independent LUs which do not require the SSCP-LU session. Thus, for independent LUs, the two address fields are combined into one field with a length of two bytes (16 bits). An additional bit in the FID2, that was previously reserved, has been added to this address structure. This bit is the ODAI (Origin address field Destination address field Assignor Indicator). Together with the 16 bits of the address fields, the OADI provides for 131 072 unique session addresses, collectively called the address space.

As mentioned in "Address Space Generation" on page 21, the address space is split into two partitions. The ODAI bit is used to identify each partition. This bit will be 0 (zero) for addresses assigned by the node containing the primary link station and 1 (one) for addresses assigned by the node containing the secondary link station.

On the same link to the boundary node, T2.1 nodes can support both dependent and independent LUs. This requires that the current address assignments for dependent LUs remain reserved. Figure 19 on page 24 depicts the address space assignment for the T2.1 nodes taking into account both dependent and independent LUs.

SIDH, SIDL	USAGE
X'00', X'00'	used for SSCP-PU sessions or not at all
X'00', X'01' ↓ X'00', X'FF'	used for SSCP-LU sessions or not at all
X'01', X'00'	used for BIND flow control
X'01', X'01' ↓ X'01', X'FF'	if dependent LUs can exist in the node, then used for dependent LU-LU sessions otherwise used for CP-CP and independent LU-LU sessions
X'02', X'00' ↓ X'FE', X'FF'	used for CP-CP and independent LU-LU sessions
X'FF', X'00' ↓ X'FF', X'FF'	reserved

Figure 19. LFSID Address Space Assignment

At session activation (BIND) time a session address is assigned by the address space manager from the address space associated with the link carrying the session traffic. Path control for that link will insert the session address in the envelope header of all the message units for that session. The session address will only be used by path control in adjacent nodes. Thus, if a session crosses more than one link, a new session address will be assigned for each link that the session traverses.

BIND Reassembly

When configuration services 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 the receiver at the adjacent node, address space manager, 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.

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

Adaptive BIND Pacing

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 handle the responses to 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*. Adaptive BIND pacing is similar to adaptive session-level pacing introduced with the LEN announcement in 1986.

Adaptive BIND pacing initially sets the maximum number of BINDs that a node may send across a particular transmission group. If the maximum number of BINDs are sent the node has to wait for a pacing response from the receiving node. The maximum number is also referred to as the window size. The pacing response from the receiving node could either increase or decrease this window for the next set of BINDs. It could also set the window size to 0 which means that no new BINDs may be sent until further notice.

2.4 Node Operator Facility

The node operator facility provides an interface between the node operator and components of the control point. For example, the node operator may activate and de-activate link stations, define and delete LUs, query control point about links and other node resources, and receive diagnostic information.

As depicted in Figure 20 on page 26 the node operator could be either:

- A human operator

  A human operator communicates with the dialog manager. The dialog manager converts the information received from the human operator into node operator facility commands and forwards these commands to the node operator facility. The node operator facility processes these commands and return the results to the dialog manager which will show the results to the human operator.

- A command file

  A command file is processed by an installation-specific file interpreter that reads the command file, converts the command file into one or more node operator facility commands, and forwards these commands to the node operator facility. The node operator facility processes these commands and return the results to the file interpreter. The file interpreter may either discard the results or log the results with the log manager.

- A transaction program

  A transaction program directly communicates with the node operator facility. The transaction program can be used to receive commands from remote locations. The transaction program forwards the commands to the node operator facility for execution. The command results that are returned to the transaction program may be propagated to the remote location. The interface between the two transaction programs will be implementation dependent, thus, the transaction program may have to convert the commands to node operator facility commands.

The node operator facility logs all commands and command results received from the node operator with the log manager. Unsolicited data received from the components will also be logged with the log manager.

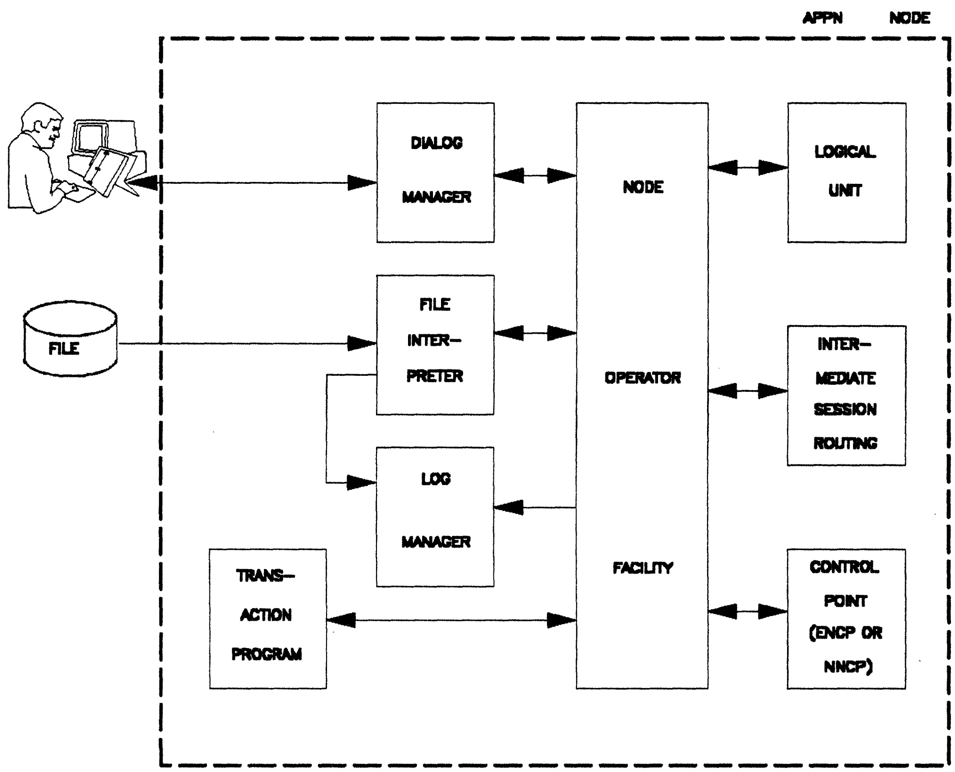

Figure 20. Node Operator Facility Interaction

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 when the node operator issues the "START_NODE" command 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 nodes resources should be registered with its network node server (APPN end node only)
- Whether mapping of mode name to class of service and transmission priority is supported (COS/TPF)
- The name of management services log file
- The name of the topology database file
- The name of the class of service (COS) definitions file

- List of resource types (only LU resource type currently supported) this node can be searched for by its network node server.

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

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 the 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", *Systems Network Architecture Type 2.1 Node Reference.*

Chapter 3. 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, or 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 deactivating 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)
- Non-activation XID exchange
- Link deactivation.

These functions are described in the following paragraphs. A big part of the link activation process works through negotiation. An example flow diagram for a negotiation process is included in this chapter. It is taken from *Systems Network Architecture Type 2.1 Node Reference*, which contains several examples and additional details on configuration services.

3.1 Initialization

The node operator facility (NOF) plays an important part in initializing configuration services and defining, starting, stopping, and querying 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. (A negotiable link station can be primary or secondary. The actual role is determined during link activation.)
- Whether or not parallel TGs are supported.

3.2 Link and Transmission Group

A link represents a connection to an adjacent node and includes the data link control (DLC), the port, and the link station.

Currently, a transmission group (TG) corresponds to a connection with a single link. The APPN architecture does not provide TGs consisting of multiple links. So, with APPN, the terms "link" and "TG" mean the same thing. 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.

Data Link Control

DLC is the process responsible for performing communications over a link using a specific data link protocol, for example, SDLC, token-ring, X.25, or CDLC (channel data link control). Each DLC process may manage one or more ports.

DLCs are defined by the node operator facility. The definition contains information such as DLC type and options supported by the DLC.

Port

A port represents the physical connection to the link hardware. The specific component it represents is sometimes referred as an adapter. Link stations are associated with a port. That includes dynamic link stations to connection networks.

Ports are defined by the node operator facility. Figure 21 on page 31 shows a port definition as part of the configuration database. The following types of information have to be defined:

- Associated DLC, such as SDLC in Figure 21.
- Port-specific information, like link station activation limits and time-out values.
- Information that is common to all link stations associated with the port; for example, if it is switched or non-switched. Some of the information is not needed for link activation, but for route calculation by route selection services. (Cost per byte and capacity are examples; "End Node with Class of Service" on page 61 will explain how they are used.)
- Information on any connection network (discussed in "Connection Networks Using a Shared-Access Transport Facility" on page 40).

Link Station

A link station is a combination of hardware and software which makes it possible to control a link. Its characteristics can either be defined through the node operator facility or established during link activation with the help of the exchange of data between adjacent link stations (XID exchange).

The following information is required for a local link station:

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

Figure 21 illustrates the components involved in link definition.

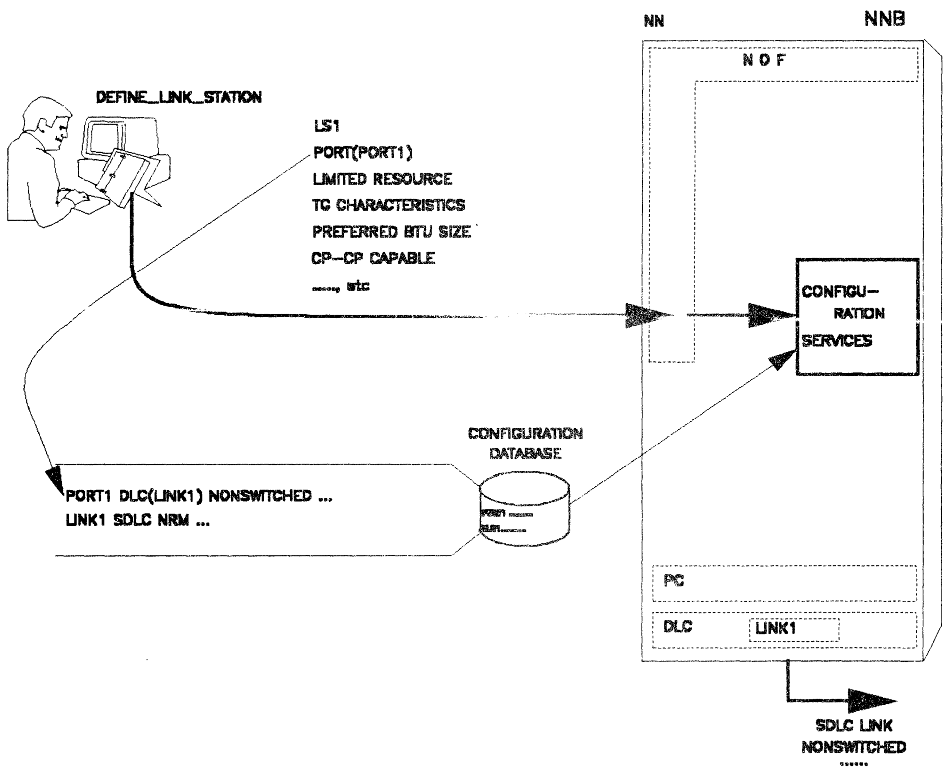

Figure 21. The Link Station Is Being Defined

Nonswitched point-to-point connections can use negotiable link stations. They can acquire information during XID3 exchange. That reduces the requirement for system definitions. The associated link station in the node on the other side of the link is called the adjacent link station (ALS). For a point-to-point link, there is only one ALS.

Nonswitched multipoint link connections have the following characteristics:

- Only one primary link station can exist on the link connection.
- One or more secondary link stations can exist on the link connection. Thus, a primary link station on a multipoint link can have more than one ALS.
- No negotiable link stations may exist on the link connection.
- Secondary station addresses must be defined.
- Apart from the null XID, APPN defines XID3 only.

Switched link connections are always point-to-point; that is, only one adjacent link station per port is allowed. Either side may dial out to establish the switched connection.

If a switched port is defined by the node operator facility as supporting auto-activation, it can be activated from session services, when an LU requests a session. Otherwise, the operator has to activate the link.

If a switched port is defined as a limited resource, it can be automatically deactivated when no sessions are active over it. The purpose is to save the cost of keeping the switched connection active when it is not needed.

Token-ring connections:
Link stations on a token-ring are always defined as negotiable. Thus, any node on the token-ring may initiate a link activation XID exchange with any other node on the token-ring.

Link stations accessed through a token-ring may or may not be defined as limited resources. If a token-ring port is defined as a limited resource, it can be deactivated automatically when no sessions are active over it.

Dynamic link stations are associated with connection networks. A connection network can be thought of as a representation of a token-ring. (Further information will be given in "Connection Networks Using a Shared-Access Transport Facility" on page 40). The dynamic link stations are not defined or activated by the node operator facility. They have a set of default parameters associated with the port through which the link station is accessed. These default parameters are defined with the port. The activation of dynamic link stations is triggered either by session services, when this node requests a session through the port (into the connection network) or by an adjacent node, when it sends a request for connection.

All dynamic link stations are limited resources. That means, when no sessions are using them, their deactivation can be initiated by session services. CP-CP sessions are therefore not supported on connections using dynamic link stations.

All dynamic link stations support auto-activation. They cannot be activated by the operator.

3.3 Link Activation

Link activation may be initiated locally or by the adjacent node. If locally, it can be through an operator command, or follow a session initiation request. Link activation may also be started in response to the link being activated by the adjacent node.

When a link is being activated, the two endpoint nodes need to compare some information about each other in order to determine certain parameters about the link. This information is passed during the XID exchange sequence initiated by the node activating the link.

Some of the information acquired at this time is:

- Whether or not the other node is active.
- What roles the ALS can have and if they are compatible with local definitions. For example, two primary non-negotiable stations are not compatible.
- TG number.

For link activation, the architecture is rather flexible. It allows several options and variations. Link activation involves three phases, two of which are optional: connect, prenegotiation, and contact.

3.3.1 The Connect Phase

During this phase, dialing and answering takes place. Modems exchange training sequences. This equalization must take place before a modem can give permission to its link station to begin transmission.

The connect phase is optional and highly DLC dependent. Information on details of this phase should be obtained from the publications describing the DLC. The "Related Publications" on page xv gives references.

3.3.2 The Prenegotiation XID Exchange

The prenegotiation phase is optional as well. If it is used, it starts with XID polling. This polling can go on until the other side is ready to respond. APPN nodes use a null XID poll to determine if the adjacent station is active. If the adjacent node received a null XID, it returns an XID3 with the exchange state indicator set to "prenegotiation". (Some implementations do not send the null XID. Instead, they send an XID3 right away with the exchange state indicator set to "prenegotiation".)

The purpose of prenegotiation is for a node to be able to verify the identity of the adjacent node, before committing itself to the values to be sent in the negotiation-proceeding XID3 (in the contact phase).

3.3.3 The Contact Phase with XID3 Negotiation

The contact phase consists of the "negotiation-proceeding" XID3 exchange and the mode setting sequence. A detailed description of the XID3 is given in *Systems Network Architecture Formats.*

The purpose of the "negotiation-proceeding" XID exchange is to reduce the requirement for system definition about the adjacent node (ease of use). 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 rules governing the negotiation process are summarized in the remaining topics of this section.

During the negotiation, the exchange state indicator is set to "negotiation-proceeding". Those fields in the XID3 that are not being determined remain stable. That is true in particular for those XID3 fields (and associated control vectors) which just communicate node properties to the adjacent node; examples of these fields are:

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

Now the primary link station is ready to send the appropriate mode setting command, such as SNRM (set normal response mode) or SABM (set asynchronous balanced mode). The contact phase is completed, when one station sends the mode setting command to the other and receives the Unnumbered Acknowledgement (UA) in reply.

Transmission Group Number Negotiation

During the contact phase, the partners determine the transmission group (TG) number that will be used as part of the link representation. The TG number must be unique between a pair of CPs. In other words, two TGs between the same pair of nodes cannot have the same TG number. This allows a TG to be identified by a pair of CP names and a TG number.

When the implementation does not support parallel TGs between two nodes, any number from 0 to 239 is allowed.

When the implementation supports parallel TGs between two nodes, any number from 1 to 239 is allowed. (0 means "any TG number".)

The numbers up to 20 are set aside for TGs that have been predefined between two nodes. In future, the numbers 240 to 255 may be used for special purposes.

The list below summarizes the rules to determine 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 multiple TGs between them are not supported, then the TG number is set to 0.
- If both nodes send TG numbers of 0 and multiple 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.

Link Station Role Negotiation

Whether a link station is primary or secondary may be predefined or negotiable. The following rules apply:

- If one side is defined as primary and the other one as secondary, the definitions will be accepted.
- If both sides are defined as primary, or both sides are defined as secondary, the XID exchange will fail with a sense code.
- If one side is defined as primary or secondary and the other side as negotiable, the first definition will be accepted and the negotiable side will take up the remaining role.
- If both sides are defined as negotiable, the link station roles are resolved based on the node identification field, consisting of block number and ID number. Whichever side has the node identification field with the higher value will become primary.

  The negotiation cannot be based on the CP name, because back-level LEN end nodes do not append the control vector that contains the CP name.

  Some nodes use a product-specific block number and an ID number unique in the network. These values will be used.

  Other products set the block number to X'000' or X'FFF' and do not use the ID number. In this case, a random number will be created in this field. A random number will also be created, if both nodes have identical node identification fields.

As a by-product of the link station role negotiation the value of the ODAI bit in each node is determined. The ODAI bit (origin destination assignor indicator) is part of the LFSID (local form session identifier). For more information see "Address Space Manager" on page 21.

For token-ring connections, the difference between primary or secondary link station is irrelevant, but the ODAI value is still important. Figure 22 is just one example of a negotiation process. It applies for token-ring. For SDLC it would be slightly different.

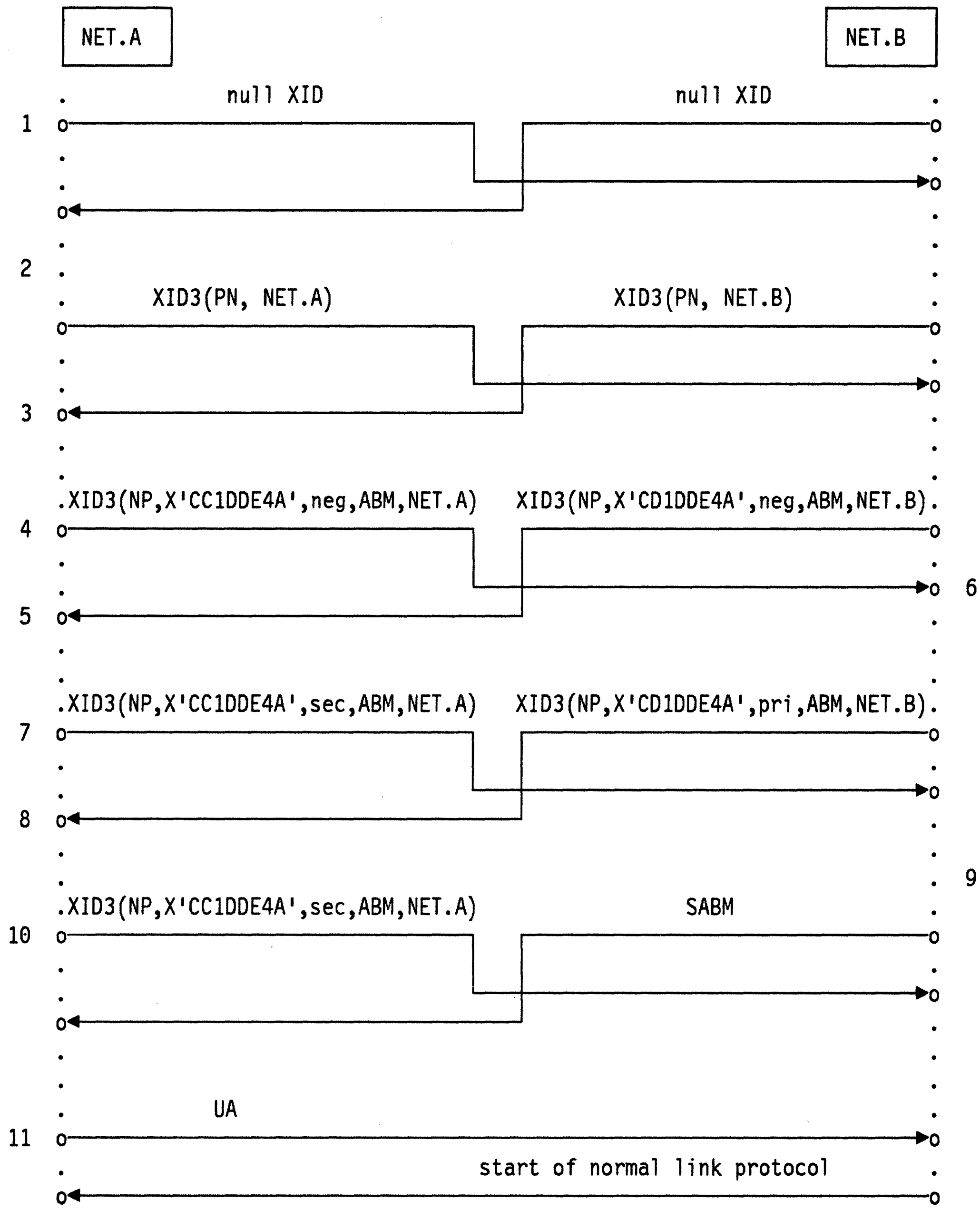

Figure 22. Link Station Role Negotiation between Nodes Containing Negotiable Link Stations.

Legend:
NET.A, NET.B network qualified CP names of adjacent nodes
PN, NP prenegotiation, negotiation proceeding
neg, pri, sec link station role: negotiable, primary, secondary
X'CC1DDE4A' value in node identification field of NET.A
X'CD1DDE4A' value in node identification field of NET.B
(S)ABM (set) asynchronous balanced mode
UA unnumbered acknowledgement

Notes referring to Figure 22:

1. Both nodes contain negotiable link stations. Since negotiable link stations can poll the adjacent node by sending XID commands, each node begins polling the other with null XIDs.
2. Each node receives the other's null XID, views it as a response to the null XID command, and sends out a prenegotiation XID command.
3. Each received prenegotiation XID3 is viewed as a response to the receiver's prenegotiation XID3 command. Each node carries out validation of the adjacent node's identity. The node can decide if it wants to accept the other node as a partner over this link.
4. Each node has sent and received a prenegotiation XID3 and can begin, therefore, to send negotiation-proceeding XID3s. Each node sends a negotiation-proceeding XID3 reflecting the defined link station role. Note that each negotiation-proceeding XID3 contains the 32-bit node identification field. The values that may be sent in this field and their significance are described in the paragraph on "Link Station Role Negotiation" on page 35.
5. On receipt of NET.B's negotiation-proceeding XID3, NET.A notes that both it and NET.B have negotiable link stations. Link station role negotiation must be possible both for nodes that do and do not send the network name control vector. Therefore, role determination is based on the node identification field sent in XID3s. NET.A compares the values in the node identification field of sent and received XID3s. An unsigned binary comparison of X'CC1DDE4A' and X'CD1DDE4A' has the node identification of NET.B greater than that of NET.A. NET.A's negotiable link station consequently makes the transition to being a secondary link station. Even though NET.A will nominally contain a secondary link station, it continues to send out XID commands, to prevent deadlock situations.
6. On receipt of NET.A's negotiation-proceeding XID3, NET.B notes that both it and NET.A have negotiable link stations. It performs the same comparison that NET.A did in step 5 and concludes that its link station has to make a transition to being a primary link station.
7. The next negotiation-proceeding XID3 sent by each node shows the results of the comparison as each declares a complementary link station role.
8. NET.A receives the primary XID3 from NET.B.
9. NET.B receives the secondary XID3 from NET.A. Since NET.B has now sent a primary XID3 and received a secondary XID3, it may initiate the mode-setting sequence if TG number negotiation is done as well. In this example, it is assumed to be the case. The next command sent by NET.B is a mode-setting command. The effect of sending this command is to ignore any subsequent XID commands that NET.B may receive during link activation from the adjacent node.
10. NET.A. continues to send XID3s. In this case, this XID3 is sent as a command at the DLC level. In other words, the assumed link station role has not changed ABM protocols at the link level.
11. When NET.A receives NET.B's mode-setting command, it cancels the requirement that its outstanding commands (like the one sent in step 10) be satisfied and prepares to respond with a UA, since all required XID negotiation has been completed.

Determine Maximum BTU Size

Each link station determines its own maximum send 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 ALS.

If the partner does not support BIND reassembly, the value must be at least 265 (or 521 if the partner is an APPN network node). Otherwise the XID will be rejected.

Disposition toward CP-CP Sessions

Whether CP-CP sessions are established or not across a particular link, depends very much on the information about the node type, the CP capabilities, and CP requirements that are exchanged in the XID3.

The rules can be summarized as follows:

- LEN end nodes (including T4/T5 nodes acting as boundary nodes) do not support CP-CP sessions.
- Between two end nodes, CP-CP sessions are not supported.
- An APPN network node may decide whether or not to support, in the XID3, CP-CP sessions across a link to a particular APPN node. This way, adjacent end nodes can be "allowed" to establish CP-CP sessions to the network node.
- An APPN network node may decide whether or not to request, in the XID3, CP-CP sessions across a link to a particular APPN network node.
- Between a network node and an end node, there may be links which do not carry CP-CP sessions, but may do so at a later time (using the functions described in "Non-Activation XID Exchange" on page 39).
- Adjacent network nodes must have the same network ID. Otherwise, the link will not be established. An end node may have a different network ID from its adjacent network node.

CP-CP sessions are used by an APPN end node to obtain services from an APPN network node server. A large subset of these services is also available to APPN end nodes without CP-CP sessions and to LEN end nodes, that are defined by the network node node operator facility as client end nodes for which the network node acts as server. In this case, APPN end nodes without CP-CP sessions to the network node need to be defined as LEN end node .. LUs in these nodes just send a BIND to the adjacent network node, which will process it as if it were from one of its own LUs.

3.4 Non-Activation XID Exchange

[3] After completion of the contact phase, information associated with the link may change. An XID3 (with the exchange state indicators set to "non-activation exchange") is used to communicate these changes. The two main reasons could be a change of the CP name caused by SSCP takeover or the request of an end node to change its network node server.

This process has certain restrictions. For example, not all implementations may support the non-activation XID3 exchange for the *secondary* link station.

End Node Changing the Server Network Node

An APPN end node may have links to one or several APPN network nodes. (These network nodes may even have different net IDs.) When an end node activates its links, it seeks network services from one of the network nodes with which it establishes a link. Once a network node server has been obtained, an end node may wish to obtain network services from some other network node with which it has an active TG. To make this change of server, the end node initiates a non-activation exchange with its current server to inform it that the CP services are no longer requested. This will result in an UNBIND of the CP-CP sessions.

When the first non-activation exchange is complete, the node initiates another non-activation exchange with the desired new network node server to request CP services.

CP Name Change (SSCP Takeover)

When an APPN node is attached to a T4 node and the SSCP owning the T4 node fails, a new SSCP may take over the ownership. In this case, the affected T4 node initiates a non-activation exchange with its attached LEN end nodes. The new CP name is attached to the XID in the network name control vector.

3.5 Link Deactivation

There are several ways link deactivation can occur:

- By operator command.
- The remote node initiated the deactivation.
- The link station is defined as limited resource, and the number of sessions falls to zero on this link.
- Failures detected on the link station or port.
- Failures in XID processing.

[3] Implementation of the architecturally defined non-activation XID exchange for end nodes required changes to low-level DLC code. Therefore, no IBM implementation of an end node actually uses the non-activation XID to change server in the architecturally allowed way, although all implementations could receive such an XID and honor it.

3.6 Connection Networks Using a Shared-Access Transport Facility

In very large token-ring networks, if you want to provide simultaneous any-to-any connectivity between all the nodes, you have to define all the token-ring addresses in each node. That causes an enormous overhead. The benefit of the concept of connection network is to eliminate the definition of connections to *each* of the nodes in the network.

A shared-access transport facility (SATF) provides the functions of the lower layers of SNA - such as a token-ring. A SATF provides any-to-any connectivity for the nodes which are attached to it. A connection between two nodes over the SATF can be seen as a link between them.

The nodes of the connection network form a subset of the nodes of the SATF. The connection network provides a means of defining attachments over a SATF without having to define, at each end node, all the nodes that this end node can reach over the SATF. The DLC signaling information (such as ring station addresses) has to be defined for the local node only, and not for all the other end nodes. Instead, the full attachment information can be obtained from the network node server during the search process prior to initiating a session that is to make use of the facility.

Connection networks offer an advantage to APPN end nodes. Since an end node reports its connection to a virtual routing node (along with DLC signaling information) to its network node server, the network node server can calculate a direct route to the target node on the same connection network, bypassing the network node server of the target node.

3.6.1 The Virtual Routing Node

A virtual routing node is not a node, but it is an easy way to define an end node's link to a connection network. An end node may represent its attachment to a connection network defined on the shared-access transport facility, by a virtual routing node (VRN). The node operator facility provides this form of definition, indicating that it is for a connection network through a local port to a transport facility.

The end node then reports this VRN connection (along with its local DLC signaling information, like MAC and SAP on a token-ring) to its network server during the search process. The information is carried in the TG vectors, which will be explained in Chapter 4 "Topology and Routing Services" on page 43. This is all the server, responsible for route computation, needs in order to determine that two nodes can communicate over a common transport facility. Both nodes indicate their attachment to the same VRN (and therefore to the same connection network) in their TG vectors. The network node server returns routing information to the origin of a search.

3.6.2 Individual and Generic Definitions of Connections

Connections to actual adjacent nodes, as well as to the virtual routing node, can be defined relative to the same shared-access transport facility. So, connections over the facility to real adjacent nodes can be defined:

Either individually in the conventional way by the node operator facility, giving detailed attachment information. In Figure 23 on page 41 this method is used between each end node and the network node server..

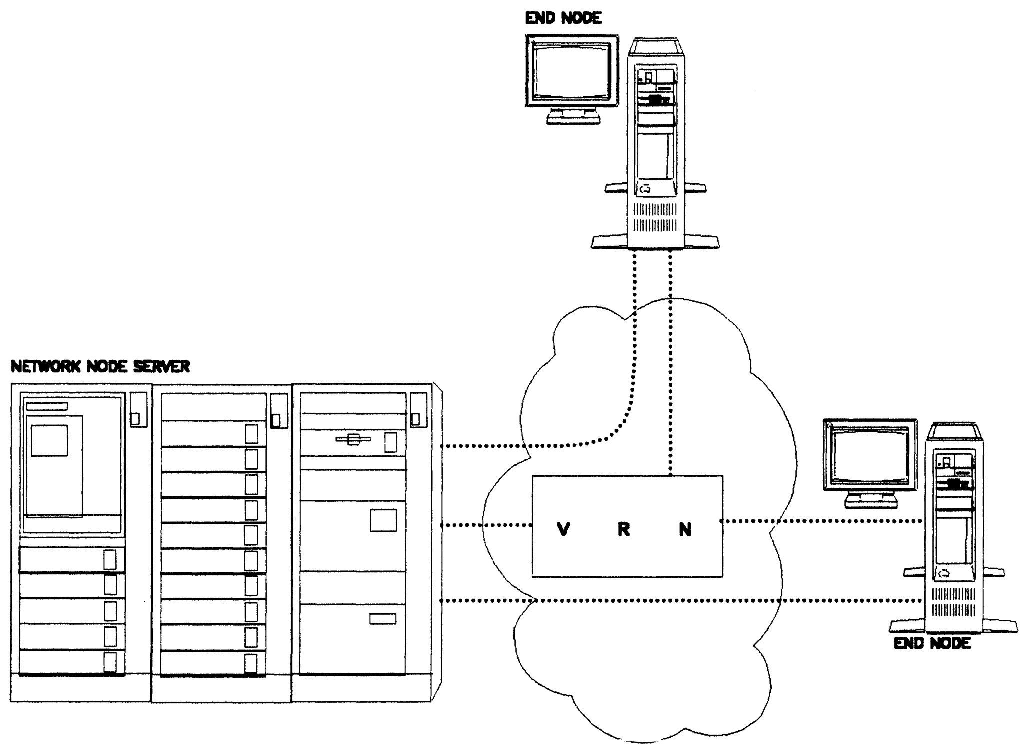

Figure 23. The SATF with Individual and Generic Connections

Or, generically using a virtual routing node to represent connectivity to an entire group of real adjacent nodes giving only the local attachment information. This applies for the connections between the end nodes.

A connection network is the view of the shared-access transport facility in this second, generic fashion. Figure 23 illustrates this dual view of the SATF:

1. As a transport facility on which multiple pairs of real nodes have their separate connecting TGs defined in the conventional manner, and
2. As a connection network consisting of a set of real nodes defined as being attached to a common virtual routing node.

3.6.3 Activating Connections through Connection Networks

Connection networks and ports to connection networks are defined by the node operator facility. In Figure 23, if the PS/2 end node on the right-hand side wanted to establish a connection to the PS/2 end node at the top, it has to use network node services of the AS/400 network node on the left first. That is why one individual connection per end node is required: the one to its network node. CP-CP sessions are not supported over the connection network.

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.

The TG number received from a real adjacent CP is ignored by configuration services. Instead, the TG number associated with the particular connection network is used.

Chapter 4. Topology and Routing Services

Topology and routing services (TRS) is a component of the CP in a T2.1 node. The main purpose of this component is to provide a route selection control vector (RSCV) containing the best route through the network. Topology and routing services is requested for this service in order to establish a session.

The actual computation of the optimal route is done by TRS's sub-component route selection services (RSS). For this task, route selection services uses the information stored in several external and internal databases, which in turn are managed by other sub-components of topology and routing services. Figure 24 shows the sub-components of topology and routing services and the databases involved.

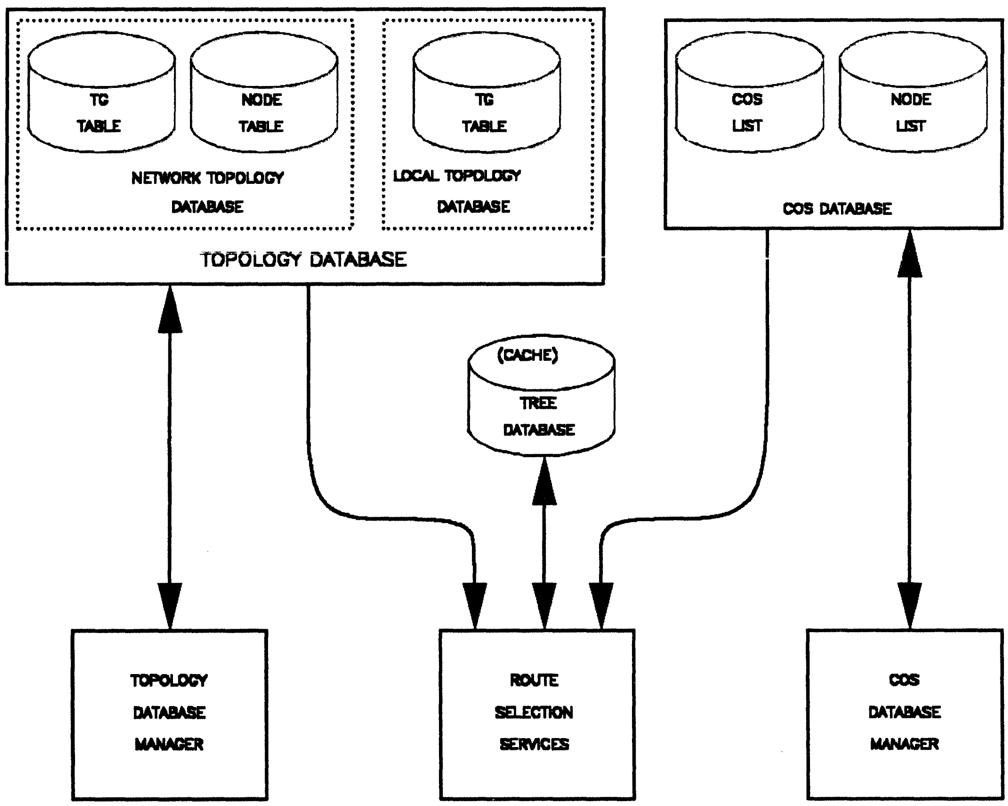

Figure 24. Databases and Sub-Components of Topology and Routing Services

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

Topology and routing services is distinct from directory services (which *locates* a session partner, before topology and routing services comes in) and from the address space manager (which assigns *local addresses* along the route selected by topology and routing services).

In this chapter, we first list the transmission group characteristics and node characteristics, which are part of the topology database and the COS database. That brings us to the description of the databases and their management. We then describe the route selection services with the help of examples, going step by step from simple to complex.

4.1 Initialization

The following parameters are passed from the node operator facility when topology and routing services is initialized:

- Type of node
- CP name of this node
- Network ID of this node
- Whether class of service is supported (COS/TPF option)
- The COS database file name
- The topology database file name.

4.2 Resource Characteristics

In order to calculate the best route, the *actual* node and transmission group(TG) characteristics have to be compared with the *required* route characteristics. Before node and TG characteristics can be compared to requirements, the qualitative node and TG characteristics have to be converted into quantitative node and TG weights. This conversion is depicted in Figure 25. The *actual* node and TG characteristics are stored in the topology database. The *required* node and TG characteristics are defined in the COS database.

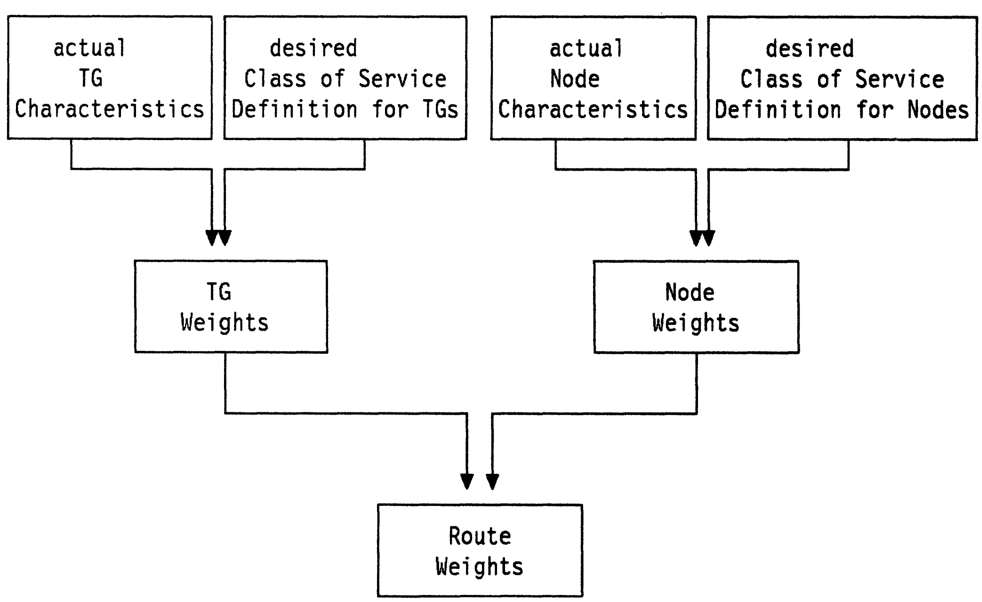

Figure 25. Transformation of Information Performed in Determining Route Weight

The resource characteristics are encoded in two different data structures:

- 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, others (such as security class) have defined values.

Note that some resource properties such as bandwidth are static while others such as congestion are dynamic and are updated periodically.

4.2.1 TG Characteristics

The TG characteristics are contained in control vector X′47′, which is described in detail in *Systems Network Architecture Formats.* Some fields are pointed out here.

Operational: flag

Congested: flag

Capacity: floating point number
The maximum throughput that can be achieved on a link without overloading it. The lowest value is 300 bps. The highest value is over 2 Gbps. The value is supplied by the network administrator. The default is X′2D′ (about 80 % of 9600). The structure of the one byte floating point number is not explained here. Just one example: 64000 bps is represented by X′46′.

Cost per time: binary number
The range is from 0 to 255. The value is supplied by the network administrator. The unit should be coordinated among the nodes. A value of 0 means that the connection can be made at no additional cost (for example, over a nonswitched line).

Cost per byte: binary number
The range is from 0 to 255. The value is supplied by the network administrator. The unit should be coordinated among the nodes. A value of 0 means that the connection can be made at no additional cost (for example, over a nonswitched line).

Security class: seven defined values
The default is X′01′, specifying ″no security″.

Propagation delay: floating point number
The network administrator has four default values to choose from:
negligible X′4C′, terrestrial X′71′, packet-switched X′91′, long X′99′.

Three user-defined fields: binary number
The network administrator may define up to three additional characteristics. The default is 128. This allows the network administrator to define more or less desirable TGs.

4.2.2 Node Characteristics

The properties of the APPN network nodes are contained in two control vectors: the node descriptor control vector X′44′, and the node characteristics control vector X′45′. They are described in detail in *Systems Network Architecture Formats*. Some fields are pointed out here.

Route addition resistance: binary number
A low value means that the node is able to accept intermediate sessions.

The following properties are encoded as flags.

Virtual routing node (VRN)

Congested

End node routing resources depleted

Network node routing resources depleted

Intermediate routing services supported
This is the case in all network nodes.

Border node functions supported
See "Border Nodes" on page 167.

4.3 Topology Database

There are two kinds of topology databases in an APPN network: the local topology database and the network topology database. Every LEN end node and APPN end node maintains a local topology database. An APPN network node maintains a network topology database. The topology database is kept in permanent storage and saved across IPLs.

The topology database is updated automatically.

The primary use of local and network topology databases is for route calculation. When an LU residing in one APPN node wishes to establish a session with an LU residing in another, topology databases enable topology and routing services to determine the best possible route to the destination LU. 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.

4.3.1 Local Topology Database

For APPN end nodes the local topology database contains information on all of the transmission groups (TGs) attached to that node. 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 local TGs to its network node server to complement the network node′s knowledge for the search and route selection processes.
3. When establishing sessions to pre-defined LUs without the help of a server network node.

ITSC Technical Bulletin Evaluation

Technical Bulletin Title: ______________________________

Technical Bulletin Form Number: ____________

This is an evaluation form to assess the quality of ITSC publications. Your feedback will help maintain the high quality of ITSC standards. Please fill out this questionnaire and send it to the address on the back of this page. No postage stamp is required if mailed in the U.S. Elsewhere, you may choose to have your IBM Marketing Representative forward your reply to the address listed on the reverse side of this form.

Date publication was ordered (MM/DD/YY) ___/___/___

Date publication was received (MM/DD/YY) ___/___/___

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

Technical Bulletin organization	___	Grammar/punctuation/spelling	___
Accuracy of the information	___	Ease of reading and understanding	___
Relevance of the information	___	Ease of finding information	___
Completeness of the information	___	Lack of redundant information	___
Value of illustrations	___		

Please answer the following questions:

a) Was the level of detail of the information adequate? Yes___ No___

b) Did you find information duplicated that was available in other IBM publications? Yes___ No___

If yes, please name the publication:

____

____

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:

____

____

Comments/Suggestions:

Thank you for your feedback.

Your name, company name, and address (optional):

____

____

____

ITSC Technical Bulletin Evaluation Form

Fold and Tape   Please Do Not Staple   Fold and Tape

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS   PERMIT NO. 40   ARMONK, N.Y.

POSTAGE WILL BE PAID BY ADDRESSEE:
IBM International Technical Support Center
Department H52, Building 930
P.O. Box 950
Poughkeepsie, New York 12602
U.S.A.

ATTN: Quality Coordinator

Fold   Fold

IBM®

PRINTED IN THE U.S.A.

The local topology database contains information on *endpoint TGs*. That is, a TG that is available for use only by sessions that terminate in the node that owns it. An endpoint TG is not included in the network topology database.

The Local Topology Database Manager
The sub-component TDM (topology database manager) creates and maintains the topology database. Entries in the topology database are created automatically, when configuration services informs of a newly activated or changed TG. 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.

4.3.2 Network Topology Database

Every APPN network node contains a copy of the network topology database. The network topology database contains information on all network nodes in the APPN network and the TGs interconnecting them (*intermediate routing TGs*). Additionally, the local copy of the network topology database contains information on the local TGs of that node. The network topology database does not include information on LEN end nodes, APPN end nodes, and the TGs attached to them.

Unlike the directory database, which is distributed among network nodes, the network topology database is fully replicated at each node. APPN protocols for the distribution of network topology information ensure that every network node is provided with a complete view of the network topology. That is why all network nodes have the same view of the network.

4.3.3 Structure

The network topology database is maintained as two tables: the node table and the TG table.

The node table
The node table contains information describing a node, such as:

- Network ID.
- CP name.
- Node characteristics (summarized in "Node Characteristics" on page 46).
- Pointer to the node-attached TGs.
- Resource sequence number, used to determine when received updates contain more recent information about a resource. See "Resource Sequence Number (RSN)" on page 49.

The TG table
The TG table contains two entries for a TG between two network nodes (A and B). One as a TG between A and B and another as TG between B and A.*:* The TG in each node is represented by a TG record and a TG vector. 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 45)
- The resource sequence number
- DLC signaling information when partner is a VRN. The DLC signaling information may refer to token-ring MAC addresses, dial-in telephone numbers, X.25 network addresses.

The Weight Index Structure
Calculating TG weights can be a time consuming process. It may have to be repeated for each session setup. For performance reasons, the architecture provides an option to cache the weights as long as they are valid. This option is called the TG weight index structure. As it is not relevant for the basic technique of route selection, it is not covered here. The weights structure is updated by route selection services. Please refer to *Systems Network Architecture Type 2.1 Node Reference* for details.

4.4 Network Node Topology Database Manager

The network node topology database manager (NNTDM) is a component that resides in every network node. Each NNTDM creates and broadcasts topology database updates (TDUs) about the node itself and locally owned TGs to other network nodes. In addition, every NNTDM stores and re-broadcasts TDUs received from NNTDMs located in other network nodes. This allows every NNTDM in the network to maintain a consistent copy of the network topology database.

4.4.1 Flow Reduction Considerations

The amount of data for topology database updates that flows between network nodes is a concern in large networks. In order to keep the impact of network overhead on the performance of LU-LU sessions at a minimum, several mechanisms are put into place to reduce this flow.

Topology Database Restricted to Network Nodes
End nodes and, more important still, the TGs connecting end nodes are not contained in the topology database. This way, it is up to the network planner to reduce the number of network nodes in order to keep the topology database small.

No Topology Database Updates Concerning Connection Networks
TG updates are not broadcast when a TG to a network node is activated or deactivated across a virtual routing node. If the TG fails (abnormal deactivation), TG updates are sent, in order to exclude this TG from route computation.

Topology Database Updates not Broadcast Back
When a network node receives a topology database update (TDU), it will re-send this TDU to all its adjacent network nodes. The network node has the responsibility not to send it to the network node from which it received the TDU.

Resource Sequence Number (RSN)
There is a different RSN for every node and TG in the network topology database, and it is assigned by the network node that owns that particular resource. Whenever the node creates a TDU for that resource, it increments the RSN to the next even value. The RSN allows a network node that receives a TDU about a particular resource to determine whether or not it has received that update before. This is the mechanism that is used to terminate the broadcast of the TDU throughout the network.

The RSN is an unsigned even integer in a circular number space. The range is 2 to 2^{32} - 2. Odd values are used to indicate "inconsistent sequence numbers" and will trigger error recovery.

Flow Reduction Sequence Number (FRSN)
The network node does not necessarily send TDU messages containing information on all NNs and TGs in its topology database. A process known as flow reduction prevents retransmission of large topology databases following temporary link outages. A record is kept at each NN as to what information has been sent previously to each of its adjacent NNs. When a link between NNs is re-established and CP-CP sessions are activated, FRSNs are exchanged and only the "new" information is transmitted.

Every network node in the network maintains its own flow reduction sequence number that it uses to time sequence all the TDUs that it broadcasts to its adjacent network nodes. The FRSN is only known by the network node and its adjacent network nodes. FRSNs are associated with TDUs. (RSNs are associated with resources.)

An adjacent network node informs a network node of the last TDU it received from the network node when a CP-CP session is activated between them. If the two nodes become connected for the first time, the adjacent network node will receive all of the network node's topology database. If the two nodes become reconnected after having been temporarily disconnected, the adjacent network node will receive very little of the network node's copy of the topology database.

Whenever a network node creates a TDU and broadcasts it to its adjacent network nodes (this includes re-broadcasting), it increments it own FRSNs. This new FRSN is included in the TDU and stored with every topology database entry that was included in that TDU.

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 all of the adjacent node's topology database.

4.4.2 Topology Updates

There are several conditions which require topology database updates.

Initial Topology Exchange

The NNTDMs in two adjacent nodes can establish conversations with each other after the CP-CP sessions between the two nodes have been activated. The TDUs are carried in GDS variable X'12C2'. It contains node characteristics, the resource sequence number, the TG identifier and (for connection networks) DLC signaling information.

Initial Connection

Before a network node joins the network, its local copy of the topology database contains only its local resources. When it is connected to the adjacent network node, it will receive a copy of the current topology database. Similarly, it will create a TDU about itself and its links to other network nodes. This TDU will be broadcast throughout the network.

Reconnection

Whenever a CP-CP session is established between two adjacent network nodes, the greatest FRSN is exchanged together with the CP capabilities. The receiver of the FRSN scans its topology database. It selects and broadcasts every entry that has an FRSN higher than the FRSN received.

Processing Local Resource Changes

Whenever a network node detects a change in its own state or in the state of locally owned TGs, it updates the entry for the TG in its local copy of the topology database. This includes incrementing the RSN for that resource to the next even value.

If the resource was either an intermediate routing TG or the node itself, NNTDM will also create and broadcast a TDU to all active and adjacent network nodes.

Node Update for Virtual Routing Nodes

Because a virtual routing node (VRN) is merely a representation and does not really exist, it cannot broadcast updates. That is why a real network node that has access to a transport facility that it represents as a virtual routing node, has to broadcast TDUs for TGs which are defined towards a virtual routing node. In addition, it must also broadcast information about the virtual routing node itself. The node characteristics for a virtual routing node have architected default values.

Processing Received Topology Database Updates

The following rules apply:

If the receiver of the TDU is the resource owner, the TDU will normally be discarded. (If RSN is odd, indicating an error, a new TDU is created and broadcast to all adjacent network nodes.)

If the receiver of the TDU is not the resource owner, then:

- If the resource is not in the topology database, it will be included.
- If the resource is in the topology database, then the RSN in the TDU is compared to the RSN in the topology database.

- If the RSN in the TDU is greater, the information is stored in the topology database and the TDU is re-broadcast to all adjacent network nodes.
- If the RSN in the TDU is less than the RSN of the resource in the topology database, then a new TDU is created and broadcast to all adjacent network nodes.
- If the RSN in the TDU is equal to the RSN of the resource in the topology database, then the information in the TDU is compared to the information in the topology database.
  - If the information is the same, the TDU is discarded.
  - Otherwise, a new TDU is created with the RSN incremented by 1 and broadcast to all adjacent network nodes. This will prevent the resource from being included in route calculations and trigger error recovery by the resource owner.

4.4.3 Other Functions

The NNTDM has some additional functions. Three of them are mentioned here.

Garbage Collection

An interval timer is associated with every resource entry in the local copy of the topology database. If no information about a resource is received for a specified time, the entry for that resource will be discarded. This applies for remote as well as for local resources.

Every NNTDM broadcasts a TDU about itself once every seven days, in order to prevent other network nodes from discarding this node from their topology databases.

Notification of Route Selection Services

Whenever NNTDM updates or deletes a resource, it notifies the route selection services component of topology and routing services. This gives route selection services a chance to delete any trees that are stored for the changed or deleted resource.

Processing Topology Database Queries

Directory services, session services, and route selection services all have interfaces to the NNTDM in order to obtain the information they need from the topology database.

4.5 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 to translate a COS name to a transmission priority. The node operator facility informs topology and routing services whether or not this option is supported. The class of service manager is a component of topology and routing services.

The COS database consists of the following items:

- List of mode names
- List of COS names
  - Definitions for TGs
  - Definitions for nodes (in APPN network nodes only)
- Pointer to weight index structure.

The COS database is maintained independently by each node. With the help of the node operator facility, it is updated by the network administrator.

Sessions have different data transmission requirements. Interactive sessions usually require faster data transmission and more predictable response times than bulk data transfers. Because sessions for several different kinds of applications can be in progress over a given route, multiple classes of service should be provided for the route.

The ability to specify a mode name and COS table at session establishment time provides an increased amount of flexibility in terms of session characteristics and route selection. However, in many cases, such flexibility may not be required. Therefore, IBM provides several pre-defined mode names and corresponding COS tables.

List of Mode Names

In a session request, a mode name is specified, not a COS name. A mode name implies a set of session characteristics, one of which is the class of service. It is the COS name that is used by route selection services to select a route. Therefore, each mode name must be mapped to the appropriate COS name.

In the COS database, each entry in the mode-COS list contains a mode name and a pointer to the corresponding COS name.

The architecturally-defined mode names that map to the architecturally-defined COS names are listed in "SNA Defined Modes and Classes of Service" on page 71.

List of Class of Service Names

Per class of service, the COS database contains one entry for TGs and another one for nodes. Each entry for TGs contains:

- COS name.
- Transmission priority:
  - Network
  - High
  - Medium
  - Low.
- Several rows for COS definitions, consisting of:
  - Ranges (pairs of high and low values) for TG characteristics. The TG characteristics are explained in "TG Characteristics" on page 45.
  - A weight field.

←——TG characteristics——→

		cost		capacity		...	user 3			
COS name	priority	low value	high value	low value	high value		low value	high value	weight	row 1
		cost		capacity		...	user 3			
		low value	high value	low value	high value		low value	high value	weight	row 2
		cost		capacity		...	user 3			
		low value	high value	low value	high value		low value	high value	weight	row 3

Figure 26. One COS Entry for a TG

As Figure 26 suggests, the COS definitions for TGs may be described as a table consisting of a number of rows. Each row contains a set of ranges for acceptable values. There is one range of acceptable values per TG characteristic. Each row has a weight field associated with the set of ranges on that row.

The characteristics defined for a class of service are used to subset TGs into two classes: those that are acceptable for the class of service and those that are not acceptable for the class of service.

The Range

A TG is *acceptable* for a particular class of service if its actual values for all characteristics fall within the range of values for those characteristics in the class of service definition. A TG is *unacceptable* for a particular class of service if an actual value for any characteristic falls outside the range of values for that characteristic in the class of service definition.

The Weight

Low weight is desirable (for routes, too). The lower the weight, the better the route.

Route selection services relates the actual values for TG characteristics to a particular COS name using the definition for that class of service. Comparing the values of the TG's characteristics to the COS definition leads to the determination of the weight for the TG. The TG weight is a quantitative measure of how well the values in the TG's characteristics satisfy the criteria specified by the COS definition. If the TG does not satisfy the criteria specified by the COS definition, it is assigned an infinite weight. The subject of weight calculation is resumed in the "Examples for Route Selection Services" on page 57. The sub-chapter "End Node with Class of Service" on page 61 gives a detailed example.

The weight field may either be a constant or a function of the characteristics.

If the weights take constant values, the table entry is composed of multiple rows with one constant weight per row. The rows with the lowest weight should be row 1: the most acceptable combination of characteristics. If a row is encountered for which all the TG's actual values lie within the acceptable ranges of values on that row, the TG is assigned the constant weight associated with the row.

If the weight is obtained from a function, the table entry is composed of a single row with the name of the function. If a row is encountered for which all the TG's actual values lie within the acceptable ranges of values on that row, the function is called to calculate the weight based on the characteristics.

For performance reasons, the COS database contains pointers to the topology database.

Definitions for Nodes

In the COS database of an APPN network node, there is also one entry for nodes for each COS name. The considerations about range and weight given for TGs apply for nodes respectively. The COS entry for nodes consists of the following subset of the node characteristics (as described in "Node Characteristics" on page 46):

- COS name
- Index
- Transmission priority
- Several rows for COS definitions, consisting of:
  - Route addition resistance
  - Congestion flag
  - A weight field.

Figure 27 on page 55 gives an example of how the COS database is updated by the operator interfacing through the node operator facility. The figure also shows the automatic update of the topology database. A change in the configuration triggers configuration services to send a topology update request to topology and routing services.

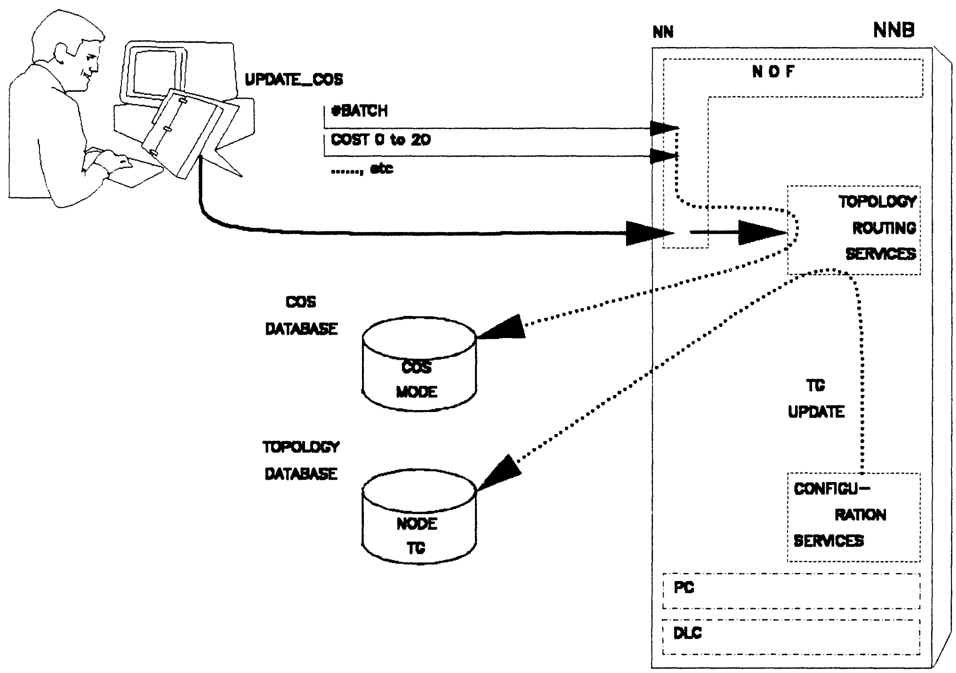

Figure 27. Update of the TRS Databases: by Operator and Automatic

4.6 Tree Data Base

The tree database is an option in an APPN network node. It stores the optimal routes to other network nodes. Routes from the network node to other network nodes form a tree structure with the origin network node at the root of the tree. Within the tree database there is one tree per class of service, and within each tree are stored the optimal routes to all network nodes for that class of service. A tree includes network nodes, their connecting TGs, and the weights assigned to both. The tree database is updated automatically.

When route selection services receives a route request from session services, route selection services first checks the tree database to see if a route has already been computed to the destination network node for the specified class of service. If so, route selection services uses the same route (together with the information provided by the end node) to construct the route selection control vector. If not, route selection services computes a new tree and stores it. Route computation delivers the shortest path tree for a specific class of service. "Route Selection for End Node with CP-CP Sessions" on page 66 will give an example for the use of the tree database and some aspects of the algorithm used to compute the trees.

When a tree database is not maintained, the trees are not cached. A new tree is computed from scratch for each route request.

The tree database is derived from the network topology database and the COS database. Whereas the network topology database is identical throughout all network nodes, the tree database is unique for each node. It shows this node's view of the network.

Figure 28 shows a small network. It contains four APPN network nodes and two APPN end nodes.

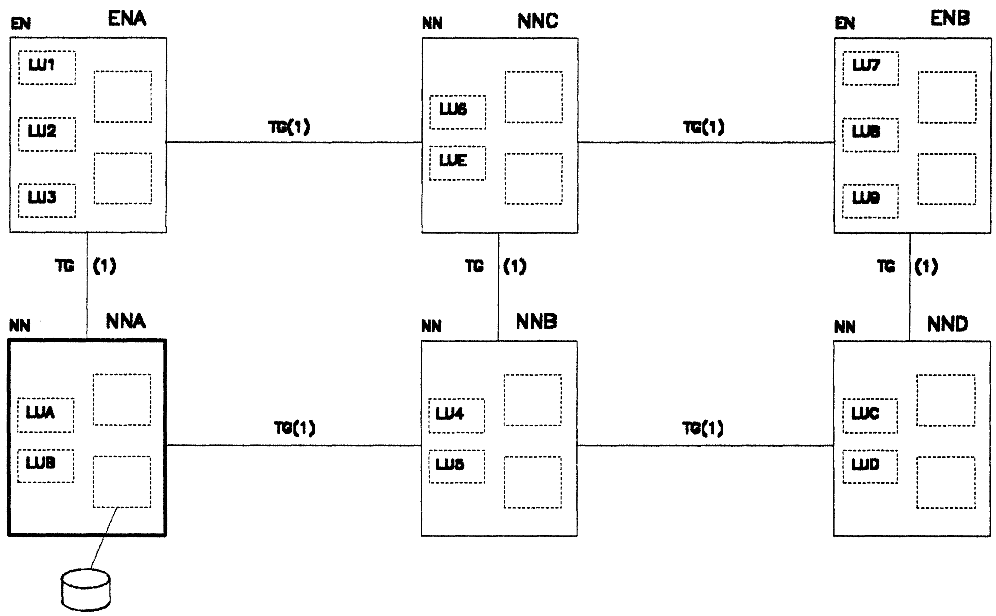

Figure 28. Network Node "NNA" as Part of an APPN Network

Figure 29 on page 57 gives an example for a tree derived from that network. It is the tree with node "NNA" as the root, for a certain class of service. (For some strange reason, these trees are usually shown with the root at the top.)

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. A new tree is added only when required by a route request.

If the database is full, the least recently used entry is deleted when a new one is added.

Another reason for deletion of a tree from the database is "overuse". Load distribution among equally low weighted routes can be achieved by randomization. Cache entries are removed when they are used more than a specified number of times, causing the tree to be re-computed.

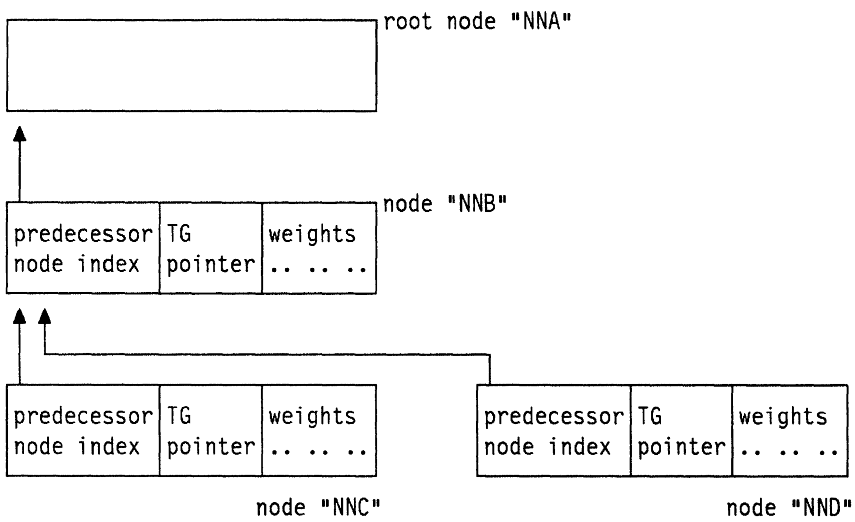

Figure 29. Network Node "NNA" as Root of a Tree

A third reason is topology changes. Cache entries may become invalid and are removed when node or TG characteristics or COS definitions change.

4.7 Examples for Route Selection Services

Route selection services is the most sophisticated sub-component of topology and routing services. In this section, the functions of route selection services will not be listed systematically, rather by using examples, progressing from simple to complex. This way, all major functions of route selection services are described and details are given only after the fundamentals have been covered.

In an APPN end node, route selection services acts for the local LUs of that node; in an APPN network node, route selection services serves the local LUs of the node and the LUs of end nodes within the network node's domain.

Generally speaking, route selection services has the following functions:

- It accepts route requests consisting of:
  - Origin node
  - Destination node
  - Class of service (COS).
- It determines the least-weighted route from the origin node to the destination node for the specified class of service.
- It returns the route in a route selection control vector as an ordered sequence of the nodes and TGs that make up the route.

To perform these functions, the route selection services has:

- A mechanism for relating *actual* values (of a node or TG) to the node or TG characteristics *required* by a mode name: the range method, as explained under "The Range" on page 53.
- An accurate knowledge of the topology of the network: The topology database. (See "Topology Database" on page 46 for a definition.)

- A route computation algorithm: the "least weight algorithm" is introduced in the following examples.

The Route Selection Control Vector
The route selection control vector (RSCV) is carried in the BIND, the LOCATE/CDINIT, and other RUs to describe the path through the APPN network. The RSCV is sent and received by APPN network nodes and APPN end nodes, but not by LEN end nodes. The RSCV appended to the BIND contains a number of TG descriptor control vectors (each of which contains a TG number and the name of the adjacent node). The RSCV also has two hop counts: the total number of hops (equal to the number of TG vectors) and the current hop count, which counts the hops (TGs) already traversed.

The complete format is found in *Systems Network Architecture Formats*.

4.7.1 The Normal APPN End Node

To establish an LU-LU session, an APPN end node will normally use the services of the adjacent network node server. The request to assist in session establishment is sent to the network node across a CP-CP session. The request is accompanied by information that the network node server needs in order to calculate the optimum route, and that the network node does not store in its own database: information on the TGs attached to the end node.

In the APPN end node, before session services sends the request to the network node, it queries its own topology database manager for its TG vectors, as is shown in Figure 30. In this example, the topology database contains just one entry. The topology database manager returns to session services all TG vectors (one in this case). Session services appends this endpoint TG vector (sometimes called TG tail vector) to the LOCATE/CDINIT, which is sent to the network node. The processing in the network node will be explained in an example of "Route Selection for End Node with CP-CP Sessions" on page 66. A more comprehensive example, putting these fragments into context is given in a later chapter under the heading "Session Initiation for APPN End Node" on page 112.

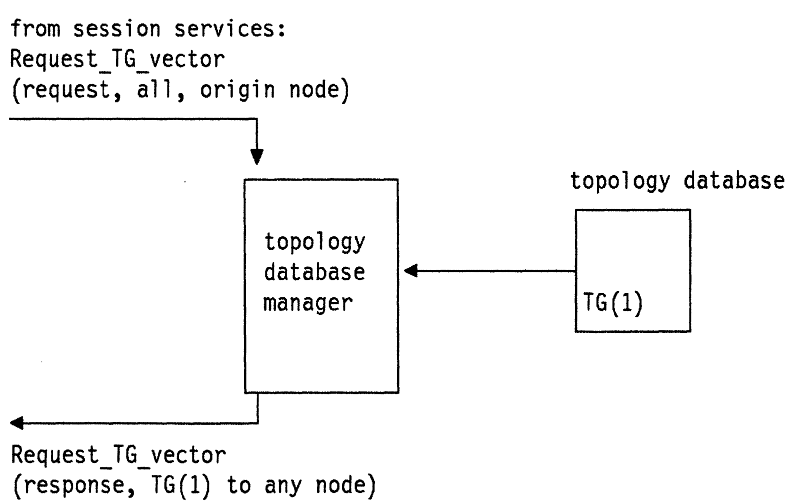

Figure 30. APPN End Node Queries Its Topology Database for TG Vectors

In this example, the APPN end node did not use its own route selection services, but those in the network node server. Yet, as we described in the section on "Local Topology Database" on page 46, there are cases where route selection services in the end node has to do the work. These cases are presented next.

In the examples given in the following sections, the basic situation is always the same: route selection services receives a request to provide the best route; the response to this request is always the RSCV with the best route. The differences are found in the processing done by route selection services, depending on the scope of functions implemented in the node.

4.7.2 The Simple End Node

The most basic case is an end node which has only one TG connected to it. The node may be a LEN end node or an APPN end node without CP-CP session. The TG could, for example, be the connection to the network node server. All remote LUs have to be pre-defined.

The example in Figure 31 assumes that session services (SS) is about to build a BIND for a session to a logical unit called DLU. The DLU has to be pre-defined as being in the network node server. This way, session services has knowledge of the CP name belonging to DLU. Session services sends an internal request to the route selection services of its node. It requests the best route towards the CP that owns the DLU. Because we are in an end node, it is a "request single hop route". The name of the CP that owns the DLU is included. Route selection services receives the request and queries the topology database.

In this simplest case, the topology database contains just one entry. The topology database manager returns one TG vector to route selection services. Route selection services has no choice and "selects" this TG vector as the best TG. As a reply to the request, route selection services builds a route selection control vector (RSCV) with the "selected" TG vector and returns it. This will tell session services which TG to use.

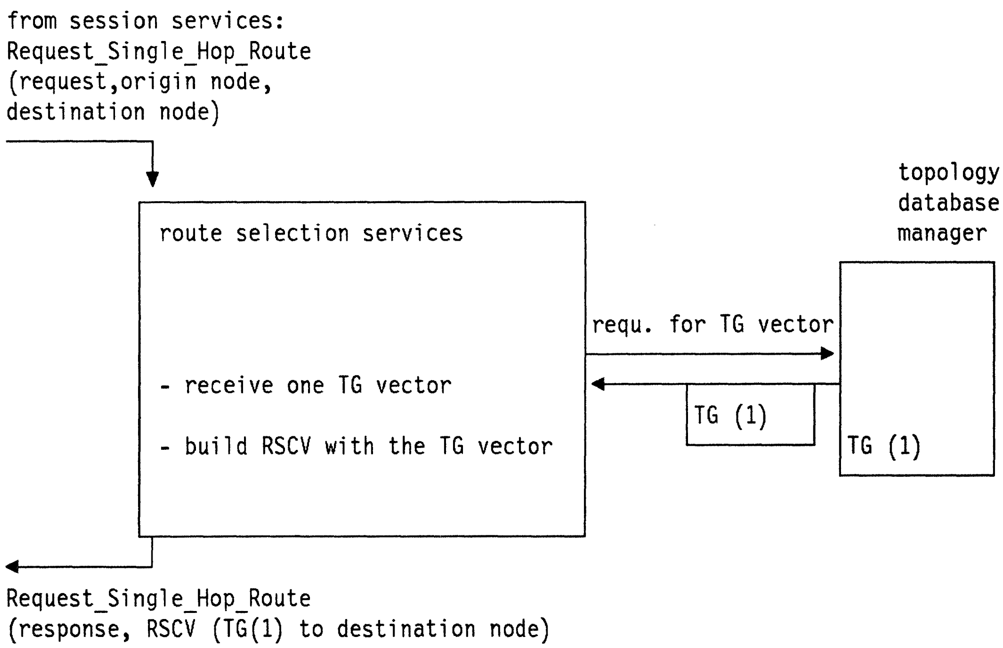

Figure 31. Route Selection in Simplest End Node with One TG

4.7.3 End Node with Multiple TGs

We now assume an end node that connects to more than one node. More important still, the connection to one of the adjacent nodes consists of three TGs. The request and response from and to session services is the same as in the previous example.

As you can see in Figure 32, the processing done by route selection services is only slightly more complex than in the previous example. The topology database contains several TG vectors. Three of these TG vectors are for TGs going to the specified node. The topology database manager presents all three TG vectors. Route selection services selects one TG vector at random.

This randomization provides a certain amount of load distribution. It comes in when there are parallel TGs or alternate routes through the network. Sessions between the same two nodes will use different routes, if there are equally preferable routes.

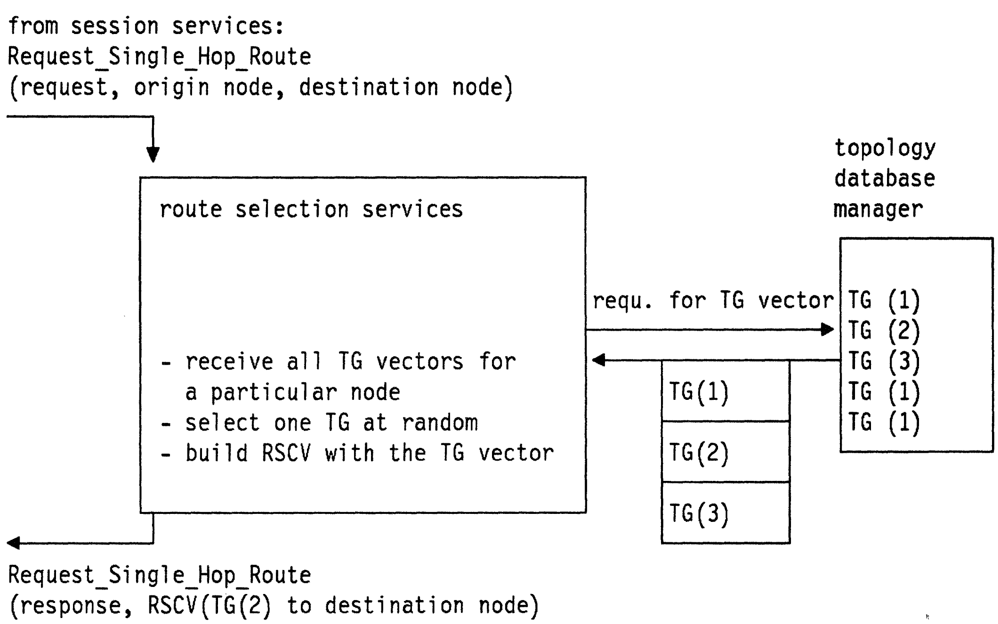

Figure 32. Route Selection in End Node with Multiple TGs

In this and the previous example, we considered an APPN end node which did not implement the COS/TPF function. Remember that this function is optional in end nodes. (It is a base function in network nodes, though.)

4.7.4 End Node with Class of Service

The example in Figure 36 on page 63 assumes that the end node implements the COS/TPF function. That means it has a COS database and route selection services uses the class of service in order to find the best route in the network.

The LU, which requests a session from session services, specifies a mode name. This has to be translated into a COS name first. Session services sends a request to the COS manager sub-function of topology and routing services. This is shown on Figure 33. The COS manager returns to session services the COS name to be included in the route request.

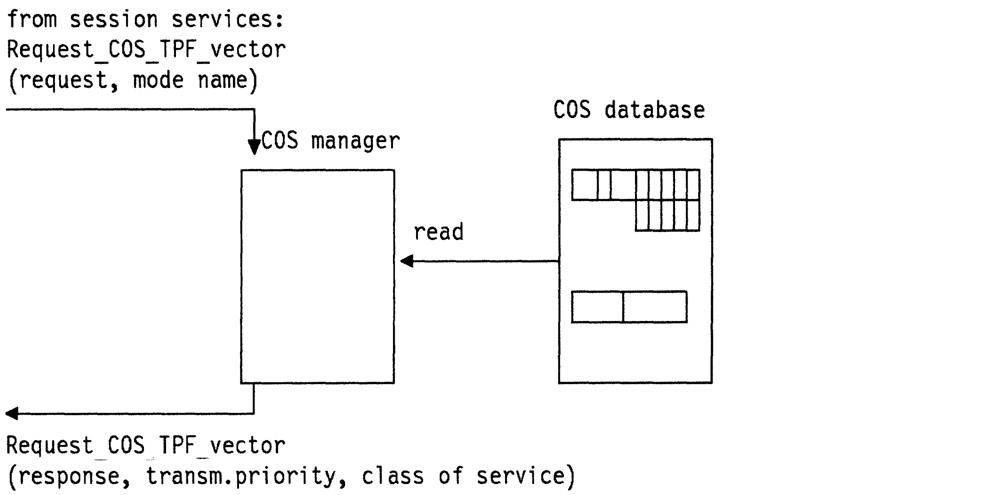

Figure 33. The Mode Name Is Used to Assign a COS Name

As in previous examples, route selection services receives a request for a single hop route. The request gives the CP name of the destination node. The request also includes a COS name, which specifies the desired characteristics for the route to be taken.

As before, the topology database manager sends all TG vectors for TGs going to that CP. Assume, there are three TG vectors. Figure 34 on page 62 gives an example for TG vectors of three TGs. This is not the actual format, but is intended to demonstrate the route selection process. Only the fields for "cost per time" and "capacity" contain relevant information. We assume that all the other fields are set to the default value (such as 128 for user3). The characteristics are explained in "TG Characteristics" on page 45. In this example, TG1 could be a 9600 bps leased line (medium capacity line with no additional cost per unit of time). TG2 is probably a 64 Kbps leased line, and TG3 a dial line.

		cost	capacity	...	user 3
TG 1	CP B	0	30	...	128

		cost	capacity	...	user 3
TG 2	CP B	0	46	...	128

		cost	capacity	...	user 3
TG 3	CP B	40	46	...	128

Figure 34. TG Vectors for Three TGs

Then, route selection services reads the relevant COS entry from the COS database. In Figure 35, a COS entry with two rows is used as an example. The characteristics "cost per time" and "capacity" are used to distinguish between classes of service. We assume that all other characteristics do not matter. So, their ranges will be 0 to 255 (such as "user3"). In this example, the class of service #BATCH requires a route with high capacity (46 or higher), and prefers a route with low cost (under 20 for the lowest weight).

#BATCH	LOW	cost		capacity		...	user 3			
COS name	priority	low 0	high 20	low 46	high 255		low 0	high 255	weight 10	row 1
		cost		capacity		...	user 3			
		low 0	high 60	low 46	high 255		low 0	high 255	weight 40	row 2

Figure 35. COS Table Entry with Two Rows

Now, route selection services is ready to do the route computation.

Route selection services starts with the TG vector for TG1. It compares the characteristics in TG1 with the ranges in row1: the cost value lies within the range; the capacity value (30) does not lie within the range (46 to 255). So, TG1 is not acceptable for row1. The comparison to row2 shows: TG1 is not acceptable here either. So, TG1 is out (it is assigned an infinite weight).

Next, the TG vector for TG2 is looked at: its values fit into all the ranges of desired values for row1. Consequently, TG2 is assigned a weight of 10.

The TG vector for TG3 comes next: the cost value does not lie in the desired range for row1. But for row2, the TG values fit into all the ranges of desired values. Consequently, TG3 is assigned a weight of 40.

The weights for the different TGs are compared now, and TG2 is found to have the lowest weight. The weight of 10 is assigned to TG2, and TG2 is chosen as the best route.

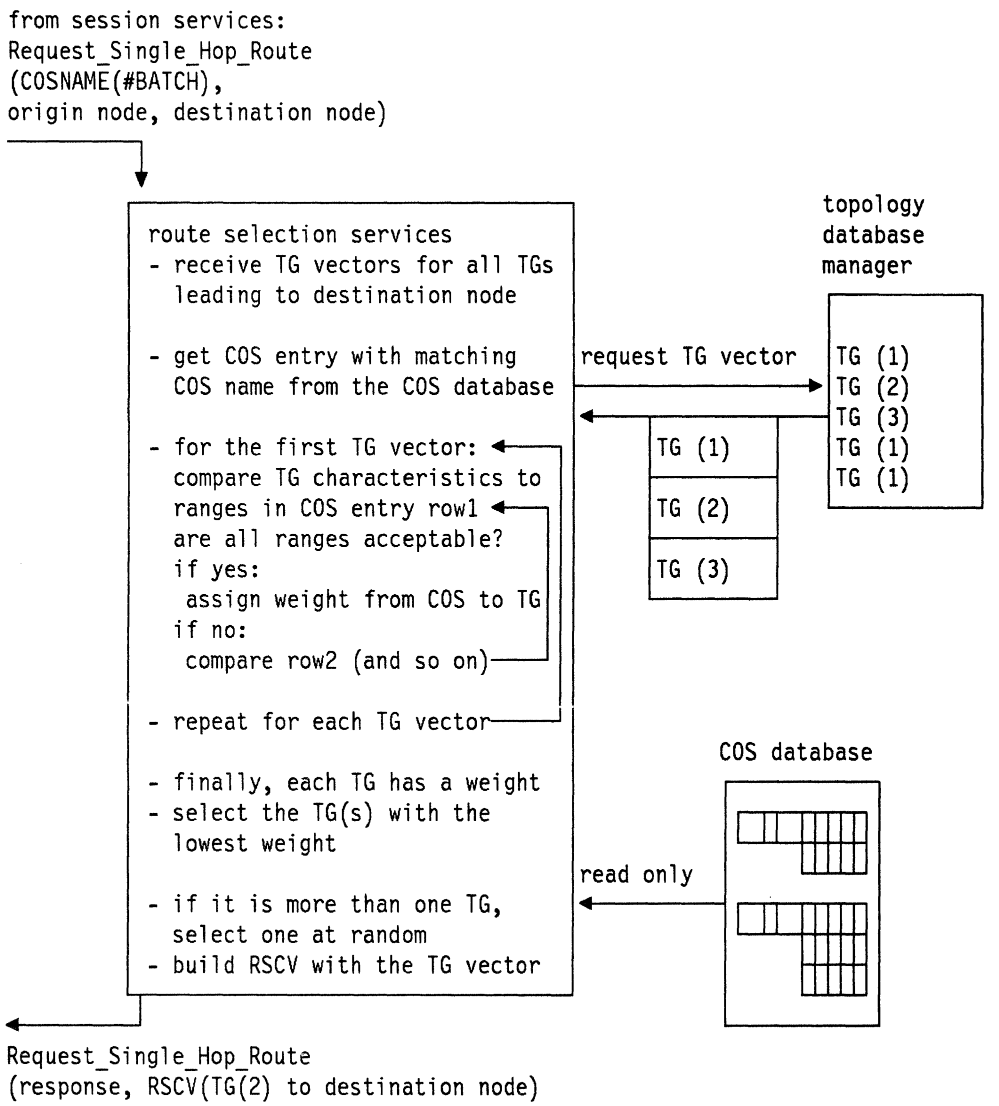

Figure 36. Route Selection with Multiple TGs Using Class of Service

4.7.5 TG Weight Calculation Functions

The example in Figure 37 is similar to Figure 36 on page 63. The difference is that the TG weight is not derived from a constant defined in the COS table, but is computed by a function. The name of the function is part of the COS table (instead of the defined weights). As the weight will be re-calculated for each request, there is no need in the COS table to distinguish the weight for the most preferred route from the weight for the second best route. The COS entry has just one row.

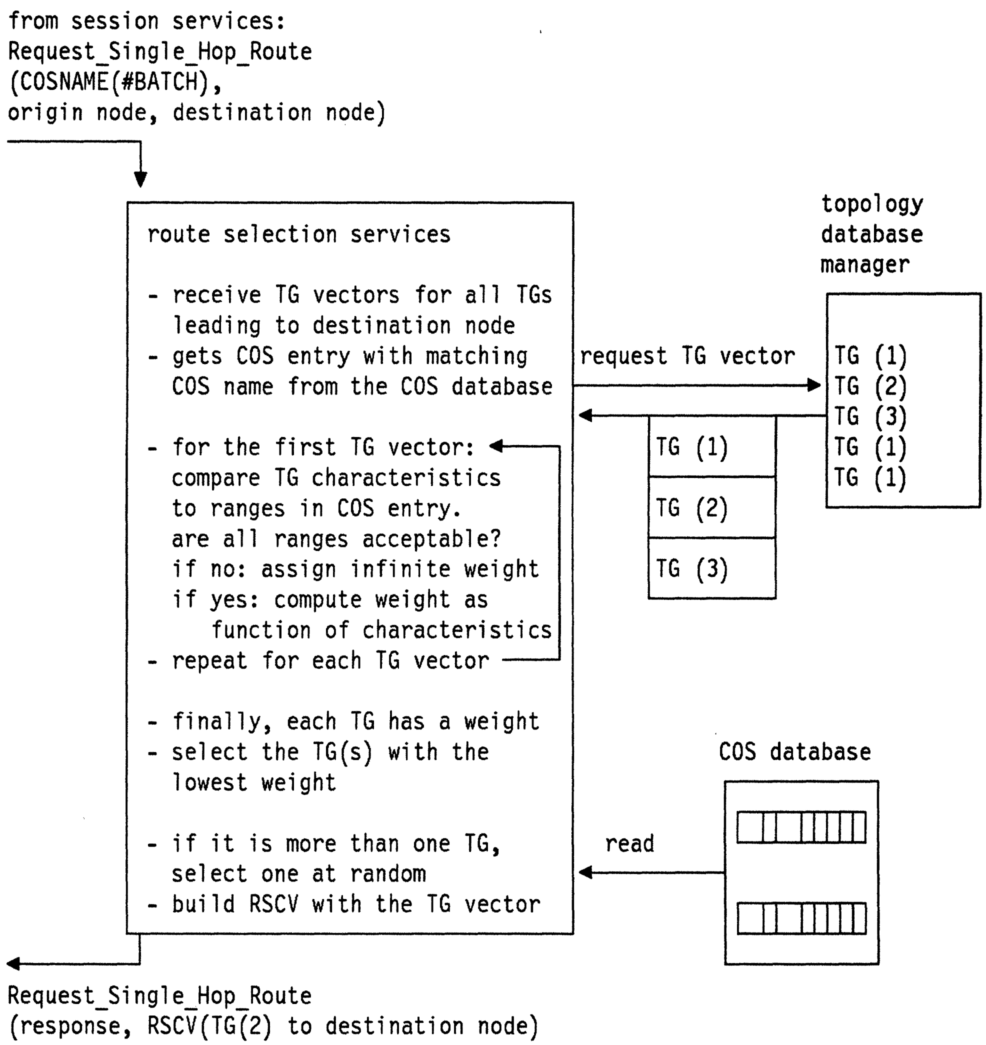

Figure 37. Route Selection with Weight Function

The previous examples all concerned APPN end nodes and the routing requests were for single hop routes. Accordingly, the responses all consisted of an RSCV with just one TG vector.

We will now turn to APPN network nodes and also show the services provided by the network node for the end nodes in its domain. The request involved is no longer a "request single hop route", but a "request route", which applies to a multi-hop route.

4.7.6 Route between Network Nodes

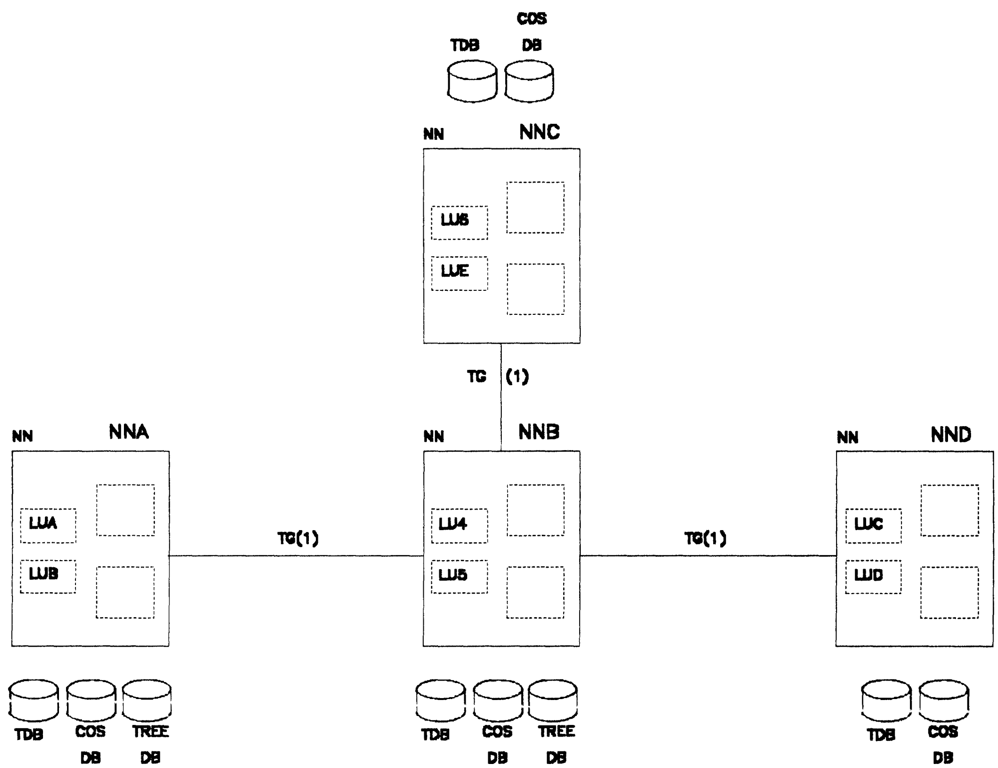

Figure 38. Example Configuration with APPN Network Nodes

The network in Figure 38 shows four interconnected network nodes. All of them have topology databases and COS databases. Two of them also have tree databases. We are in network node "NNA". LUA in node "NNA" requests a session with LU6 in node "NNC". The session is for file transfer, so the LU has chosen mode #BATCH. Session services requests the best route from "NNA" to "NNC" for class of service #BATCH.

Node "NNA" has implemented the tree database option. Route selection services gets the request and first reads the tree database. It actually finds a tree for class of service #BATCH, originating in "NNA", and containing node "NNC". The tree had been used previously to calculate a route. That saves any further calculations. Route selection services reads the required node and TG vectors and constructs the RSCV.

The RSCV is a hop-by-hop representation of the route from the origin node to the destination node. It consists of an ordered sequence of (TG number, CP name) pairs. The tree is kept for further use. It is put in a list as "most recently used" tree.

In this example, the information in the tree database was used to find the least weight route. But, it did not explain how the tree got into the tree database in the first place. This will become clear in the next example.

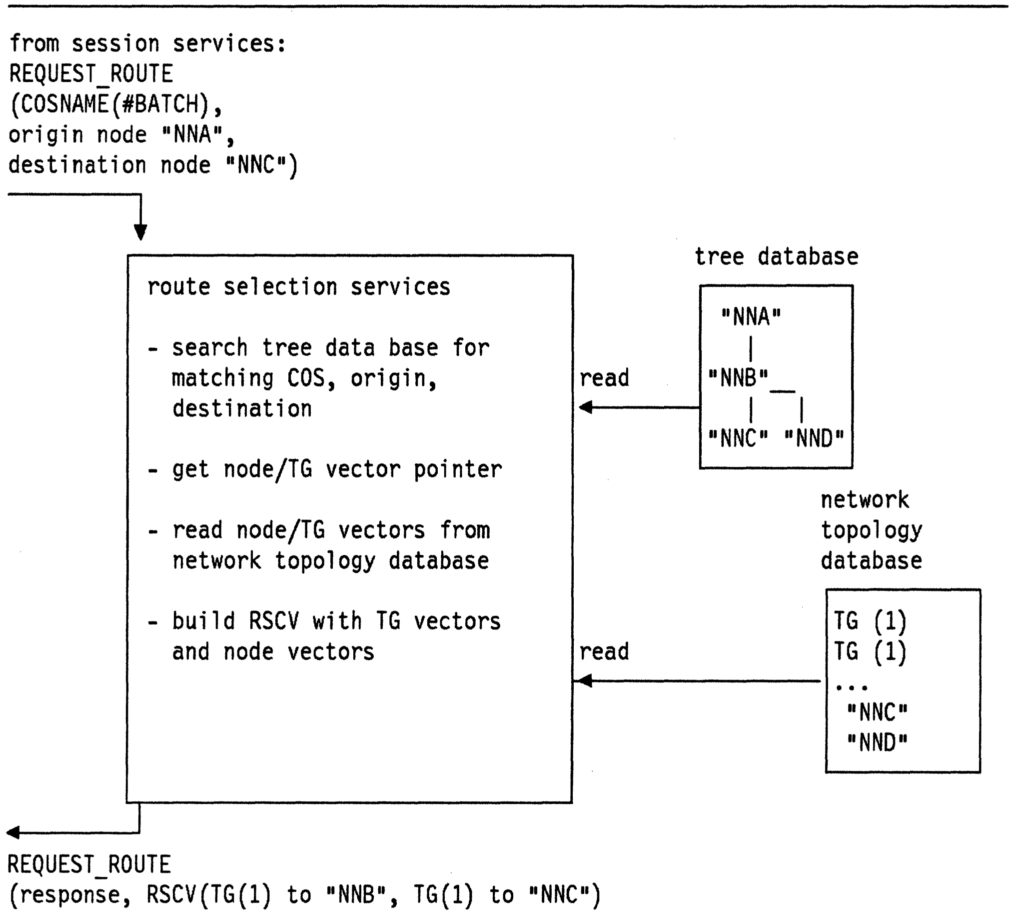

Figure 39. Route Selection in a Network Node Using the Tree Database

4.7.7 Route Selection for End Node with CP-CP Sessions

We use the network in Figure 40 on page 67 for the next example. Like the previous one, it has four interconnected APPN network nodes. Additionally, it has two APPN end nodes each connected to two network nodes. End node ″ENA″ has CP-CP sessions to network node ″NNA″. End node ″ENB″ has CP-CP sessions to network node ″NND″. Still, we are in network node ″NNA″ and consider the processing done by its route selection services as a service for end node ″ENA″.

LU1 in end node ″ENA″ requests a session to LU7. A process (to be considered later in "Session Initiation Using Network Node Server" on page 117) locates the destination LU. It finds out that LU7′s CP is called ″ENB″. A request for the best route is sent to route selection services in the network node server.

If the network node ″NNA″ had just the information that is in the topology database, it would probably come up with a route going through nodes ″NNA″, ″NNB″, ″NND″ to node ″ENB″. This is most likely not the best route. So, route selection services in the network node needs additional information: This information is carried in the route request, which not only contains origin node, destination node, and class of service, but also TG vectors of the endpoint TGs (TG tail vectors). The origin APPN end nodes include TG vectors for those of their TGs that go to APPN network nodes or to virtual routing nodes. In this case, the tails are from "ENA" to "NNA", and to "NNC", and from "ENB" to "NND" and to "NNC".

One TG tail vector of end node "ENA" points route selection services to network node "NNA", for which there is a tree in the database. The tree can be extended by the tail vectors. The weights of the TGs can be computed according to class of service and added to the weights in the tree, resulting in the weights of the new routes.

The other TG tail vector of end node "ENA" points route selection services to network node "NNC" for which (we assume) there is no tree in the database. This tree has to be calculated from scratch. Part of the building process of a tree is the computation of TG weights for a certain class of service. This process was described in an earlier example on page 62.

Looking at Figure 41 on page 68, you notice that now we use a *network* topology database, which also includes node vectors. (The *local* topology database contains TG vectors, only.)

The node weights are computed in a similar process, as presented for TG weights. Node weights and TG weights are then added to form the route weight. Several routes are possible, and the tree with the lowest weight routes is stored.

Note, that for performance reasons, an implementation may choose to compute a tree for the whole network or to stop the building of the tree as soon as the

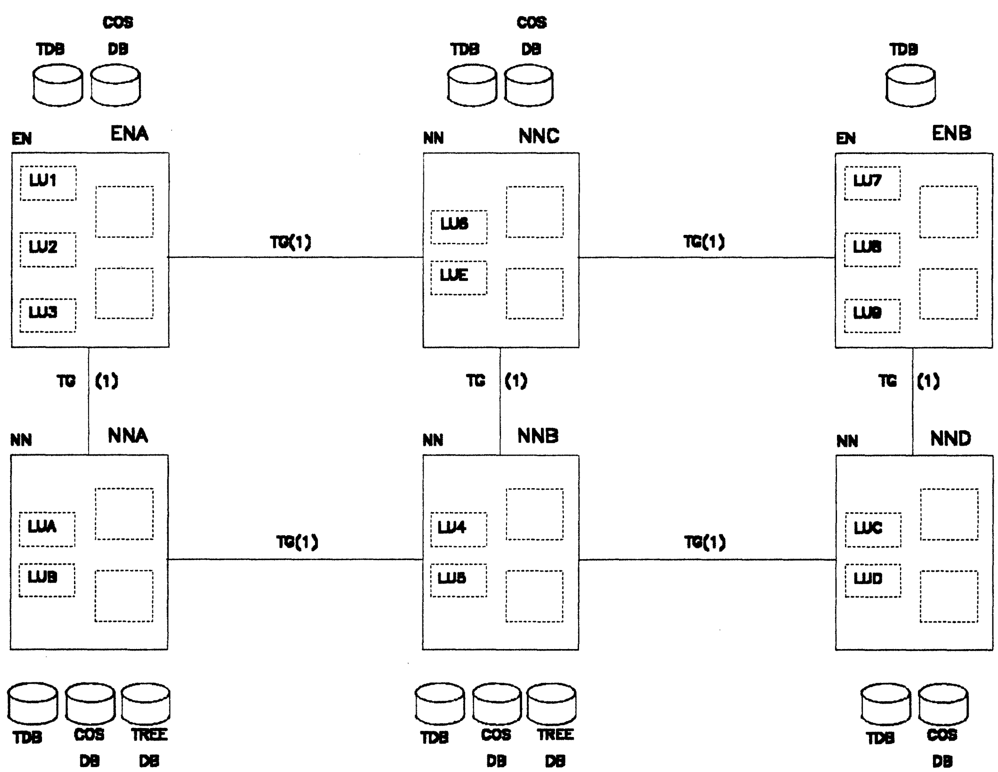

Figure 40. Example Configuration with APPN Network Nodes and APPN End Nodes

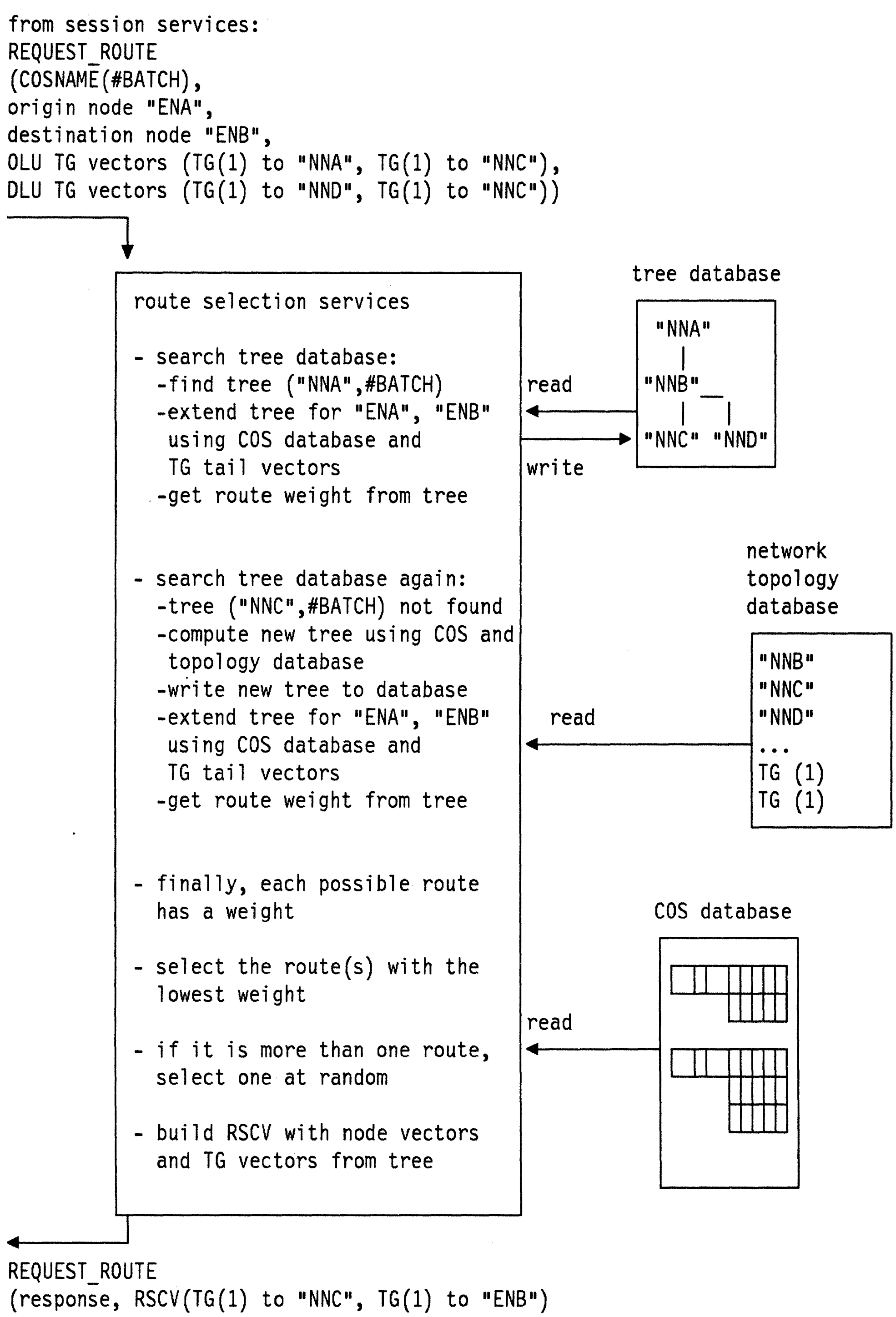

Figure 41. Route Selection in a Network Node Computing New Trees

required node is reached. In our example, the tree computed for node "NNC" may be very small. Anyway, the TG tail vectors are added to the new tree and the weight for the route from "ENA" through "NNC" to "ENB" can be calculated. The weights of the routes are compared and the one with the lowest weight is chosen.

The Least Weight Algorithm
The route computation algorithm is a modified version of the "shortest path algorithm". Given a root network node and a class of service, the algorithm constructs a tree whose branches contain every other topologically reachable network node in the routing portion of the network, under the constraints of the class of service. The tree represents the least weight paths from the network node at the root to each network node in the tree for the class of service.

Details of the "least weight algorithm" are found in scientific literature. The "Related Publications" on page xv gives a reference. In this document, we make no attempt to prove the validity of the algorithm. Only a few interesting features are pointed out:

- The tree may be computed in part. As soon as the requested destination node is put on the tree, the computation may be stopped.
- The tree is expandable. When a new network node joins the network, in many cases it can be stuck onto existing trees without complete re-computation of the tree. Similarly, the TG tail vectors of end nodes can be added to existing trees. This is an important performance aspect.
- The running time of the algorithm is proportional to the number of TGs. The number of network nodes is less significant.
- Once a tree is computed, it may be cached and used for subsequent route requests, having the same network node as the root and using the same class of service.

4.7.8 Special Cases

The following functions are described only briefly. They are based on the examples given above.

Route Selection for End Node without CP-CP Sessions

In the previous example, we assumed that there were CP-CP sessions between the APPN end nodes and their serving APPN network nodes. The CP-CP session is used to send a LOCATE/CDINIT from the end node to the network node, in order to obtain the best route through the network. (The LOCATE/CDINIT is explained in "Session Initiation Using Network Node Server" on page 117.) The network node will return the best route in an RSCV.

Now, let's assume an end node that does not have a CP-CP session to the adjacent network node. The configuration is still as shown in Figure 40 on page 67. As before, LU1 in the end node "ENA" requests a session to LU7 in end node "ENB". End node "ENA" selects a route to adjacent network node "NNA" (as described in detail, LU7 is defined as being in "NNA"). "ENA" selects a route to "NNA" (as described in detail on pages 58 to 64). End node "ENA" sends a BIND to network node "NNA" and lets network node "NNA" find the destination LU. Network node "NNA" is able to do that and the session will be established. Here, "NNA" acts as pseudo-server for "ENA".

But:
as opposed to LOCATE/CDINIT, the BIND does not carry TG tail vectors. That is why network node "NNA" does not know that there is a "shortcut" between end node "ENA" and end node "ENB". The session will go through network node "NNA".

The solution, in this case, will be for end node "ENA" to pre-define the destination LU as being in network node "NNC". This way, the route from "ENA" through "NNC" to "ENB" can be enforced.

Route Selection through Network Nodes for LEN End Node

Whereas end nodes have the choice, whether or not they establish CP-CP sessions to their adjacent network node, a LEN end node never has CP-CP sessions to adjacent nodes. A LEN end node relies completely on pre-definition. For session establishment it can only send a BIND. As far as routing is concerned, it works much the same as explained above in "Route Selection for End Node without CP-CP Sessions" on page 69. When the BIND response comes back to the network node, it will pass it on to the LEN end node. As seen from the LEN end node, LU7 is in the adjacent network node.

Route Selection Using Virtual Routing Nodes

Just as endpoint TGs to network nodes, endpoints to virtual routing nodes must be incorporated into the route request to route selection services.

When TG vectors exist for endpoint TGs to virtual routing nodes from the origin node and from the destination node, route selection services must check to see if the origin endpoint TG to a virtual routing node and the destination endpoint TG to a virtual routing node share a common hidden virtual node as partner node. A hidden virtual routing node is not included in the trees constructed by route selection services.

A visible virtual routing node is included in the trees constructed by route selection services. It does not require special processing.

Route Computation for Directory Services

In addition to computing routes for sessions, route selection services also computes routes for LOCATE/FINDs that directory services sends when attempting to verify the location of a resource. The route for LOCATE/FIND is a sequence of CP names from the origin node to the destination node.

Using a COS name of CPSVCMG, route selection services constructs a tree that represents the least weight route from the network node at the root to all network nodes in the tree. Thus, route selection services uses the same algorithm as for session routes.

4.8 Source Routing versus Hop-by-Hop Routing

Source routing is the method of routing where the description of the route is part of the message. Thus, the nodes that do the routing do not need any information on how to reach the destination node. The origin node (source node) has predetermined the route. The routing of the BIND using the route selection control vector (RSCV) would be an example. Source routing is favored where the intermediate nodes cannot do much processing. This may be either because they lack intelligence or because the links are too fast.

Hop-by-hop routing, on the other hand, uses information contained in the routing nodes. The message contains an identifier by which the routing node can determine over which link the message is to be transmitted next. The routing of normal data on LU-LU sessions is done using this routing method, with the session-connectors being the information in the routing node and the local-form session identifier (LFSID) being the identifier in the message. Hop-by-hop routing requires some intelligence in the intermediate node. It has the advantage of predictable response times and reliable methods for problem isolation.

APPN uses both kinds of routing.

4.9 SNA Defined Modes and Classes of Service

Generally speaking, each installation is free to choose its mode names and COS names. When a large number of installations connect, that can lead to considerable mode and COS tables. This is caused by the fact that nodes exchange and check mode names. To avoid unnecessary table maintenance, SNA has now defined mode names and related COS names which satisfy the requirement for differentiated classes of service.

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.

Below is a list of the SNA-defined names. The contents of the COS tables are described in *Systems Network Architecture Type 2.1 Node Reference*. The contents of the modes are described in *Systems Network Architecture LU 6.2 Reference: Peer Protocols*.

Mode Name	Corresponding COS Name
default	#CONNECT
#BATCH	#BATCH
#INTER	#INTER
#BATCHSC	#BATCHSC
#INTERSC	#INTERSC
CPSVCMG	CPSVCMG
SNASVCMG	SNASVCMG

The character "#" represents the hexadecimal value X'7B'.

Chapter 5. Directory Services

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

5.1 Initialization

The directory services component is created and initialized by the node operator facility at node initialization time. The node operator facility starts the initialization process when the "START_NODE" command is issued by the node operator. The node operator facility derives specific parameters from the "START_NODE" command and passes these parameters to directory services, such as:

- 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 with the network node server (APPN end node only).

For example, in Figure 42 on page 74, the node operator issues the "START_NODE" command. On receipt of the "START_NODE" command the node operator facility initializes the node. One of the node operator facility initialization functions is to create and initialize directory services. The node operator facility derives the following parameters from the "START_NODE" command and passes these parameters to directory services:

- APPN end node
- The control point name is "ENB"
- The network ID is ITSC
- Register resources with network node server.

At the end of its initialization phase, directory services stores the control point name in its local directory database.

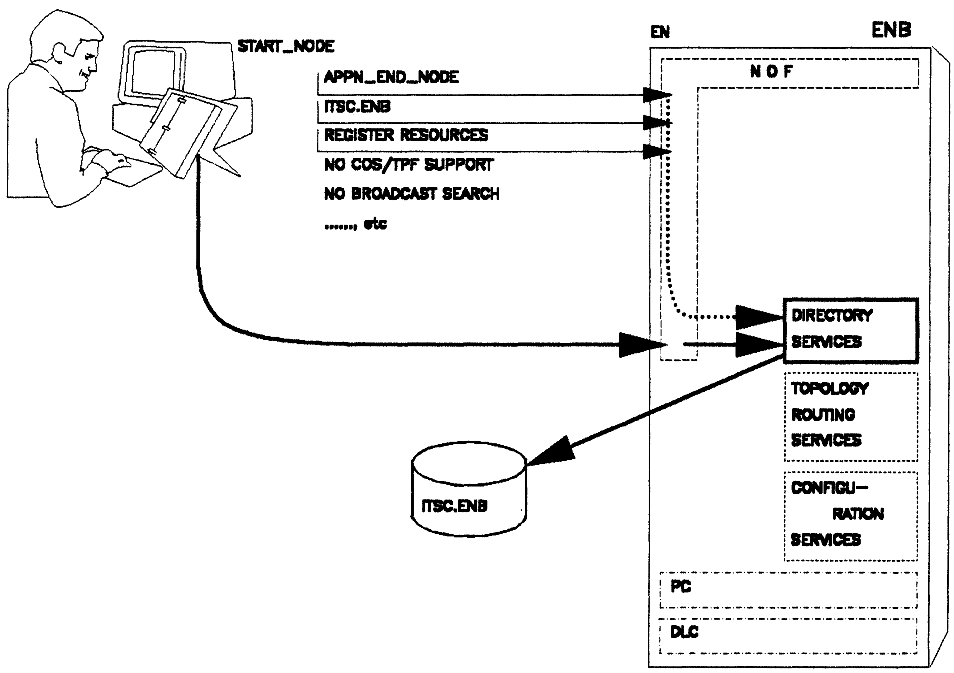

Figure 42. Directory Services Initialization

5.2 Local Directory Database

The local directory database is used by directory services to maintain awareness about network resources and their location in the APPN network. These resources may be located in the node itself, in adjacent nodes, or non-adjacent nodes. In the latter case, directory services derives the resource name and location from successful search requests.

The local directory database can be viewed as a collection of directory entries where each directory entry describes the name and location of a resource. The lay-out of a directory entry is as follows:

- The network qualified name of the resource
- The resource type, either:
  - "LU" (logical unit)
  - "ENCP" (APPN end node or LEN end node)
  - "NNCP" (APPN network node)

  Other resource types, for example user ID, transaction program (TP), may be easily added to the directory database in future extensions to APPN.
- Parent "ENCP" or "NNCP" entry (for an LU entry)

- An indicator that specifies whether the resource needs to be registered with its network node server (APPN end nodes only)
- The registration status, either:

  Resource is not registered
  Resource registration in progress
  Resource is registered.

Conceptually, the directory database hierarchy could look like this:

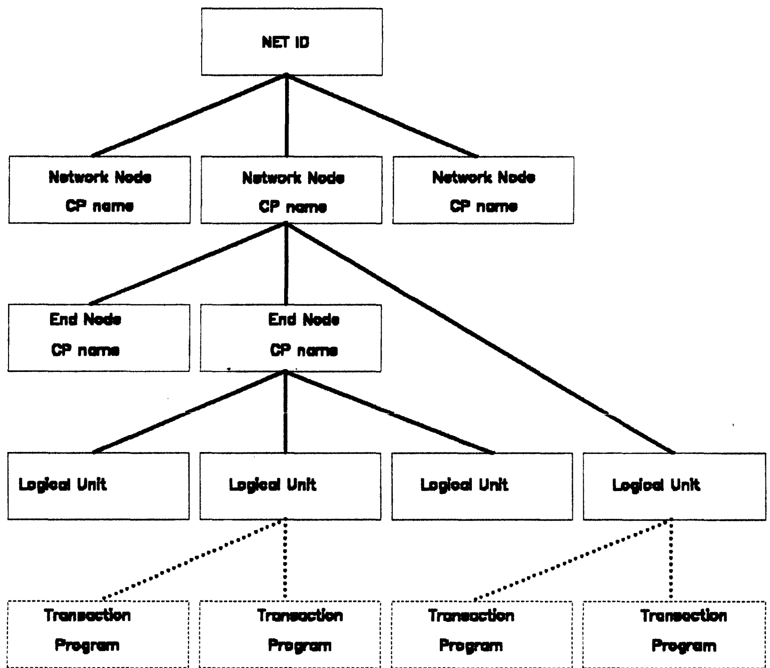

Figure 43. Directory Database Hierarchy

Three types of directory entries can be distinguished in a directory database:

- System defined directory entry
- Registered directory entry
- Cached directory entry.

The following sections describe each type of directory entry in more detail using an example configuration as depicted in "APPN Network Configuration used in this Document" on page 12.

5.2.1 System Defined Directory Entry

The system defined directory entries are created by node operators. The node operator interfaces with the node operator facility to define the resources in the local directory database. Each resource that can be searched for in the APPN network must have been defined once by a system defined directory entry. The definition is done on the node that owns the resource. Once the resource is defined the distributed search facilities of the network node server can be used to locate the resource on any APPN node in the network.

The node operator has four commands at his disposal that update the directory database. They are:

- ″START_NODE″ command

  The primary function of ″START_NODE″ command is to initialize the node and its components (see "Initialization" on page 73 for details on directory services initialization). At the end of the initialization phase ″START_NODE″ command generates a ″DEFINE_DIRECTORY_ENTRY″ command to register the control point name in the directory database.
  The ″DEFINE_DIRECTORY_ENTRY″ command is described in more detail later.

- ″DEFINE_ADJACENT_NODE″ command

  The ″DEFINE_ADJACENT_NODE″ command is used to define an adjacent node to the local node. The parameters that may be specified are:

  - The network qualified name of the adjacent node
  - Whether CP-CP sessions are supported with the adjacent node
  - Adjacent node type, that is LEN end node, APPN end node, or APPN network node
  - List of LU names local to the adjacent node.

    This parameter is not needed if adjacent node LUs can be located through services provided by an APPN network node. However, if these services cannot be obtained through an APPN network node, then this parameter must specify all the adjacent node LUs that are referenced by the local node. For example, a LEN end node does not support CP-CP session with an APPN network node. Therefore, a LEN end node must specify all adjacent node LUs that are referenced by the LEN end node. Alternatively, an APPN node (end node or network node) that connects to an adjacent LEN end node must define the LEN end node LUs that are referenced by the APPN node.

  In Figure 44 on page 77, APPN end node ″ENB″ could have defined network nodes ″NNC″ and ″NND″ as follows:

  ```
  DEFINE_ADJACENT_NODE  ITSC.NNC
                        No CP-CP sessions supported
                        NETWORK NODE

  DEFINE_ADJACENT_NODE  ITSC.NND
                        CP-CP sessions supported
                        NETWORK NODE
  ```

The definition for APPN end node "ENB" in APPN network node "NND" could have been:

```
DEFINE_ADJACENT_NODE  ITSC.ENB
                      CP-CP sessions supported
                      END NODE
```

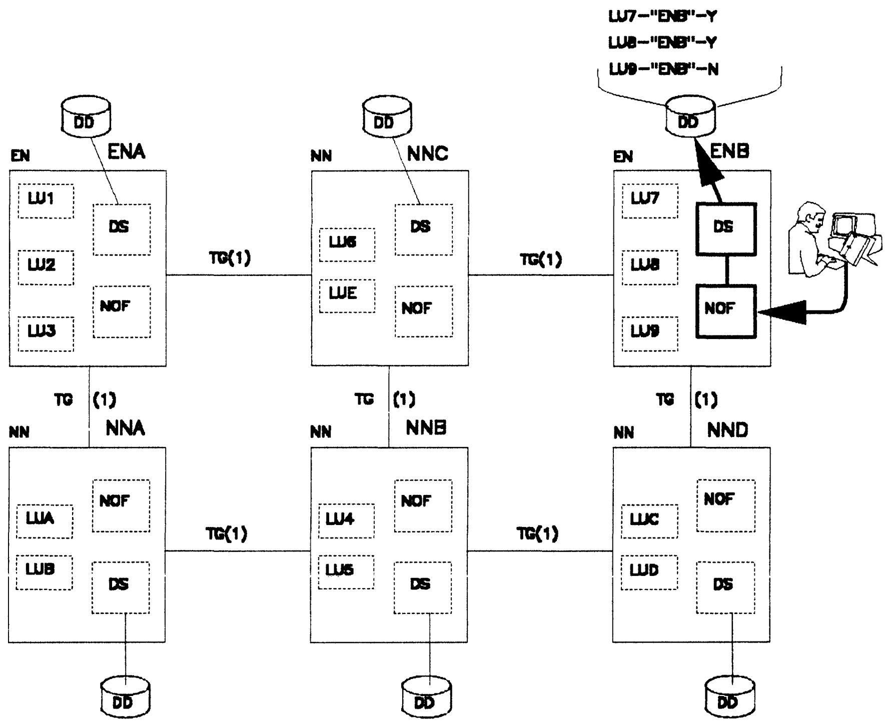

Figure 44. System Defined Directory Entries for Local Resources

The "DELETE_ADJACENT_NODE" command will remove the adjacent node definition from the local directory database. It requires only one parameter and that is the network qualified name of the adjacent node.

- "DEFINE_DIRECTORY_ENTRY" command

  The "DEFINE_DIRECTORY_ENTRY" command is used to define or change an existing directory entry. The parameters specified with the "DEFINE_DIRECTORY_ENTRY" command are:

  - The network qualified name of the resource
  - The type of resource
  - Whether the LU should be registered with the network node server
  - The name of the parent resource (optional)
  - The parent's resource type (optional).

The "DELETE_DIRECTORY_ENTRY" command removes the entry from the directory database. It requires the following parameters:

- The name of the resource to be deleted
- The resource type
- The name of the parent resource (optional)
- The parent's resource type (optional).

- "DEFINE_LOCAL_LU" command

  The "DEFINE_LOCAL_LU" command defines a new local LU to the node. Besides updating the node's internal tables and activating the LU for use by a transaction program (TP), the LU is also registered in the local directory database. Some of the parameters used for registration are:

  - The network qualified name of the logical unit
  - Whether the LU should be registered with the network node server

  For example, in Figure 44 on page 77, the node operator at "ENB" defined resources LU7, LU8, and LU9 in the local directory database. The resources LU7 and LU8 were defined as registerable with the network node server, but LU9 was not.

```
DEFINE_LOCAL_LU  LU7
                 REGISTERABLE
                 ... etc

DEFINE_LOCAL_LU  LU8
                 REGISTERABLE
                 ... etc

DEFINE_LOCAL_LU  LU9
                 NOT REGISTERABLE
                 ... etc
```

  The node operators at the other nodes use the same process to define their local LUs in their node's directory database.

  The "DELETE_LOCAL_LU" command will remove the local LU definition from the node's internal tables and directory database. The only parameter to be specified is the network qualified name of the resource to be removed.

A node operator may also choose to define the resources of an adjacent node into its local directory database. This allows directory services to locate resources on adjacent nodes without using the directory services from its network node server. This latter approach is needed for LEN end nodes. A LEN end node does not support CP-CP sessions with an adjacent APPN network node. Thus, it cannot use the directory services of an APPN network node to search for LUs unknown to the directory services at the LEN end node.

Consequently, a LEN end node requires that all resources of adjacent nodes that are accessed by its node, are registered in its local directory database. Alternatively, if the adjacent node needs access to LEN end node resources, then these resources must be defined at the adjacent node.

For example, in Figure 45 on page 79, LEN end node "LNX" connects to APPN network node "NNA". The LEN end node cannot establish CP-CP sessions with "NNA" and therefore cannot request assistance from network node "NNA" to locate resources unknown to the LEN end node. If, for example, LUX at "LNX" requests a session with LUB, then directory services at "LNX" cannot locate resource LUB. As a result, the session request will be denied by "LNX".

However, if LUX at "LNX" requests a session with LUA, then directory services at "LNX" can locate LUA in its local directory database and derive from the directory entry that LUA is located at "NNA". Directory services at "LNX" was capable of locating network resource "LUA" because the node operator at "LNX" defined LUA, owned by "NNA", in its local directory database:

```
DEFINE_DIRECTORY_ENTRY LU(LUA)
                       PARENT RESOURCE("NNA")
                       PARENT RESOURCE TYPE(NNCP)
```

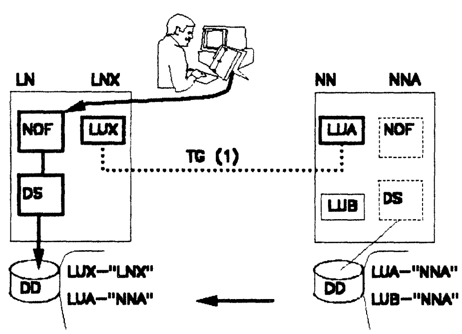

Figure 45. System Defined Directory Entry for Remote Resources on LEN End Nodes

The definition of remote resources is not required for APPN nodes, although there could be reasons to define remote resources on APPN nodes too. For example, if an LU on an APPN end node frequently requires a session with an LU on an adjacent node, then the definition of that LU in the local directory database of the APPN end node will bypass the request for assistance from the network node server and thus reduce the overhead imposed on the network node server.

For example, in Figure 46 on page 80, LU2 at "ENA" frequently requires a session with LU6 at "NNC". If the node operator at "ENA" does not define remote resource "LU6" in its local directory database, then directory services at "ENA" needs assistance from its network node server "NNA" to locate the resource. To reduce the number of assists from directory services at "NNA", for locating the network resource, the node operator at "ENA" decides to define LU6 (owned by "NNC") in its local directory database:

```
DEFINE_DIRECTORY_ENTRY LU(LU6)
                       PARENT RESOURCE("NNC")
                       PARENT RESOURCE TYPE(NNCP)
```

This way directory services at "ENA" is able to locate the resource in its local directory database and the session can be activated without the assistance of directory services at its network node server.

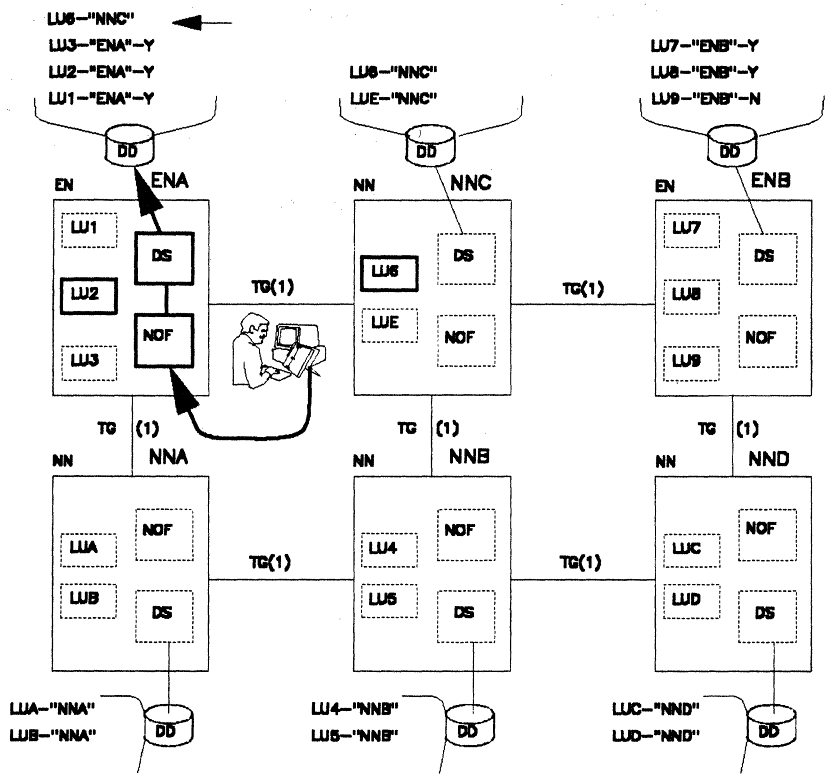

Figure 46. System Defined Directory Entry for Remote Resources on APPN End Nodes

5.2.2 Registered Directory Entry

The registered directory entries are temporary entries in the local directory database of an APPN network node. These entries represent the local LUs of an adjacent APPN end node for which the APPN network node acts as network node server. As soon as an APPN end node establishes CP-CP sessions with its network node server, the APPN end node may register its local resources with the network node server. The APPN end node can register all its local resources with the network node server or a selective part. At resource definition time the node operator at an APPN end node defines for each resource, whether or not it needs to be registered with the network node server.

Directory services at the APPN end node uses the "Register Resource GDS variable" (X'12C3') to register its resources with the network node server. The resources that need to be registered with the network node server are derived from the local directory database, enveloped in one or more control vectors, and added to the "Register Resource GDS variable" data stream. In addition, directory services changes the status of the selected resource in its local directory database to *pending*.

If all resources are processed, or the maximum "Register Resource GDS variable" length is reached (1024 characters), the variable is sent across the CP-CP session to the network node server. On positive response from the network node server, directory services at the APPN end node changes the status of the selected resources from *pending* to *registered*.

For example, in Figure 47, the node operator at "ENB" has defined resources LU7 and LU8 to be registerable (Y) and LU9 not (N). As soon as "ENB" has established CP-CP sessions with its network node server "NND", directory services at "ENB" will register resources LU7 and LU8 with its network node server. The "Register Resource GDS variable" from "ENB" looks like this:

```
Register Resource GDS variable((LU7),(LU8))
```

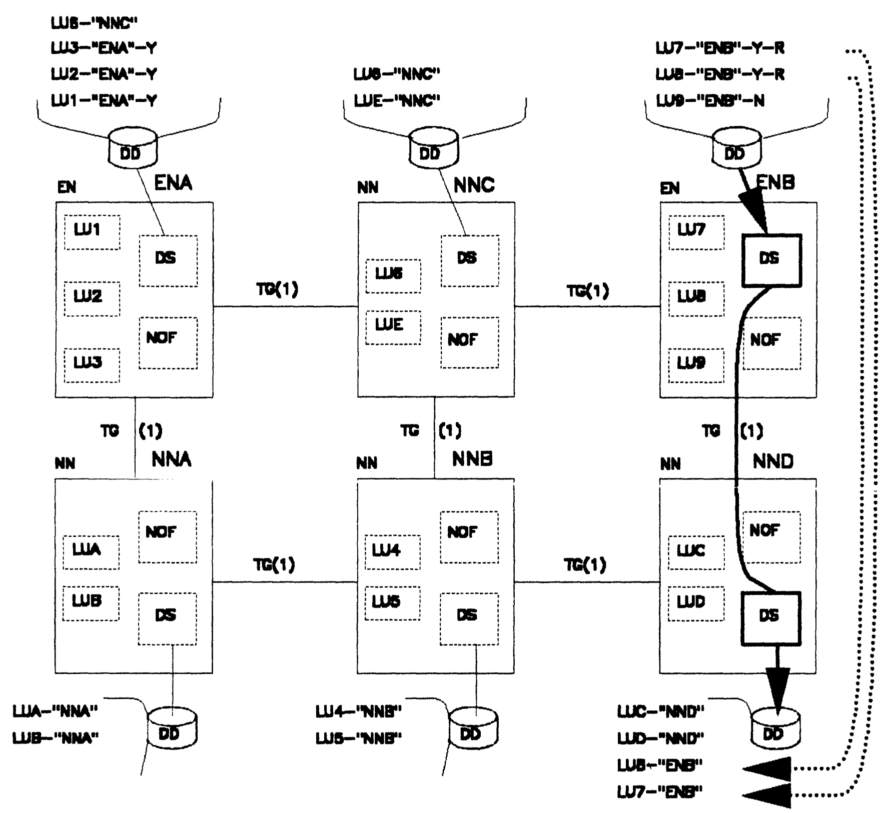

Figure 47. Registered Directory Entry

Directory services at the APPN end node will be responsible for maintaining the correct information for its resources in the network node server directory database. Thus, if the node operator at the APPN end node adds, deletes, or updates local resources, directory services updates its local directory database and its resource information with the network node server.

For example, if a node operator adds a new resource to its local directory database and identifies the resource as registerable, directory services will add the resource to its local directory database and to the network node server directory database using the "Register Resource GDS variable" with control vectors identifying the new resource.

The node operator can also delete a local resource that is registered with the network node server. In that case, directory services will delete the resource from its local directory database and from the network node server directory database using the "Delete GDS variable" (X'12C9') with control vectors identifying the resource to be deleted.

Finally, the node operator may also update a local resource that is registered with the network node server. As there is no update variable available, directory services will first delete the resource with the "Delete GDS variable" and then add the resource with the updated information using the "Register Resource GDS variable".

As soon as the CP-CP sessions with the APPN end node are deactivated, the network node server automatically deletes all the registered entries for that APPN end node.

5.2.3 Cached Directory Entry

An APPN network node may also register resource information for resources that were previously unknown to the node. For example, in Figure 48, LU9 at "ENB" initiates a session with LUD.

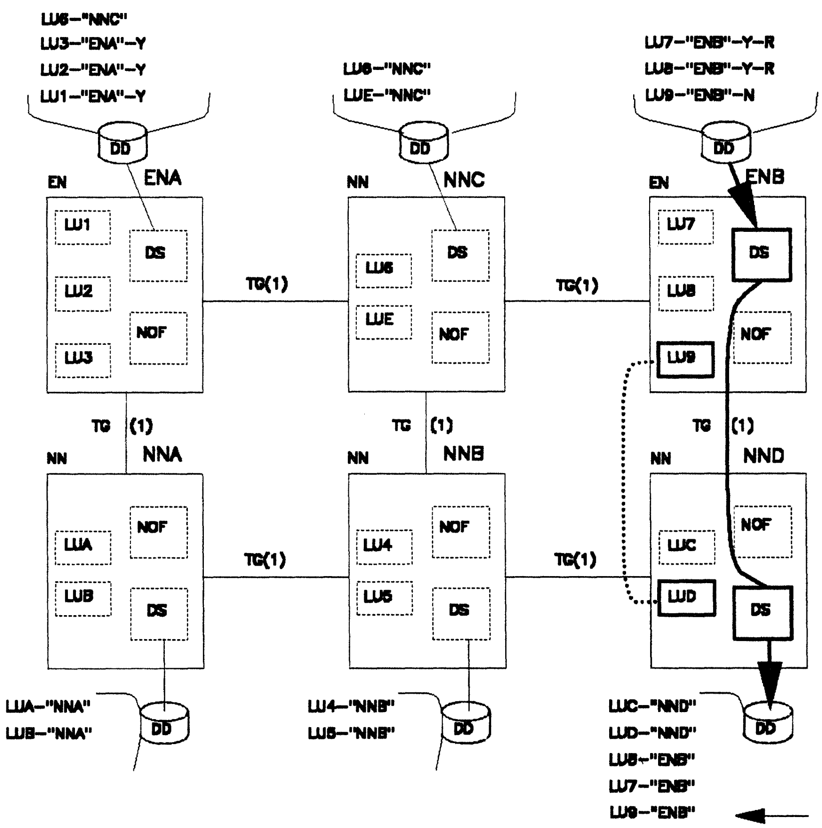

Figure 48. Cached Directory Entry

Directory services at "ENB" cannot locate resource LUD and therefore requests assistance from its network node server to locate resource LUD. Upon receiving the request, directory services at "NND" may register LU9, owned by "ENB". in its local directory database.

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. Even the node's ability to search for resources may be severely affected if no provisions are made to reduce the number of cache entries.

However, it is up to the implementation of the APPN network node function to decide whether cache entries are created, whether the cache entries will be saved and stored, and how the cache entries will be maintained.

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.

A network node that caches resource entries which 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.

5.3 Distributed Directory Database

The distributed directory database can be regarded as a "virtual" database that contains a list of all the resources in the network. In the example APPN network in Figure 49 on page 84, all nodes have established CP-CP sessions with their adjacent nodes. Connecting the APPN nodes together creates, for each APPN node, a distributed directory database that consists of a union of local directory databases. Each node can search for a resource in its local directory database and use the "locate search request" to search the directory databases of all other nodes in the network.

5.4 Locate Network Resources

The primary function of directory services is to search the distributed directory database for specific network resources. If the network resource is located, directory services obtains location specific information from the node owning the network resource and returns this information to the initiator of the request. Location specific information includes the resource name, the control point name of the owning node, and the control point name of its network node server.

The originator of the request can be session services that assist an origin LU (OLU) in establishing a session with a destination LU (DLU). Session services defines the request using an internal "Locate_Message" command that includes, amongst others, the name of the DLU and OLU, and the "Cross-Domain Initiate GDS variable". This variable is a specific session services variable that is destined for session services at other nodes. Directory services will only transport the "Cross-Domain Initiate GDS variable", concatenated to the "Locate GDS variable", back and forth between session services at origin and end node. For more information, see "Session Initiation for APPN End Node" on page 112.

The directory services function on LEN end nodes is restricted to the local directory database of the LEN end node. If the LEN end node cannot locate the resource on its local directory database, directory services at the LEN end node returns a "locate failure" to the initiator of the request.

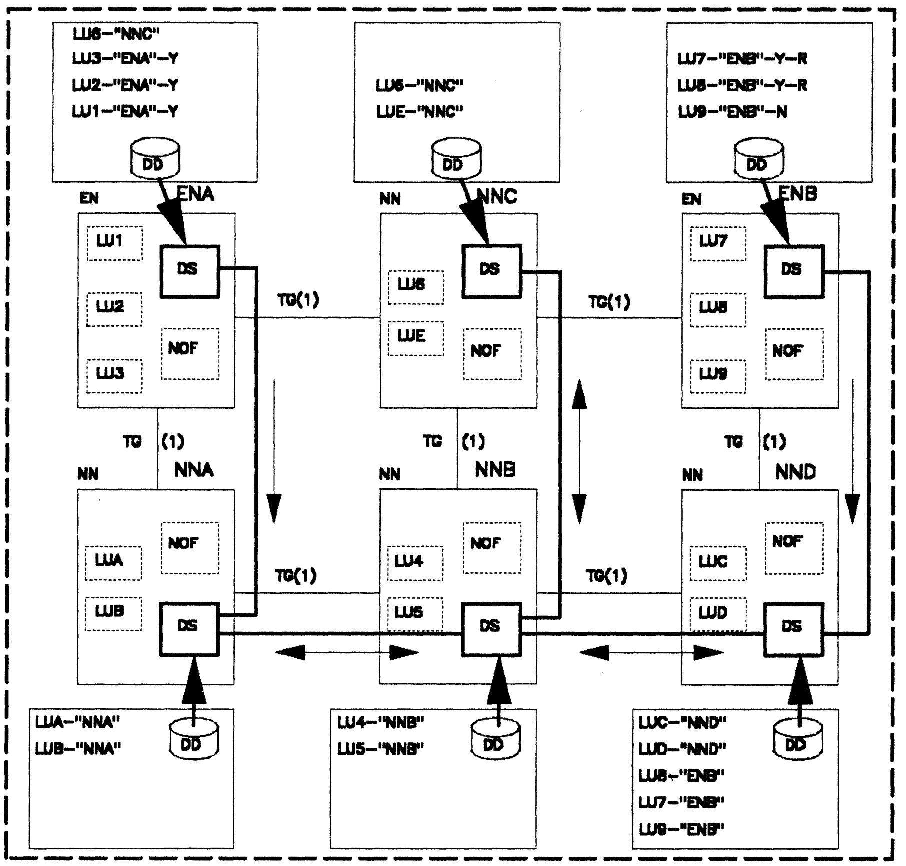

Figure 49. Distributed Directory Database

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 the search request fails at the APPN end node, the APPN end node automatically forwards the search request to its network node server. Directory services at the APPN network node is responsible for a distributed search through the network to locate the resource. If the DLU is located in the network, the availability of the DLU is verified and the DLUs location-specific information is returned to the OLU. In the following sections the search capability of LEN end nodes are addressed first, then the search capabilities of APPN end nodes, and finally the distributed search capabilities of APPN network nodes.

Logical Unit Name Equals Control Point Name

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

Before a network node server for an OLU searches its local directory database for the requested DLU, it checks with topology and routing services to see whether the LU name is equal to one of the control point names for the network nodes in the topology database. If the LU name is equal to an active network node, then directory services does not need to perform the directory search, nor to verify whether the DLU is available. An active topology database entry means the resource is located and available.

Please note: the topology database only contains network nodes; therefore, the LU needs to be located on a network node to bypass the directory search.

5.4.1 LEN End Node Locate Search

LEN end nodes do not support CP-CP sessions with an APPN network node; thus, a LEN end node cannot use the distributed search capabilities provided by a network node server. The search for a resource on a LEN end node therefore, is restricted to its local directory database.

Session services at a LEN end node requests directory services at its node to search for a resource using the "Request_Local_Search" command. This command contains two parameters. They are:

The fully qualified procedure correlation ID (FQPCID)

The requested destination LU (DLU) name.

On receipt of the command, directory services searches its local directory database for the requested DLU. In case the DLU is located, directory services returns the owning control point name of the DLU to session services. If not located in the local directory database, directory services will return the appropriate sense code to session services indicating "resource not found".

The following two examples will use the configuration as described in Figure 50. In the first example, LUX at "LNX" initiates a session with LUA. Directory services at "LNX" receives the following request from session services:

```
Request_Local_Search(request,FQPCID(1),DLU(LUA))
```

Directory services at "LNX" searches its local directory database for resource LUA. The resource is located and the reply returned to session services will be:

```
Request_Local_Search(reply,FQPCID(1),DLU(LUA at "NNA"))
```

In the second example, LU2 at "LNX" initiates a session with LUB. Directory services at "LNX" receives the following request from session services:

```
Request_Local_Search(request,FQPCID(2),DLU(LUB))
```

Directory services at "LNX" searches its local directory database for resource LUB. The resource cannot be located in the local directory database, therefore directory services will return the following reply to session services:

```
Request_Local_Search(reply,FQPCID(2),sense("resource not found"))
```

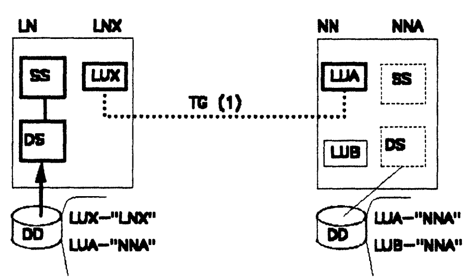

Figure 50. LEN Local Search

5.4.2 APPN End Node Locate Search

The search request received by directory services at an APPN end node differs from the search request received by directory services at a LEN end node. At a LEN end node the "Request_Local_Search" command was used by session services to locate a resource.

However, for APPN end nodes session services uses the "Locate_Message" command. The command contains, amongst others, the FQPCID, the origin and destination LU names, and the "Cross-Domain Initiate GDS variable".

Session services uses the "Cross-Domain Initiate GDS variable" to exchange information with the session services components at the destination node and their respective network node servers. For more information on "Cross-Domain Initiate GDS variable" see "Session Initiation for APPN End Node" on page 112.

Directory services regards the "Cross-Domain Initiate GDS variable" as user data, hence, it will not process the variable. It merely includes the variable in its search request to its network node server.

On receipt of the "Locate_Message" command from session services, directory services at an APPN end node will first search its local directory database for the DLU. If the search is successful, directory services will derive the DLU's owning control point name from the directory database and include it in the "Locate_Message" command. Directory services will further change the request status to reply in the command and discard the "Cross-Domain Initiate GDS variable". The session services variable is discarded because the variable is destined for session services at the network node server.

In the following example in Figure 51 on page 87 LU2 at "ENA" requests a session with LU6 at "NNC". As you may recall, in "System Defined Directory Entry" on page 76, the node operator at "ENA" defined resource LU6 at "NNC" in its local directory database to bypass the frequent searches on its network node server. Directory services receives the following command from session services:

```
Locate_Message(request,FQPCID(3),OLU(LU2 at "ENA"),DLU(LU6)),
  Cross-Domain Initiate GDS variable(OLU=PLU,mode name(#INTER),FQPCID(3),
        TG vector(TG(1) to "NNA",TG(1) to "NNC"))
```

Directory services at "ENA" searches its local directory database for resource LU6, locates the information, and returns the following response to session services:

```
Locate_Message(reply,FQPCID(3),OLU(LU2 at "ENA"),DLU(LU6 at "NNC"))
```

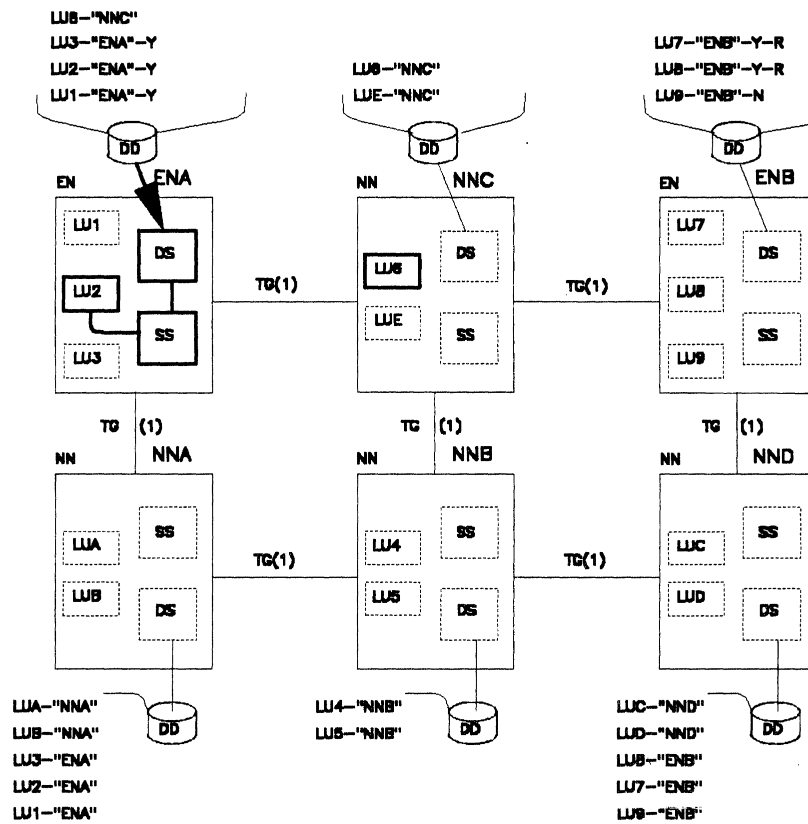

Figure 51. End Node Locate Search for Remote Resource

However, if the DLU cannot be located in the local directory database, then directory services at the APPN end node generates the "locate search request". The "locate search request" consists of the following three variables:

- The "Locate GDS variable" that contains, amongst others, the FQPCID and an indicator that this is a request.
- The "Find Resource GDS variable" that contains vectors that describe the origin LU (OLU) and its owning control point name, and a vector that describes the destination LU (DLU).
- The "Cross-Domain Initiate GDS variable" is a session services variable that will be regarded as user data by directory services.

Directory services at the APPN end node forwards the "locate search request" to its network node server. In case the DLU is located by the network node server, directory services at the APPN end node will receive the "locate search reply" The "locate search reply" consists of the following three variables:

- The "Locate GDS variable" that contains, amongst others, the FQPCID and an indicator that this is a reply.
- The "Found Resource GDS variable" that contains the DLU name, its owning control point name, and the control point name of DLU's network node server.
- The "Cross-Domain Initiate GDS variable" reply from session services at the network node server.

In the following example, Figure 52, the node operator at "ENA" has deleted resource LU6, owned by "NNC", from its local directory database.

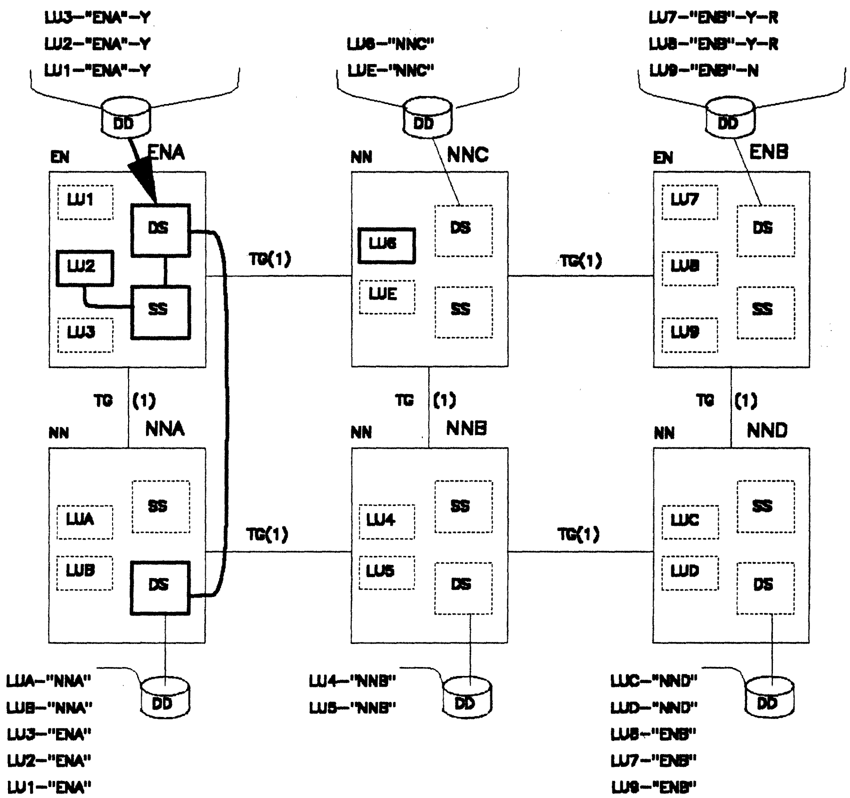

Figure 52. End Node Locate Search for Resource on Network Node

If LU2 at "ENA" now requests a session with LU6, directory services at "ENA" receives the following request from session services:

```
Locate_Message(request,FQPCID(4),OLU(LU2 at "ENA"),DLU(LU6)),
  Cross-Domain Initiate GDS variable(OLU=PLU,mode name(#INTER),FQPCID(4),
           TG vector(TG(1) to "NNA",TG(1) to "NNC"))
```

Directory services at "ENA" cannot locate resource LU6 in its local directory database. It will now generate the "Locate GDS variable" and "Find Resource GDS variable" and send the variables together with the "Cross-Domain Initiate GDS variable" across the CP-CP session towards directory services at its network node server ("NNA"):

```
Locate GDS variable(request,FQPCID(4)),
  Find Resource GDS variable(OLU(LU1 at "ENA"),DLU(LU6)),
  Cross-Domain Initiate GDS variable(OLU=PLU,mode name(#INTER),FQPCID(4),
           TG vector(TG(1) to "NNA",TG(1) to "NNC"))
```

If the DLU is located by "NNA", directory services at "ENA" receives the following response from its network node server:

```
Locate GDS variable(reply,FQPCID(4)),
  Found Resource GDS variable(DLU(LU6 at "NNC")),
  Cross-Domain Initiate GDS variable(OLU=PLU,FQPCID(4),COS/TPF(2,#CONNECT),
            RSCV(TG(1) to "NNC"))
```

As a result, directory services at "ENA" returns the following reply to session services:

```
Locate_Message(reply,FQPCID(4),OLU(LU2 at "ENA"),DLU(LU6 at "NNC")),,
  Cross-Domain Initiate GDS variable(OLU=PLU,FQPCID(4),COS/TPF(2,#CONNECT),
            RSCV(TG(1) to "NNC"))
```

In case network node server "NNA" could not locate the resource, directory services at "ENA" receives the following "locate search reply" from directory services at its network node server:

```
Locate GDS variable(reply,FQPCID(4),sense("resource not found"))
```

As a result, directory services at "ENA" would return the following response to session services:

```
Locate_Message(reply,FQPCID(4),sense("resource not found"))
```

5.4.3 Broadcast Search

In the previous section, directory services at "ENA" requested assistance from its network node server "NNA" to locate a network resource. If a network node server receives a "locate search request" from one of its end nodes, then directory services at the network node server first tries to locate the resource in its own local directory database. If the resource cannot be located, directory services at the network node server forwards the "locate search request" to all its adjacent network nodes requesting these nodes to assist the network node server to locate the resource. This process is known as *"broadcast search"*.

Each network node that receives a "broadcast search" from its adjacent network node is responsible for:

- Checking whether the "broadcast search" has already been received from one of the other network nodes. If so, the network node informs the sender that the resource can not be located at this node.

  For example, in Figure 53 there are four network nodes. They are NNa, NNb, NNc, and NNd. NNa and NNd are connected to both NNb and NNc. If NNa initiates the "broadcast search", it will send the "locate search request" to NNb and NNc, and both nodes will propagate the "locate search request" to node NNd. Assuming NNd will first receive the "locate search request" from NNb then NNd will send "resource not found" reply to the "locate search request" from NNc. It may well be that NNd, or a node beyond NNd, owns the requested resource. In that case, NNd will send the positive reply to NNb.

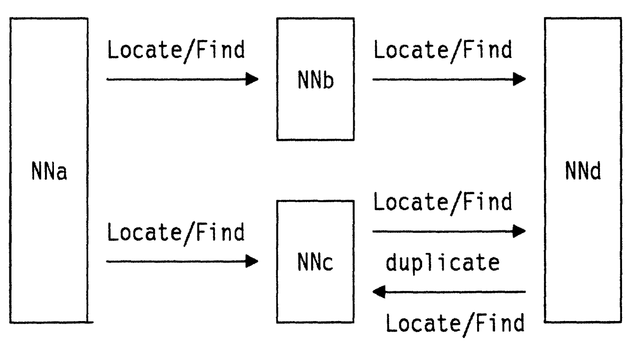

Figure 53. Network Node Broadcast Search

- Propagation of the "broadcast search" to all its adjacent network nodes excluding the node from which the "broadcast search" is received. The propagation of "broadcast search" is irrespective of the fact that the DLU may be located in the network node's directory database.
- Maintain a status of all "broadcast search" requests that are sent to adjacent network nodes. The responses from the adjacent nodes are consolidated, and as soon as all responses are received, the reply is sent to the originator of the "broadcast search".
- If the DLU is located by one of the nodes, then the originator of the "broadcast search" is informed immediately. However, the network node keeps on collecting responses from other network nodes for the same "broadcast search" until all responses are received. As soon as the last response is received the network node sends the consolidated reply to the originator of the request.

We will describe the "broadcast search" in more detail using the example from the configuration in Figure 54. In this example LU1 at "ENA" requires a session with LU7.

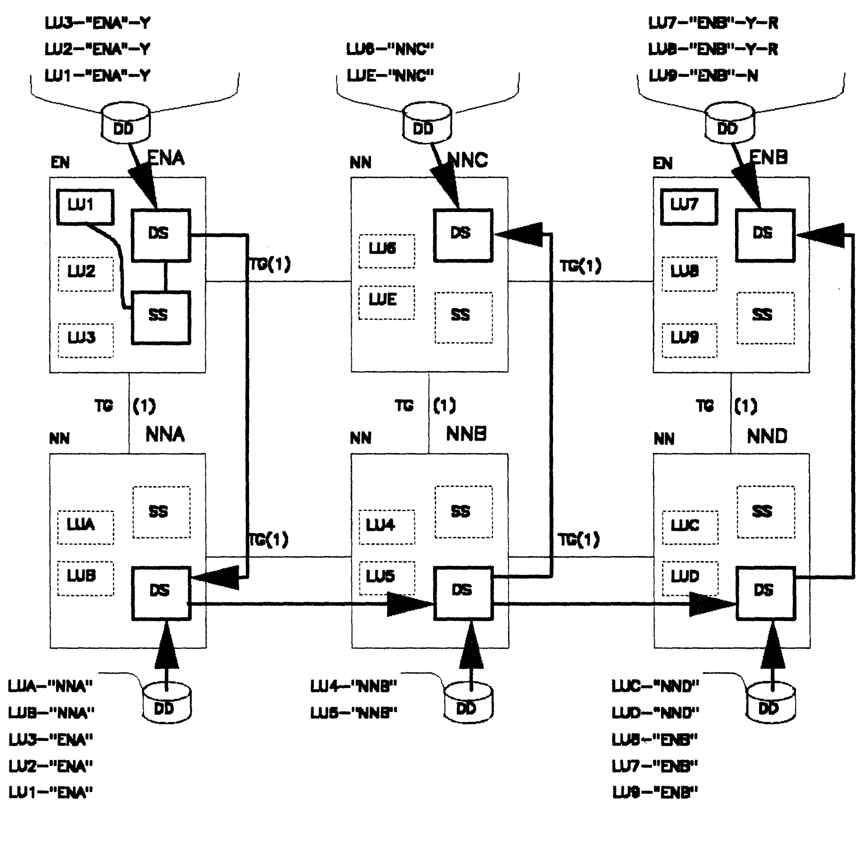

Figure 54. End Node Locate Search for Resource on End Node

Session services at "ENA" generates the following request for directory services:

```
Locate_Message(request,FQPCID(5),OLU(LU1 at "ENA"),DLU(LU7)),
  Cross-Domain Initiate GDS variable(OLU=PLU,mode name(blanks),FQPCID(5),
         TG vector(TG(1) to "NNA",TG(1) to "NNC"))
```

Directory services at "ENA" cannot locate DLU LU7 in its local directory database. Therefore, directory services generates the "Locate GDS variable" and "Find Resource GDS variable", and forwards the variables together with the "Cross-Domain Initiate GDS variable" to directory services at its network node server "NNA". The "locate search request" looks like this:

```
Locate GDS variable(request,FQPCID(5)),
  Find Resource GDS variable(OLU(LU1 at "ENA"),DLU(LU7)),
  Cross-Domain Initiate GDS variable(OLU=PLU,mode name(blanks),FQPCID(5),
         TG vector(TG(1) to "NNA",TG(1) to "NNC"))
```

On receipt of the "locate search request" from "ENA", directory services at "NNA" searches its local directory database for DLU LU7. As the result of the search is negative, directory services at "NNA" will now initiate the "broadcast search" to its adjacent network nodes. Conceptually this "broadcast search" flows between network nodes as follows:

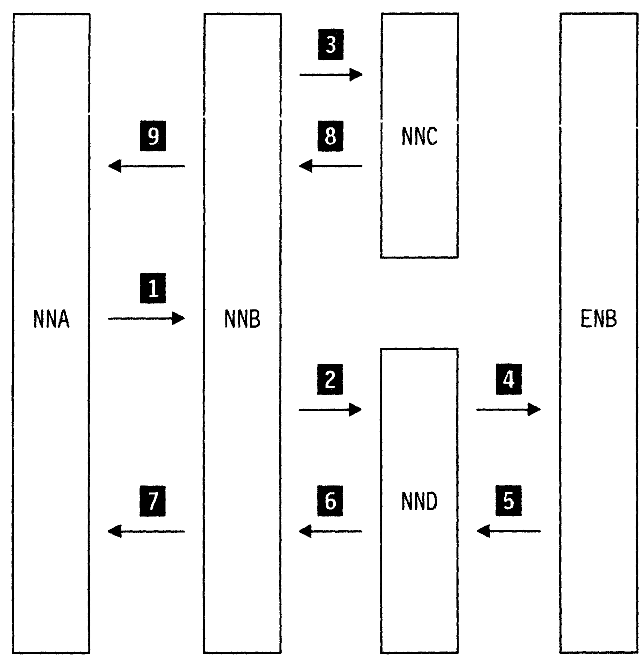

Figure 55. Conceptual View of Broadcast Search

1 "NNA" forwards the "locate search request" to adjacent network node "NNB".

2 "NNB" forwards the "locate search request" to adjacent network node "NND".

3 "NNB" forwards the "locate search request" to adjacent network node "NNC". "NNB" then searches its local directory database for DLU LU7. Although the DLU is not located, "NNB" will withhold his response to "NNA" until replies have been received from adjacent nodes "NNC" and "NND".

4 ″NND″ locates DLU LU7 in its local directory database. APPN end node ″ENB″ is identified as owner therefore the ″locate search request″ is sent to ″ENB″ to verify whether DLU LU7 is still available on ″ENB″

5 ″ENB″ generates the ″Found Resource GDS variable″ and includes in the ″Found Resource GDS variable″ the requested location information about the DLU. The request status in the ″Locate GDS variable″ is changed to reply and the ″Locate GDS variable″ together with the ″Found Resource GDS variable″ is returned to ″NND″.

6 ″NND″ may cache the location information of the origin LU that is OLU name LU1, owning control point name ″ENA″, and the control point name of network node server ″NNA″. ″NND″ sends the ″locate search reply″ from ″ENB″ to ″NNB″ together with the information that ″NND″ has completed the ″broadcast search″ (all responses received).

7 Although ″NNB″ is still waiting for a reply from the ″broadcast search″ to ″NNC″, ″NNB″ sends the ″Locate GDS variable″ with the ″Found Resource GDS variable″ from ″ENB″ to ″NNA″ and includes in his reply an indicator that ″NNB″ did not complete the ″broadcast search″.

8 ″NNC″ does not have any connections to network nodes other than the connection to ″NNB″. As ″NNB″ has sent the request to ″NNC″, ″NNC″ will not broadcast the request to ″NNB″ anymore. As ″NNC″ cannot locate the resource in its local directory database, ″NNC″ informs ″NNB″ that the ″broadcast search″ has completed and that the resource can not be located using ″NNC″.

9 On receipt of the response from ″NNC″, ″NNB″ has completed its ″broadcast search″ and informs ″NNA″ that the resource has not been located. Please note that the ″Locate GDS variable″ with the ″Found Resource GDS variable″ from ″ENB″, sent to ″NNA″ earlier, will not be repeated in ″NNB′s″ final reply.

On receipt of the positive reply from ″NNB″, ″NNA″ may cache the location dependent information for the DLU. That is, resource LU7, owning control point name ″ENB″, and its network node server ″NND″. In Figure 56 on page 93, both ″NNA″ and ″NND″ have cached the information.

Directory services at ″ENA″ receives the following ″locate search reply″ from directory services at its network node server:

```
Locate GDS variable(reply,FQPCID(5)),
  Found Resource GDS variable(DLU(LU7 at "ENB" served by "NND")),
  Cross-Domain Initiate GDS variable(OLU=PLU,FQPCID(5),COS/TPF(2,#CONNECT),
            RSCV(TG(1) to "NNC",TG(1) to "ENB"))
```

Consequently, directory services at ″ENA″ returns the following reply to session services:

```
Locate_Message(reply,FQPCID(5),OLU(LU1 at "ENA"),DLU(LU7 at "ENB")),
  Cross-Domain Initiate GDS variable(OLU=PLU,FQPCID(5),COS/TPF(2,#CONNECT),
            RSCV(TG(1),"NNC",TG(1),"ENB"))
```

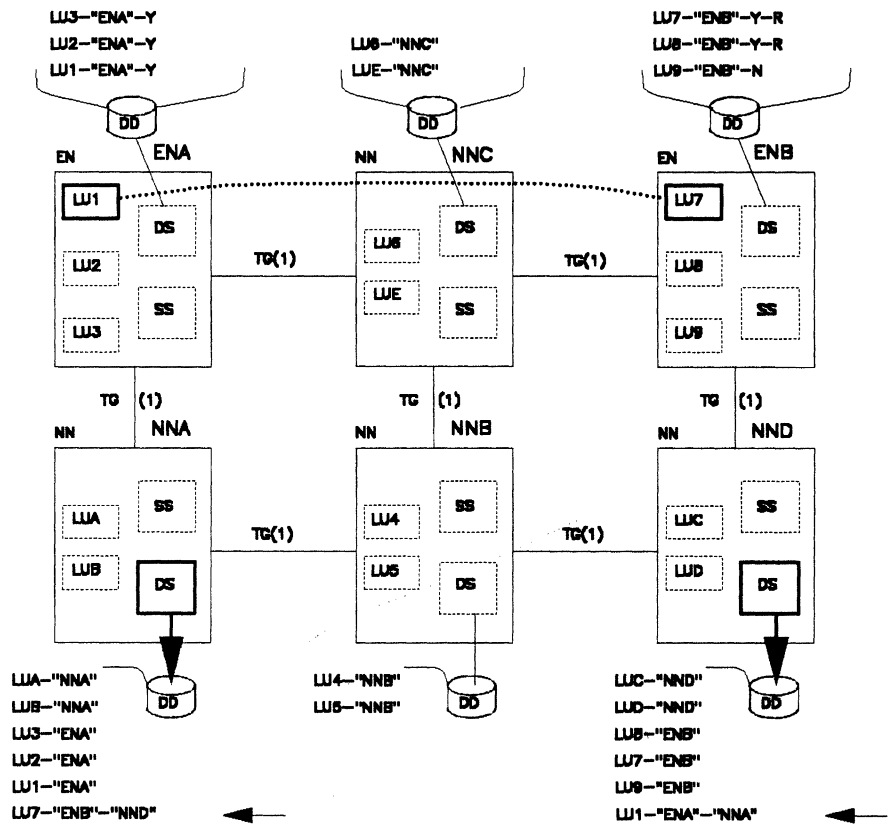

Figure 56. Successful Broadcast Search

5.4.4 Directed Search

When a network node server receives a locate ("locate search request") from one of its LUs, directory services at the network node server first searches its local directory database for the requested DLU. The result of the search could be either successful or not. The previous section, "Broadcast Search" on page 89, describes the latter situation. In this section, the successful result is described, that is, the DLU is located in the network node server's directory database.

If directory services at the network node server locates the DLU in its local directory database, directory services obtains the DLUs network node server control point name from the directory entry and requests topology and routing services to provide the route towards this control point. The route selection control vector returned by topology and routing services is included in the "Locate GDS variable", and directory services at the network node server forwards the request towards the first network node described in the route selection control vector.

Each network node receiving the request sends the "locate search request" to the next network node identified in the route selection control vector. The final destination network node verifies the availability of the DLU with the DLU's owning node. The DLU's owning node generates the "Found Resource GDS variable" and includes the "Found Resource GDS variable" in the "locate search reply" that is sent back to the OLU's network node server. On receipt of the "locate search reply", directory services at the network node server forwards the reply to the originator of the "locate search request".

In the previous section, "Broadcast Search" on page 89, LU1 at "ENA" successfully initiated a session with LU7 at "ENB". As a result of the successful initiation the network node servers, "NNA" and "NND", cached in their local directory database LU7 and LU1 respectively.

The next time LU1 at "ENA" initiates a session with LU7, directory services at "ENA" receives the following search request:

```
Locate_Message(request,FQPCID(6),OLU(LU1 at "ENA"),DLU(LU7)),
  Cross-Domain Initiate GDS variable(OLU=PLU,mode name(blanks),FQPCID(6),
         TG vector(TG(1) to "NNA",TG(1) to "NNC"))
```

Please note, an APPN end node does **not** cache information about DLUs that were previously engaged in a session with one of its local LUs. Directory services at "ENA" cannot locate DLU LU7 in its local directory database. Therefore, directory services generates the "Locate GDS variable" and "Find Resource GDS variable", and forwards the variables together with the "Cross-Domain Initiate GDS variable" to directory services at its network node server "NNA". The "locate search request" looks like this:

```
Locate GDS variable(request,FQPCID(6)),
  Find Resource GDS variable(OLU(LU1 at "ENA"),DLU(LU7)),
  Cross-Domain Initiate GDS variable(OLU=PLU,mode name(blanks),FQPCID(6),
         TG vector(TG(1) to "NNA",TG(1) to "NNC"))
```

On receipt of the "locate search request", directory services at "NNA" searches its local directory database for DLU LU7 and finds that LU7 is owned by "ENB" and that the network node server for "ENB" is "NND". Next, Directory services at "NNA" sends the "REQUEST_ROUTE" command to topology and routing services and requests topology and routing services to provide the route between "NNA" and "NND":

```
REQUEST_ROUTE(request,Origin-Node("NNA"),
              Destination-Node('NND"))
```

The "REQUEST_ROUTE" command reply from topology and routing services is:

```
REQUEST_ROUTE(reply,sense(0),
             RSCV(TG(1) to "NNB",TG(1) to "NND"))
```

On receipt of the reply from topology and routing services, directory services includes the route selection control vector in the "Locate GDS variable". As destination LU LU7 is located on end node "ENB", directory services generates the network name control vector that identifies "ENB" as the final destination node, and adds the network name control vector to the "Locate GDS variable". The "locate search request" is then forwarded to the first node in the route selection control vector, in our example "NNB":

```
Locate GDS variable(request,FQPCID(6),Network Name control vector("ENB"),
         RSCV(TG(1) to "NNB",TG(1) to "NND")),
  Find Resource GDS variable(OLU(LU1 at "ENA"),DLU(LU7)),
  Cross-Domain Initiate GDS variable(OLU=PLU,mode name(blanks),FQPCID(6),
         TG vector(TG(1) to "NNA",TG(1) to "NNC"))
```

When "NND" receives the "locate search request", it derives from the network name control vector that the "locate search request" is destined for end node "ENB". Consequently, "NND" strips off the network name and route selection control vector from the "Locate GDS variable" and forwards the "locate search request" to "ENB" to verify whether the LU is still available. Ultimately, the "Found Resource GDS variable" is created by "ENB" and returned to the OLU's network node server.

5.4.5 Directed Search Failure

In the previous section "Directed Search" on page 93, network node "NNA" initiated a successful directed search request for DLU LU7, a resource that was previously cached by "NNA". If the network node server builds and sends a directed search to the destination network node server, the situation could arise that DLU LU7 is no longer available or that one of the nodes (or the TG towards one of the nodes) is no longer available. Hence, directory services at "NNA" receives a negative response on the directed search.

As a result of the negative response, directory services at "NNA" deletes the cached entry for LU7 and initiates a "broadcast search" for DLU LU7. The outcome of this command is dependent on the situations which led to the negative response for the directed broadcast. The DLU may be found again or the DLU may still not be available. Both cases are handled in "Broadcast Search" on page 89.

Let us take an obvious example. APPN end node "ENB" in Figure 56 on page 93 deactivates its CP-CP sessions with "NND". This implies that network node "NND" will remove all registered entries for "ENB" from its local directory database. In Figure 57 on page 96, the configuration is described that exists if "ENB" has completed its deactivation sequence.

Now LU1 at "ENA" initiates a session with LU7. Directory services at "ENA" can not locate LU7 in its local directory database; therefore, "ENA" forwards the "locate search request" to its network node server "NNA". "NNA" learns from its local directory database that LU7 is owned by "ENB" and that "NND" is the network node server of "ENB". Directory services at "NNA" requests topology and routing services to provide the route to "NND", generates the network name control vector, and adds both vectors to the "Locate GDS variable". "NNA" uses the route selection control vector, returned by topology and routing services, to forward the "locate search request" to destination node "NND".

On receipt of the request, "NND" derives from the network name control vector that the "locate search request" is destined for end node "ENB". As end node "ENB" has deactivated its CP-CP sessions, the "locate search request" can not be delivered by "NND" to its final destination node "ENB". As a result, directory services at "NND" sends the negative reply to "NNA" with the status condition "specified end node no longer served". On receipt of the negative reply, "NNA" discards the cached entry for LU7 and initiates a "broadcast search" to locate LU7 in the network.

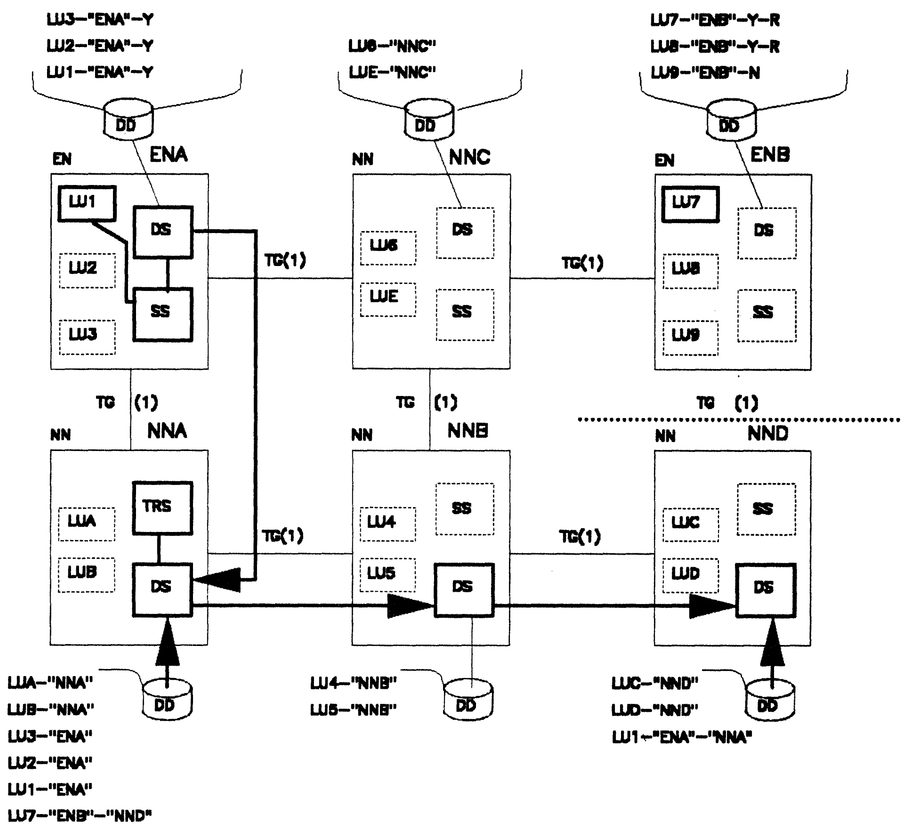

Figure 57. APPN End Node "ENB" Disconnects

5.4.6 APPN End Node Search Support

Normally, APPN end nodes register their local resources with their network node server when CP-CP sessions are established. The resources of an APPN end node remain registered with the network node server as long as the CP-CP sessions between end node and network node remain active. The network node automatically removes the registered entries for an end node when the CP-CP sessions with the APPN end node are deactivated.

The APPN architecture provides the capability for APPN end nodes not to register its resources with the network node server, but request its network node server to propagate the "broadcast search" requests to the APPN end node for resources unknown to the network node server. This command is also referred to as the "domain search" because the APPN end node, unlike APPN network nodes, does not broadcast the search request to its adjacent nodes. On receipt of the "domain search" from its network node server, the end node merely checks whether the requested DLU is one of its local LUs and responds accordingly.

At node initialization time the node operator of the end node specifies whether the "domain search" support will be requested from the network node server. For more information see "Initialization" on page 103 and "Control Point Capabilities" on page 106.

Whether or not to use the "domain search" may be dependent upon the network configuration or the product implementing the function. For example, if the end node is a large processor that controls a large number of resources the registration flow between end node and network node may take a long time when the CP-CP sessions are activated. Thus, this type of end node could be a good candidate for requesting its network node server to be searched for resources unknown to the network node server. Another example could be a network node with limited database capacity such as a 3174. An end node with a large number of resources may deplete the database capacity of its 3174 network node server when it registers its resources with the network node.

5.4.7 Multiple LUs Found with Broadcast Search

In the previous chapters we have assumed that the network qualified LU name would be unique throughout the APPN network. However, to accomplish a unique LU name space for a network ID requires an LU naming convention, agreed and adhered to by all network administrators when defining the LUs for the nodes under their control.

Although most network installations have some naming convention in place naming conflicts may still arise. The most obvious cause is human error when defining the LU.

Therefore, the situation could arise that the broadcast search results in more than one node reporting the "resource found" condition. As you may recall from "Broadcast Search" on page 89, the "broadcast search" request is always propagated to adjacent network nodes irrespective of whether the LU is located on that node. In addition, the network node keeps on collecting broadcast search replies from its adjacent nodes despite that node's awareness of a previous "resource found" condition. Each "resource found" condition is reported back to the network node server of the origin LU.

In the current APPN architecture, the **first** "resource found" condition received by the network node server of the origin LU is accepted for session activation. Each subsequent "resource found" condition received by the network node server is simply discarded and an alert for this condition is generated.

For example, in Figure 54 on page 90 ("Broadcast Search"), LU1 at "ENA" requested a session with LU7. Let us assume that the node operator at node "NNC" defined LU7 on its node instead of LUE. That would result in a configuration as depicted in Figure 58 on page 98.

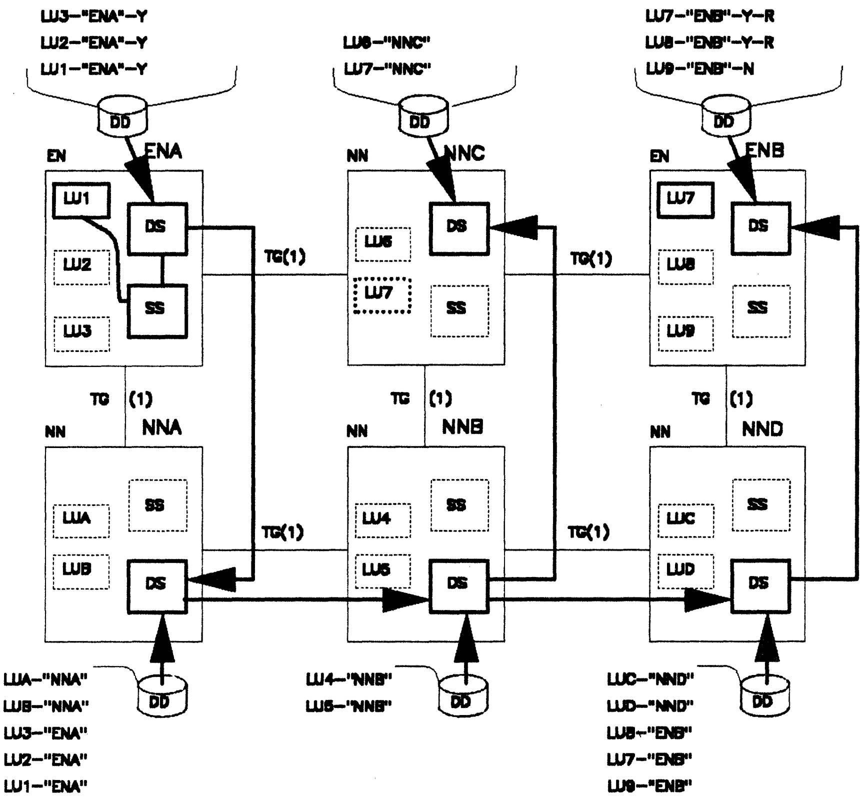

Figure 58. Non-Unique LU Definitions

In Figure 55 on page 91 the conceptual view was shown for the broadcast search of LU7 in the network. Assuming that the "resource found" condition from "ENB" would arrive first at node "NNB", then steps **1** through **7** would remain the same. Steps **8** and **9** will change as follows:

- **8** "NNC" does not have any connections to network nodes other than the connection to "NNB". As "NNB" has sent the request to "NNC", "NNC" will not broadcast the request to "NNB" anymore. As "NNC" locates the resource in its local directory database, it generates the "Found Resource GDS variable" and includes in the "Found Resource GDS variable" the requested location information about the DLU. The request status in the "Locate GDS variable" is changed to reply and the "locate search reply" is returned to "NNB".
- **9** On receipt of the response from "NNC", "NNB" sends the "locate search reply" from "NNC" to "NNA" together with the indicator that "NNB" has completed the "broadcast search" (all responses received). On receipt of the response from "NNB", "NNA" will discard the "resource found" condition from "NNC" as the reply from LU7 at "ENB" was received earlier.

5.4.8 Wildcards

LEN end nodes require that all network resources located on adjacent nodes are 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. A subarea domain normally supports a large number of network resources, either under its own control or controlled by attached subarea domains. As the subarea complex can only be supported as a LEN end node, connection of the subarea to a network node could result in a huge manual registration burden at the network node.

To reduce this registration burden, directory services provides a facility called wildcards. Directory services distinguishes two types of wildcards. They are:

- Full wildcards

  Full wildcards are represented with an asterisk (*). An asterisk results in a match for each network resource that is searched for by directory services. Therefore, the use of the full wildcard should be restricted to one network node only. If the full wildcard is defined on more than one network node, then each node returns the "resource found" condition for the broadcast request.

- Partially specified 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***.

When directory services receives a broadcast search it uses a specific order to search for the resource in its directory database:

- Directory services first searches its directory database for the explicit resource name entry.
- If it fails to find the explicit resource name, directory services will search all partially specified names.
- If that also fails, directory services will search for a full wildcard. If there is a full wildcard, directory services generates the "Found Resource GDS variable" and identifies in the variable that the resource has been found using a full wildcard.

On receipt of "Found Resource GDS variable" with the full wildcard indicator, the network node server for the origin LU temporarily stores the "Found Resource GDS variable" until all the broadcast search responses are received. As soon as all responses are received and if there is more than one "Found Resource GDS variable" for the resource, then the network node server selects the variable without the full wildcard indicator first.

The configuration in Figure 59 consists of five nodes. They are:

1. APPN network node "RED" is connected to network nodes "RAL" and "RTP". It owns one resource and that is REDLU1.
2. APPN network node "RAL" connects to network node "RED" and to LEN end node "ITSC". It has defined in its directory database three resources all pointing to LEN end node "ITSC":
   - Resource WTCR19
   - Partial wildcard ITSC*
   - Full wildcard (*).
3. APPN network node "RTP" connects to network node "RED" and to LEN end node "WTC". It has defined, in its directory database, two resources all pointing to LEN end node "WTC":
   - Partial wildcard WTC*
   - Full wildcard (*).
4. LEN end node "ITSC" connects to network node "RAL". It owns three resources. They are ITSCLU1, ITSCIL1, and WTCR19.
5. LEN end node "WTC" connects to network node "RTP". It owns three resources. They are WTCR195, WTCR197, and WTLU1.

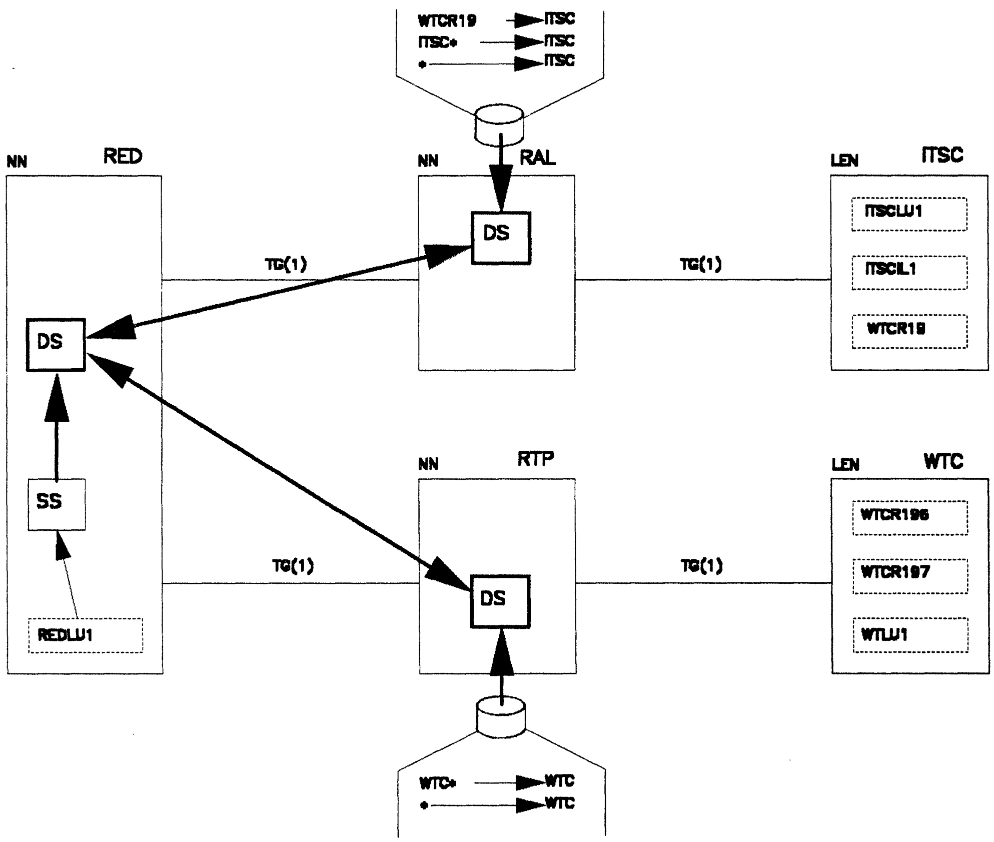

Figure 59. Directory Search Using Wildcards

In the first example, "REDLU1" at "RED" requests a session with "ITSCLU1". Directory services at network node "RED" cannot locate the resource and forwards a "broadcast search" to network nodes "RTP" and "RAL". Network node "RTP" finds the resource in its directory database using a full wildcard. It creates the "Found Resource GDS variable" and identifies in the variable that the resource was found using a full wildcard. The variable is forwarded to "RED". On receipt, network node "RED" will temporarily store the "Found Resource GDS variable" from "RTP" and await the replies from the other nodes, in our case "RAL".

Network node "RAL" finds the resource using a partially specified name. It creates the "Found Resource GDS variable" and indicates in the variable that the resource was located using an explicit name. The variable is forwarded to "RED". On receipt, network node "RED" will obtain the stored "Found Resource GDS variable" from "RTP" and select the "Found Resource GDS variable" that was found using the explicit name. Thus "RED" discards the variable received from "RTP".

In the next example, "REDLU1" at "RED" requests a session with "WTCR19". Directory services at network node "RED" cannot locate the resource and forwards a "broadcast search" to network nodes "RTP" and "RAL". Network node "RTP" finds the resource in its directory database using a partially specified name. It creates the "Found Resource GDS variable" and identifies in the variable that the resource was found using an explicit name. The variable is forwarded to "RED". On receipt, network node "RED" does not wait for the replies from the other nodes as the resource was located using an explicit name. "REDLU1" will be informed that "WTCR19" is located through "RTP". The subsequent BIND from "REDLU1" will fail because the resource is not available via "RTP".

Network node "RAL" finds the resource using the explicit name. It creates the "Found Resource GDS variable" and indicates in the variable that the resource was located using an explicit name. The variable is forwarded to "RED". On receipt, network node "RED" will discard the "Found Resource GDS variable" from "RAL" as the partner was already located on another node.

In the last example, "REDLU1" at "RED" requests a session with "WTLU1". Directory services at network node "RED" cannot locate the resource and forwards a "broadcast search" to network nodes "RTP" and "RAL". Network node "RAL" finds the resource in its directory database using a full wildcard. It creates the "Found Resource GDS variable" and identifies in the variable that the resource was found using a full wildcard. The variable is forwarded to "RED". On receipt, network node "RED" will temporarily store the "Found Resource GDS variable" from "RAL" and await the replies from the other nodes, in our case "RTP".

Network node "RTP" finds the resource in its directory database using a full wildcard. It creates the "Found Resource GDS variable" and identifies in the variable that the resource was found using a full wildcard. The variable is forwarded to "RED". On receipt, network node "RED" obtains the stored "Found Resource GDS variable" from "RAL". Both variables indicate that the resource was located using a full wildcard, hence, directory services selects the first reply from "RAL". As only one full wildcard is allowed in an APPN network, network node "RED" generates an error message and discards the "Found Resource GDS variable" from "RTP".

Off course the BIND that is sent by "REDLU1" to "RAL" fails, because "WTLU1" is not available through "RAL".

Use of wildcards

These examples were certainly not provided to discourage the use of wildcards. On the contrary, the use of wildcards is highly recommended, especially in those cases where the registration burden can be reduced significantly. However, **the use of wildcards requires a consistent naming convention across all nodes that allow the use of wildcards**.

Chapter 6. Session Services

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

6.1 Initialization

The session services component is created and initialized by the node operator facility at node initialization time. The node operator facility starts the initialization process when the ″START_NODE″ command is issued by the node operator. The node operator facility derives specific parameters from the ″START_NODE″ command and passes these parameters at initialization time to session services:

- The node type (LEN end node, APPN end node, or APPN network node)
- The control point name of the node
- The network ID of the node
- Whether the mapping of mode name to class of service and transmission priority Field, in short COS/TPF, is supported (end node only).

  For more information see "Class of Service Database" on page 52.
- Whether the ″domain search″ capability is supported (APPN end node only).

  For more information see "APPN End Node Search Support" on page 96.

Session services will save these parameters for later reference.

For example, in Figure 60 on page 104, the node operator facility derives the following parameters from the ″START_NODE″ command and passes these parameters to initialize session services:

- APPN end node
- Control point name ″ENB″
- Network ID ″ITSC″
- No COS/TPF support
- No ″domain search″

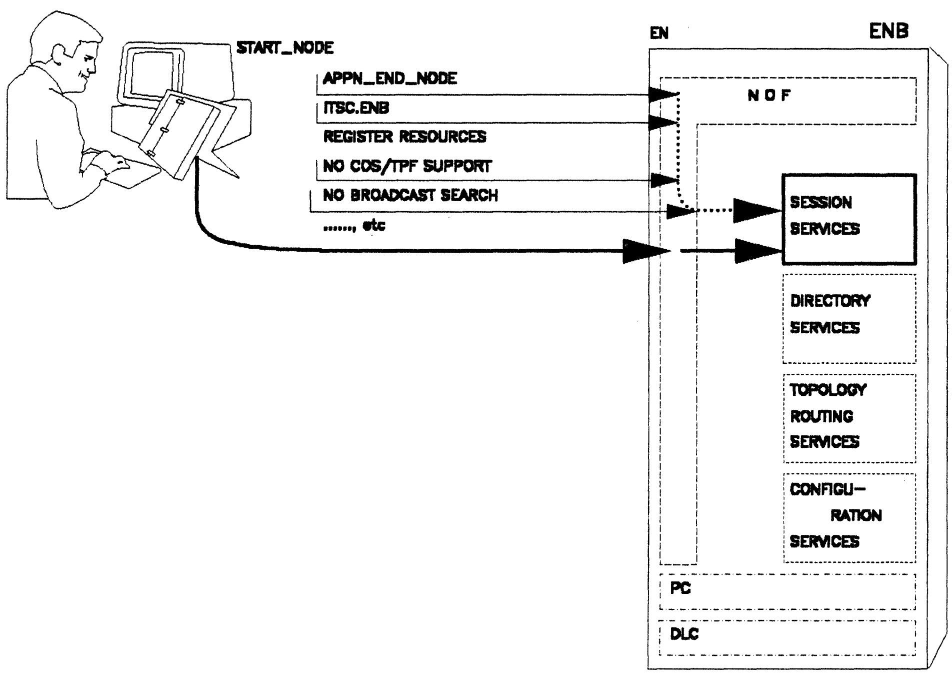

Figure 60. Session Services Initialization

6.2 Activate CP-CP Session

The control point components, that is topology and routing services, directory services, and session services, use CP-CP sessions with adjacent APPN nodes to exchange network information with their respective counterpart in the adjacent nodes. Examples of CP-CP session usage include the exchange of network topology updates between topology and routing services on adjacent network node, the distribution of LU search requests between directory services on adjacent nodes, and the exchange of CP capabilities between session services on adjacent nodes.

The underlying protocol for CP-CP sessions is logical unit type 6.2 (LU 6.2). For more information on LU 6.2 protocols see *Systems Network Architecture LU 6.2 Reference: Peer Protocols*. Under this protocol, a contention situation can arise if both session partners attempt to allocate a conversation at the same time. This situation is resolved by designating one of the session partners the contention-winner and the other the contention-loser. The LU 6.2 protocol assigns the contention-winner role automatically to the session partner activating the session.

If CP-CP sessions between adjacent APPN nodes are needed then two parallel CP-CP sessions are required. Each node will activate a session with its partner node. This will make each node the contention-winner on the activated session and the contention-loser on the session activated by the partner node. The node will use the contention-winner session to transmit data and the contention-loser session to receive data.

Figure 61 gives an example of how a connection can be initiated with another node.

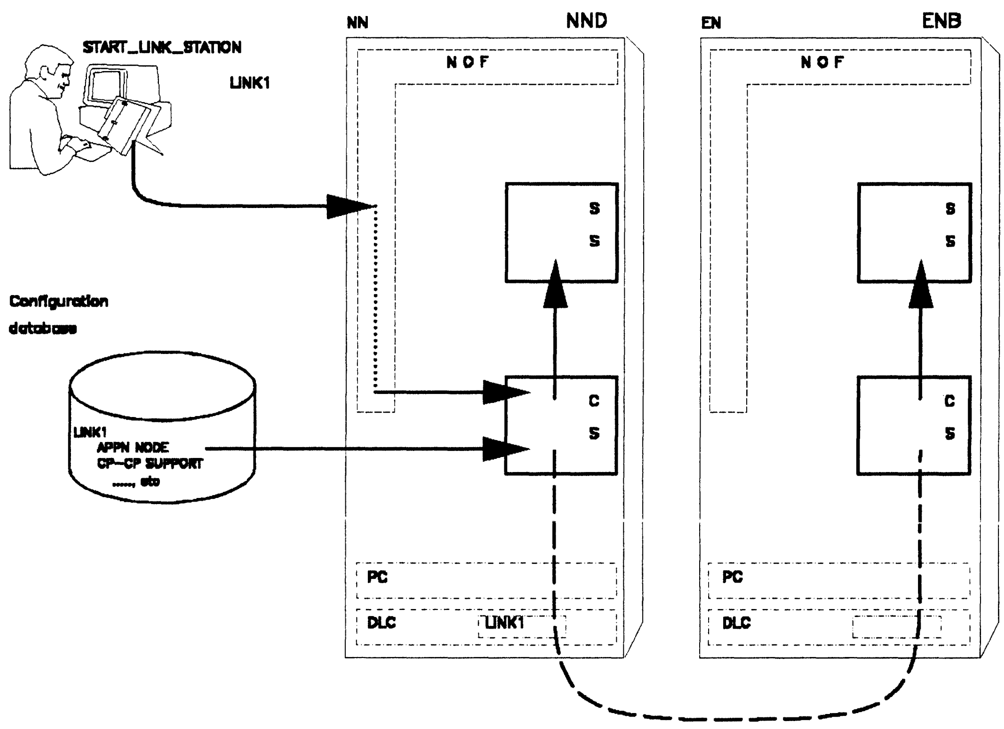

Figure 61. Connection Initiated with Adjacent Node

In this example the link station, called "LINK1", has been defined by the node operator and stored by configuration services in its configuration database. Next, the node operator issues the "Start_Link_Station" command and refers to the definition of "LINK1" in the configuration database. As a result of this command, the node operator facility triggers configuration services to activate the link with the adjacent node.

Once two nodes have established a connection, configuration services determines from the LINK1 definition and the XID exchange with the adjacent link station that CP-CP sessions are supported and required (for more information see "The Contact Phase with XID3 Negotiation" on page 34). Consequently, configuration services notifies session services that another node has been contacted and that CP-CP sessions are required.

Figure 62 on page 106 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 specifying the network qualified name of the session partner, as well as (but not shown in Figure 62 on page 106) its own network qualified name together with the session characteristics required. 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. The exchange of control point capabilities is done using service transaction programs.

Session services on each node notifies directory services that its CP-CP contention-winner session is pending active when it is getting ready to send the BIND, and notifies directory services that either CP-CP session is active after successful exchange of control point capabilities. For each CP-CP session, session services identifies whether it is a contention-winner or contention-loser session.

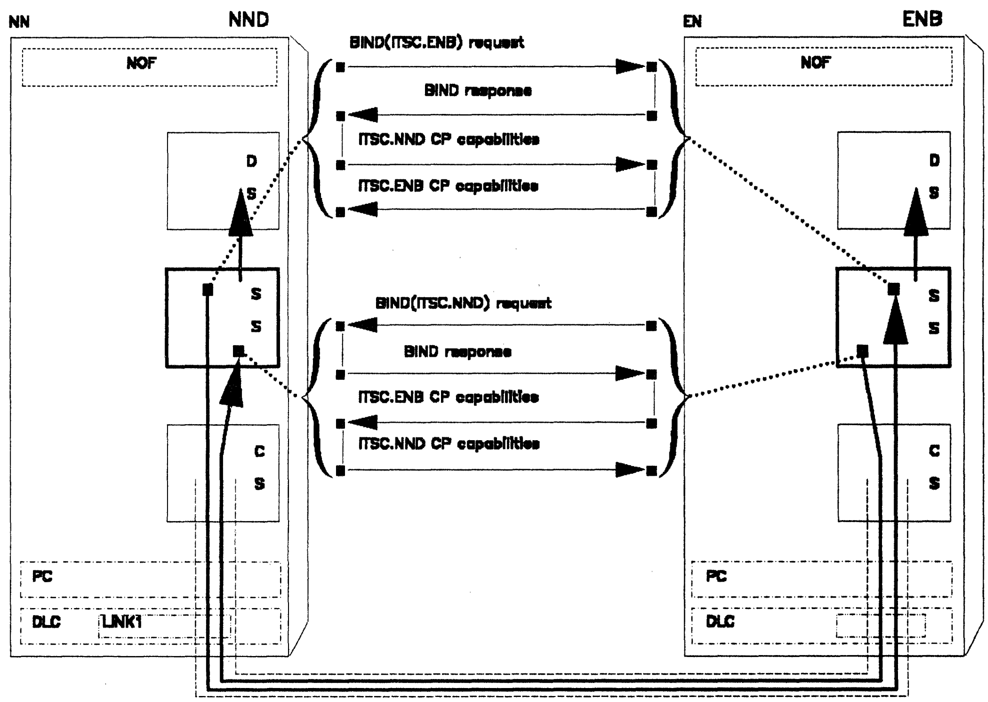

Figure 62. Session Services Activate CP-CP Session

Control Point Capabilities

After the CP-CP session has been established and the BIND completed successfully, the nodes exchange their control point capabilities. A node encodes its CP capabilities in the "CP Capabilities GDS variable" and sends them across the CP-CP session towards the adjacent node. Information contained in the "CP Capabilities GDS variable" is, amongst others:

- MS capabilities exchange supported

  This parameter is set by APPN nodes that support the receipt and replies to management services capabilities vector.

- Flow reduction sequence number

  This parameter is set by network nodes that also supports topology database updates to be exchanged with the adjacent nodes. For more information see "Flow Reduction Sequence Number (FRSN)" on page 49.

- Resource search capability

  This parameter is set by APPN end nodes that support the "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 there is only one resource type supported and that is "LU". For more information see "APPN End Node Search Support" on page 96.

6.3 Deactivate CP-CP Session

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 an implementation option.

6.4 Generate FQPCID

In Figure 63 on page 108, APPN network node "NND" establishes a CP-CP session with APPN end node "ENB". This session may be identified using the network qualified names of both session partners, that is ITSC.NND and ITSC.ENB.

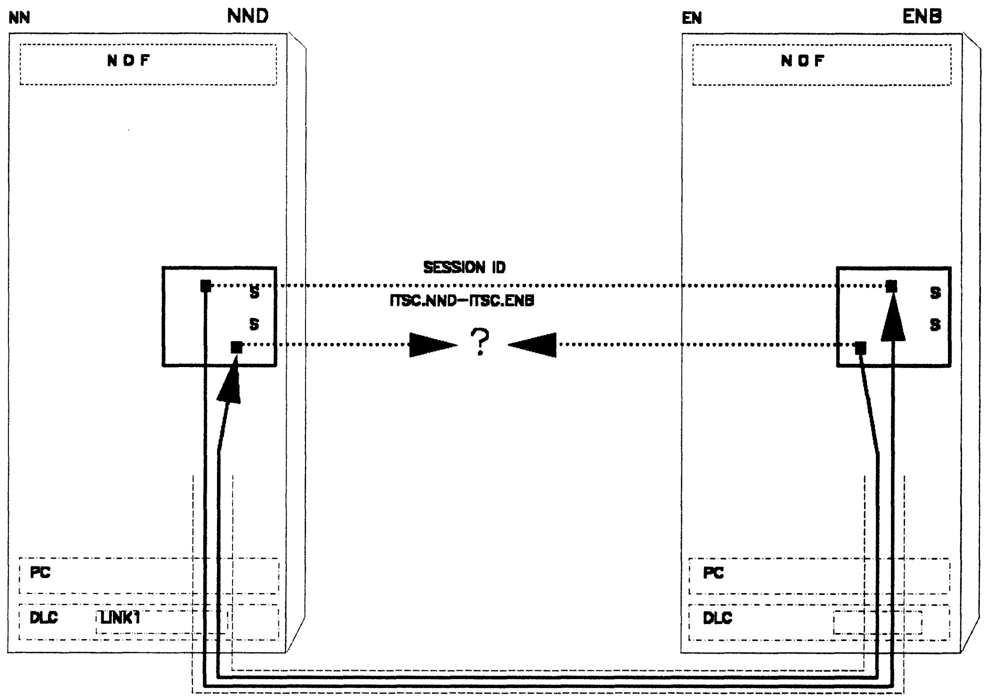

Figure 63. Session Identification

However, if the second (parallel) CP-CP session is activated by "ENB" it is no longer possible to uniquely identify each individual session with the network qualified names. Therefore, session services at the originating node is responsible for generating a network-unique session identifier, hereafter referred to as FQPCID (fully qualified procedure correlation ID). The FQPCID will be used to correlate messages between nodes, for example the BIND request and BIND response, and to identify a particular session for problem determination, auditing, performance monitoring, and/or accounting.

The FQPCID is an eight character field which is first generated at node initialization time. Appended to this generated value is the network qualified name of the control point. However, this value will not be used to guarantee uniqueness merely to identify the control point generating the FQPCID. To ensure uniqueness between sessions from a particular node, the FQPCID is incremented by 1 each time the FQPCID is assigned to a session. A detailed description of the generation process can be found in the Chapter 5 section "FQPCID Generation" of *Systems Network Architecture Type 2.1 Node Reference.*

Although the FQPCID is intended to be unique, the FQPCID may collide in one of the nodes involved in session initiation. When a collision occurs the session initiation request is rejected by the node and the rejected request is returned to the control point that generated the FQPCID. This control point will be responsible for generating a new FQPCID and re-initiating the session.

6.5 Activate LU-LU Session

An LU-LU session distinguishes two roles for LUs after the session has been established: the role of the PLU (primary logical unit) and the role of the SLU (secondary logical unit). In Systems Network Architecture the role of PLU is automatically assigned to the LU that sends the BIND request and the SLU role to the LU that responds to the BIND request with a BIND response. Because only one of the session partners in an LU-LU session is assigned the PLU role (the SLU role is assigned to its session partner) an LU-LU session can also be referred to as a PLU-SLU session.

Session activation is done by a PLU that sends a BIND command across the network to its session partner. The PLU specifies in the BIND command information like:

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

Normally a PLU does not maintain information about the route towards the SLU, nor is it able to generate the FQPCID. Therefore, the PLU requests assistance from session services to complete the BIND information, a process that is also referred to as the session initiation phase.

The session initiation phase is started by an LU when it informs session services that location and routing information is required about another LU. The LU requesting the session initiation is known as the ILU (initiating logical unit). The first step in the session initiation phase is to locate the requested LU. The LU that needs to be located is known as the DLU (destination logical unit) while the other LU, on whose behalf the locate is made, is known as the OLU (origin logical unit). When the DLU is located, the optimum route between OLU and DLU is calculated and finally, either OLU or DLU, will be triggered to activate the session (BIND).

The APPN architecture, as announced in March 1991, recognizes the LU roles as mentioned above; however, the architecture has restricted the roles of ILU, OLU, and PLU to the same LU. In other words, the PLU must be the OLU and the OLU must be the ILU. Future enhancements to the APPN architecture may remove these restrictions.

A session between two LUs may cross multiple nodes. For a session, four nodes may play a role at session initiation time, they are:

- The node that owns the OLU
- The network node server for the OLU
- The network node server for the DLU
- The node that owns the DLU.

Whether or not a particular node plays a role at session initiation time is dependent on the node type of the OLU. For example:

- If the OLU is owned by a LEN end node, then only session services at the LEN end node will be involved in the session initiation request.
- If the OLU is owned by an APPN end node, then the involvement of session services on other nodes is dependent on the DLU's owning node, for example:
  - If the DLU is located in the local directory database of the owning node, then only session services at that node is involved.
  - If the DLU is located on another APPN end node, then not only session services at both end nodes are involved, but also their respective network node servers. This could be one network node server if both end nodes are served by the same network node server or two network node servers.

In the following sub-chapters the role of session services is further described in each of the above mentioned situations.

6.5.1 Session Initiation for LEN End Node

Before an LU requests assistance from session services to initiate a session, the LU first obtains an FQPCID from session services. This FQPCID will be associated with the session initiation request and subsequent activation of the session. For more information see "Generate FQPCID" on page 107. After the LU has obtained the FQPCID from session services, the LU requests session services to locate the DLU and return to the LU the route selection control vector. The route selection control vector describes the route to be taken between OLU and DLU. For more information on route selection control vector see "Examples for Route Selection Services" on page 57.

The request from the LU to session services contains the FQPCID, the OLU name, the DLU name, and the mode name.

On receipt of the session initiation request, session services executes the following steps:

- If the node supports COS/TPF mapping, session services requests topology and routing services to provide the COS/TPF control vector for the mode name that is supplied in the session initiation request from the ILU. For more information see "Class of Service Database" on page 52.
- Next, session services builds the "Request_Local_Search" command and includes in the command the following information (supplied by the ILU):
  - The FQPCID
  - The DLU name.

  After the "Request_Local_Search" command is built, session services sends the command to directory services.
- If the DLU cannot be located by directory services in its local directory database, directory services sends the "Request_Local_Search" reply and includes in the reply the sense code "resource not found". Session services sends a negative reply to the OLU and includes in the reply the sense code from the "Request_Local_Search" reply.
- If, however, the DLU is located by directory services, then directory services returns the DLU's owning CP name to session services.

- On receipt of DLU's CP name, session services checks whether the CP names of OLU and DLU are the same. If so, session services completes the session initiation phase and returns the requested information to the OLU.
- If the CP names are not the same, then session services builds the "Request_Single_Hop_Route" message and sends the message to topology and routing services. The "Request_Single_Hop_Route" message requests topology and routing services to provide the route to the adjacent node that provides the priority and service as requested in the COS/TPF control vector. The "Request_Single_Hop_Route" contains the following information:

  The CP name of the OLU
  The CP name of the DLU
  The COS/TPF control vector (optional).
- The "Request_Single_Hop_Route" reply from topology and routing services contains the first and only hop, that is, the transmission group (TG) towards the CP of the DLU. Session services builds the "Activate_Route" command and includes in the command the destination CP name and the designated TG that it derives from the "Request_Single_Hop_Route" reply. Session services sends the command to configuration services to activate the TG towards the destination CP.
- On successful activation, configuration services returns to session services the path control identifier (path control element address) that represents the designated TG on the node.
- Finally, session services sends its reply to the OLU's session initiation request and includes in the reply the path control element address obtained from configuration services.
- The OLU completes the BIND information and forwards the BIND to path control, including in the request, the path control element address that identifies the specific TG to be used on the node.

The following example will describe the session initiation request for an LU located on an adjacent node. It will be the same example as was used in "LEN End Node Locate Search" on page 85.
In Figure 64, LUX at "LNX" initiates a session with LUA.

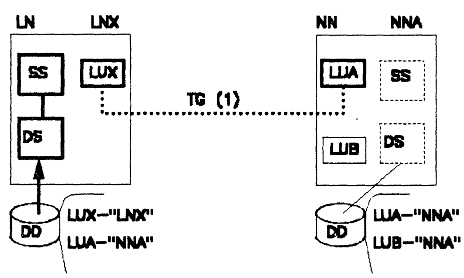

Figure 64. LEN Session Initiation

First LUX requests a FQPCID from session services. Let us assume that session services returns FQPCID(1) to the OLU. Next, the OLU initiates a session and requests session services for assistance:

```
Init_Signal(request,FQPCID(1),OLU(LUX),DLU(LUA))
```

As COS/TPF mapping is not supported by session services at "LNX", session services requests directory services to locate the DLU and return the CP name of the DLU:

```
Request_Local_Search(request,FQPCID(1),DLU(LUA))
```

Directory services at "LNX" searches its local directory database for resource LUA. The resource is located and directory services returns the following reply to session services:

```
Request_Local_Search(reply,FQPCID(1),DLU(LUA at "NNA"))
```

The control point names of OLU and DLU, "LNX" and "NNA" respectively, are not the same; therefore, session services will request topology and routing services for the route selection control vector that describes the link towards "NNA". Please note that "LNX" did not support COS/TPF conversion; hence, the COS name could not be obtained from the COS/TPF control vector.
Therefore, session services will set the default COS name (all blanks) in the "Request_Single_Hop_Route":

```
Request_Single_Hop_Route(request,COS name(        ),
                         ORIGIN-NODE("LNX"),
                         DESTINATION-NODE("NNA"))
```

The "Request_Single_Hop_Route" response from topology and routing services will be:

```
Request_Single_Hop_Route(reply,RSCV(TG(1) to "NNA"))
```

Session services derives the first and only hop (TG(1) to "NNA") from the route selection control vector, build the "Activate_Route" command, and send the command to configuration services to activate the TG:

```
Activate_Route(request,FQPCID(1),Destination Node("NNA"),TG(1))
```

If TG(1) to "NNA" is not active, then configuration services activates the TG. From the (activated) TG to "NNA", configuration services derives the path control element number and returns this identifier in the "Activate_Route" reply to session services:

```
Activate_Route(reply,FQPCID(1),PC element(number))
```

On receipt of the "Activate_Route" reply, session services builds the reply ("Cinit_Signal") for LUX and includes in the reply the path control element number. The reply to the LU will not contain the route selection control vector as this vector was obtained locally:

```
Cinit_Signal(reply,FQPCID(1),PC element(number))
```

Next, LUX sends the BIND, without route selection control vector, to path control and provides path control with the element number that identifies the TG towards "NNA".

6.5.2 Session Initiation for APPN End Node

After obtaining the FQPCID from session services, the LU requests session services to locate the DLU and return to the LU the route selection control vector. The request from the LU contains the FQPCID, the OLU name, the DLU name, and the mode name.

On receipt of the session initiation request, session services executes the following steps:

1. If the node supports COS/TPF mapping, session services requests topology and routing services to provide a COS/TPF control vector for the mode name supplied in the session initiation request from the OLU. For more information see "Class of Service Database" on page 52.

2. Then session services requests topology and routing services to provide all the end node's TG vectors towards adjacent network nodes and connection networks. For more information see "The Normal APPN End Node" on page 58.

3. Next session services builds the "Cross-Domain Initiate GDS variable" and includes the following information:

   - An indicator that specifies whether the OLU or DLU will be the PLU. According to the APPN architecture the indicator will always read OLU=PLU.
   - The mode name supplied by the OLU in the session initiation request.
   - The FQPCID supplied by the OLU in the session initiation request.
   - The end node's TG vectors (from step 2)
   - The COS/TPF control vector (from step 1).

   The "Cross-Domain Initiate GDS variable" is destined for session services components on other nodes that will be involved in the session initiation and activation request. Other nodes in this case may be session services on the network node server for OLU and DLU, and the DLU's owning node.

4. Session services now builds the "Locate_Message" command that contains, amongst others, the following information:

   - The FQPCID supplied by the OLU.
   - The network qualified OLU name.
   - The network qualified DLU name.
   - The control point name of the node containing the OLU.
   - The "Cross-Domain Initiate GDS variable" built in the previous step.

   When the "Locate_Message" command is built, session services sends the command to directory services with a request to locate the DLU and return the CP name of the owning node.

5. After directory services has processed the request, three possible replies can be returned to session services:

   - Directory services was not able to locate the DLU; therefore, directory services returns the "Locate_Message" reply to session services with the appropriate sense code that identifies the type of failure. For more information about possible reason codes see the Chapter 8 section "Locate Search Failures" in *Systems Network Architecture Type 2.1 Node Reference*. Session services sends a negative response to the OLU and includes in its response the sense code from the "Locate_Message" reply.
   - Directory services has located the DLU in its local directory database. The "Locate_Message" reply from directory services will contain the CP name of the node that owns the DLU. The "Cross-Domain Initiate GDS variable" that was part of the "Locate_Message" command has not been delivered by directory services to session services on another node, thus directory services discards the variable. As a consequence, the "Locate_Message" reply does not contain a "Cross-Domain Initiate GDS variable".

The route selection control vector describing the route between OLU and DLU is only present in the "Cross-Domain Initiate GDS variable". Therefore a "Locate_Message" reply without a variable implies that session services will have to obtain the route selection control vector from topology and routing services itself. "Session Initiation for Pre-Defined Remote Resource" further describes the steps performed by session services in case the "Cross-Domain Initiate GDS variable" is missing.

- Directory services has located the DLU using the distributed search capabilities of its network node server. The "Locate_Message" reply from directory services contains the CP name of the node owning the DLU and the "Cross-Domain Initiate GDS variable" prepared by session services at its network node server.

  The "Cross-Domain Initiate GDS variable" contains, amongst others, the route selection control vector that describes the route between OLU and DLU. "Session Initiation Using Network Node Server" on page 117 further describes the steps performed by session services in case the "Cross-Domain Initiate GDS variable" is present in the reply.

6.5.3 Session Initiation for Pre-Defined Remote Resource

The "Locate_Message" reply from directory services did not contain the "Cross-Domain Initiate GDS variable". This implies to session services that the DLU has been located by directory services in its local directory and that the DLU is either located at its own node or an adjacent node.

If the control point names of OLU and DLU are not the same, then session services builds the "Request_Single_Hop_Route" message and sends the message to topology and routing services. This requests topology and routing services to provide the route to an adjacent destination node that provides the priority and service as requested in the COS/TPF control vector. The "Request_Single_Hop_Route" contains the following information:

- The CP name of the OLU
- The CP name of the DLU
- The COS/TPF control vector (optional).

The "Request_Single_Hop_Route" reply from topology and routing services contains the first and only hop, that is, the TG towards the CP of the DLU. Session services then builds the "Activate_Route" command and includes in the command the destination CP name and the designated TG. The command is then sent by session services to configuration services to activate the TG.

On successful activation, configuration services returns to session services the path control element address of the designated TG in the "Activate_Route" reply. On receipt of the reply, session services completes the session initiation request and sends the "Cinit_Signal" reply to the OLU. It includes in the reply the path control element number obtained from configuration services, but not the route selection control vector as this vector was obtained locally.

The OLU now completes the BIND command and sends the BIND to path control. It includes in the request the path control element number from the "Cinit_Signal" that identifies to path control the specific TG to be used.

The following example describes the session initiation request for a predefined remote resource. The same example was used in "APPN End Node Locate Search" on page 86.

In Figure 65 on page 116, LU2 at "ENA" starts the session initiation sequence by requesting session services for a FQPCID. Let us assume that session services returns FQPCID(3) to LU2. Next, LU2 initiates a session request for LU6. Session services receives the following request from LU2:

```
Init_Signal(request,FQPCID(3),OLU(LU2),DLU(LU6),mode name(#INTER))
```

Session services at "ENA" does not support COS/TPF mapping; therefore, topology and routing services will not be requested to map the mode name to a COS/TPF control vector. As "ENA" is an APPN end node, session services requests topology and routing services to supply the TG vectors for all the connections this end node has to network nodes and connection networks. In the request, session services specifies its own CP name and that it needs *all* transmission groups towards network nodes and connection networks. Session services issues the following request to topology and routing services:

```
Request_TG_vector(request,all,"ENA")
```

"ENA" is connected to two network nodes: "NNA" (its network node server) and "NNC". The reply from topology and routing services will be:

```
Request_TG_vector(reply,((TG(1) to "NNA"),(TG(1) to "NNC"))
```

On receipt of the reply from topology and routing services, session services builds the "Cross-Domain Initiate GDS variable" and the "Locate_Message" command. It sends the "Locate_Message" command with the "Cross-Domain Initiate GDS variable" to directory services, requesting directory services to locate the DLU:

```
Locate_Message(request,FQPCID(3),OLU(LU2 at "ENA"),DLU(LU6)),
  Cross-Domain Initiate GDS variable(OLU=PLU,mode name(#INTER),FQPCID(3),
           TG vector(TG(1) to "NNA",TG(1) to "NNC")
```

Directory services locates the DLU in its local directory database and returns the following reply to session services:

```
Locate_Message(reply,FQPCID(3),OLU(LU1 at "ENA"),DLU(LU6 at "NNC"))
```

On receipt of the reply from directory services, session services looks for the "Cross-Domain Initiate GDS variable" in the "Locate_Message" reply. The reply does not contain the "Cross-Domain Initiate GDS variable", which implies that the resource was located in the local directory database.

Next, session services checks whether the CP names of OLU and DLU are the same. The CP names, "ENA" and "NNC" respectively, are not the same; therefore, session services builds the "Request_Single_Hop_Route" command and sends the command to topology and routing services. Session services includes in the command the default COS name (all blanks) because the end node did not support COS/TPF mapping.
The "Request_Single_Hop_Route" command looks like:

```
Request_Single_Hop_Route(request,COS name(        ),
                Origin Node("ENA"),
                Destination Node("NNC"))
```

There is only one TG towards "NNC", therefore, the reply from topology and routing services will be:

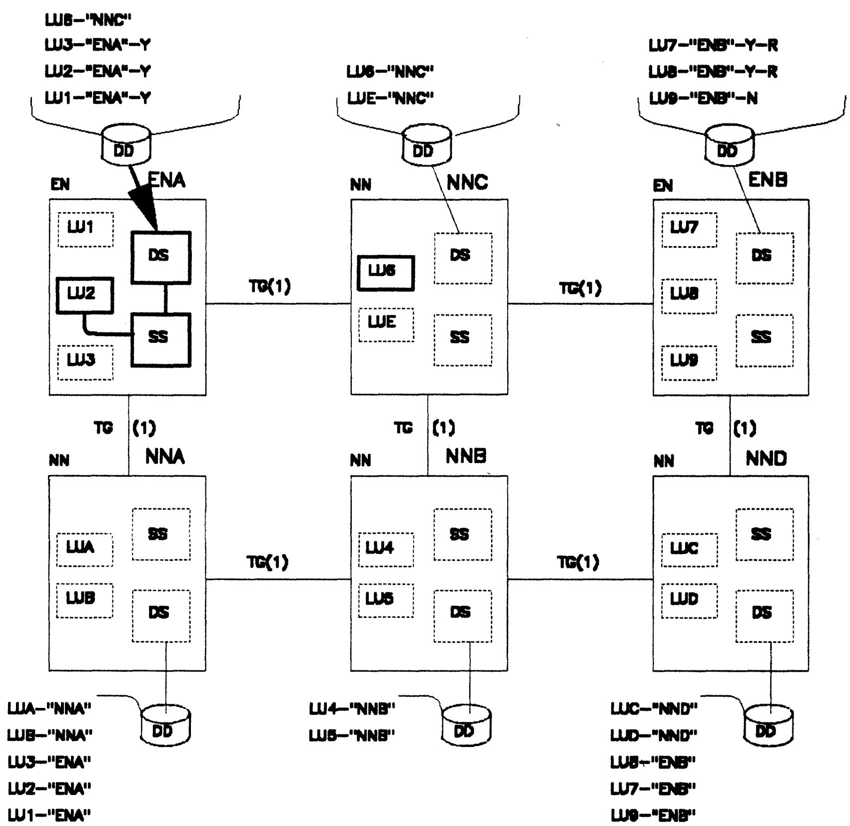

Figure 65. APPN End Node Local Session Initiation for Remote Resource

```
Request_Single_Hop_Route(reply,RSCV(TG(1) to "NNC"))
```

Session services derives the first and only hop, TG(1) to node "NNC", from the route selection control vector, builds the "Activate_Route" command, and sends the command to configuration services to activate the TG:

```
Activate_Route(request,FQPCID(3),Destination Node("NNC"),TG(1))
```

If TG(1) to "NNC" is not active, then configuration services activates the TG. From the active TG to "NNC", configuration services derives the path control element address that identifies the specific TG in the node and includes the address in the "Activate_Route" reply to session services:

```
Activate_Route(reply,FQPCID(3),PC element address)
```

On receipt of the "Activate_Route" reply, session services builds the reply ("Cinit_Signal") for LU2 and includes in the reply the path control element address. The reply to LU2 will not contain the route selection control vector as this vector was obtained locally.

```
Cinit_Signal(reply,FQPCID(3),PC element address)
```

On receipt of the reply from session services, LU2 completes the BIND request and sends the BIND to path control together with the path control element address from the "Cinit_Signal" that identifies the designated TG towards "NNC".

6.5.4 Session Initiation Using Network Node Server

The ″Locate_Message″ reply from directory services contained the ″Cross-Domain Initiate GDS variable″. This implies that the ″Cross-Domain Initiate GDS variable″, which was part of the ″Locate_Message″ command, has been delivered to session services at the network node server, and that session services at the network node server returns a ″Cross-Domain Initiate GDS variable″ that contains the session route between OLU and DLU and a COS/TPF control vector.

In this section we will not only describe the roles of session services at the network node server, and its interfaces to other control point components, but also the roles and functions provided by session services on the destination node and its network node server. To better describe these roles we will use the example in Figure 66 on page 118, which is the same example as used in "Broadcast Search" on page 89.

In this example LU1 at ″ENA″ requests a FQPCID from session services. Let us assume that session services returns FQPCID(5). Next, LU1 at ″ENA″ initiates the session with ″LU7″:

```
Init_Signal(request,FQPCID(5),OLU(LU1),DLU(LU7),mode name(blanks))
```

Session services on ″ENA″ does not support COS/TPF mapping; therefore, topology and routing services will not be requested to map the mode name to a COS/TPF control vector. As ″ENA″ is an APPN end node, session services requests topology and routing services to supply the TG vector for all the connections this end node has to network nodes and connection networks. In the request, session services specifies that it needs *all* transmission groups towards network nodes and connection networks. Session services issues the following request to topology and routing services:

```
Request_TG_vector(request,all)
```

″ENA″ is connected to two network nodes: ″NNA″ (its network node server) and ″NNC″. The reply from topology and routing services will be:

```
Request_TG_vector(reply,((TG(1) to "NNA"),(TG(1) to "NNC"))
```

On receipt of the reply from topology and routing services, session services builds the ″Cross-Domain Initiate GDS variable″ and the ″Locate_Message″ command. It sends the ″Locate_Message″ command with the ″Cross-Domain Initiate GDS variable″ to directory services, requesting directory services to locate the DLU:

```
Locate_Message(request,FQPCID(5),OLU(LU1 at "ENA"),DLU(LU7)),
  Cross-Domain Initiate GDS variable(OLU=PLU,mode name(blanks),FQPCID(5),
            TG vector(TG(1) to "NNA",TG(1) to "NNC")
```

Directory services at ″ENA″ cannot locate the DLU in its local directory database, thus directory services sends the ″locate search request″ to its network node server (″NNA″).

In Figure 67 on page 119, that part of the broadcast search flow is depicted that resulted in a ″resource found″ condition. Only those nodes are shown in the search path that required the assistance of session services on that node.
The following functions were performed by each node:

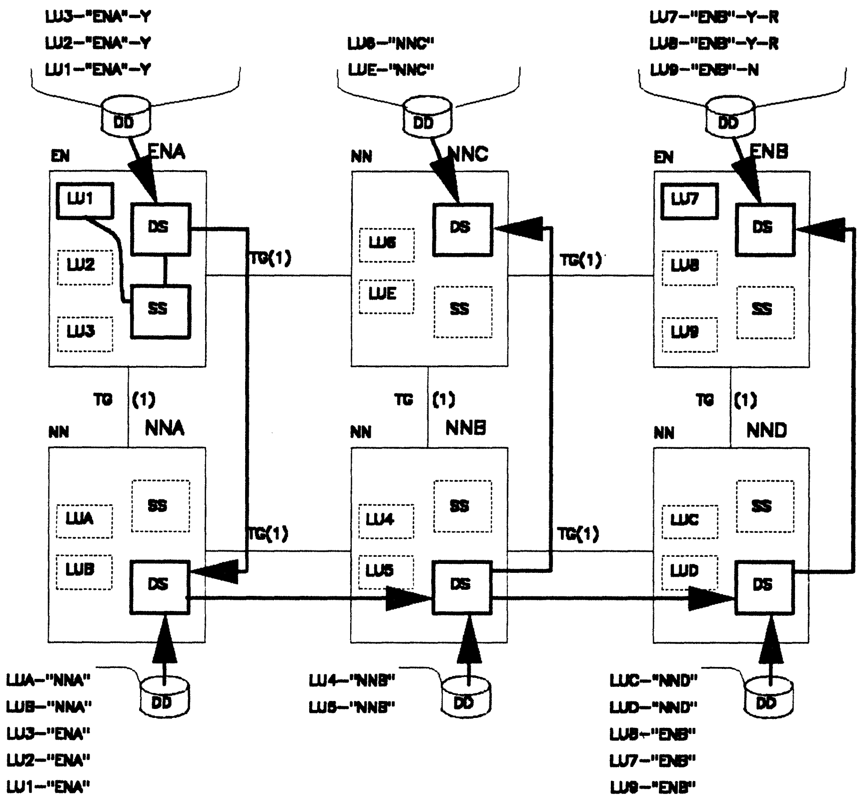

Figure 66. Session Initiation for Remote LU

1 Directory services at "NNA" detects that the OLU is from an end node for which it acts as network node server, builds the "Locate_Message" command, and sends it to session services:

```
Locate_Message(request,FQPCID(5),OLU(LU1 at "ENA"),DLU(LU7)),
  Cross-Domain Initiate GDS variable(OLU=PLU,mode name(blanks),FQPCID(5),
          TG vector(TG(1) to "NNA",TG(1) to "NNC")
```

Session services at "NNA" detects that the "Cross-Domain Initiate GDS variable" does not contain the COS/TPF control vector, builds the "Request_COS/TPF vector" command and includes in the vector the mode name. The request is sent to topology and routing services at "NNA":

```
Request_COS/TPF vector(request,mode name(blanks))
```

Topology and routing services maps the mode name to a transmission priority and COS name, and includes both in the COS/TPF control vector that is returned to session services. For more information see "Class of Service Database" on page 52.
The reply from topology and routing services to session services at "NNA" looks like:

```
Request_COS/TPF vector(reply,SENSE(0),COS/TPF(2,#CONNECT))
```

Session services at "NNA" adds the COS/TPF control vector to the "Cross-Domain Initiate GDS variable" and removes the TG vector that described all the connections the end node has to network nodes and

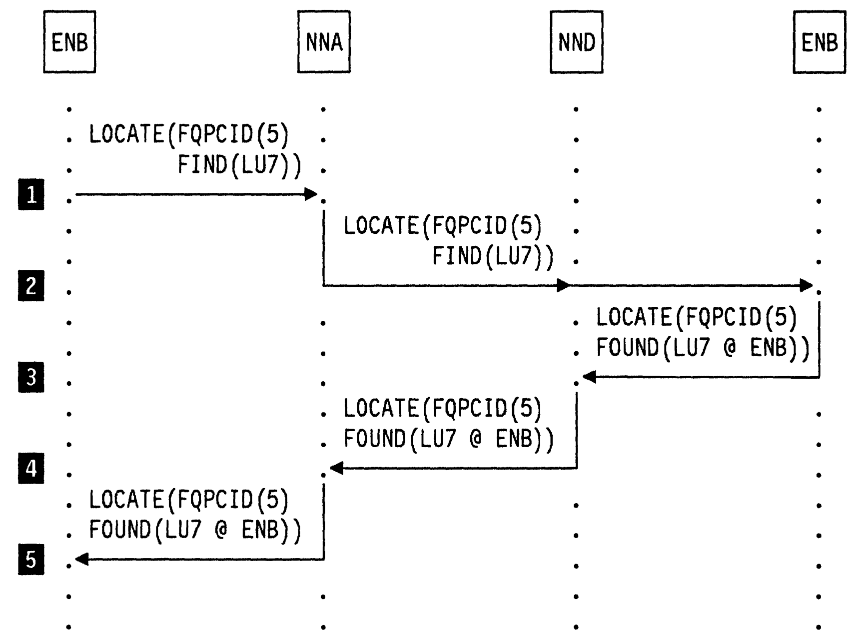

Figure 67. Broadcast Search Flow

connection networks. Session services at "NNA" saves the TG vector for later use. Next, session services sends the updated "Locate_Message" command back to directory services:

```
Locate_Message(request,FQPCID(5),OLU(LU1 at "ENA"),DLU(LU7)),
  Cross-Domain Initiate GDS variable(OLU=PLU,COS/TPF(2,#CONNECT),FQPCID(5))
```

Directory services at "NNA" cannot locate the resource in its local directory database and initiates the "broadcast search".

2 The search request from "NNA" will also be received by directory services at end node "ENB", the owner of resource LU7. Directory services at "ENB" builds the "Locate_Message" reply and sends it to session services at its node:

```
Locate_Message(reply,FQPCID(5),OLU(LU1 at "ENA"),DLU(LU7 at "ENB")),
  Cross-Domain Initiate GDS variable(OLU=PLU,COS/TPF(2,#CONNECT),FQPCID(5))
```

On receipt of the "Locate_Message" reply, session services builds the "Request_TG_vector" command and requests topology and routing services to provide all the end node's TG vectors towards network nodes, connection networks, and any TG vector towards the destination node, being "ENA". For example, if "ENB" would have had a direct link to "ENA", then topology and routing services will also provide a TG vector for that link:

```
Request_TG_vector(request,all,
               destination node("ENA"))
```

"ENB" is connected to two network nodes: "NND" (its network node server) and "NNC". It does not have a direct connection to "ENA". The reply from topology and routing services will be:

```
Request_TG_vector(reply,(TG(1) to "NND",TG(1) to "NNC")
```

Session services at "ENB" adds the TG vector to the "Cross-Domain Initiate GDS variable" and sends the "Locate_Message" reply back to directory services:

```
Locate_Message(reply,FQPCID(5),OLU(LU1 at "ENA"),DLU(LU7 at "ENB")),
  Cross-Domain Initiate GDS variable(OLU=PLU,COS/TPF(2,#CONNECT),FQPCID(5),
          TG vector(TG(1) to "NND",TG(1) to "NNC")
```

Directory services now sends the "Locate_Message" reply back in the direction of the OLU.

3 The first node on the route towards the OLU is the network node server of the DLU, in our case "NND". Directory services at "NND" sends the "Locate_Message" reply to session services. Session services checks whether the endpoint TG vectors are included in the "Cross-Domain Initiate GDS variable". If there are no endpoint TG vectors in the "Cross-Domain Initiate GDS variable", session services at "NND" interfaces with topology and routing services at its node to obtain the TG vectors. In our example the endpoint TG vectors are already in the "Cross-Domain Initiate GDS variable", hence, session services returns the "Locate_Message" reply without any changes.

4 As the "Locate_Message" reply flows towards the OLU, it will also arrive at the network node server for the OLU, in our example "NNA". Directory services at "NNA" sends the "Locate_Message" reply to session services:

```
Locate_Message(reply,FQPCID(5),OLU(LU1 at "ENA"),DLU(LU7 at "ENB")),
  Cross-Domain Initiate GDS variable(OLU=PLU,COS/TPF(2,#CONNECT),FQPCID(5),
          TG vector(TG(1) to "NND",TG(1) to "NNC"))
```

Session services at "NNA" obtains the TG vector for the OLU that was saved earlier by session services, builds the "REQUEST_ROUTE" command, and sends the command to topology and routing services. It requests topology and routing services to calculate the route between OLU and DLU. The information provided in the "REQUEST_ROUTE" command are the CP names and endpoint TG vectors for both OLU and DLU, and the COS/TPF control vector. The "REQUEST_ROUTE" command looks like:

```
REQUEST_ROUTE(request,COS/TPF(2,#CONNECT)
              origin node("ENA"),destination node("ENB"),
              OLU TG vectors(TG(1) to "NNA",TG(1) to "NNC"),
              DLU TG vectors(TG(1) to "NND",TG(1) to "NNC"))
```

Topology and routing services computes the optimum route and returns the "REQUEST_ROUTE" response that includes the route selection control vector describing the route between OLU and DLU. The reply looks like:

```
REQUEST_ROUTE(reply,sense(0),
        RSCV(TG(1) to "NNC",TG(1) to "ENB")
```

Session services builds the "Cross-Domain Initiate GDS variable" and includes in the variable the COS/TPF control vector and the route selection control vector received from topology and routing services. The "Locate_Message" reply sent by session services to directory services is:

```
Locate_Message(reply,FQPCID(5),OLU(LU1 at "ENA"),DLU(LU7 at "ENB")),
  Cross-Domain Initiate GDS variable(OLU=PLU,COS/TPF(2,#CONNECT),FQPCID(5),
        RSCV(TG(1) to "NNC",TG(1) to "ENB"))
```

5 Finally, the reply arrives at "ENA", the owner of the OLU. Directory services sends the "Locate_Message" reply to session services:

```
Locate_Message(reply,FQPCID(5),OLU(LU1 at "ENA"),DLU(LU7 at "ENB")),
  Cross-Domain Initiate GDS variable(OLU=PLU,COS/TPF(2,#CONNECT),FQPCID(5),
        RSCV(TG(1) to "NNC",TG(1) to "ENB"))
```

Session services derives the first hop from the route selection control vector in the "Cross-Domain Initiate GDS variable", that is, TG(1) to "NNC", builds the "Activate_Route" command, and sends the command to configuration services to activate the route:

```
Activate_Route(request,FQPCID(5),destination node("NNC"),TG(1))
```

If TG(1) to "NNC" is not active, then configuration services activates the TG. From the active TG to "NNC", configuration services derives the path control element address that identifies the specific TG in the node and returns it in the "Activate_Route" reply to session services:

```
Activate_Route(reply,FQPCID(5),PC element address)
```

On receipt of the "Activate_Route" reply, session services builds the reply ("Cinit_Signal") for LU1 and includes in the reply the path control element address, the COS/TPF control vector, and the route selection control vector. Both COS/TPF control vector and route selection control vector are derived from the "Cross-Domain Initiate GDS variable" that was built by session services at the network node server. Next, session services sends the "Cinit_Signal" to LU2:

```
Cinit_Signal(reply,FQPCID(5),PC element address,
         COS/TPF(2,#CONNECT),RSCV(TG(1) to "NNC",TG(1) to "ENB"))
```

On receipt of the reply, the OLU (in our example LU1) completes the BIND, that is, it will append the COS/TPF control vector and the route selection control vector to the BIND image. The LU then sends the BIND to path control and includes in the request the path control element number that represents TG(1) to "NNC" on that node.

Chapter 7. Network Management

Network management is the process of planning, organizing, monitoring, and controlling an APPN network.

7.1 Initialization

The management services component is created and initialized by the node operator facility at node initialization time. The node operator facility starts the initialization process when the "START_NODE" command is issued by the node operator. When the management services is created, node operator facility passes all the parameters, specified by the node operator when issuing the "START_NODE" command, to the management services component to initialize management services

7.2 Network Management Categories

Network management can be divided in four major categories. They are:

- Configuration management
- Problem management
- Change management
- Performance and accounting 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 and release plus maintenance level (software), vendor and service organization, and others.

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 effect 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 a loss of availability of a system resource that is visible to the end user. Problems may originate in hardware, software, or as 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 being worked upon. 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 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 trial 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.

7.3 Management Services Concepts

The management services concept is based upon entry points and focal points. This concept allows customers to centralize their network management. It allows for multiple focal points where each focal point may provide centralized control for one or more network management categories.

7.3.1 Entry Point

An entry point is a node that manages and controls its local resources, such as links, link stations, CP components, and LUs. It sends SNA formatted network management data about itself and its resources to a focal point for centralized processing. For communication with a focal point it uses the distributed network management capabilities provided by multiple domain support (MDS). From its focal point it receives and executes network management requests that control the node's resources. The entry point returns the results of these requests to the requesting focal point.

Typical entry point nodes are the APPN end node and APPN network node. Besides controlling its own local resources the entry point for a network node is also responsible for the end node resources for which it acts as network node server. When the network node establishes a relationship with one or more focal points, the network node automatically notifies its end nodes about the focal point relationship. This way the end node uses the same focal points as its network node server without having to establish a relationship with the focal point itself.

In the configuration in Figure 68 on page 126, the following entry points are created:

- Entry point "NNA" including the end node "ENA" for which it acts as network node server
- Entry point "NNB"
- Entry point "NNC"
- Entry point "NND" including the end node "ENB" for which it acts as network node server.

Please note the end nodes "ENA" and "ENB" are by definition entry points. However, these end nodes do not establish relationships with focal points. Logically, the end node entry point is part of the network node server entry point.

7.3.2 Focal Point

A focal point is an entry point that provides centralized management and control for a set of network management categories. It provides this support for one or more entry points. The relationship between a focal point and entry point is established by exchanging management services capability vectors that specify the specific categories for which the focal point provides centralized management services.

The set of entry points that have such a relationship with a focal point is known as the *sphere of control* (SOC) of a focal point. Each entry point in its sphere of control is known as a *SOC node*. End nodes need not establish a relationship with a focal point as the end node obtains these services through its network node server. This reduces the network management definitions for SOC nodes in a focal point. However, an end node may be a SOC node itself and establish a relationship with a focal point.

Entry points are either assigned to a focal point's sphere of control or are dynamically acquired by a focal point. The focal point that has entry points assigned to it is referred to as an assigned focal point and the focal point that acquires entry points is referred to as a default focal point.

Default Focal Point

The default focal point learns about the identity of SOC nodes by examining the network topology database. It sends the management services capabilities vector to each node it finds in the topology database. In the management services capabilities vector it offers its focal point services to an entry point. As only network nodes appear in the network topology database, a default focal point only acquires network nodes.

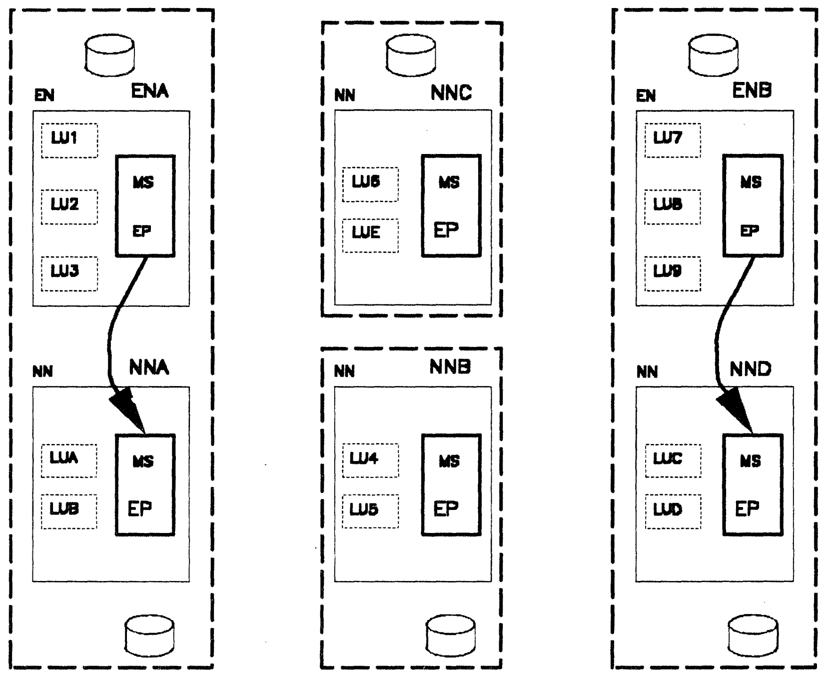

Figure 68. Network Management Entry Points

The default focal point concept could be useful in single network environments that consist solely of network node SOCs. If end nodes would be part of the network, then these end nodes would obtain focal point services only when they are connected to their network node server.

For example, in Figure 69 on page 127 network node "NNC" has been defined as default focal point. When the network nodes connect to the network, topology and routing services propagates the CP name to all network nodes. As the topology database update request reaches node "NNC", management services is informed about the network node and forwards the management services capabilities vector to the network node. It asks the network node whether "NNC" could be its default focal point. If the network node does not have another focal point it will accept "NNC" as focal point. As a result of this focal point relationship, "ENA" will be informed by network node "NNA" that "NNC" is the focal point.

However, if a network node fails or is taken out-of-service the default focal point will not only remove the network node from its sphere of control, but also the end nodes that used the network node as network node server. As shown in Figure 69, network node "NNC" was deactivated. This implies that node "NNA", including "ENA", is no longer in the sphere of control of "NNC". However, end node "ENA" could still be part of the network and establish sessions directly through node "NNC", but cannot obtain network management services from a focal point. This latter requires "ENA" to use "NNC" as network node server.

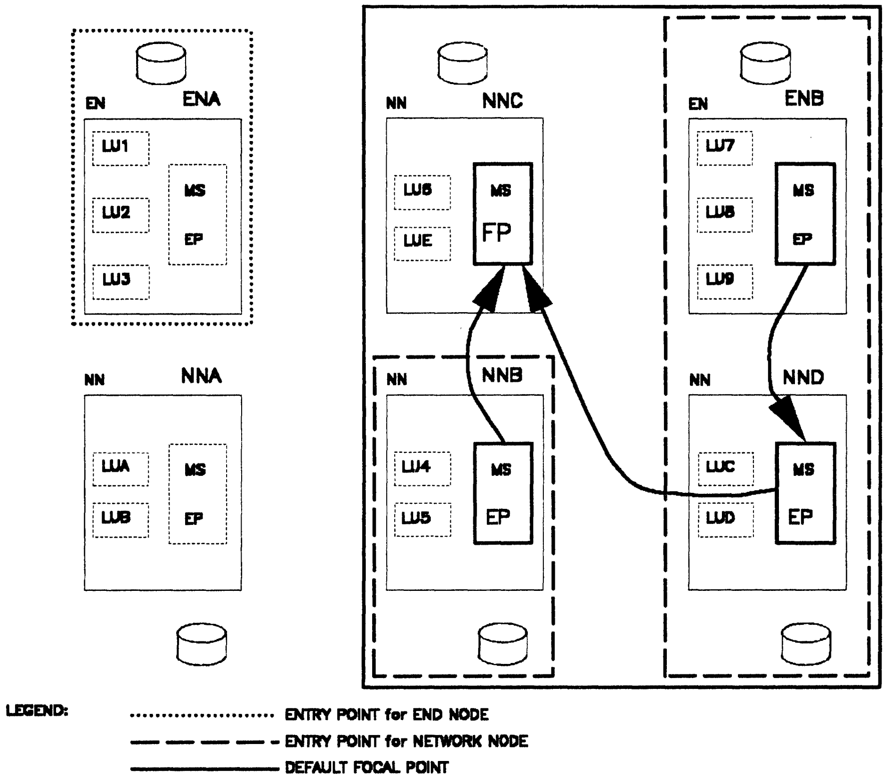

Figure 69. Network Management Default Focal Point

Assigned Focal Point

An assigned focal point can be assigned in two ways:

1. Explicitly

   An explicitly defined node is a node that is defined at the focal point as being in its sphere of control. The focal point will be responsible for initiating and establishing the focal point to entry point relationship.

2. Implicitly

   An implicitly defined node is a node that is defined at the entry point as being in the sphere of control of a focal point. The entry point node will be responsible for initiating and establishing the focal point to entry point relationship. The focal point learns about the implicitly defined nodes as the node requests focal point management services by sending the management services capabilities vector.

Unlike default focal points, the assigned focal point requires network management definitions at the focal point and/or entry point. However, the assigned focal point could be preferred if the network consists of multiple focal points, and focal point relationships are required with end nodes.

For example, in Figure 70 on page 128 there are two focal points: "NNC" and "NNB". Focal point "NNC" controls SOC node "NNA", and focal point "NNB" controls "NND". This configuration can only be accomplished using assigned focal points, either explicitly or implicitly defined.

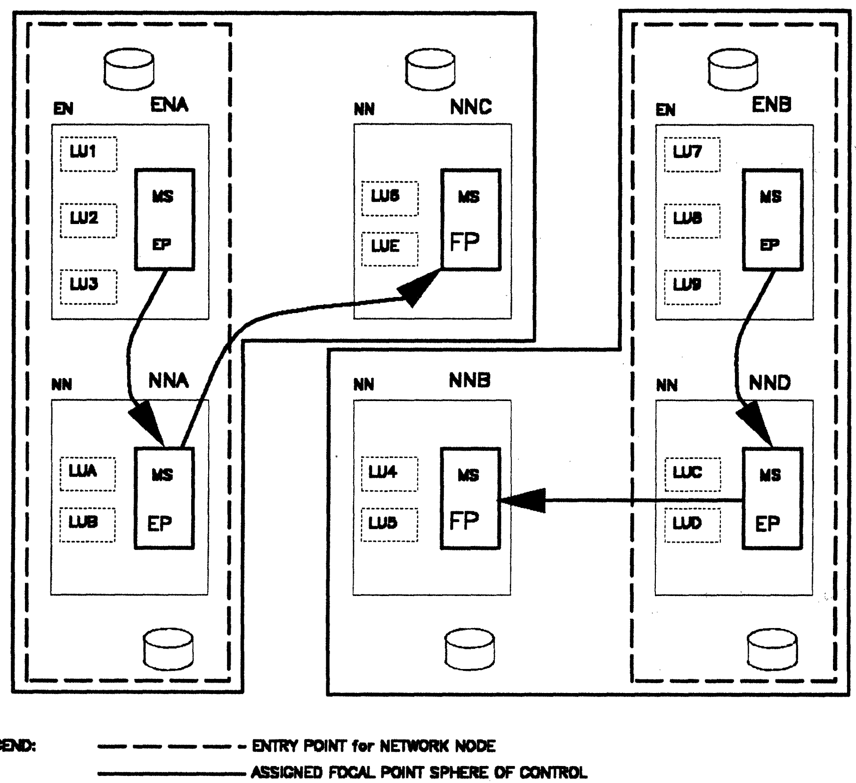

Figure 70. Network Management Assigned Focal Points

7.4 Management Services Components

Management services distinguishes three components. They are:

- Local management services, hereafter referred to as LMS
- Physical unit management services, hereafter referred to as PUMS
- Control point management services, hereafter referred to as CPMS.

7.4.1 Local Management Services

LMS is the network management portion that is implemented in components and layers of a T2.1 node. 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.

7.4.2 Control Point Management Services

Control point management services (CPMS) is a CP component in the 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.

7.4.3 Physical Unit Management Services

Physical unit management services (PUMS) is the component of the physical unit (PU) responsible for providing general management services for the node and its associated resources. PUMS requires an SSCP-PU session with its controlling SSCP to forward network management data to the SSCP or receive network management requests from the SSCP. The management services commands (network management vector transport) 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 and sent across the SSCP-PU session to the SSCP.

7.5 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 individual 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. The function sets are divided into two categories: the *base function set* and the *optional subset*.

7.5.1 Function Sets

The management services function sets can be subdivided into a general management category and a specialized management category for entry points. The general management category distinguishes the following function sets:

- Multiple Domain Support function set
- Management Services Capabilities function set
- File Services Support function set
- Send Data SSCP-PU function set
- Send Request SSCP-PU function set
- Receive Request SSCP-PU function set
- Receive Data SSCP-PU function set
- Database Support function set
- SOC Manager function set.

The specialized management category distinguishes the following function sets:

- EP_Alert function set
- EP_RTM function set
- EP_QPI function set

- EP_Change_MGMT function set
- EP_Common_Operations_Services function set.

Multiple Domain Support

The multiple domain support (MDS) base subset is required in both end nodes and network nodes. It provides the capability to send management services requests and data between management functions in the same or different nodes. The optional subsets are:

- Optional Subset 1 (End Node Support)

  The end node support is applicable to network nodes only. It consists of the MDS router functions for the entry point.

- Optional Subset 2 (Network Node Support)

  The network node support is applicable to network nodes only. It consists of the MDS router functions for network nodes.

- Optional Subset 3 (High Performance Option)

  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.

- Optional Subset 4 (Transport Confirmation Option)

  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.

Management Services Capabilities

The multiple domain support (MDS) base subset is required in both APPN end nodes and network nodes. 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, through using an LU-LU session, or, indirectly through its network node server.

- Optional Subset 1 (Have a backup or Implicit FP)

  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.

- Optional Subset 2 (Be a SOC End Node)

  Support for sphere of control end node is applicable to end nodes and network 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.

- Optional Subset 3 (Base Network Node Support)

  Support for base network node support is required for network nodes. It provides the support to a network node of being a SOC node and sending and receiving MS capabilities from the entry point side of the relationship.

- Optional Subset 4 (Have a Subarea Focal Point)

  Support for subarea focal point is applicable for 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.

- Optional Subset 5 (Be a Focal Point)

  Focal point support is applicable for network nodes only. It provides the functions to support a focal point application and the ability to establish explicit focal point relationships.

File Services

The file services function set is applicable to both end nodes and network nodes. It provides the support to route management services requests and bulk data between nodes using SNA distribution services.

- Optional Subset 1 (Network Operator Support)

  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.

- Optional Subset 1 (File Deletion Support)

  File deletion support is only applicable to network nodes. It provides the support to interact with the network operator to delete files.

Send Data SSCP-PU

The send data SSCP-PU function set is applicable to both end nodes and network nodes. It provides the support for sending network management vector transport RUs across an SSCP-PU session to a subarea CPMS.

Send Request SSCP-PU

The send request SSCP-PU is only applicable to network nodes. It provides the support to receive requests from the Multiple Domain Support function set, converts them to network management vector transport RUs, and routes them to PUs in its domain. Alternatively, it receives response network management vector transports from PUs, converts them to Multiple Domain Support management services units, and passes them to Multiple Domain Support.

Receive Request SSCP-PU

The receive request SSCP-PU function set is applicable to both end nodes and network nodes. It provides the support to receive network management vector transport RUs and pass the vector to the appropriate function group set.

Receive Data SSCP-PU

The receive data SSCP-PU function set is applicable to network nodes. It provides the support for sending network management vector transport RUs across an SSCP-PU session to a subarea CPMS.

- Optional Subset 1 (Alert Filtering)

  Support for alert filtering is only applicable to network nodes. It allows a focal point to cease processing certain alerts, based upon one or more operator specified filters.

Database Support

The database support function set is only applicable to network nodes. 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.

SOC Manager

The SOC manager function set is only applicable to network nodes. It provides the support to initiate and subsequently update and monitor the spheres of control for each focal point. The SOC manager function determines when to initiate the sphere of control, when to retry failing nodes, and monitors the status of the nodes under its sphere of control.

- Optional Subset 1 (Default Focal Point)

  The default focal point subset allows a focal point to automatically include in its sphere of control all network nodes which are not serviced by other, primary focal points. The inclusion of a node in the default sphere of control is based on information in the topology database and is not dependent on manual input.

- Optional Subset 2 (Backup Focal Point)

  The backup focal point allows a focal point to define a backup focal point in the SOC table. This information is sent to SOC nodes for recovery should they not be able to reach this focal point.

EP_ALERT

The EP_ALERT function set 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:

- Optional Subset 1 (Problem Diagnosis Data)

  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.

- Optional Subset 2 (Delayed Alert)

  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 re-established the held alerts will be forwarded to the focal point.

- Optional Subset 3 (Held Alert for PUMS)

  Support for held alert for PUMS means that the entry point is capable of holding alerts until the session with the PUMS is re-established.

- Optional Subset 4 (Operator-Initiated Alert)

  Support for operator-initiated alerts provides 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.

- Optional Subset 5 (Qualified Message Data)

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 re-construct the message from its local message table. For example, if the national language differs between focal point and &ep, this subset allows the focal point and entry point to generate the alert message in their own national language.

- Optional Subset 6 (Text Message)

  Support for text message provides the capability to include in the alert a character string of 236 characters.

- Optional Subset 7 (LAN Alert)

  Support for LAN alert provides the capability to send alerts for errors detected at the MAC layer of a token-ring, ethernet, or bridged LANs.

- Optional Subset 8 (SDLC/LAN LLC Alert)

  Support for SDLC/LAN LLC alerts provides the capability to send alerts for problems detected on SDLC and LAN logical link level control.

- Optional Subset 9 (X.21 Alert)

  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.

- Optional Subset 10 (Hybrid Alert)

  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 down level version.

- Optional Subset 11 (X.25 Alert)

  Support for X.25 alerts provide the capability to send alerts for problems detected on X.25 link connections.

- Optional Subset 11 (Held Alert for CPMS)

  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

Support for EP_RTM provides the capability measure and monitor end users response times.

- Optional Subset 1 (Local Display)

  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

Support for the EP_QPI provides the capability to physically identify the SNA node and attached devices upon request.

EP_CHANGE_MGMT

Support for the EP_CHANGE_MGMT provides the capability to respond to change control and activation requests from a change management focal point or local operator interface.

- Optional Subset 1 (Production-Only Activation)

  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

Support for EP_COMMON_OPERATIONS_SERVICES provides the capability to support communication between network operators and served network management applications.

Chapter 8. APPN Implementations

Several products implement T2.1 functions, either as APPN nodes or as LEN end nodes. Among them, the AS/400 stands out, as it implemented APPN functions as an extension to the Systems Network Architecture T2.1 node. Now, APPN has been announced as part of Systems Network Architecture and is published along with the other functions of Systems Network Architecture.

The characteristics of the implementations in the AS/400, in the 3174, in the PS/2 as APPN nodes, and in VTAM/NCP as a LEN end node are summarized in this chapter. Some architectural considerations for VTAM/NCP as APPN nodes are added in "SNA Functions in Addition to APPN" on page 163.

For several years, all of the above products have been supporting dependent LUs on the basis of the T2.0 node architecture. They continue to do so. These functions are not considered in this document.

Some other IBM products with APPN implementations are not described in this document (such as the S/36, or DPPX/370 Release 3). APPN implementations of other manufacturers (Apple Computer, Inc; NOVELL, Inc; Systems Strategies, Inc; Siemens/Nixdorf Informationssysteme AG) are not covered here either.

The architecture does not limit the size of the APPN network. The implementations have certain limitations, though, caused by the requirement for internal or external storage and processor utilization. The limitations that are given in this chapter specify the maximum values, which can used during system definition. That does not mean that these values can actually be achieved. For performance reasons, lower values may be recommended.

For each product some important implemented functions are listed. A "Summary of Implemented Functions" on page 147 combines this information. All the functions are explained in this document. A ″yes″ in the table means that a node implements this function or cooperates with this function implemented in another node. A ″no″ in the table indicates that a node does not implement this function or that this function does not apply to that particular node type. The tables reference the pages where more information can be found.

The evolution of SNA will continue. That includes enhancements to APPN. The implementation list is based on the current software. Future releases of current products or new products will provide additional functions.

8.1 AS/400

APPN was implemented in the first release of the AS/400 when it was announced in 1988. The core functions had already been there in the S/36, the AS/400's predecessor. In that sense, for the AS/400, APPN is a "proven" architecture. The AS/400 functions as network node or as an end 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 is an integral part of the Operating System of the AS/400.

8.1.1 Terminology

The term "location" is used for LU (logical unit).

A remote node is also called "controller" or "control unit".

A "device" is the representation of a remote location (LU) in the local node.

Wildcard routing is also referred to as "*ANY" routing.

Congestion
The maximum number of intermediate routing sessions supported by a network node may 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.

Pacing
Support is provided by the system to change the pacing values based on the system's buffer resources.

8.1.2 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 LEN end nodes
- LUs in APPN end nodes without CP-CP session, if the LU name is different from CP name
- LUs in APPN end nodes that do not register and do not allow domain broadcast
- LUs, for which session security is defined
- Single session LUs.

Controller descriptions for LAN devices are created automatically. That includes other AS/400s or PS/2s, but excludes VTAM/NCP. (The reason is, that connection network support is only for independent LUs.)

Limitations

The number of conversations from a local LU to a remote LU must not exceed 512 per mode.

The maximum number of intermediate sessions in a network node is 2000.

The maximum number of active user modes allowed between a local LU and a remote LU is 14.

The maximum number of devices that can be associated with a controller is 254.

The maximum RU length is 16,384.

The maximum length of a BIND is 512. The maximum length of the RSCV on a BIND is 255. The number of hops in the RSCV depends on the length of the CP names.

8.1.3 Implemented Functions

	Mentioned on	AS/400 NN	AS/400 EN
Session Services			
CP-CP Sessions	page 17	yes	yes
Multiple CP-CP Session Partners	page 9	yes	no
Mode to COS Mapping	page 71	yes	yes
Limited Resource	page 32	yes	yes
BIND Segmenting	page 24	yes	yes
BIND Reassembly	page 24	yes	yes
User LU Name same as CP Name	page 84	yes	yes
Intermediate Session Routing			
Intermediate Session Routing	page 18	yes	no
Adaptive and Fixed Pacing	page 19	yes	no
Intermed. Session RU Segmenting	page 18	yes	yes
Intermed. Session RU Reassembly	page 18	yes	yes
Border Node	page 167	yes	no
Directory Services			
Broadcast Searches	page 89	yes	yes
Participate in Broadc. Search	page 96	yes	yes
Directed Searches	page 93	yes	yes
Directory Caching	page 82	yes	no
Predefinition of Remote LUs	page 79	yes	yes
Safe/Store of DS Cache	page 83	yes	no
Register LUs with NN Server	page 80	yes	yes
Multiple Founds Multitail LEN	page 97	no	no
Wildcard Search Reply	page 99	yes	yes
Topology and Routing Services			
Topology Broadcast	page 50	yes	no
Initial Topology Exchange	page 50	yes	no
Safe/Store of TDB	page 46	yes	yes
Garbage Collection of TDB	page 51	yes	yes
Randomized Route Computation	page 60	yes	yes
COS/TPF Option	page 52	yes	yes
Tree Caching	page 55	yes	no
Incremental Updates to Trees	page 69	no	no
Management Services			
Multiple Domain Support	page 130	yes	yes
MS for EN in NN (RQE1)	page 125	yes	yes
MS for EP to Explicit FP	page 127	yes	yes
MS for EP to Implicit FP	page 127	no	no
Held Alerts	page 133	yes	yes
Receive NMVT on SSCP-PU Sess.	page 131	yes	yes
Send NMVT on SSCP-PU Session	page 131	yes	yes
Connectivity			
Connection Network	page 40	yes	yes
Transmission Priority	page 53	yes	yes
Multiple TGs	page 30	yes	yes
Parallel TGs	page 30	yes	yes
Multiple TGs to Subarea Network	page 30	yes	yes

8.2 3174

The 3174 support for Advanced Peer-to-Peer Networking was announced in March 1991. The 3174 acts as a network node only. For detailed information, refer to *3174 Planning Guide Configuration Support C*.

APPN is a feature of 3174 configuration support C Release 1. It allows application programs that use the APPC interface (LU 6.2) to communicate with other APPC programs in the APPN network or on the host.

Previously, the standard 3174 acted as a T2.0 node only. In 1990, an RPQ became available, which could provide gateway functions for independent LUs in downstream T2.1 nodes, too. Neither the RPQ nor the T2.0 functions are further described in this document.

8.2.1 Terminology

In the 3174 context, the term "gateway" is likely to apply to the 3174 token-ring gateway feature. An APPN gateway node doesn't exist today.

A shared link is used for T2.1 traffic and for T2.0 traffic at the same time. In this case, the SSCP-PU session will be requested in the XID3.

The 3174 contains limited resources for "dynamic links". When the session count goes to zero, the link is taken down. That does not apply to CP-CP sessions, as they are persistent.

The 3174 assumes that all end nodes are authorized.

Network node routing resources depletion
When this condition is detected, sessions are directed away from the node. It is detected automatically, when the configured maximum number of sessions allowed has been reached. A node is considered no longer depleted, when the number of intermediate sessions drops below the maximum number.

Congestion
Congestion is automatically calculated when a large portion of available buffers have been assigned. Resource depletion and congestion still allow additional sessions to be routed through the network node, if no better routes can be found. Yet, the probability of additional sessions is reduced, as the weight of this node will be increased. The information is broadcast each time a node enters or leaves congestion or depletion state.

8.2.2 System Definitions

The 3174 does not have an on-line node operator facility. Any NOF or definitions are done offline through customization.

Limitations

The safe/store cache function is supported, if the 3174 has a hardfile.

Upstream (through the host), any LU 6.2 can be reached. Downstream support is limited to LU 6.2 on token-ring or LU 6.2 on LEN end nodes using Peer Communication. IBM intends to extend the support to APPN end nodes on coax.

Upstream for SDLC and channel links, the 3174 is always the secondary station (not negotiable).

The node's "route addition resistance" is fixed to 128.

The maximum number of nodes in the topology database is 297.

The maximum RU size is 8 KB.

The maximum number of intermediate LU 6.2 sessions is 1000.

Generally speaking, the maximum number of links is 225. If the 4Mbps token-ring adapter is used, the limit is 140; however, when the 8 KB frame size is used, then the limit is 100.

The maximum number of adjacent network nodes is eight.

Up to 120 CPs may specify LU names. A maximum of four LUs may be defined per CP name.

8.2.3 Implemented Functions

	Mentioned on	3174 NN
Session Services		
CP-CP Sessions	page 17	yes
Multiple CP-CP Session Partners	page 9	yes
Mode to COS Mapping	page 71	yes
Limited Resource	page 32	yes
BIND Segmenting	page 24	yes
BIND Reassembly	page 24	yes
User LU Name same as CP Name	page 84	yes
Intermediate Session Routing		
Intermediate Session Routing	page 18	yes
Adaptive and Fixed Pacing	page 19	yes
Intermed. Session RU Segmenting	page 18	yes
Intermed. Session RU Reassembly	page 18	yes
Border Node	page 167	no
Directory Services		
Broadcast Searches	page 89	yes
Participate in Broadc. Search	page 96	yes
Directed Searches	page 93	yes
Directory Caching	page 82	yes
Predefinition of Remote LUs	page 79	yes
Safe/Store of DS Cache	page 83	yes
Register LUs with NN Server	page 80	yes
Multiple Founds Multitail LEN	page 97	no
Wildcard Search Reply	page 99	yes
Topology and Routing Services		
Topology Broadcast	page 50	yes
Initial Topology Exchange	page 50	yes
Safe/Store of TDB	page 46	yes
Garbage Collection of TDB	page 51	yes
Randomized Route Computation	page 60	yes
COS/TPF Option	page 52	yes
Tree Caching	page 55	no
Incremental Updates to Trees	page 69	no
Management Services		
Multiple Domain Support	page 130	yes
MS for EN in NN (RQE1)	page 125	yes
MS for EP to Explicit FP	page 127	no
MS for EP to Implicit FP	page 127	no
Held Alerts	page 133	yes
Receive NMVT on SSCP-PU Sess.	page 131	yes
Send NMVT on SSCP-PU Session	page 131	yes
Connectivity		
Connection Network	page 40	yes
Transmission Priority	page 53	yes
Multiple TGs	page 30	yes
Parallel TGs	page 30	no
Multiple TGs to Subarea Network	page 30	no

8.3 PS/2

The PS/2 support for LEN end nodes was announced in 1988. The PS/2 support for APPN network nodes and APPN end nodes was announced in March 1991. For detailed information, refer to *NS/2 Installation and Network Administrator's Guide*.

The LEN end node software for the PS/2 is part of OS/2* EE. The support for APPN end nodes and APPN network nodes is in a separate product called Networking Services/2, which is based on the OS/2 Communication Manager.

8.3.1 Terminology

Substitute network node server for end node A, is a network node to which end node A does not have a CP-CP session. But it is specified as a full wildcard, and all BINDs flow there, when end node A does not have a CP-CP session with its network node server.

Congestion status and adaptive session pacing are controlled by the utilization of the send/receive storage. The node considers itself as congested, if less than 128KB of send/receive storage remains available. The node considers itself as no longer congested, if more than 37% of send/receive storage is available. The congestion status change is broadcast.

Adaptive session pacing is implemented according to the following rules:

- When more than 50% of send/receive storage is available, the pacing window size can be increased.
- When less than 50% of send/receive storage is available, the pacing window size is not increased.
- When less than 37% of send/receive storage is available, the pacing window size is reduced to half its previous value.
- When between 64KB and 128KB of send/receive storage is available, the pacing window size is set to 1.
- When less than 64KB of send/receive storage is available, the pacing message is held.

8.3.2 System Definitions

NS/2 supports receipt of XID0, in which case it acts as a primary link station.

Limitations

Only one network node can be specified as server. But, another server can be designated as substitute server (by using the end node's wildcard function).

Route addition resistance is fixed to 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.

8.3.3 Implemented Functions

	Mentioned on	PS/2 NN	PS/2 EN	PS/2 LEN
Session Services				
CP-CP Sessions	page 17	yes	yes	no
Multiple CP-CP Session Partners	page 9	yes	no	no
Mode to COS Mapping	page 71	yes	yes	no
Limited Resource	page 32	yes	yes	yes
BIND Segmenting	page 24	yes	yes	no
BIND Reassembly	page 24	yes	yes	no
User LU Name same as CP Name	page 84	yes	yes	no
Intermediate Session Routing				
Intermediate Session Routing	page 18	yes	no	no
Adaptive and Fixed Pacing	page 19	yes	no	no
Intermed. Session RU Segmenting	page 18	yes	no	no
Intermed. Session RU Reassembly	page 18	yes	no	no
Border Node	page 167	no	no	no
Directory Services				
Broadcast Searches	page 89	yes	yes	no
Participate in Broadc. Search	page 96	yes	yes	no
Directed Searches	page 93	yes	no	no
Directory Caching	page 82	yes	no	no
Predefinition of Remote LUs	page 79	yes	yes	yes
Safe/Store of DS Cache	page 83	yes	no	no
Register LUs with NN Server	page 80	yes	yes	no
Multiple Founds Multitail LEN	page 97	yes	no	no
Wildcard Search Reply	page 99	yes	yes	no
Topology and Routing Services				
Topology Broadcast	page 50	yes	no	no
Initial Topology Exchange	page 50	yes	no	no
Safe/Store of TDB	page 46	no	no	no
Garbage Collection of TDB	page 51	yes	no	no
Randomized Route Computation	page 60	yes	no	no
COS/TPF Option	page 52	yes	yes	no
Tree Caching	page 55	yes	no	no
Incremental Updates to Trees	page 69	yes	no	no
Management Services				
Multiple Domain Support	page 130	yes	yes	no
MS for EN in NN (RQE1)	page 125	yes	yes	no
MS for EP to Explicit FP	page 127	yes	yes	no
MS for EP to Implicit FP	page 127	no	no	no
Held Alerts	page 133	yes	yes	no
Receive NMVT on SSCP-PU Sess.	page 131	yes	yes	yes
Send NMVT on SSCP-PU Session	page 131	yes	yes	yes
Connectivity				
Connection Network	page 40	yes	yes	no
Transmission Priority	page 53	no	no	no
Multiple TGs	page 30	yes	yes	yes
Parallel TGs	page 30	yes	yes	yes
Multiple TGs to Subarea Network	page 30	yes	yes	yes

8.4 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. Being part of SAA* (System Application Architecture*), VTAM and NCP will implement APPN in a future release. The architectural aspects of APPN for the subarea network are covered in "SNA Functions in Addition to APPN" on page 163.

Information in this chapter is based on VTAM V3 R4 and NCP V5 R4. The LEN function can be provided by VTAM and NCP together, or by VTAM on its own.

8.4.1 Terminology

Most APPN functions have analogies in the subarea network. Unfortunately, different terms are used for similar functions.

Altough, from the point of view of APPN, the VTAM/NCP complex is considered a LEN end node, VTAM and NCP implement numerous functions that are not included in the basic LEN architecture. Intermediate session routing is an example.

Dynamic cross domain resource
Dynamic cross domain resource provides reduced definitions like the cache directory.

CDINIT rerouting
CDINIT rerouting can be seen as a sequence of directed searches. For an adjacent LEN end node, VTAM can act as a pseudo-server. It can route a BIND to an APPN network node, where a new search can be started.

Transmission priority and class of service
The class of service name is part of the logon mode table. The class of service is used to select an operational route from a list of pre-defined routes. The transmission priority is associated to the route. Transmission priority refers to routes between subareas and to route extensions (boundary links).

Casual Connect
Towards an APPN node the subarea network presents itself as a LEN end node. The boundary function that supports the link to the APPN node may be either in VTAM or in the NCP. Two subarea networks may be connected through each side's boundary function. Thus, each side sees the adjacent side as one LEN end node. This LEN connection between two NCPs is not as powerful as the regular link between two subareas (full-duplex, multi-link TGs) and is practically restricted to independent LU 6.2 sessions. The term "casual" hints at these restrictions. This function complements the wildcard function on the APPN side.

Adjacent link station (ALS) selection function
The ALS selection function in the VTAM session management exit can be programmed to select a route to an LU when multiple links exist in a LEN connection. This is equivalent to selective wildcard routing.

8.4.2 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 which provide dynamic network access and eliminate the need for double definitions for certain classes of LUs.

Self-defining independent LUs
When a BIND from a LEN end node enters the subarea network, the OLU is automatically defined (dynamic CDRSC). In an excellent way, this function complements the wildcard search function in APPN networks. The DLU can be pre-defined, a dynamic CDRSC, or automatically defined by a VTAM exit. See "Adjacent link station (ALS) selection function" on page 144.

Dynamic switched definition support
Dynamic switched definition support simplifies adding switched devices to the network, including leased link attached token-ring devices. This support is for physical units and dependent or independent logical units. For dial-in support, reusable model definitions in conjunction with an installation exit routine are used.

Limitations

The main limitation is, of course, that VTAM and NCP currently implement the LEN end node function only.

Route selection between APPN networks and subarea networks is not seamless, as independent algorithms apply.

Multiple links from the APPN network to the subarea network require VTAM V3 R4 / NCP V5 R4 or later releases.

Notes referring to "Implemented Functions" on page 146:

y#1 see "Transmission priority and class of service" on page 144.

y#2 see "CDINIT rerouting" on page 144.

y#3 see "Dynamic cross domain resource" on page 144.

8.4.3 Implemented Functions

	Mentioned on	VTAM/NCP LEN
Session Services		
CP-CP Sessions	page 17	no
Multiple CP-CP Session Partners	page 9	no
Mode to COS Mapping	page 71	y#1
Limited Resource	page 32	yes
BIND Segmenting	page 24	no
BIND Reassembly	page 24	no
User LU Name same as CP Name	page 84	no
Intermediate Session Routing		
Intermediate Session Routing	page 18	yes
Adaptive and Fixed Pacing	page 19	yes
Intermed. Session RU Segmenting	page 18	yes
Intermed. Session RU Reassembly	page 18	no
Border Node	page 167	no
Directory Services		
Broadcast Searches	page 89	no
Participate in Broadc. Search	page 96	y#2
Directed Searches	page 93	y#2
Directory Caching	page 82	y#3
Predefinition of Remote LUs	page 79	y#3
Safe/Store of DS Cache	page 83	no
Register LUs with NN Server	page 80	no
Multiple Founds Multitail LEN	page 97	no
Wildcard Search Reply	page 99	no
Topology and Routing Services		
Topology Broadcast	page 50	no
Initial Topology Exchange	page 50	no
Safe/Store of TDB	page 46	no
Garbage Collection of TDB	page 51	no
Randomized Route Computation	page 60	no
COS/TPF Option	page 52	no
Tree Caching	page 55	no
Incremental Updates to Trees	page 69	no
Management Services		
Multiple Domain Support	page 130	no
MS for EN in NN (RQE1)	page 125	no
MS for EP to Explicit FP	page 127	no
MS for EP to Implicit FP	page 127	no
Held Alerts	page 133	no
Receive NMVT on SSCP-PU Sess.	page 131	yes
Send NMVT on SSCP-PU Session	page 131	yes
Connectivity		
Connection Network	page 40	no
Transmission Priority	page 53	yes
Multiple TGs	page 30	yes
Parallel TGs	page 30	yes
Multiple TGs to Subarea Network	page 30	yes

8.5 Summary of Implemented Functions

	Mentioned on	AS/400 NN	AS/400 EN	3174 NN	PS/2 NN	PS/2 EN	PS/2 LEN	VTAM LEN
Session Services								
CP-CP Sessions	page 17	yes	yes	yes	yes	yes	no	no
Multiple CP-CP Session Partners	page 9	yes	no	yes	yes	no	no	no
Mode to COS Mapping	page 71	yes	yes	yes	yes	yes	no	yes
Limited Resource	page 32	yes	yes	yes	yes	yes	yes	yes
BIND Segmenting	page 24	yes	yes	yes	yes	yes	no	no
BIND Reassembly	page 24	yes	yes	yes	yes	yes	no	no
User LU Name same as CP Name	page 84	yes	yes	yes	yes	yes	no	no
Intermediate Session Routing								
Intermediate Session Routing	page 18	yes	no	yes	yes	no	no	yes
Adaptive and Fixed Pacing	page 19	yes	no	yes	yes	no	no	yes
Intermed. Session RU Segmenting	page 18	yes	yes	yes	yes	no	no	yes
Intermed. Session RU Reassembly	page 18	yes	yes	yes	yes	no	no	no
Border Node	page 167	yes	no	no	no	no	no	no
Directory Services								
Broadcast Searches	page 89	yes	yes	yes	yes	yes	no	no
Participate in Broadc. Search	page 96	yes	yes	yes	yes	yes	no	yes
Directed Searches	page 93	yes	yes	yes	yes	no	no	yes
Directory Caching	page 82	yes	no	yes	yes	no	no	yes
Predefinition of Remote LUs	page 79	yes	yes	yes	yes	yes	yes	yes
Safe/Store of DS Cache	page 83	yes	no	yes	yes	no	no	no
Register LUs with NN Server	page 80	yes	yes	yes	yes	yes	no	no
Multiple Founds Multitail LEN	page 97	no	no	no	yes	no	no	no
Wildcard Search Reply	page 99	yes	yes	yes	yes	yes	no	no
Topology and Routing Services								
Topology Broadcast	page 50	yes	no	yes	yes	no	no	no
Initial Topology Exchange	page 50	yes	no	yes	yes	no	no	no
Safe/Store of TDB	page 46	yes	yes	yes	no	no	no	no
Garbage Collection of TDB	page 51	yes	yes	yes	yes	no	no	no
Randomized Route Computation	page 60	yes	yes	yes	yes	no	no	no
COS/TPF Option	page 52	yes	yes	yes	yes	yes	no	no
Tree Caching	page 55	yes	no	no	yes	no	no	no
Incremental Updates to Trees	page 69	no	no	no	yes	no	no	no
Management Services								
Multiple Domain Support	page 130	yes	yes	yes	yes	yes	no	no
MS for EN in NN (RQE1)	page 125	yes	yes	yes	yes	yes	no	no
MS for EP to Explicit FP	page 127	yes	yes	no	yes	yes	no	no
MS for EP to Implicit FP	page 127	no	no	no	no	no	no	no
Held Alerts	page 133	yes	yes	yes	yes	yes	no	no
Receive NMVT on SSCP-PU Sess.	page 131	yes	yes	yes	yes	yes	yes	yes
Send NMVT on SSCP-PU Session	page 131	yes	yes	yes	yes	yes	yes	yes
Connectivity								
Connection Network	page 40	yes	yes	yes	yes	yes	no	no
Transmission Priority	page 53	yes	yes	yes	no	no	no	yes
Multiple TGs	page 30	yes	yes	yes	yes	yes	yes	yes
Parallel TGs	page 30	yes	yes	no	yes	yes	yes	yes
Multiple TGs to Subarea Network	page 30	yes	yes	no	yes	yes	yes	yes

Chapter 9. Sample Configurations

The previous chapters provided a tutorial on the APPN architecture and a summary of APPN functions installed in the various products. This chapter addresses some of the aspects of connecting APPN end nodes to one network node server, connecting an APPN and subarea network in a multi-domain environment, and sample configuration that address the APPN multi-network capabilities. Finally, a configuration is depicted that shows the benefits of APPN using connection networks.

9.1 Routing between APPN End Nodes

The configuration in Figure 71 shows an APPN network node and three APPN end nodes. They are interconnected any-to-any, except for one connection: there is no link between ″ENA″ and ″ENC″. CP-CP sessions exist between ″NND″ and ″ENA″, also between ″NND″ and ″ENB″.

Definitions

All nodes define their local LUs and the links to their adjacent nodes.

″ENC″ also defines those remote LUs that it may want to establish sessions with. (That is because ″ENC″ does not have a network node server.) LUB is defined as being in ″ENB″. LUA is defined as being in ″NND″. As we can see, LUA is not really in ″NND″, but from the point of view of ″ENC″, ″NND″ is the owner. In other words, ″NND″ acts as pseudo-server for ″ENC″.

In ″NND″, LUC is pre-defined as being in ″ENC″.

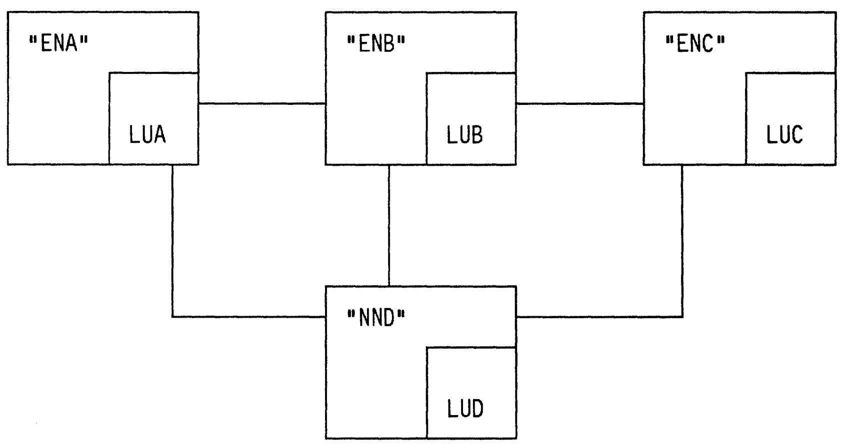

Figure 71. Routing between APPN End Nodes

Routing

Sessions between ″ENA″ and ″ENB″ are established with assistance of the network node server in ″NND″. They use the direct link between ″ENA″ and ″ENB″.

Sessions between ″ENA″ and ″ENC″ always go through ″NND″.

Sessions from "ENB" to "ENC" go through "NND", as long as the LUs of "ENC" are not known in "ENB".

Sessions from "ENC" to "ENB" use the direct link, because LUB is pre-defined.

Highlights

- APPN end nodes with CP-CP sessions have the benefit of minimal system definitions.
- APPN end nodes with CP-CP sessions have the benefit of optimal routing to adjacent nodes.
- APPN end nodes without CP-CP sessions can still use the adjacent network node as pseudo-server, but do not get the advantages of full APPN.
- This configuration can be thought of as being on a token-ring. In that case, with the same number of definitions, additional connectivity can be achieved.

9.2 Resource Definition for Multi-Tail APPN Subarea Connection

Connecting an APPN network to a subarea network requires definitions in both the APPN node that connects to the subarea node, and in the subarea node that connects to the APPN node. Currently these definitions must be synchronized with each other, otherwise, the subarea node does not accept session initiation requests from the originating logical unit. In the following two examples, the APPN network has multiple connections to the subarea network. In the first example, the subarea domains are controlled by VTAM Version 3 Release 3. In the second example, the subarea domains are controlled by the recently announced VTAM Version 3 Release 4. This release of VTAM fully supports multi-tail connections from an APPN network, and further reduces the system definition for independent LUs.

9.2.1 Resource Definition for APPN Subarea Connection (VTAM V3R3)

In Figure 72 on page 151, the APPN network connects to the subarea network with two links. One link attaches a 3174 and the other link an AS/400. Both, the 3174 and the AS/400 act as network nodes in the APPN network, and both connect to the subarea as T2.1 nodes. The 3174 supports, on the same link, two dependent LUs (for example 3179s).

Subarea Definitions

The network administrator has defined all independent LUs on the link towards the AS/400. The link is controlled by LEN1 and has the following definitions:

```
LN1LN   LINE  .......
LN1PU   PU    .......
NNCL1   LU    LOCADDR=0           <----located on network node NNC
EN3L1   LU    LOCADDR=0           <----located on end node EN3
```

For the link to the 3174, the network administrator has defined two dependent LUs and no independent LUs. The link is controlled by LEN2:

```
LN2LN   LINE  ....
LN2PU   PU    ....
DPLU1   LU    LOCADDR=1
DPLU2   LU    LOCADDR=2
```

APPN Definitions

The network administrator for "NNA" (AS/400) has defined the full wildcard "ITSC.*ANY" for LEN end node "LEN1". This implies that all session initiation requests for destination LUs with network ID "ITSC", which cannot be located in the APPN network, are assumed to be available at LEN end node "LEN1". Any BIND that is received by "NNA" as a result of the full wildcard, is sent across the link to "LEN1".

On "NNB" (3174) the APPN administrator did not define any resource or a wildcard that points to the LEN end node "LEN2". Thus, the 3174 will not enable independent LUs to establish a session across the link to "LEN2".

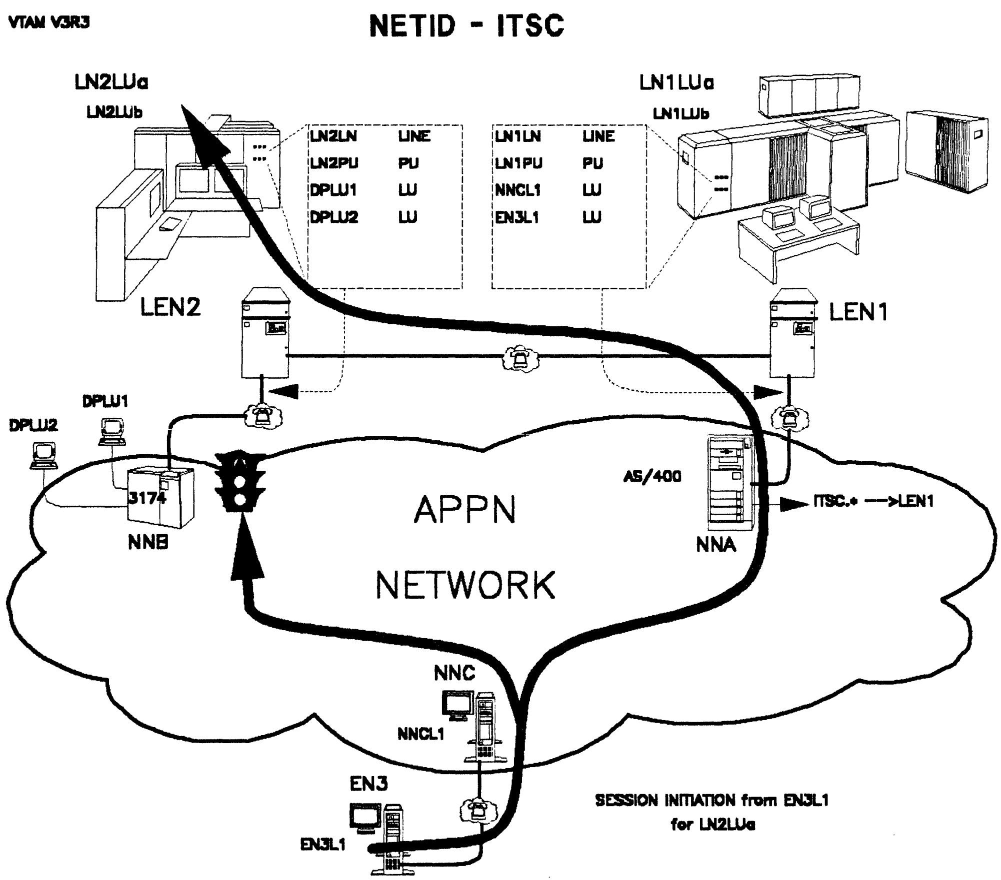

Figure 72. Resource Definition for APPN and Subarea Connection (VTAM V3R3)

Routing

If, for example, "ITSC.EN3L1" at "EN3" initiates a session (BIND) with "ITSC.LN2LUa", network node server "NNC" initiates a "broadcast search" to locate LU "ITSC.LN2LUa". On receipt of the "broadcast search", network node "NNB" (3174) replies that the resource cannot be located. However, "NNA" replies that "ITSC.LN2LUa" might be available on LEN end node "LEN1". It includes in the reply an indicator that the match was found in its directory database using a full wildcard. As no other node in the APPN network replied that the resource was known at its node, "ITSC.EN3L1" forwards the session activation request (BIND) to "NNA".

"NNA" forwards the BIND across the link to subarea node "LEN1". "LEN1" verifies whether the originator of the request (ITSC.EN3L1) corresponds with one of the LU definitions for the line, and, if the network ID of the origin LU corresponds with the network ID of subarea node "LEN1". If either the LU or network ID does not correspond, the BIND is rejected, else, subarea node "LEN1" searches its tables for resource "ITSC.LN2LUa". "LEN1" cannot locate the resource in its domain and requests its adjacent subarea domain "LEN2" whether it has knowledge of the resource. "LEN2" derives from its tables that the resource is located at its node and informs "LEN1" accordingly. As a consequence, "LEN1" forwards the BIND to "LEN2", which then delivers the BIND to "ITSC.LN2LUa".

The following example addresses the session activation request (BIND) from the subarea node. LU "ITSC.LN1LUa" at "LEN1" activates a session (BIND) with "ITSC.NNCL1". The subarea searches its tables for resource "ITSC.NNCL1". It derives from its tables that the resource is located on a link towards "NNA". As a consequence the (BIND) is sent across the link to node "NNA"

"NNA" does not own the resource, nor does it find any information in the BIND request that tells "NNA" to whom the BIND should be routed. Therefore, "NNA" initiates a "broadcast search" through the APPN network to locate resource "ITSC.NNCL1". Network node "NNC" informs "NNA" that it is the owner of the resource. "NNA" calculates the route to "NNC", includes it in the BIND, and forwards the BIND to "NNC".

9.2.2 Resource Definition for APPN Subarea Connection (VTAM V3R4)

With VTAM Version 3 Release 4, two new T2.1 features were announced. They are:

- T2.1 LU dynamics

  The network administrator at the subarea node no longer needs to define the independent LUs for a T2.1 nodes. The independent LU is automatically created by VTAM when the independent LU sends the session activation (BIND) request to the subarea.

- T2.1 multi-tail

  With T2.1 multi-tail support, the T2.1 node can have multiple connections into the subarea network. Thus, independent LUs can establish sessions in or through the subarea network, across all available connections.

Subarea Definitions

For the following example, we use the configuration as depicted in Figure 73 on page 153. Using the above mentioned VTAM Version 3 Release 4 features, the minimal definitions that need to be made by the network administrator for subarea node "LEN1" are:

```
LN1LN    LINE  .......
LN1PU    PU    .......
```

The network administrator for subarea node "LEN2" still needs to define the two dependent LUs:

```
LN2LN    LINE  ....
LN2PU    PU    ....
DPLU1    LU    LOCADDR=1
DPLU2    LU    LOCADDR=2
```

APPN Definitions

The network administrator for "NNA" (AS/400) has defined the full wildcard "ITSC.*ANY" for LEN end node "LEN1". This implies that all session initiation requests for destination LUs with network ID "ITSC", which cannot be located in the APPN network, are assumed to be available at LEN end node "LEN1". Any BIND that is received by "NNA" as a result of the full wildcard, is sent across the link to "LEN1".

The network administrator for network node "NNB" (3174) has defined a partial wildcard. The partial wildcard directs all session requests for destination LUs, starting with "ITSC.LN2", to LEN end node "LEN2". For all other session initiation requests, network node "NNB" informs the originating network node that the resource cannot be located.

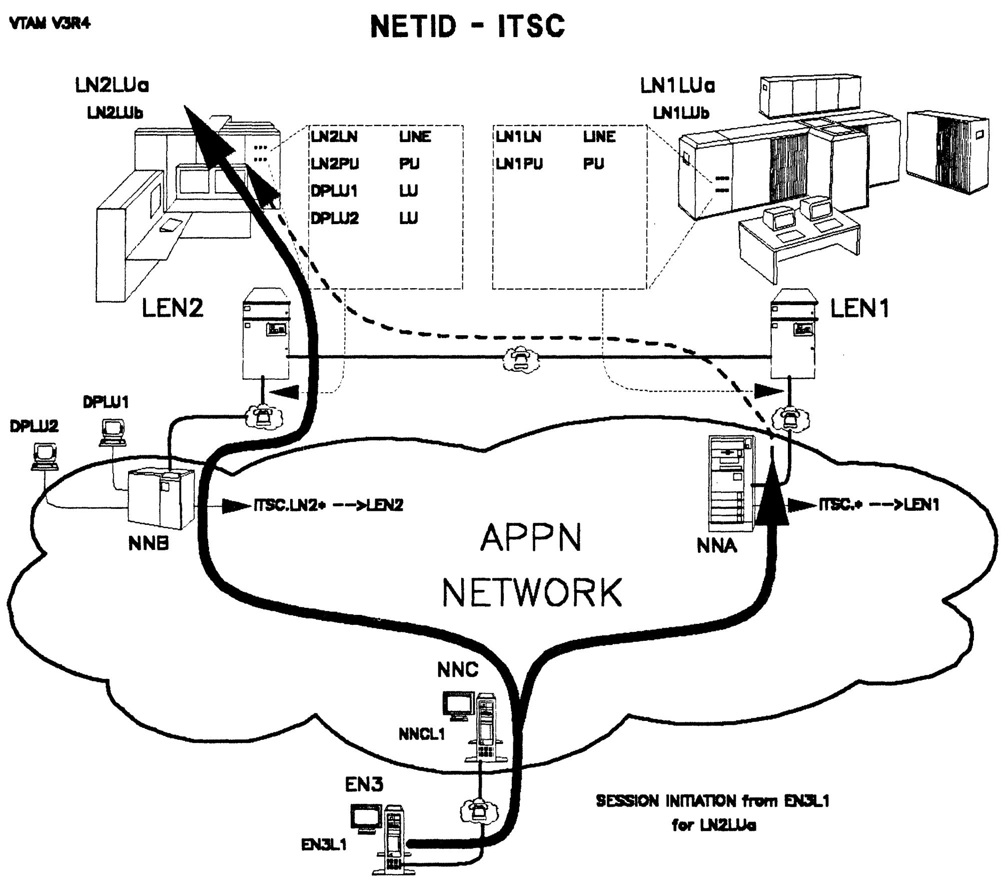

Figure 73. Resource Definition for APPN Subarea Connection (VTAM V3R4)

Routing

If "ITSC.EN3L1" at "EN3" initiates a session with "ITSC.LN2LUa", its network node server "NNC" initiates the "broadcast search" to "NNB" and "NNA". Network node "NNA" replies that the resource might be available on LEN end node "LEN1", but that it used the full wildcard to reach that descision. Thus, "NNC" temporarily stores the reply until all replies to the "broadcast search" have been received.

Network node "NNB" replies that the resource is available on LEN end node "LEN2", and that it located the resource using the explicit name. As a consequence, the reply from "NNA" is discarded by "NNC", and the session activation request (BIND) is forwarded to "NNB". "NNB" sends the BIND over the link towards "LEN2", which locates destination resource "ITSC.LN2LUa" at its node, and delivers the BIND to it.

In addition, "LEN2" generates a so-called cross domain resource (CDRSC). The CDRSC identifies LU "ITSC.EN3L1", and that the LU is located on node "NNB".

The following two examples handle session activation requests from the subarea node.

- In the first example, LU "ITSC.LN1LUa" at "LEN1" activates a session (BIND) with "ITSC.EN3L1". The subarea node "LEN1" searches its tables for "ITSC.EN3L1". It does not find the resource so it requests whether its adjacent subarea domain "LEN2" has knowledge of resource "ITSC.EN3L1". "LEN2" derives from its tables (CDRSC that was previously generated) that the resource is located on a link towards "NNB" and supplies the information to "LEN1". As a consequence, the (BIND) is sent from "LEN1" to "LEN2". "LEN2" forwards the BIND on the designated link towards "NNB".

  "NNB" locates the resource in the APPN network, calculates the route towards destination LU "ITSC.EN3L1", and forwards the BIND to its destination.

- In the second example, LU "ITSC.LN2LUa" at "LEN2" activates a session (BIND) with "ITSC.NNCL1". The subarea node "LEN2" searches its tables for "ITSC.NNCL1". It does not find the resource, so it requests its adjacent subarea domain "LEN1" whether it has knowledge of resource "ITSC.NNCL1". "LEN1" does not find the resource in its tables either and informs "LEN2" accordingly. As a result, "LEN2" informs "ITSC.LN1LUa" that the resource cannot be located.

The above mentioned examples show that if the destination LU is located in the APPN network, manual definitions may still be needed. With VTAM Version 3 Release 4 these definitions should be done using CDRSC definitions. This allows independent LUs, like "ITSC.EN3L1" and "ITSC.NNCL1", to enter session initiation requests or respond to BINDs from the subarea network using either "NNA" or "NNB" as entry point to the subarea network.

9.3 APPN Multi-Network Capabilities

An APPN network requires that all network nodes that are part of this network have the same network ID. In the following sections, two possible solutions are shown to connect two independent APPN networks. The examples only show those parts of each network that are relevant to these examples.

9.3.1 APPN Multi-Network Using VTAM

In Figure 74 on page 156, two independent APPN networks, "NETA" and "NETB", are connected to a subarea network with network ID "NETC". The subarea network is controlled by VTAM Version 3 Release 3 with its non-native network connection (NNNC) feature. This feature allows an APPN network to connect to the subarea network with a network ID that differs from the network ID of the subarea.

APPN Definitions

The network administrator for network node "NN1" has defined adjacent node "LEN1" in "NETC" as LEN end node and specified a full wildcard "*.*---> LEN1" for this link. This implies that all session initiation requests for destination LUs with *any* network ID, which cannot be located in the APPN network, are assumed to be available at node "LEN1". Any BIND, that is received as a result of the full wildcard reply, is sent by "NN1" across the link towards "LEN1".

The network administrator for network node "NNB" has defined adjacent node "LEN1" in "NETC" as LEN end node. The administrator defined two full wildcards for this link. They are:

- "NETB.* --> "LEN1"
- "NETC.* --> "LEN1"

This implies that all session initiation requests for destination LUs with either network ID "NETB" or "NETC", which cannot be located in the APPN network, are assumed to be available at node "LEN1". Any BIND, that is received as a result of the full wildcard reply, is sent by "NNB" across the link towards "LEN1".

Subarea Definitions

The network administrator for the subarea network "NETC" has defined the following line definitions for connecting the two APPN nodes.
For connecting node "NN1" in network "NETB":

```
LN1LN    LINE  ....
LN1PU    PU    XNETALS=YES,NETID=NETB
APPL1    LU    LOCADDR=0 ....
```

For connecting node "NNB" in network "NETA":

```
LN2LN    LINE  ....
LN2PU    PU    XNETALS=YES,NETID=NETA ....
APPL2    LU    LOCADDR=0 ....
```

Routing

When LU "NETB.APPL1" at "EN1" initiates a session with "NETA.APPL2", its network node server "NN2" initiates the "broadcast search" to locate the resource throughout network "NETB". On receipt of the request, network node "NN1" replies that resource "NETA.APPL2" is available at LEN end node "LEN1", and that it used a full wildcard to reach that decision. As no other node in network "NETB" found the resource, the session activation request (BIND) is sent to "NN1", which forwards the BIND to LEN end node "LEN1".

On receipt of the BIND, subarea node "LEN1" checks whether the origin LU "NETB.APPL1" has been defined for the line, and whether the network ID of the origin LU, "NETB", corresponds with the network ID defined for the physical unit (PU) on the link towards "NN1". In our example, the network ID defined for the PU is "NETB". If either the LU or network ID does not correspond, the BIND will be rejected else subarea node "LEN1" searches its tables for resource "APPL2". "APPL2" is located and the BIND is forwarded by the subarea node on the link to "NNB".

"NNB" derives the destination LU name, "NETA.APPL2", from the BIND and searches its directory database. The destination LU is not located, thus, "NNB" initiates a "broadcast search" to locate the destination LU. "NNC" replies that LU "NETA.APPL2" is located on end node "ENA". "NNB" calculates the route

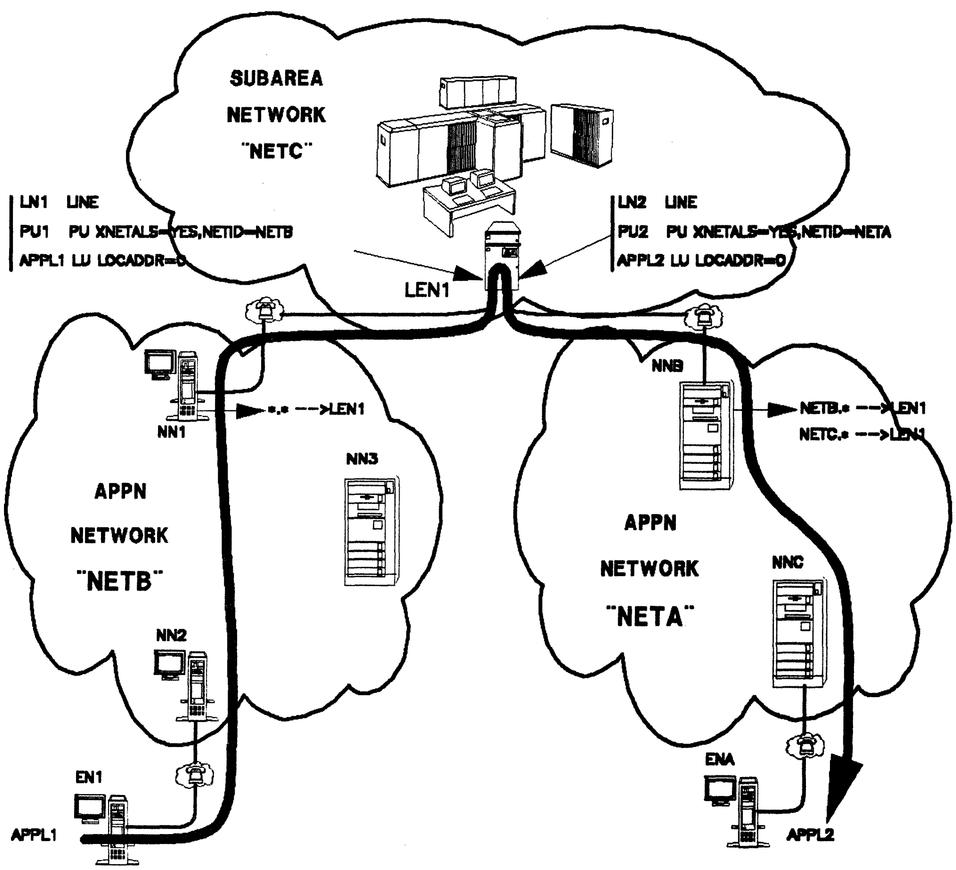

Figure 74. APPN Multi Network using VTAM

towards the end node, includes it in the BIND and forwards the BIND to its destination.

9.3.2 APPN Multi-Network Using Border Node

In Figure 75 on page 157, the two APPN networks are connected using the border node. Currently, the AS/400 is the only product that has implemented the border node function on top of the network node function.

APPN Definitions

The network administrator for "NN3" in network "NETB" has defined node "NNA" in network "NETA" as adjacent APPN end node for which it acts as network node server.

The network administrator for "NNA" in network "NETA" has defined "NN3" in network "NETB" as network node. When "NNA" connects to "NN3" it is informed that "NN3" is a network node but that there is a mismatch in network ID. Being a border node, "NNA" automatically switches its image towards "NN3" and informs "NN3" that it is an APPN end node. It requests network node "NN3" to be its network node server and to receive "broadcast search" requests for resources unknown to network node "NN3". Unlike a normal APPN end node, "NNA" does not own resources.

Thus, node "NNA" presents a network node image towards the nodes in its own network "NETA", and the end node image to its network node in network "NETB".

Internally the network node function in "NNA" will pass search requests to the end node function in "NNA" for resources, that cannot be located in the directory database by the network node function of "NNA".

The end node function of "NNA" passes these search requests to "NN3". "NN3" searches its directory database and initiates the "broadcast search" if the resource cannot be located in its directory database. Search requests that are received by "NN3", and that cannot be located by "NN3", are forwarded to the end node function of "NNA", as requested by the end node, when it established CP-CP sessions with its network node server.

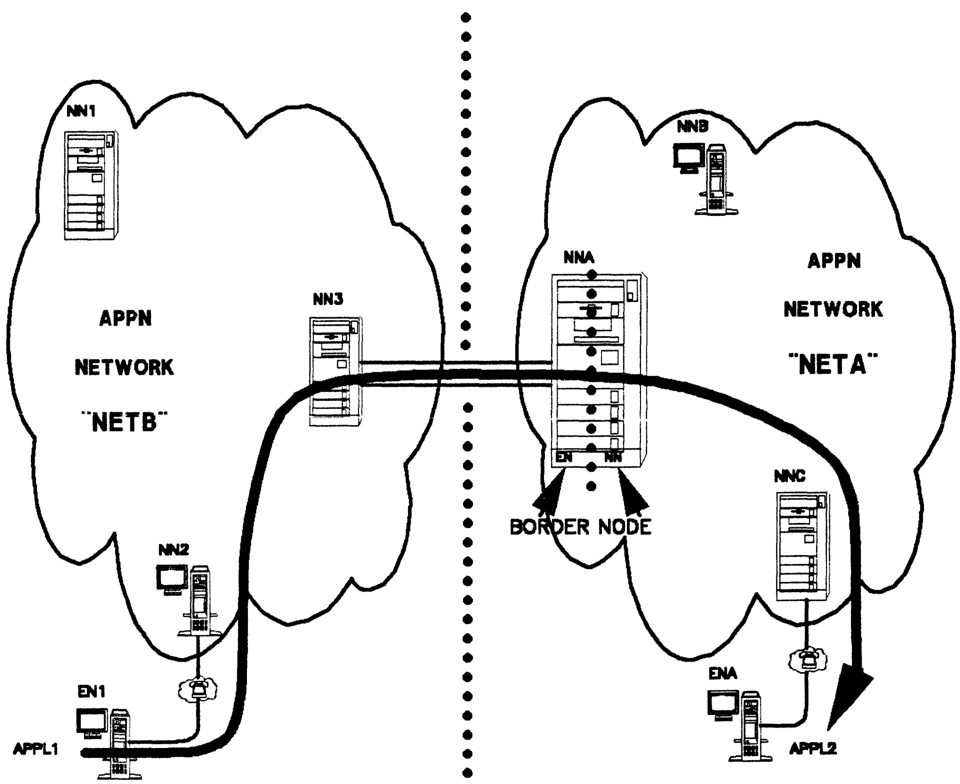

Figure 75. APPN Multi-Network using Border Node

Routing

When LU "NETB.APPL1" at "EN1" initiates a session with "NETA.APPL2", its network node server "NN2" initiates the "broadcast search" to locate the resource throughout network "NETB". On receipt of the request, network node "NN3" searches its local directory database for resource "NETA.APPL2". As the resource cannot be located in the local directory database, network node "NN3" searches its domain. One of its end nodes, "NNA", requested to receive the domain "broadcast search" for resources unknown to "NN3", hence the "broadcast search" for resource "NETA.APPL2" is forwarded to its APPN end node "NNA".

The end node function of "NNA" automatically transfers the search request to the network node function in "NNA". "NNA", in turn, initiates a "broadcast search" in network "NETA". Network node "NNC" informs "NNA" that the resource is located on its end node "ENA". This positive reply is carried across the border node to "NN3", which informs the network node server of "NETB.APPL1". Finally the BIND is completed and sent across border node "NNA" towards its destination "NETA.APPL2" at "ENA".

9.4 APPN and Connection Networks

The T2.1 node that is connected to the LAN establishes direct connections to other T2.1 nodes, using only one LAN interface address. Normally, each T2.1 node defines the LAN address of every T2.1 node with which it establishes a connection. In a LAN network with a large number of T2.1 nodes, it would require a significant administrative overhead on each T2.1 node.

With the introduction of APPN, and specifically the connection network, the administrative overhead is reduced to **two** definitions for each APPN end node; these are the LAN address of its network node server, and its access to a virtual routing node in a connection network using its own LAN address.

9.4.1 Connection Network with Virtual Routing Node

The configuration in Figure 76 on page 159 shows a shared access transport facility (SATF), such as a token-ring. Each node has one physical port into the SATF with several sub-channels.

All APPN end nodes have CP-CP sessions with network node "NNN". "NNM" is just another network node on the SATF. LEN end node "LEN" uses the network node "NNN" as its pseudo-server.

Where is the virtual routing node? It is not in Figure 76 on page 159 because it does not physically exist. The virtual routing node is a means to define a connection network.

Definitions

As depicted in Figure 77 on page 159, all end nodes define two links into the SATF:

1. To the network node server "NNN"
2. To the virtual routing node "VRNX".

The serving network node "NNN" defines the connections to the end nodes in its domain. It also defines the connection to the LEN end node "LEN", and any LUs in "LEN" that can be used as destination LUs in the network. Any other network nodes have to be defined only if they are adjacent, such as network node "NNM".

The network node "NNM" defines the adjacent network node "NNN" and its link to the connection network represented by "VRNX".

LEN end node "LEN" defines all its connections and all remote session partners.

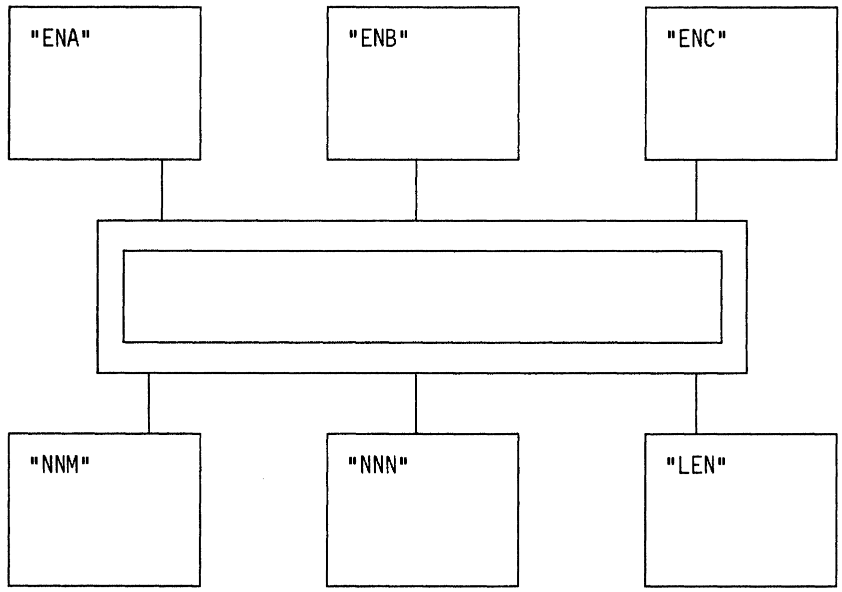

Figure 76. Connection Network: What You Install

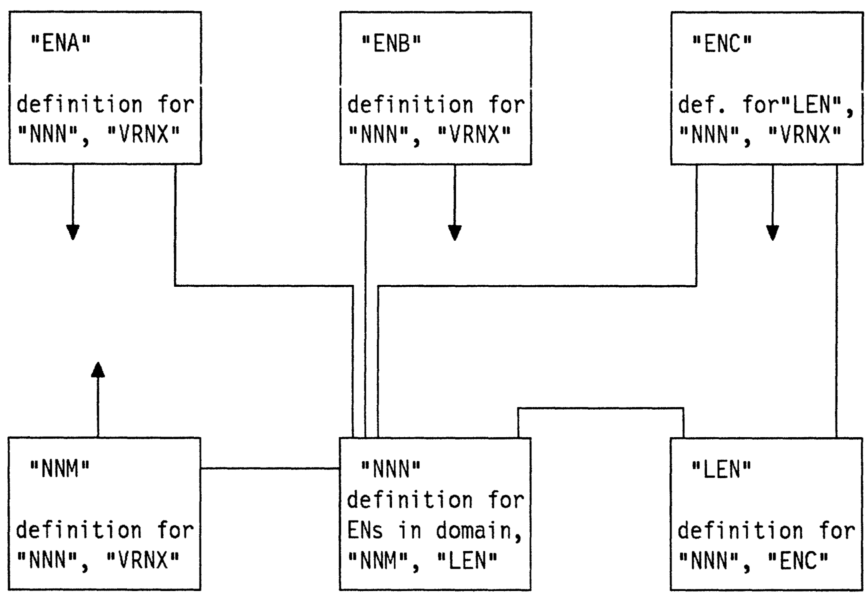

Figure 77. Connection Network: What You Define

Routing

Sessions from the end nodes are established with the assistance of the network node server in "NNN". Sessions between the end nodes use the single hop route across the SATF (no intermediate nodes in between).

The connection between "LEN" and "ENC" is used for sessions between LUs on those nodes. All other sessions have to go through "NNN" as the pseudo-server for "LEN" (multiple hop route).

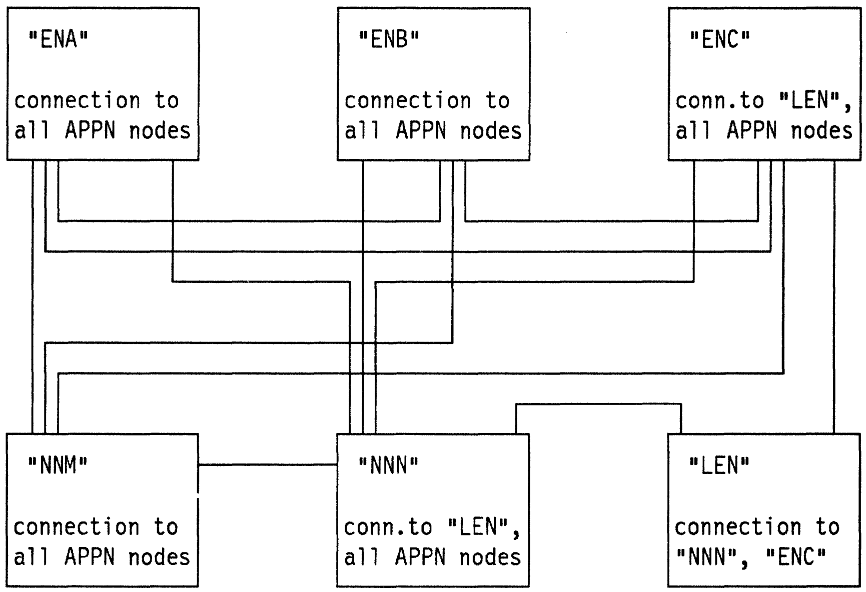

Figure 78. Connection Network: What You Get

Highlights

- All connections in this configuration use the SATF.
- Only those that define the virtual routing node "VRNX" use the connection network.
- CP-CP sessions do not use the connection network.
- The LEN end node "LEN" gets only what it defines.
- The end nodes (which represent the majority of nodes in the network) get the real benefit of the connection network. With minimal definitions, they get the maximum connectivity and the shortest routes.

9.4.2 Sample APPN Configuration in a LAN Environment

This example describes a possible LAN configuration using a 3174, two AS/400 systems, and three PS/2s.

APPN Definitions

Each APPN end node (EN1, EN2, EN3, EN4, and EN5) has defined network node "NNA" as its network node server. Each end node has defined the LAN address of "NNA" and its access to a virtual routing node, in our case IBMTR, in the connection network.

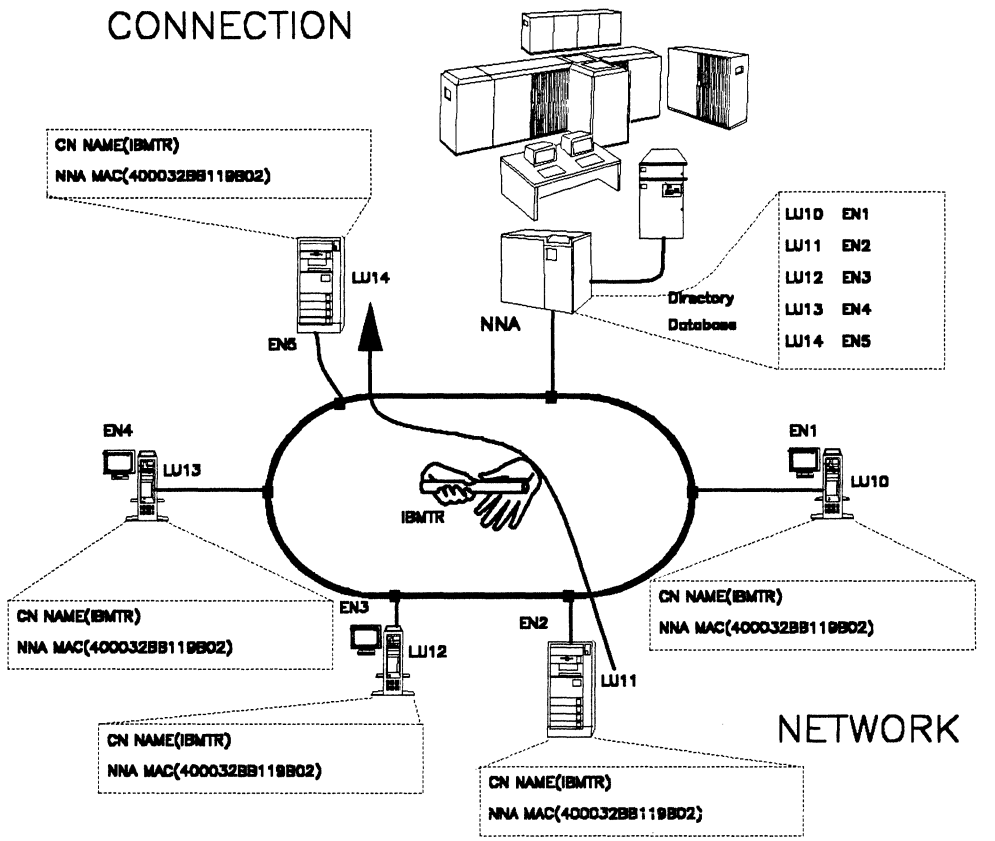

Figure 79. Considerations for APPN in a LAN environment

Routing

In Figure 79, each APPN end node has established CP-CP sessions with its network node server "NNA", and has registered its resources with the network node server. When resource "LU11" at "EN2" initiates a session with "LU14", its end node "EN2" searches its local directory database for the destination resource. As the resource is not located, "EN2" requests assistance from its network node server "NNA" to locate the resource. In its request to the network node server, "EN2" supplies information about its links to network nodes and connection networks. In our case, "EN2" supplies information about its link to "NNA" and its connection to virtual routing node "IBMTR" together with its addressing information to the virtual routing node.

Network node "NNA" locates the destination resource "LU14" in its directory database and derives that "LU14" is owned by "EN5". "NNA" forwards the request to "EN5" and requests the end node to verify whether the resource is available, and if so, to supply the end node's links towards network nodes and connection networks. The resource is available, thus, "EN5" replies that it has a link to network node "NNA" and a link to a virtual routing node called "IBMTR" together with its addressing to a virtual routing node.

On receipt of the information from "EN5", network node "NNA" calculates the route between the two endpoints "LU11" and LU14". As both have access to the same virtual routing node, "NNA" supplies "EN2" with the addressing information (in our example a token-ring address) of the target node. "EN2" derives the token-ring address from the reply and establishes a direct link with "EN5", and "LU10" establishes its session with "LU14" across this link.

Chapter 10. SNA Functions in Addition to APPN

The previous chapters of this document are based on the APPN architecture, as announced in March 1991. It seems logical, but not certain, that every IBM communication product should sooner or later participate in APPN, especially VTAM and NCP. This chapter lists some of the required changes that would have to be made to the APPN architecture in order to make it suitable for VTAM and NCP. We also mention functions that have been suggested to improve the efficiency of connection networks end very large networks.

It should be mentioned that the functions in this chapter are not part of the current APPN architecture nor are they planned to be included in future VTAM, NCP, or APPN products.

10.1 Seamless Connection between APPN Network and Subarea Network

The combination of APPN networks and subarea networks can be done with the current software. But this combination is done on the level of a LEN end node, thus, the full functions of both subarea networking and APPN networking cannot be exploited. The main reason for this restriction is the lack of CP-CP sessions.

In order to provide a seamless connection, there must be nodes that support the full functions of both flavors of Systems Network Architecture (subarea networking and Advanced Peer-to-Peer Networking). Preferably these nodes should perform tasks such as:

- Present an APPN appearance to APPN nodes
- Present a subarea appearance to subarea nodes.
- Transform search procedures

10.2 Dependent LU Support

A dependent LU may reside in either a T2.0 or T2.1 node. Like for the T2.0 node, the T2.1 receives network services for its dependent LUs only from an SSCP in a T5 node. The SSCP first establishes an SSCP-PU session with the PU function in the T2.1 node, and then establishes an SSCP-LU session with each dependent LU in that node. The SSCP-PU session is primarily used for network management, and the SSCP-LU session is used to initiate sessions and keep the SSCP aware of the status of the LU.

When a node with a dependent LU is attached to an APPN network, an SSCP is no longer available to provide network services. Preferably, the services provided by an APPN network node should be extended with the network services currently provided by an SSCP for dependent LUs. The dependent LU could either be located on the network node itself, or, on an adjacent T2.0 or T2.1 node for which the network node acts as network node server.

10.3 Initiating SLU Support

In subarea networking SLU-initiated sessions are widely used. The SLUs, that initiate sessions in a subarea network, are located on T2.0 and T2.1 nodes, and sometimes on T5 nodes. In the SNA literature the SLUs are also referred to as dependent LUs. These SLUs can not act as PLU in a session and they can be either LUs of type LU 6.2, or, LUs of type LU1, LU2, or LU3. For example, the LU2 type protocol is commonly used by 3270 type displays.

The LUs, with which these SLUs initiate sessions, are host applications that reside on T5 nodes. The host applications activate the session (BIND) and act therefore as PLU of the session.

Some, older, applications can only act as the PLU in a session. Thus, although these applications may support the LU 6.2 protocol and therefore can establish sessions with independent LUs, they require the independent LU to act as SLU in the session.

The current APPN architecture provides functions which support PLU-initiated sessions only. Therefore, the APPN architecture should be extended to support SLU-initiated sessions from both independent as well as dependent LUs.

10.4 Third Party Initiate

Another commonly used function in subarea networking is third party initiate. The third party initiate function can only be requested by a application program that resides on a T5 node. The application program requests the function on behalf of an SLU that has established a session with that application. Triggering this function, CLSDST PASS in subarea terminology, the application program requests the controlling SSCP to establish a session between the SLU and a new target application, and to deactivate the session between the SLU and the requesting application. The application that requests the function must be the PLU of the session while the SLU, on whose behalf the third party initiate is requested, remains the SLU in the new session.

Several application programs use the third party initiate because of their internal program design. That is, although an application program LU may have multiple concurrent sessions with SLUs using only **one** application LU, this application requires an unique LU for each SLU that initiates a session with the application. Good examples of such applications are time sharing option (TSO) and NetView*, IBM's widely used network management application. Both applications require an unique application LU for each SLU that initiates a session with the application. Each application is known with one LU name throughout the subarea network. The LU name is assigned to the main task of the application. The SLU initiates a session with that main task but, as soon as the main task has succesfully established a session with that SLU, it selects one of the LUs, assigned to that application. Using the third party intiate function, it requests its controlling SSCP to establish a session between the SLU, with which it has a session, and the selected application LU.

Another good example of a third party initiate application is a security application, for example NetView/Access. An enterprise may enforce SLUs to establish a session with a security application to verify the identity of the end user that uses the SLU, before the end user may use the network resources. If the end user is successfully authenticated, the end user could select one of the applications for which the end user is authorized. As a result, the security application uses the third party initiate function to request its controlling SSCP to establish a session between SLU and selected application.

Currently, the third party initiate is not supported for LU 6.2 sessions. However, due to increased focus on security (especially for inter-enterprise connections), authentication of LU 6.2 end users may be required on network boundaries, before these end users are allowed to access network resources of an enterprise. This latter would also require the third party initiate function for LU 6.2 sessions.

10.5 Dynamic Discovery of Network Node Server

This possible function applies to shared access transport facilities (SATF), as described in section "Connection Networks Using a Shared-Access Transport Facility" on page 40. Currently, the connection network is of benefit for connections between end nodes on one SATF. For them, the system definition of a virtual routing node makes definitions for any other end node dispensable. Yet, the network node server still has to be defined by the end nodes.

System definition requirements are reduced further, if the end node can discover the network node server automatically. This is possible, whenever the transport facility supports broadcast protocols.

APPN network node servers on the SATF could send heartbeat messages at regular intervals. APPN end nodes which are added to the SATF would listen for the heartbeat and send an XID to the network node server to obtain its services for any further connections.

10.6 Multi-Network Connectivity

A single APPN network requires the same network ID for all its participating network nodes. Network nodes that do not have the same network IDs cannot establish a connection with each other.

For end nodes is not required to have the same network ID as their network node server. In principle each end node can have its own network ID.

Partitioning networks with different network IDs is common practice in SNA subarea networks. The terminology used in subarea networking for partitioning networks is SNA Network Interconnect (SNI). SNI was announced for subarea networking in 1983. Currently it is widely used by customers in both single- as well as multi-enterprise environments.

Large enterprises use SNI to build independent networks based upon their organizational structure. These networks are then merged into a single logical network for the purpose of data exchange. This way the management and design of these networks remain independent from each other while providing any-to-any connectivity.

The need for multi-network capability is even stronger between enterprises. One of the main contributors to multi-enterprise networking is electronic data interchange (EDI) with trading partners. An enterprise network, that may already be a multi-network itself, connects to other enterprise networks, either directly or through the use of a value added network (VAN) provider such as IBM Information Network. Hence the multi-network may span two or more networks.

The growing demand for interconnecting these independent networks will also be required for APPN networks. Interconnection of APPN networks should be accomplished without losing the APPN distributed search capabilities, routing facilities, and preferably without additional system definitions. The general term that is used for a node that interconnects two distinct networks is *gateway node.*

Preferably the APPN multi-network capability should provide for:

- Network topology isolation

  Network topology information that is normally exchanged between network nodes in a distinct APPN network should not cross network boundaries.

- Support any multi-network configuration

  The multi-network capability should not place any restrictions on the type of multi-network configurations. For example the multi-network configuration could be a cascaded, star-shaped, or a meshed multi-network

- Multiple gateways between two networks

  The interconnection of two networks should allow for one or more gateway nodes. This allows each network to have its own gateway node and thus control the access to its own network. Eventually a network may have multiple gateways to one network to allow for load balancing and backup.

- Transparent search for resources

  The search for resources in a multi-network should be transparent to origin LU and its end user. However end user should be aware that resources in another network should be network qualified in the session initiation request. Resources that are not network qualified receive the network ID of the originating node.

- Route calculation

  In a single APPN network topology and routing services calculates the route from origin to destination LU. This requires however network topology information from all participating networks. The network topology information is not exchanged across network boundaries thus topology and routing services at the origin node can only calculate the route towards its gateway node. Topology and routing services at the gateway node should provide the route through the adjacent network, either from gateway node to destination node or from gateway node to gateway node.

  Another item that should be taken into account is the mapping of COS names. COS names may not exist in another network or may have different properties assigned to it. For example a COS name may be low priority in one network and have a high priority in another network.

- Gateway customization capabilities

  The APPN distributed search capabilities (broadcast search) are automatically sent across network boundaries. This could introduce a significant overhead in interconnected networks. Therefore the gateway should provide a facility to selectively allow or deny these searches. The selection could be based upon origin LU name, destination LU name, network ID, or even mode and COS name.

- Network management procedures

Network management concept should not be restricted by network boundaries. That is a focal point should be able to establish a relationship with an entry point in a another network.

However in a multi-enterprise environment a focal point to entry point relationship may not be desirable because of security related issues.

Border Nodes

The first system that implemented a subset of multi-network connectivity is AS/400. The AS/400 multi-network capability is part of the OS/400* Version 2 Release 1 announcement (April 1991). This subset is also referred to as Border Node.

The border node provides additional functions on top of the APPN network node. It enables:

- LUs to establish sessions with LUs in an adjacent network
- Isolation of network topology information between adjacent networks
- Session problem determination across adjacent networks.

A network node with border node capabilities, hereafter referred to as border node, acts as a network node in its own network. Thus the border node receives the network topology information and the distributed search requests from its adjacent network nodes. The border node connects to a network node in the adjacent network. As two network nodes cannot currently connect to each other with different network ID, the border node presents an end node image to the network node. An end node can establish a session with a network node using a different network ID

The end node function in the border node establishes CP-CP sessions with the network node and indicates in its CP capabilities that it requires end node search support. For more information see "APPN End Node Search Support" on page 96. As a consequence the network node automatically sends any broadcast search for a resource that cannot be located in its local directory database, to the end node function in the border node. This way directory search requests can be exchanged between the two networks. Network topology information is not sent between the two networks as this function is not supported for end nodes.

In the example, in Figure 80, there are three interconnected networks. They are NETA, NETB, and NETC. NETA connects to both NETB and NETC. The border node in NETA connects as end node to network nodes in NETB and NETC. Please note that the end node function in a border node may have more than one network node server. Normally an end node can only have one network node server.

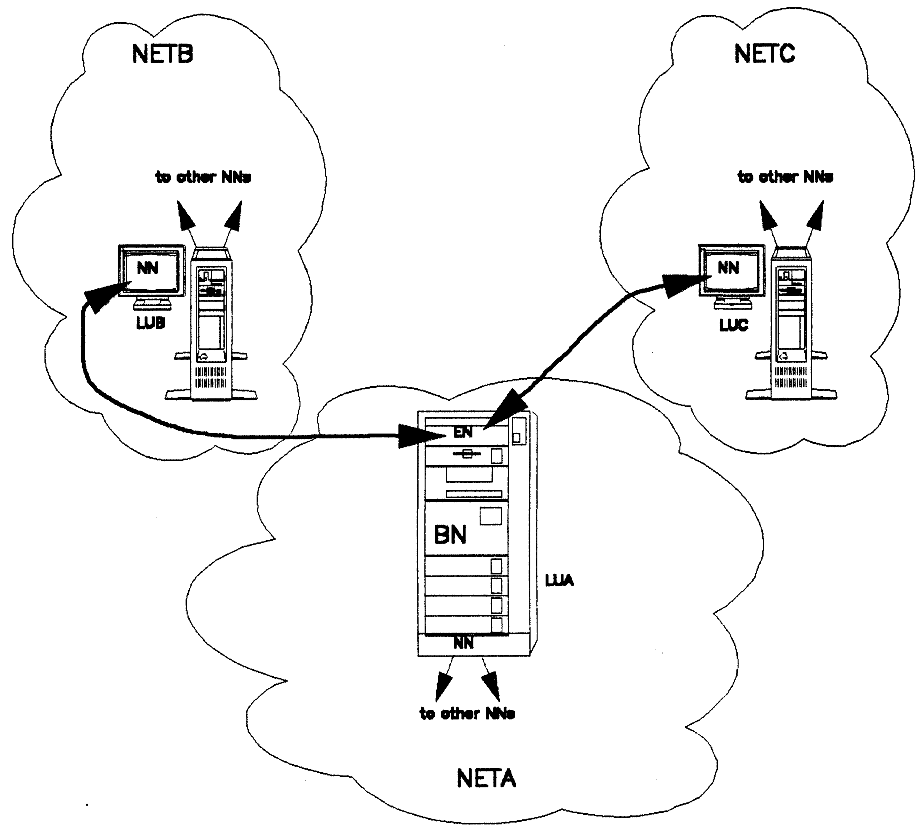

Figure 80. Multi-Network Connection using Border Node

LUA in NETA can establish sessions with both NETB.LUB and NETC.LUC. However LUB in NETB cannot establish a session with LUC in NETC as these networks are **not** adjacent. The network with the border node does not propagate search requests from NETB destined for NETC or vice versa. Nor will it act as intermediate node for a session between an LU in NETB and NETC.

Moreover if an end node is connected to NETA using a different network ID then the border node does not allow session requests from either NETB or NETC for an LU on that end node. The border node only allows directory services requests and session activation (BIND) requests to pass if the network ID of the destination LU is the same as the adjacent network.

Abbreviations

ABBREVIATION	MEANING
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
CS	configuration services
DAF	destination address field
DD	directory database
DLU	destination logical unit
DLC	data link control
DS	directory services
EN	end node
ENCP	end node control point
EP	enntry point
FID	format identifier
FQPCID	fully qualified process control id
FRSN	flow reduction sequence number
GDS	generalized 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
LN	low entry networking node
MAC	medium access control
MSU	management services unit
NAU	network accessible unit
NCP	network control program
NN	network node
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
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

READER'S COMMENT FORM

Title: **APPN Architecture and Product Implementations Tutorial**
GG24-3669-00

You may use this form to communicate your comments about this publication, its organization, or subject matter, with the understanding that IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

Your comments:

If you would like a reply, please supply your name and address on the reverse side of this form.

Thank you for your cooperation. No postage stamp is necessary if mailed in the U.S.A.. (Elsewhere, an IBM office or representative will be happy to forward your comments or you may mail directly to the address in the Edition Notice on the back of the title page.)

Reader's Comment Form

Fold and Tape — Please Do Not Staple — Fold and Tape

NO POSTAGE NECESSARY IF MAILED IN THE UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 40 ARMONK, N.Y.

POSTAGE WILL BE PAID BY ADDRESSEE:
IBM International Technical Support Center
Department 985, Building 657
P.O. Box 12195
Research Triangle Park, NC 27709
U.S.A.

Fold — Fold

Return address:

Your Name ____________________

Company Name ____________________ *Department* __________

Street Address ____________________

City ____________________

State ____________________ *Zip Code* __________

IBM Branch Office serving you ____________________

IBM®

GG24-3669-00

APPN Architecture and Product Implementations Tutorial

PRINTED IN THE U.S.A.

READER'S COMMENT FORM

Title: **APPN Architecture and Product Implementations Tutorial**
GG24-3669-00

You may use this form to communicate your comments about this publication, its organization, or subject matter, with the understanding that IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

Your comments:

If you would like a reply, please supply your name and address on the reverse side of this form.

Thank you for your cooperation. No postage stamp is necessary if mailed in the U.S.A.. (Elsewhere, an IBM office or representative will be happy to forward your comments or you may mail directly to the address in the Edition Notice on the back of the title page.)

Reader's Comment Form

Fold and Tape — Please Do Not Staple — Fold and Tape

NO POSTAGE NECESSARY IF MAILED IN THE UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 40 ARMONK, N.Y.

POSTAGE WILL BE PAID BY ADDRESSEE:
IBM International Technical Support Center
Department 985, Building 657
P.O. Box 12195
Research Triangle Park, NC 27709
U.S.A.

Fold — Fold

Return address:

Your Name ______________________

Company Name ______________________ *Department* __________

Street Address ______________________

City ______________________

State ______________________ *Zip Code* __________

IBM Branch Office serving you ______________________

IBM®

GG24-3669-00

PRINTED IN THE U.S.A.

READER'S COMMENT FORM

Title: **APPN Architecture and Product Implementations Tutorial**
GG24-3669-00

You may use this form to communicate your comments about this publication, its organization, or subject matter, with the understanding that IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

Your comments:

If you would like a reply, please supply your name and address on the reverse side of this form.

Thank you for your cooperation. No postage stamp is necessary if mailed in the U.S.A.. (Elsewhere, an IBM office or representative will be happy to forward your comments or you may mail directly to the address in the Edition Notice on the back of the title page.)

Reader's Comment Form

Fold and Tape — Please Do Not Staple — Fold and Tape

NO POSTAGE NECESSARY IF MAILED IN THE UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 40 ARMONK, N.Y.

POSTAGE WILL BE PAID BY ADDRESSEE:
IBM International Technical Support Center
Department 985, Building 657
P.O. Box 12195
Research Triangle Park, NC 27709
U.S.A.

Fold — Fold

Return address:

Your Name ______________________

Company Name ______________________ *Department* __________

Street Address ______________________

City ______________________

State ______________________ *Zip Code* __________

IBM Branch Office serving you ______________________

IBM®

GG24-3669-00

APPN Architecture and Product Implementations Tutorial

PRINTED IN THE U.S.A.

GG24-3669-00

APPN Architecture and Product Implementations Tutorial

GG24-3669-00

PRINTED IN THE U.S.A.

GG24-3669-00