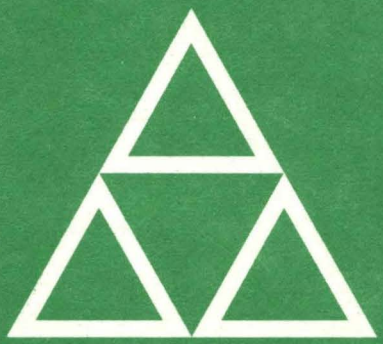
NSD Systems Center
Washington

An Overview:
Advanced Peer-to-Peer Networking

Robert J. Hines



Technical Bulletin

NSD Systems Center - Washington
Gaithersburg, Maryland
Technical Bulletin

An Overview
Advanced Peer-to-Peer Networking

Robert J. Hines

First Edition: This Technical Bulletin is being made available to IBM and customer personnel. It has not been subject to any formal review and is distributed on an "As Is" basis without any warranty, either expressed or implied. The exact organization and implementation of functions described will vary from installation to installation and must be individually evaluated for applicability.

References in this publication to IBM products, programs or services do not imply that IBM intends to make these available outside the United States.

A form for reader's comments is provided at the back of this publication. If this form has been removed comments may be addressed to:

Robert J. Hines
IBM Corporation
NSD Systems Center - Washington
Dept. 76X
18100 Frederick Pike
Gaithersburg, MD 20879

IBM may use or distribute any of the information you supply in any way that it believes appropriate without incurring any obligation whatever. You may of course continue to use the information you supply.

DAPS Code 0895
August 25, 1988

An Overview

Advanced Peer-to-Peer Networking

August 25, 1988

Robert J. Hines

About this Bulletin

SNA Low Entry Networking (SNA LEN) is a new extension to the SNA architecture that supports peer-to-peer communication. This support is the basis for Systems 36 (S/36) and the newly announced Application System 400 (AS/400) Advance Peer-to-Peer Networking (APPN) feature. The APPN feature is provided via Type 2.1 node services and an enhanced LU 6.2, Advanced Program-to-Program Communication (APPC). The discussions in this bulletin will be limited primarily to the S/36 implementation.

APPN provides a method to establish and maintain a network of S/36 and AS/400 systems in a powerful easy to use manner. Users in this environment can install and maintain complex networks without requiring highly skilled programmers and operators.

The intent of this bulletin is to present a broad base technical overview of the new services implemented in the S/36 Advance Peer-to-Peer Networking (APPN) feature and a brief look at the physical connectivity SNA LEN provides when an APPN network is attached to the subarea network. Throughout this bulletin I make a constant effort to point out that APPN is not a part of the SNA LEN architecture but represents new services implemented in the S/36 and AS/400. These new services are compatible and may coexist with the various connections and protocols in use today.

In the last half of this bulletin, we turn our attention to the unique problems presented when performing session problem determination (PD) in an LU 6.2 environment. Considerable complexity is added to this process because logical units are capable of having multiple session partners. The network operator can no longer depend on having a one-to-one relationship between an end user and a LU when gathering PD data. Therefore, the data needed and the methods used to collect this data are explored. Finally, enhancements to the diagnostics aids currently available are discussed with regards to their use in this process.

I would liked to point out that the technical content of this bulletin is directed primarily toward those readers who have prior SNA experience and is intended to be informative not instructional.

Preface

Acknowledgements

I would like to take this opportunity to personally thank all individuals who reviewed drafts of this technical bulletin. My very special thanks go out to John Waclawsky, NSDSC - Washington, Charles Woods, and Fred Fletcher both from Advanced Design, Communication Product Division, for their suggestions and comments in helping to make this bulletin as technically accurate as possible.

Contents

Figures

Chapter 1. SNA Low Entry Networking

Introduction

In 1974, Systems Network Architecture (SNA) was introduced as IBM's blueprint for the future design of telecommunication networks. In its original design and implementation, SNA users were limited to single host networks with attached programmable controllers and terminals. Since that time, there have been a number of extensions made to the SNA architecture to allow continued network growth and to satisfy the telecommunication needs and requirements of a great number of its users.

When we look at some of the more recent extensions to SNA, we think of both SNA Network Interconnection (SNI) and Extended Network Addressing (ENA). Both of these extensions were significant and had a tremendous impact on the growth potential of networks.

SNI allowed those SNA users who had become address constrained because of the interdependences between the subarea/element address split to divide a large network into a number of smaller networks. Those smaller networks were then interconnected through a Gateway function which gave addressability across the network boundary while maintaining the autonomy of the interconnected networks. SNI greatly expanded the SNA addressability of an enterprise. SNI's implementation was not a simple task and required a certain level of expertise that we associate with a complex environment.

SNA users who had very large networks soon found VTAM storage utilization to be a critical factor. In many cases it was on the verge of being totally depleted because of a substantial increase in VTAM's working set size. ENA, through Virtual Storage Constraint Relief (VSCR), provided a solution in this area. VSCR allowed VTAM to operate above the 16 mega-byte line and to more effectively manage this valued resource. Additionally, ENA theoretically gave the SNA user an unlimited addressing potential within a network by removing the inter-dependency between the subarea/element address split. ENA's implementation was extremely simple when compared to SNI and did not place any special requirements on the user's level of expertise.

This new extension to the architecture, SNA Low Entry Networking (SNA LEN) and the S/36 Advanced Peer-to-Peer Networking (APPN) feature will incorporate a whole new community of SNA users. The new services of APPN will allow for the peer connection of systems of various sizes into networks of arbitrary topology. The goal of this extension is to better serve the needs of small systems users, down to the current generation of personal computers. SNA LEN, along with APPN, allow the direct peer connection between adjacent nodes in conjunction with the dynamic definition of network resources. Simply stated, this implementation merges control point and routing functions into a single node, thereby eliminating the external configuration services flows needed between VTAM and NCP. The main topics in this bulletin are directed toward the functional implementation of the new APPN services and cover in some detail the following:

- Terminology
- SNA Environment
- APPN Environment
- Why SNA LEN
- Sys/36 APPN Services
- LU 6.2 Session PD

Terminology

You have probably heard both SNA Low Entry Networking and Advanced Peer-to-Peer Networking mentioned in the same breath and it may have left you a little confused as to what each meant. Before proceeding, I will define some of the new terminology used and how it will apply to this discussion.

- **SNA Low Entry Networking (LEN)** - With the emergence of intelligent work stations having increased processing power and storage, a need is created to extend the capabilities of peripheral nodes as they are known in today's SNA environment. This new extension, SNA LEN, is the term used to describe the direct peer-to-peer connection between adjacent T2.1 nodes.
- **Advanced Program-to-Program Communication (APPC)** - APPC (LU 6.2) first announced in 1983, has been enhanced and supports both Independent and dependent LUs.
  1. **Dependent LU** - A dependent LU receives an activate logical unit (ACTLU) from the host SSCP, can act only as a secondary LU, and can have only one LU-LU session active at a time. Dependent LUs are supported by SNA LEN for migration purposes only.
  2. **Independent LU** - An independent LU does not receive an ACTLU from the host SSCP, may act as a primary or secondary LU, and may have a session limit greater than one.
- **Advanced Peer-to-Peer Networking (APPN)** - Advanced Peer-to-Peer Networking (APPN) is the System 36 implementation as a T2.1 node using LU 6.2. The S/36, previously limited to the role of a T2.0 peripheral node, and the new AS/400 are currently the only systems that have this capability. Both may be implemented as either a network node or an end node acting as a primary or secondary session partner.

  I believe to further understand SNA Low Entry Networking and S/36 APPN, we need to understand the types of nodes that can exist in this configuration.

  - A T2.1 peripheral node (PN) is a non-subarea node that may be attached to a subarea node thru a 3725, an SDLC link, or Token Ring.
  - An end node (EN) in the S/36 APPN environment is a node that does not provide any network services to other nodes. All sessions that enter an end node terminate within that end node except for local terminals and LUs which are considered inside the end node as far as the network structure is concerned.
  - A network node (NN) in the S/36 APPN environment is a node that provides intermediate routing functions, route selection services, and directory services to end nodes. It is analogous to a subarea network domain which has been compressed into a single node.
  - A composite end node (CEN) is a VTAM domain that supports T2.1 protocols. To the attached APPN node (EN,NN) the collection of nodes that comprises the VTAM domain appears as a single T2.1 node.

Again, SNA Low Entry Networking is the direct peer connection between adjacent T2.1 nodes. When implemented with the S/36 APPN feature, those nodes functioning as NNs have the capability to dynamically define and reconfigure network resources as they enter and leave the network.

SNA Backbone Network

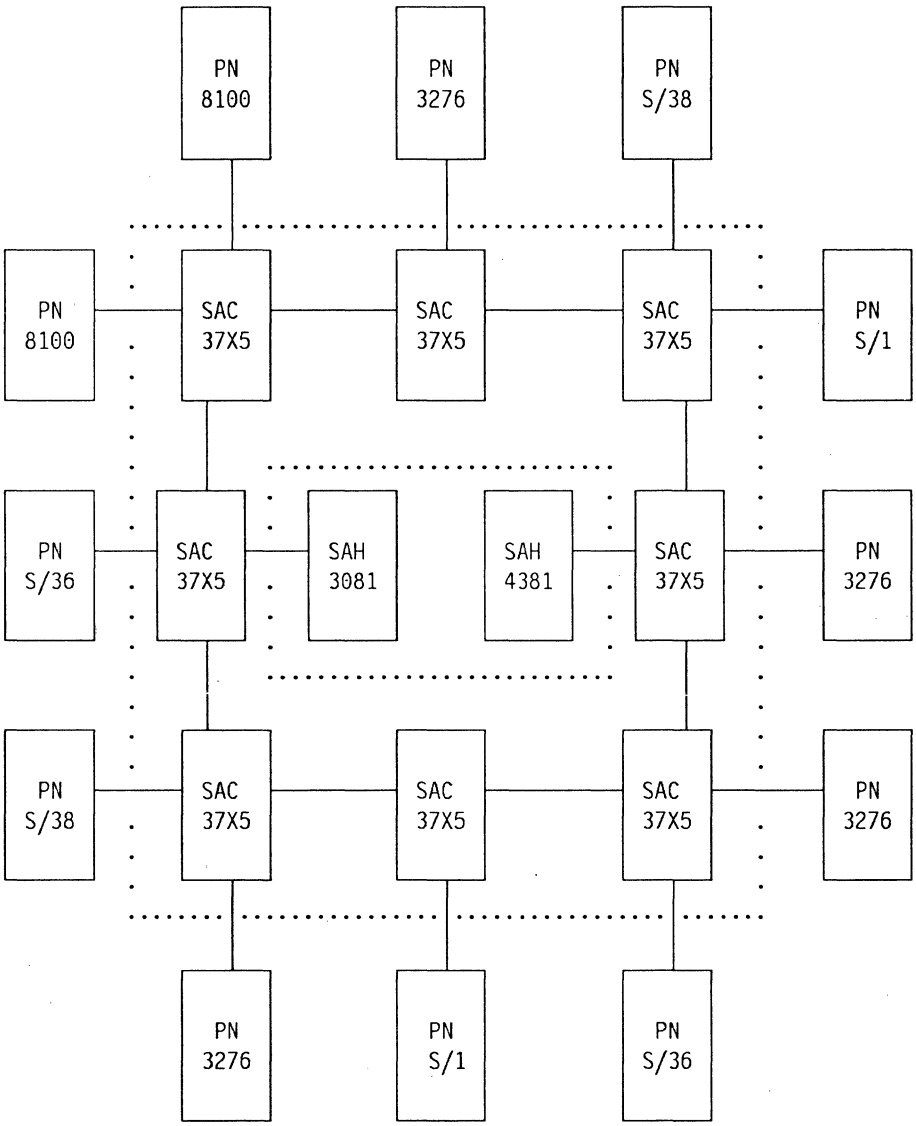


Figure 1-1. SNA Backbone Configuration

Figure 1-1 is the SNA backbone network as we know it today, with the three levels of hierarchy shown outlined by the dotted areas. These levels of hierarchy are the sub-area host nodes (SAH), sub-area communication controller nodes (SAC), and the traditional peripheral nodes (PN) attached through SAC nodes are the entities of a typical SNA backbone network. All resources within this network must be predefined before they can become a part of, or utilize the resources in the network. An inherent characteristic of the SNA backbone network is that a loss of a host that owns most of the resources in the network would severely impact network operations.

Also shown are some of the devices that are commonly used as PN's, SAC's, and SAH's. Now, with the decreasing cost of hardware, primarily memory and computing power, a small system such as the S/36 is capable of using the APPN feature.

APPN Network

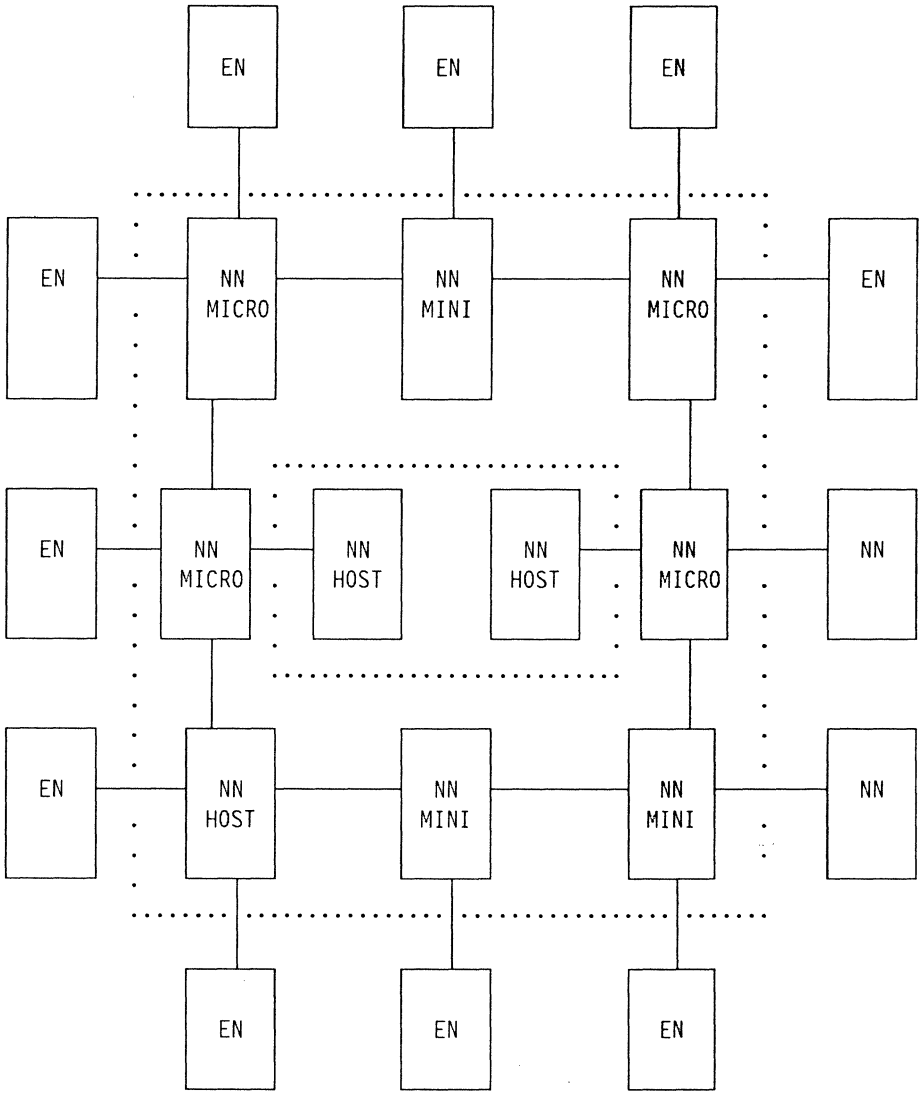


Figure 1-2. APPN Networking Configuration

The S/36 APPN feature exploits LU 6.2 use in a T2.1 node. This is accomplished primarily by providing distributed functions and services to peer connected nodes. In Figure 1-2, I again use the dots to indicate the hierarchical structure of the network. In an APPN network there are only two levels, network nodes (NN) and end nodes (EN). Pre-definition of resources in this environment is not a requirement because the network nodes dynamically reconfigure and define resources as they enter and leave the network. This environment is composed of peer T2.1 nodes and a loss of any node is limited to its locally owned resources and would not have the same effect as the loss of an SSCP in a CMC.

A network node may be defined as one that provides intermediate routing functions, route selection services, and directory services to other nodes. Typically, we build communication networks from the top down. With APPN this philosophy may be reversed by starting with a group of independent small systems. These small systems, if desired, may evolve into a APPN environment and eventually attach to a SNA backbone

network. Giving the small systems user this flexibility, plus other significant enhancements, were primary considerations in the SNA LEN design. We may conclude that the new APPN services allows small systems that were once limited to the role of peripheral nodes to become full network nodes.

Why APPN?

- **Easy to use** - When considering the needs of users of small networking systems, a primary requirement would be a system that is easy to implement and use by this customer set. APPN functionally lends itself to simple implementation, modification, and use by the small systems user.

- **Peer Decentralized Network Control** - Another important consideration is peer decentralized control. By that, we mean that autonomous control is maintained throughout the network and each node may determine when it enters or leaves the network. We do not want to confuse control with network management; it is still desirable to have a centralized network management function for the overall network. However, it is essential in this environment, that the levels of functions provided in any given network match the levels of centralization or decentralization that a customer desires.

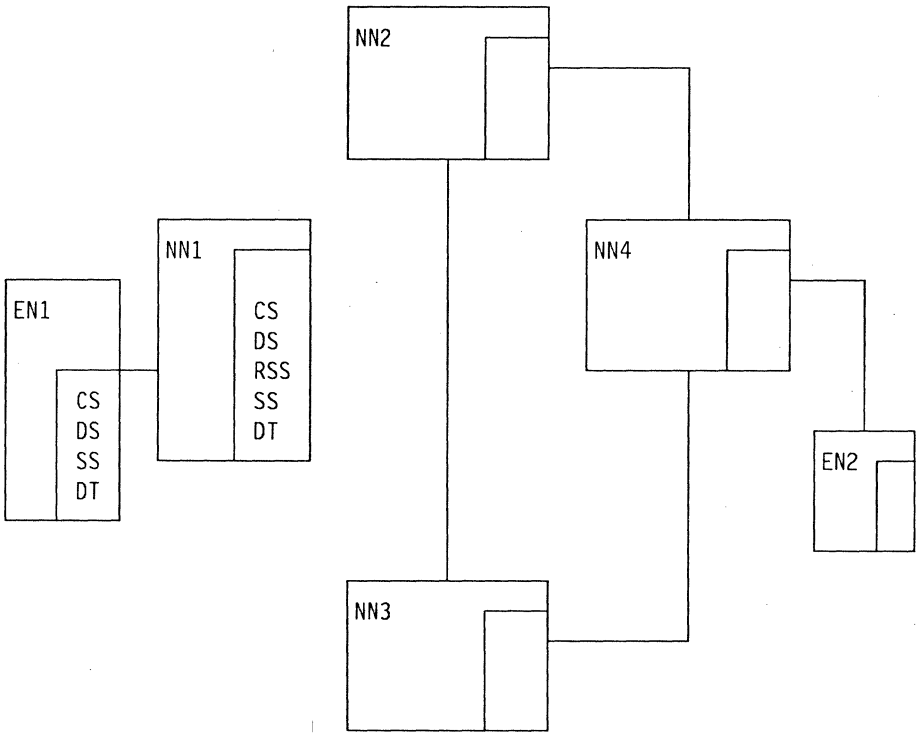
- **Arbitrary Topologies** - Today's telecommunication market has created a requirement for a more flexible architecture that will allow users to build SNA networks of small sizes and at a low cost. These networks must be able to support a variety of connections (LANs, X.25, X.21, Switched, etc.), interconnect with existing SNA networks, and be unrestricted in their growth. In this environment, you would want to be able to configure any topology easily. Having two levels of hierarchy, rather than three, was an important consideration in satisfying this requirement.

- **Continuous Operations** - Finally, an enterprise at some time may find it desirable to commit its business to a network of small systems. This would make reliability as equally a key factor as it is in today's large complex environments. Avoiding the 'single point of failure' which is characteristic of a centralized system, is exploited in this environment and will allow APPN to be as reliable as today's SNA backbone networks.

What Is S/36 APPN?

S/36 APPN is its implementation as a T2.1 node using an enhanced Advanced Program-to-Program Communication (LU 6.2). Every APPN node has within it a control point (CP) that is responsible for managing and coordinating the control functions with other nodes in the network through CP-to-CP sessions. CP-to-CP sessions are created during BIND processing and there may be either one or two CP-to-CP sessions between an adjacent node pair. The CP services associated with adjacent session partners are:

- Connectivity Services (CS) - is responsible for connecting a new link or node into an existing network;
- Directory Services (DS) - maintains a local and remote (cache) directory of network resources;
- Route Selection Services (RSS) - is a NN service which determines the preferred route to a remote resource;
- Session Services (SS) - is responsible for session activation and BIND negotiation;
- Data Transport (DT) - controls the flow of data traffic in the network.



- **Connection Services (CS)**
- **Directory Services (DS)**
- **Route Selection Services (RSS)**
- **Session Services (SS)**
- **Data Transport (DT)**

Figure 1-3. S/36 APPN Services

NN1, as shown in Figure 1-3, is not currently connected into the existing network of NN2, NN3, and NN4. Next, I will utilize each of these services to connect NN1 into the network and bring up a session.

Connection Services

As previously stated, networks must be able to accommodate a variety of connections (X.25, X.21, Switched) and DLC protocols in order to be compatible with the many SNA configurations that are available today. Regardless of the connection or the protocol being used, the first thing that must happen in order for NN1 to become a part of this network, is to activate the link. In Figure 1-4 on page 1-7, we may assume that links A & B between NN1 and NN2/NN3 to be SDLC point-to-point switched lines. Therefore, in our example, NN1 will only need a telephone number to make the physical connection and become part of the network. During this discussion, I will only make reference to NN1 and NN2, you may assume the same exchange will occur between NN1 and NN3.

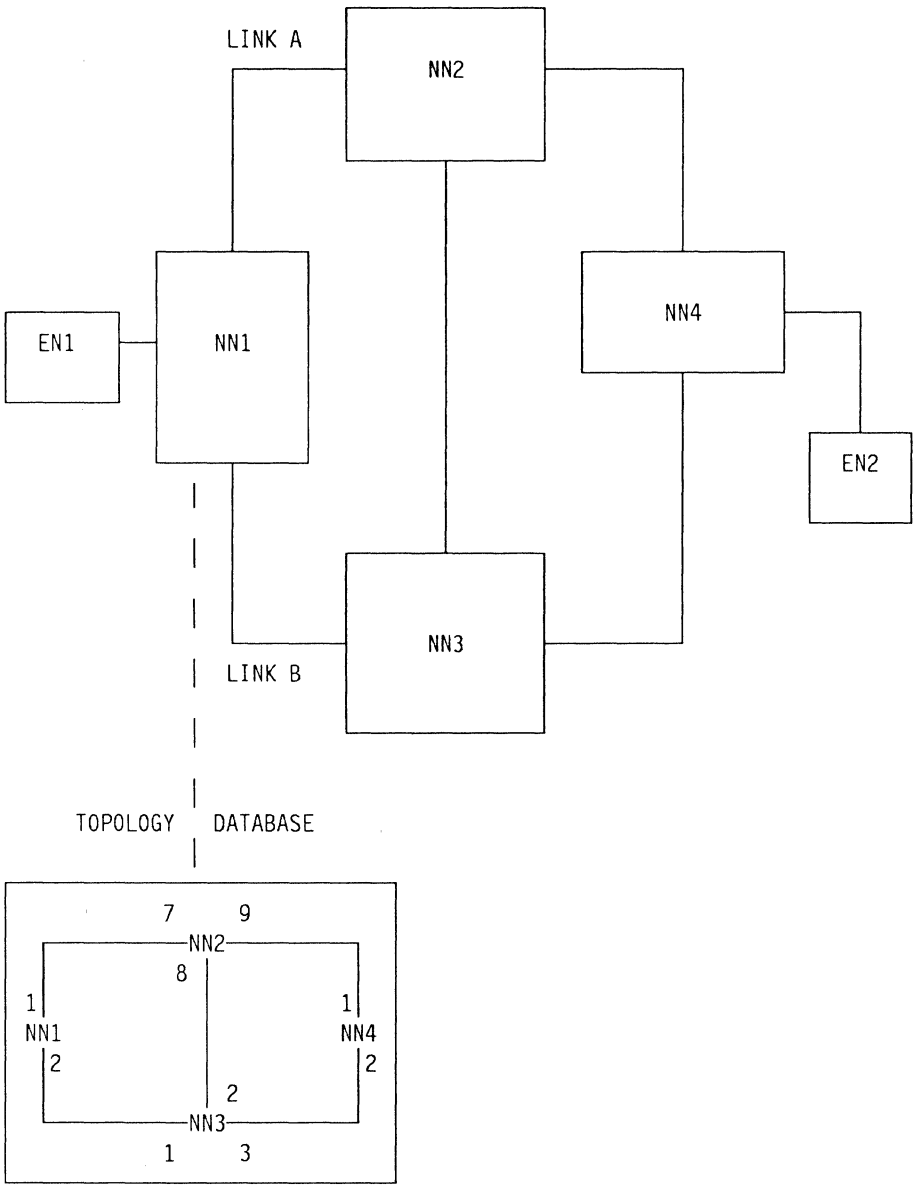


Figure 1-4. Connects The Node Into The Network

Once NN2 has been dialed and the physical connection made, there is an information exchange between NN1 and NN2 via XID 3 commands. These commands will include an information field that describes the node, its transmission characteristics, and whether it supports primary or secondary SDLC stations. The next step is to activate the DLC protocol and a subsequent CP-to-CP session between the two control points in each NN. This CP-to-CP session, an LU 6.2 session, is used by the nodes at this time to update their respective topology databases with the information exchanged in the XID commands. Later, NN1 and NN2 will propagate their updated database over CP-to-CP sessions to all adjacent nodes.

Whenever a NN has a change in its network configuration, the control point is required to store the updated information in its local database and to propagate the change via an update message to all adjacent nodes. This update message, called a resource control vector (RCV), is created by the NN that owns the resource and contains the following information:

- A resource identifier

- A set of resource characteristics
- A resource sequence number (RSN)

Topology Database Update

The topology database (TDB), as shown in Figure 1-4 on page 1-7, is a fully replicated database distributed throughout the APPN network and physically located in each network node. A network node receives a copy of the current database at the time it joins the network and while connected, its local copy will be consistent with the databases of the other NNs.

The topology database update (TDU) function may be regarded as a transaction program using an LU 6.2 CP-to-CP session. One or more RCVs may be encoded in a TDU general datastream (GDS) variable, which is later sent to adjacent network nodes to indicate a change in the network configuration. The GDS variable is the data structure used to communicate between control points on LU 6.2 sessions. Generally, a TDU will contain multiple RCVs and each must be processed separately.

There is always the possibility that the updates could arrive out of sequence because of line speed, route traversed, or some other variable. As the receiver of a TDU message, a network node will check to see if the message received is for a new resource or to indicate a change in status for a known resource before propagating that update to adjacent network nodes.

Directory Services

Now that NN1 has entered the network, let us go through a scenario where LU A in EN1 wants to have a session with LU D in EN2 but LU D's location is unknown to LU A. First, LU A will initiate the directory search function by sending a locate request to NN1 over the CP-to-CP session that exists between itself and NN1 to find LU D. SNA LEN node identification is accomplished by using the NETID.CPNAME. Directory Services is responsible for returning the location (NETID.CPNAME) of the requested logical resource.

In SNA today, the network qualified name is represented by two 8 character names, the network ID and the resource name. We generally use the term 'NETID.NAME' when referring to the network qualified name. Today, the network qualified name is architected to flow on all LU 6.2 flows (e.g., BIND) and all cross domain flows. In a subarea environment, the NETID identifies a complete network (i.e., multiple SSCPs use the same NETID if they are part of the same network).

Each network node is responsible for maintaining a directory of its local resources. An end node served by a network node is a local resource. There is an entry in the directory for every resource (LU) in the network, and the source of that entry is the owner of the resource; i. e., the NN where the resource is physically located. Directory entries are accumulated in one of three ways:

1. Through system definition input.
2. By retaining the results of an earlier successful search (caching).
3. Through run-time registration of EN resources with the server NN. Run-time registration provides the capability for an authorized EN to register its network accessible resources with its serving network node.

In addition to the local resource directory that every T2.1 node maintains, there are three kinds of directory entries in a network node:

- Entries for network node CP domain resources contain the CP name for which the resource is located. For resources other than LUs, the entry will also contain the LU associated with that resource.

- Cross-domain resource entries contain the CP name at which the resource is located and the network node CP that provides directory and session services for that control point.
- Cross-network resource entries contain the network node CP providing services for sessions between origin LU and destination resource.

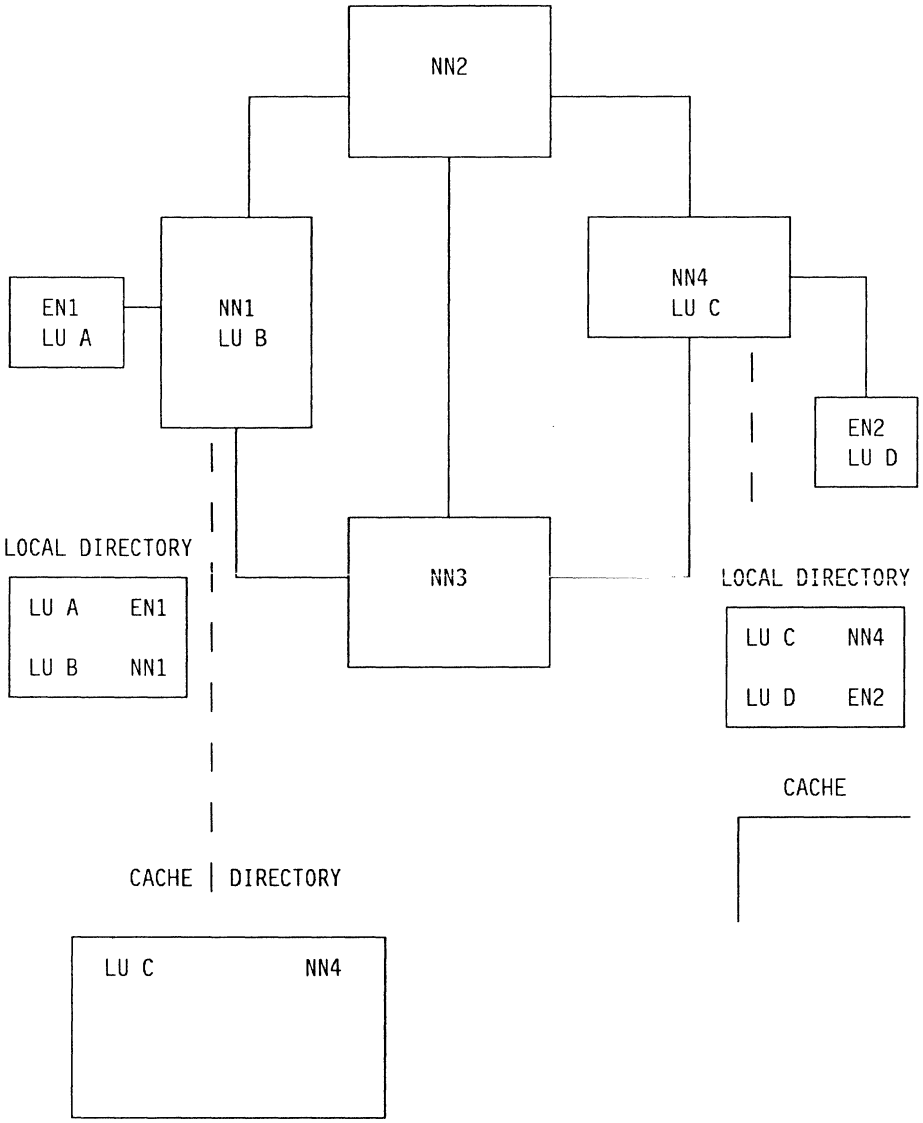


Figure 1-5. Maintains a Local and Cache Directory

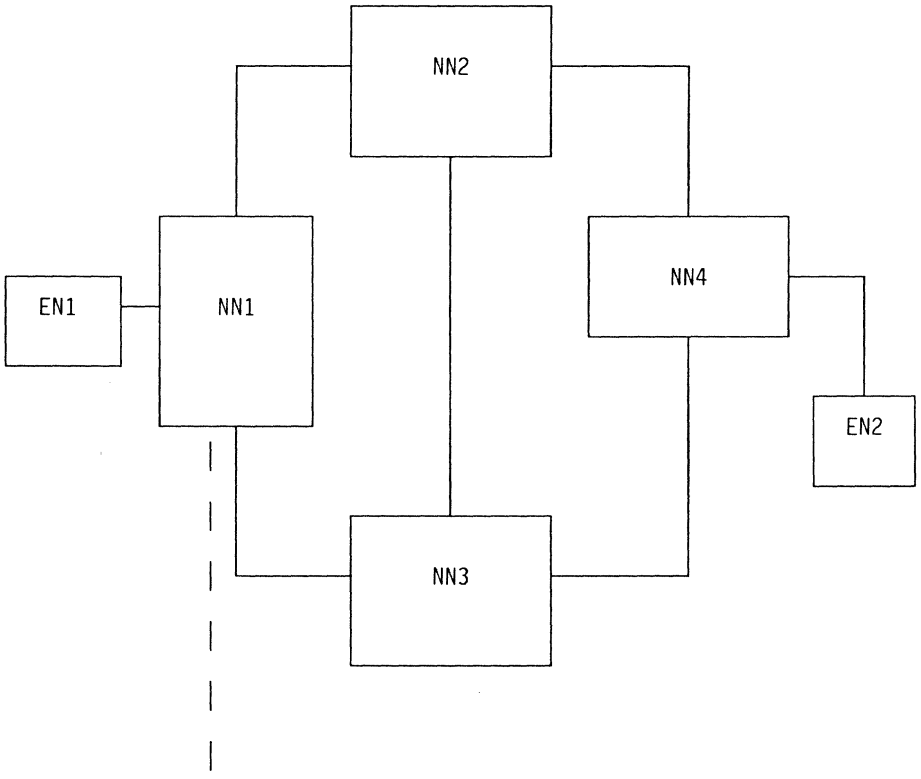
When NN1 receives the locate request from LU A to find LU D, the search order is as follows:

1. Local Directory - LU D could be a local resource.
2. Cache Directory - Was there a previous session with LU D?
3. Issue a network wide 'undirected' search to all adjacent nodes to locate LU D and wait for either a:
   - Positive reply - which indicates resource found

- Negative reply - which indicates resource not found

A negative reply is also generated when two adjacent NNs receive a locate request from a third party and send this same request to one another. This would be the case in the exchange between NN2 and NN3 after having received the locate request from NN1. Once the location, NETID.CPNAME, of the resource has been determined, it is saved in the originating NN's cache and returned to the requester (EN1).

Route Selection Services



ROOTED TREE DATABASE

BATCH	INTERACTIVE
NN1 | NN2 | NN3 | NN4	NN1 | NN2 / \ NN3 NN4

Figure 1-6. The Preferred Paths to Other NNs

At this point, EN1 knows where LU D is located, and the next thing that must happen before an LU-to-LU session can exist between the two is for Route Selection Services to compute the preferred path between the LU partners. Route selection is always done in the NN that is adjacent to the origin EN (NN1 in our

example). A modified version of the shortest path algorithm uses the origin NN, the class of service (COS), and the current information in the topology data base to construct a rooted tree. This tree, whose root is the origin NN, will have branches that will contain every other reachable node in the network for that COS.

The Rooted Tree Database (RTD) identifies the preferred route from the local node to other NNs for the various predefined classes of service. The information from the RTD is then used to build a Route Selection Control Vector (RSCV) which contains a list of nodes and links along the path to the destination. This RSCV is later used during BIND processing.

Session Services/Activation

Once the RSCV has been built, one other important operation must be performed during session activation before the session between LU A and LU D is established. Session connectors must be placed in each intermediate network node along the route.

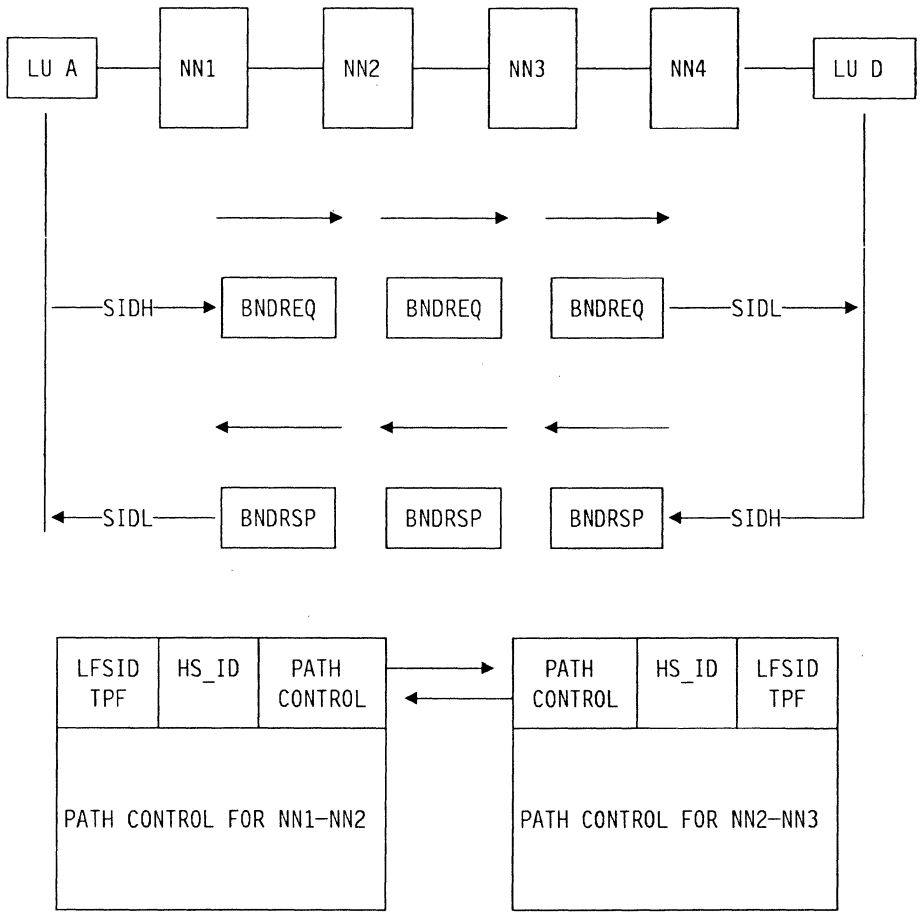


Figure 1-7. Activates Sessions and Data Flow Control

A T2.1 node address is made up of an origin and destination assignment indicator (ODAI) bit plus two 8 bit address fields, session ID high (SIDH) and session ID low (SIDL). As the BIND flows from NN to NN these fields in the FID 2 transmission header (TH) are concatenated to form a 16 bit local form session identifier (LFSID). In the BIND request and all PIUs for the associated session that flow in that direction, the destination address field (DAF) contains the SIDL and the orgin address field (OAF) the SIDH. For the

BIND response and PIU flow in the opposite direction, these fields are reversed as indicated in Figure 1-7, DAF/SIDH and OAF/SIDL.

The BIND continues on from node to node but with new fields (LFSID) in the TH. Transmission priority is specified on a per session basis in the BIND request. As each connection of half sessions along the route (NN1-NN2, NN2-NN3, NN3-NN4) is completed, entries are made in the session connectors and path control element (a T2.1 path control element exists for each transmission group) of each NN. The LFSID values are only valid for the duration of the session and are available for reuse by a new session. Therefore, the relationship between a network addressable unit (NAU) and an address in the TH is of a dynamic nature. The transmission priority field (TPF) of the BIND is stored in the session connectors and half sessions along the session path and does not have to be present in every subsequent PIU related to that session.

LEN-level BINDs exchanged by SNA LEN nodes, either T2.1 or subarea nodes, contain information to provide new features such as adaptive session pacing and session information for network management. All LEN-level nodes support extended BINDs. Regardless of the LU-LU session type, an extended BIND includes:

- Control Vectors and the Control Vector included indicator (CVII).
- The Fully Qualified Procedure Correlation Identifier (FQPCID) - Is used for cross-domain session initiation and termination when the partner SSCP supports network-qualified names.
- The Adaptive Session Pacing Indicator (ASPI) - Adaptive session pacing allows an endpoint to determine the transmit window dynamically and include the window size in the pacing response sent to the partner.
- The Whole BIUs Required Indicator (WBRI).
- The Network-Qualified Name - The network qualified name will be included in the network services name field of the extended BIND if the secondary logical unit and the other SSCPs in the session path support network-qualified names.
- The RSCV is always included in an extended BIND from a T2.1 node when the secondary LU is not adjacent to the origin LU. Peer-to-peer session such as CP-to-CP sessions between adjacent nodes need not have a RSCV in the BIND.
- Class Of Service/Transmission Priority Field (COS/TPF) - The COS/TPF is in the BIND request to specify the route selection and path control parameters used in the T2.1 network.

A BIND without this information is known as a non-extended BIND. Non-extended BINDs are exchanged by all pre-SNA LEN subarea nodes, T2.0 nodes, and when a T2.1 node communicates with a T2.0 node.

Data Transport

Once the session is established, the session connectors and the LFSID are used to route the user's data through the network. The transmission header of each incoming message is examined to determine the LFSID and is used as an index to the session connector for the outgoing link and LFSID.

- **Segmenting** - One of the primary concerns in regards to transporting data within an APPN network is the buffer capacity of adjacent nodes along the route. Remember, we are talking about an environment that could conceivably include a mixture of every device capable of being an NN. Therefore, when a session is established there is a minimum amount of buffer space that needs to be reserved for that session at each node along the path. The maximum response unit (RU) size is negotiated between LUs at BIND time and any RU that exceeds this length is segmented. Other reasons why the sender may segment are:
  - The route characteristics are such that PIUs of certain size are transmitted more efficiently

- Or, the sending of PIUs of a particular length may reduce processing time for one or both LU partners.

- **Reassembly** - There are two types of segment reassemblers:
  - Station reassemblers - reassemble segments without checking what session the segment belongs to and only functions correctly when segments from multiple sessions are not interleaved.
  - Session reassemblers - reassemble using the address in each TH and are able to reassemble segments interleaved from different sessions. Any T2.1 node that supports reassembly must do so on a per session basis so that it can reassemble the various segments interleaved from different sessions.

- **Priorities** - Levels of priority in SNA LEN are the same as those for subarea SNA nodes and are defined during BIND processing. Transmission priority is specified on a per session basis in the BIND request in the Mode/TPF (X'2C') control vector as one of three priorities. The TPF field of the TH will allow a user to assign a low, medium, or high priority to sessions using the same link. A fourth and highest priority, network priority, is also defined. Path control maintains a queue for each priority level and utilizes an aging algorithm to ensure that lower priority messages are not preempted indefinitely.

- **Flow Control** - The flow control mechanism makes sure that no messages are lost and eliminates deadlocks because of buffer capacities being exceeded. SNA LEN uses variable window session pacing on each hop and changing the window size is the responsibility of each receiving node along the session path. Given this control, the receiver is required to pre-allocate buffers that it commits, thus providing freedom from deadlock situations. Flow control within an APPN environment is a very complex subject and will not be further discussed in this overview.

Chapter 2. LU 6.2 Session PD

Advanced Program-to-Program Communication, LU 6.2 (APPC), provides a set of protocols that allow the application program LU to have single, multiple, and/or parallel sessions with an LU partner or partners. After a session is established between LU partners, the conversation is the network resource used for communication between the session partners. Session partners may have multiple conversations active during a session with the conversations being spread out over multiple sessions.

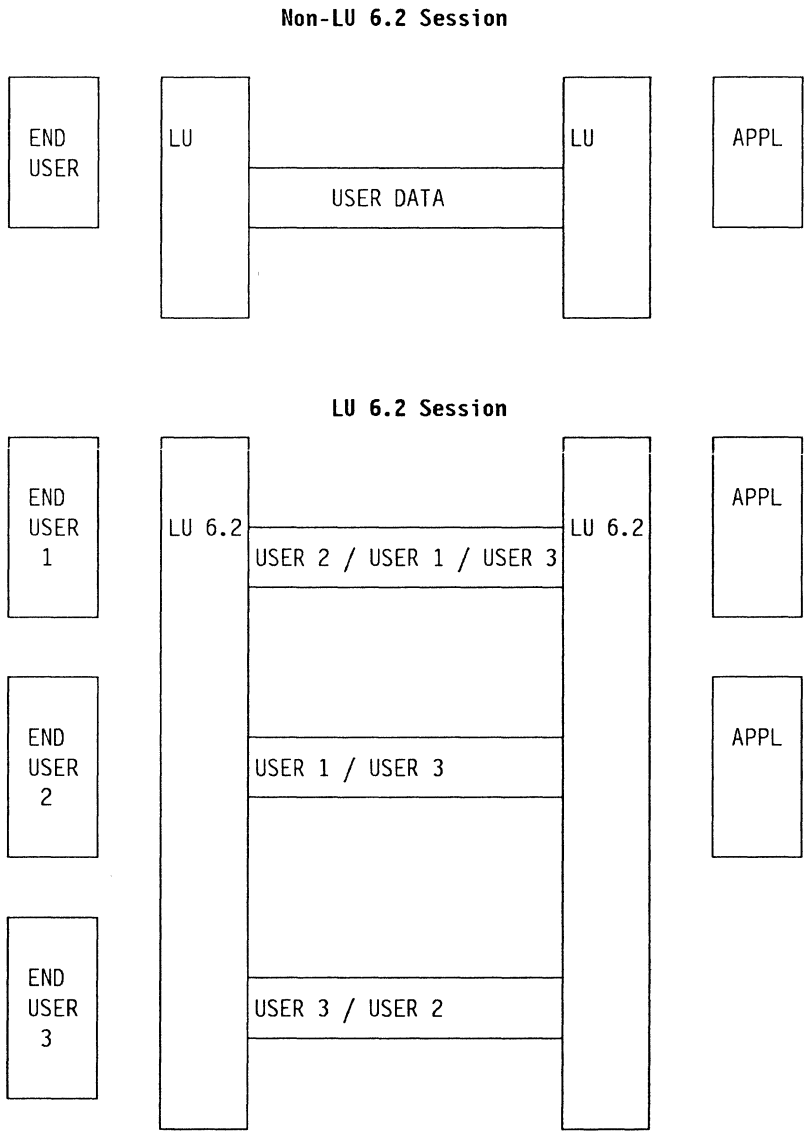


Figure 2-1. Multiple End Users and Sessions

Figure 2-1 illustrates the session/end user relationship in a non-LU 6.2 session and in an LU 6.2 session. In the non-LU 6.2 session, there is always a one-to-one relationship between the end user and session. The network operator need only know the LU name to retrieve session information. Additionally, since a one-

to-one relationship exists between the end user and session, any trace data received for a session belongs to that end user.

For an LU 6.2 session this one-to-one relationship does not exist between the end user and session. An LU may have multiple end users and/or sessions active at any one time. It is no longer sufficient for the end user to identify only the LU name to retrieve session information since it would be impossible to tell which session was used to transport the data or if the data was really sent. In an LU 6.2 environment, the data from multiple end users is multiplexed on different sessions.

LU 6.2 (APPC) requires a new method of performing problem determination when an end user experiences communication problems. A primary concern at this time is the ability to provide the data necessary to perform session problem determination (PD) and problem source identification (PSI) in this environment.

Once again, T2.1 nodes are capable of supporting two types of LUs:

1. **Dependent LU** - A dependent LU receives an activate logical unit (ACTLU) from the host SSCP, can act only as a secondary LU, and can have only one LU-LU session active at a time. Dependent LUs are supported in a SNA LEN environment for migration purposes.
2. **Independent LU** - An independent LU does not receive an ACTLU from the host SSCP, may act as a primary LU or secondary LU, and may have a session limit greater than one.

The fact that an independent LU is able to have multiple and concurrent sessions active, adds complexity to the data gathering process. In this section we will examine some of the innate problems associated with the PD/PSI process using APPC.

PD Data

When we think about collecting problem determination data in this complex environment, we have to first answer the following questions:

- What data is needed?
- Where is it found or kept?
- How do we retrieve it?

Before this is possible, we first have to know what information must flow on the session to allow the operator to determine what data belongs to the end user experiencing the problem. At any point in time, we must be able to provide the end-to-end data necessary to perform session PD/PSI. This data must be accessible and retrievable at the node level or at a focal point host level application such as NetView.

Figure 2-2 on page 2-3 is a chart listing some of the more common failures encountered in today's SNA networking environment and the minimum data needed to perform session PD/PSI for similar failures reported in an APPN environment. These same common failures will require precise information for resolution because of the inherent nature of an LU 6.2 session. Later in this section, we will look at this PD data as it relates to the resources allocated by the LU in this environment (conversations and sessions) and enhancements made to existing service aids to facilitate this process.

P/D DATA	Session set-up failure	Session outage	Active Session problems
Route Selection Control Vector (RSCV)	X	X	X
Fully Qualified Procedure Correlation ID (FQPCID)	X	X	X
RU Image involved in failure	X	X	X
Node ID involved in failure	X	X	X(1)
Sense code associated with the problem	X	X	X(1)
Last 10 PIUs received		X	X
Congestion indicators			X
Session pacing values			X
Session sequence numbers			X
User correlation data	X	X	X

Figure 2-2. Session Protocol Violation

WHERE:

X data is required for this type of problem

(1) If error was a result of a protocol violation

Configurations

Regardless of the network configuration, the two most important characteristics of any PD application are its ability to capture first failure data and to generate an error record of the failure. For the sake of this discussion, we will assume that there is some host application that will collect the PD data generated. Next, we will examine the 3 basic configuration types used in routing a LU 6.2 session in an APPN environment.

Configuration 1

In our first sample network, Figure 2-3 on page 2-4, as in our earlier scenario, the LU 6.2 session is routed entirely within the APPN network. Again, LU A could be in session or attempting a session with some other LU in the network when the failure occurs.

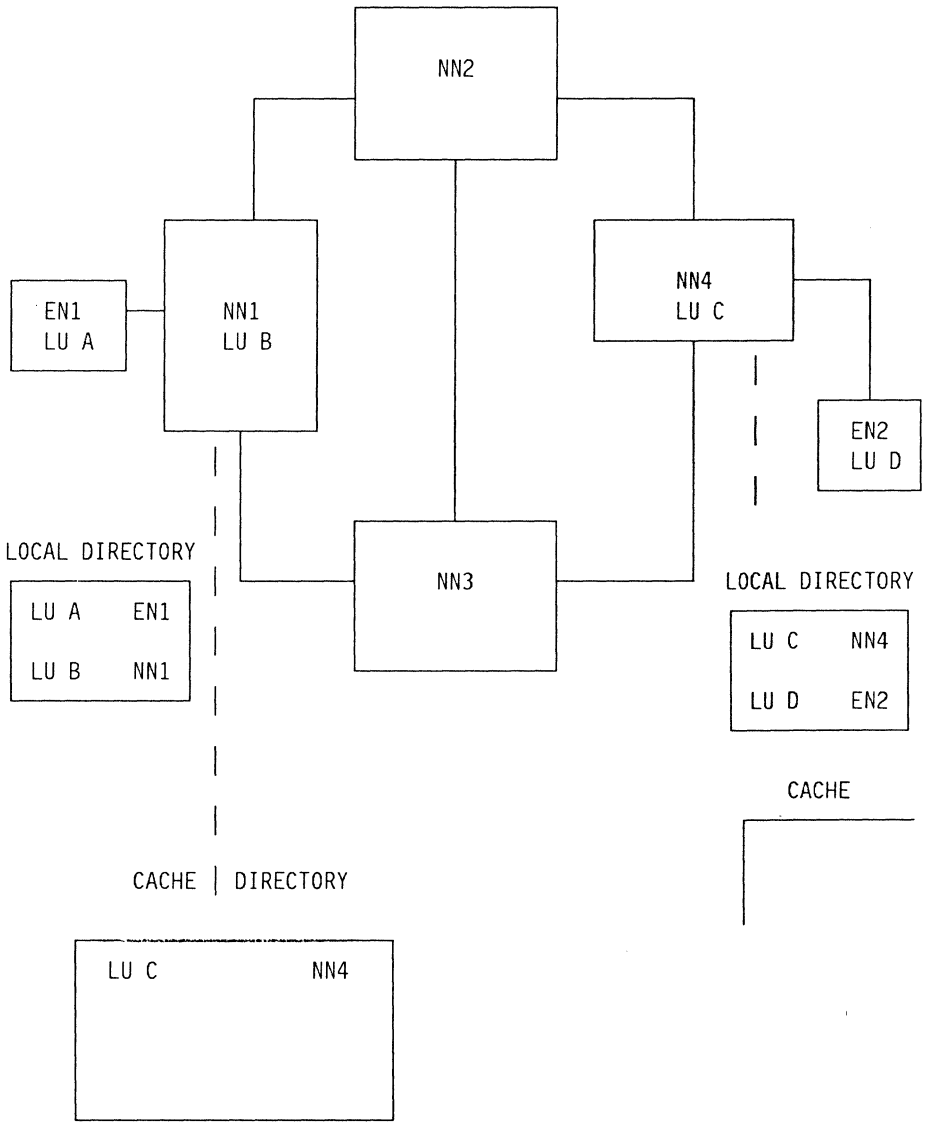


Figure 2-3. Session Routed Within the APPN Network

Configuration 2

In Figure 2-4 on page 2-5, a host application in CEN A or B could be in session with an LU in any end node in the APPN network. Any session failure between LU partners in this configuration would be experienced at the host level and appropriately indicated. The boundary network node (NN C) in this configuration could be designated a collection point for the APPN network, responsible for collecting and forwarding error data of failures associated with the other nodes to a host application such as NetView.

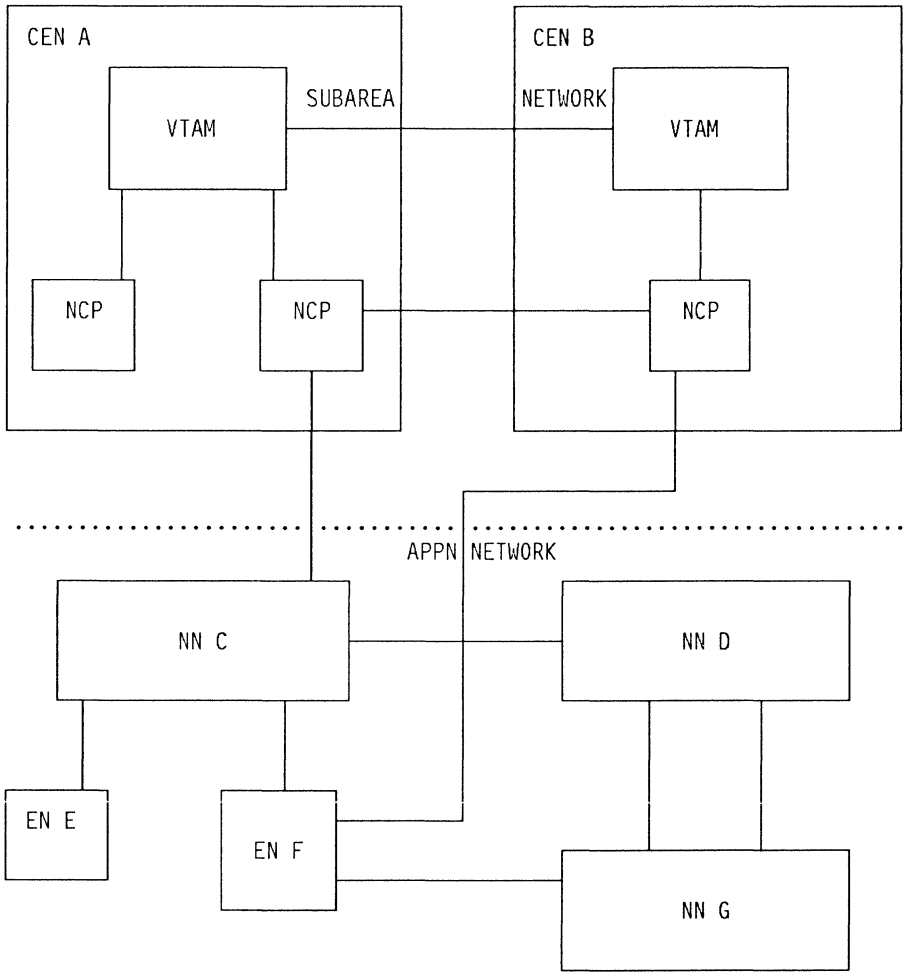


Figure 2-4. Session has an End Point in the Subarea Network

Configuration 3

Our third and final configuration, Figure 2-5 on page 2-6, an LU in APPN network A could be in session with an LU in APPN network B, using the subarea network strictly as a transport network. For any failure in the subarea network associated with that session, the PD data is available at the host. However, for any failure in the APPN networks, the PD data must be available at a boundary node for collection by the subarea network for analysis. To the subarea, the session path terminates at the boundary node and it may or may not know about the session partners in APPN networks A and B.

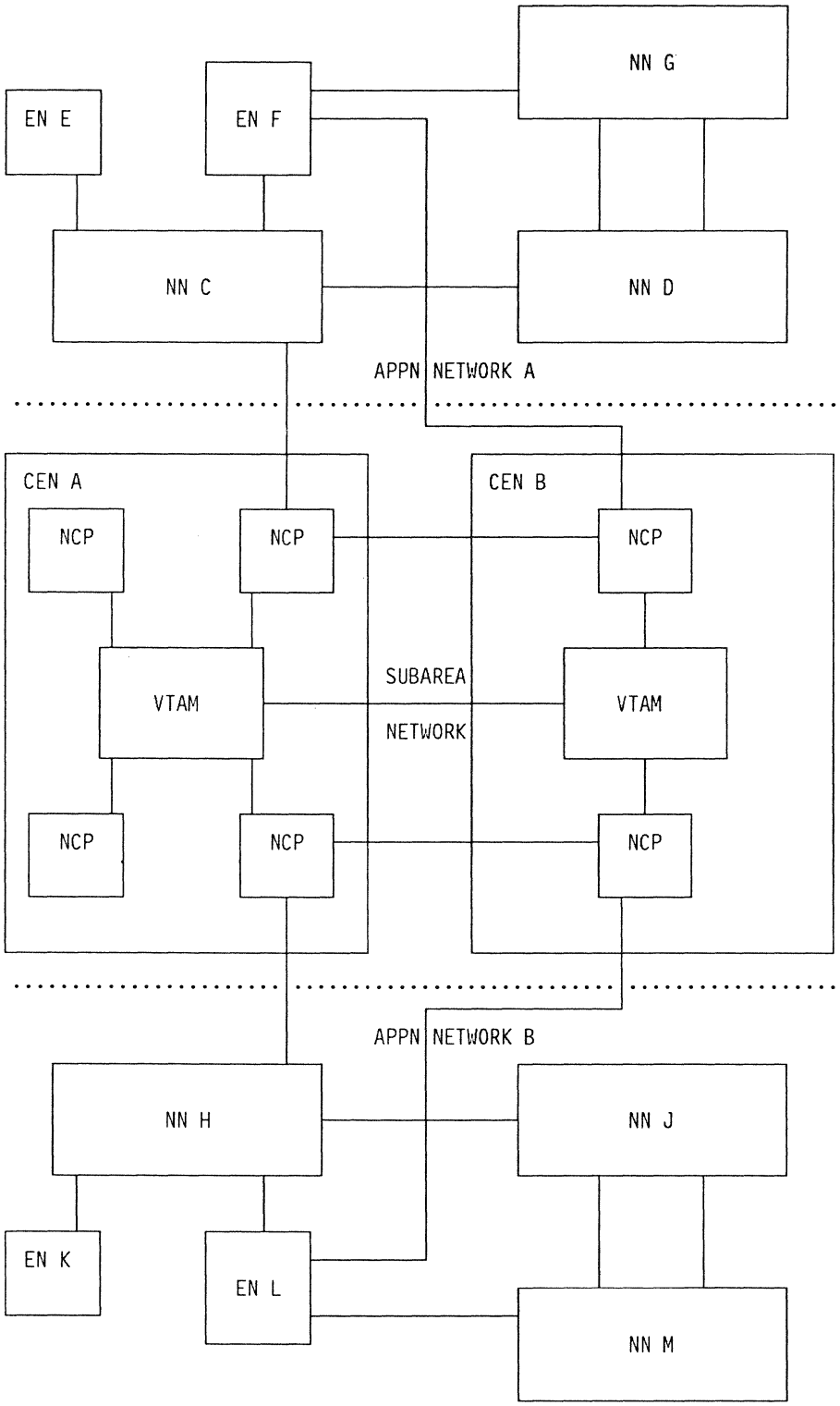


Figure 2-5. Session Traverses Subarea Network

The previous three figures illustrate the connectivity and session paths in SNA LEN. An LU may have multiple sessions active for any number of end users at any one time. The network resources assigned to the end users are conversations and sessions. The LU dynamically assigns conversations to available sessions,

creating and terminating sessions as required. If a session is not available, the LU may queue the conversation until a session becomes available. When there is a problem in the network, the network operator may not know which conversation or session was used by the end user. In order to make this determination, the following data is required for an LU 6.2 session when performing session PD; Relational end-user data, Conversation PD data, and Session PD data. We will now take a closer look at the data and the information included.

Relational End-User Data

- MAPS END-USER TO CONVERSATION
- ONE-TO-ONE MAPPING DOES NOT EXIST
- REQUIRED IN LU 6.2 ENVIRONMENT

PD DATA	Sent by the Network Operator	Supported by the Node	Returned by the Node
Physical ID	Optional	Optional	Required if supported
User Logon ID	Optional	Optional	Required if supported
Application ID	Optional	Optional	Required if supported
LU Name	Optional	Required	Required
LUW ID	Optional	Required	Required
Conversation Correlator	Not Sent	Not Sent	Required

Figure 2-6. Relational End User Data

Relational end-user data maps the nodal end user identities to the conversations actually used by the end users. This mapping between end user and conversation need not be a one-to-one mapping. The purpose of this data is to allow the network operator to find all conversations associated with the nodal end users (human, application, or work station). The end users may be identified by either the physical location, logon ID, application name, or any combination thereof. This information will enable the network operator to find out which network resources are actually being used. Relational end user data is required in an LU 6.2 environment for session PD and is:

- Available to the end user
- Maintained by the T2.1 node so that a request to the control point will initiate the retrieval of the following associated data.
  - LU name

- LU of Work ID (LUW ID) - Identifies all LUs involved in a single transactions. The LUW ID can be used to determine the status of all conversations in all LUs involved with the transaction
- Conversation Correlator - Identifies a unique conversation
- Active status - Identifies an active or recently active conversation.

Since each node keeps a "wrap-around" log of all conversations, this log may be used to access historic session or conversation PD data residing within the node or elsewhere in the network. Every node that supports multiple end users also provides a method for identifying an individual end user. Retrieval of this data is done through the use of certain optional or required keys supplied by the network operator or the node. Figure 2-6 on page 2-7 is a chart showing this relational end-user data and its source.

All LU 6.2 nodes must support relational end-user data, with only one exception. The node that always maps a single end-user to a single LU and the LU maps to a single session does not have to support it.

Conversation PD Data

- DEFINES THE CONVERSATION
- USED TO ACCESS THE SESSION DATA
- REQUIRED IN LU 6.2 ENVIRONMENT

Conversation PD data is also required in a LU 6.2 environment and is used to determine if the end user's problem is a problem with a conversation. This data defines the conversation, provides its current status within the LU, and identifies the session used by the conversation. The following conversation PD data is kept by the origin LU:

1. Conversation correlator, used as a key to retrieve the rest of the data
2. Fully qualified name of the LU partner
3. Origin Transaction Program Name
4. Destination Transaction Program Name
5. Logical Unit of Work ID
6. Fully Qualified Procedure Correlator Identifier (FQPCID)
7. Conversation Status
   - Waiting for a free session
   - Conversation bound to a session
   - Waiting to inform the transaction program about termination of a conversation
   - Conversation terminated with/without an error
8. Termination sense data, if the conversation terminated with an error.

The destination LU is not required to keep the same data as the origin LU. In addition to its partner LU name, it must also keep items 1, 2, 6, and 7 listed previously.

If a conversation ended abnormally, the PD data is retrievable from the wrap-around log. The node is not required to log conversations that terminated successfully.

The new FQPCID has been enhanced and is a useful aid when diagnosing session related problems. It is included in the extended BIND request/response and may be displayed at any node that participates in the session. In the enhanced format, the FQPCID is a globally unique identifier for a session and is used for:

- Correlating session services between CPs
- Identifying and resolving unbind problems (cleanup)
- Identifying a session for accounting, auditing, and performance monitoring purposes
- Identifying a session for problem determination, resolution, and the key to accessing session PD data.

Session PD Data

- REQUIRED IN T2.1 NETWORKS
- INDEPENDENT FROM RELATIONAL AND CONVERSATION

Session PD data is required for PD on sessions in a T2.1 node and is used to determine if the problem being reported by the end user is a problem with the session. If a T2.1 node can determine that it has a session problem and the FQPCID is available, session PD is possible. However, if the problem is not with the session itself and is internal to the node or LU, then this data would not be useful in resolving the end user problem.

Session PD data will allow the network operator to determine the status of a session and determine which node is most likely the cause of the problem. The following session PD data is required in a T2.1 node:

1. Full session path (RSCV) - Identifies all nodes and transmission groups from the origin to the destination. This information may be examined to determine the status of the session in all nodes in the path and if any alerts were received from any of the nodes.
2. First 12 bytes of the last 10 PIUs flowing on the session - Includes the Transmission Header, Request Header, and the first 3 bytes of the the Request/Response Unit.
3. BIND image, sent or received - The BIND image sent is saved until a BIND image is received, then that is saved. It may be used to determine if there are conflicts between the negotiated BIND and the application requirements.
4. Extended Sense Data - Contains the sense data and the node that generated the data.
5. Session pacing data from each node in the path - Consists of the window size and the residual count in both directions and is helpful when trying to identify congestion points in a network.
6. The last two PIU in and out sequence numbers from each end of the session path along with a normal or expedited flow indicator.

Cascaded Transactions

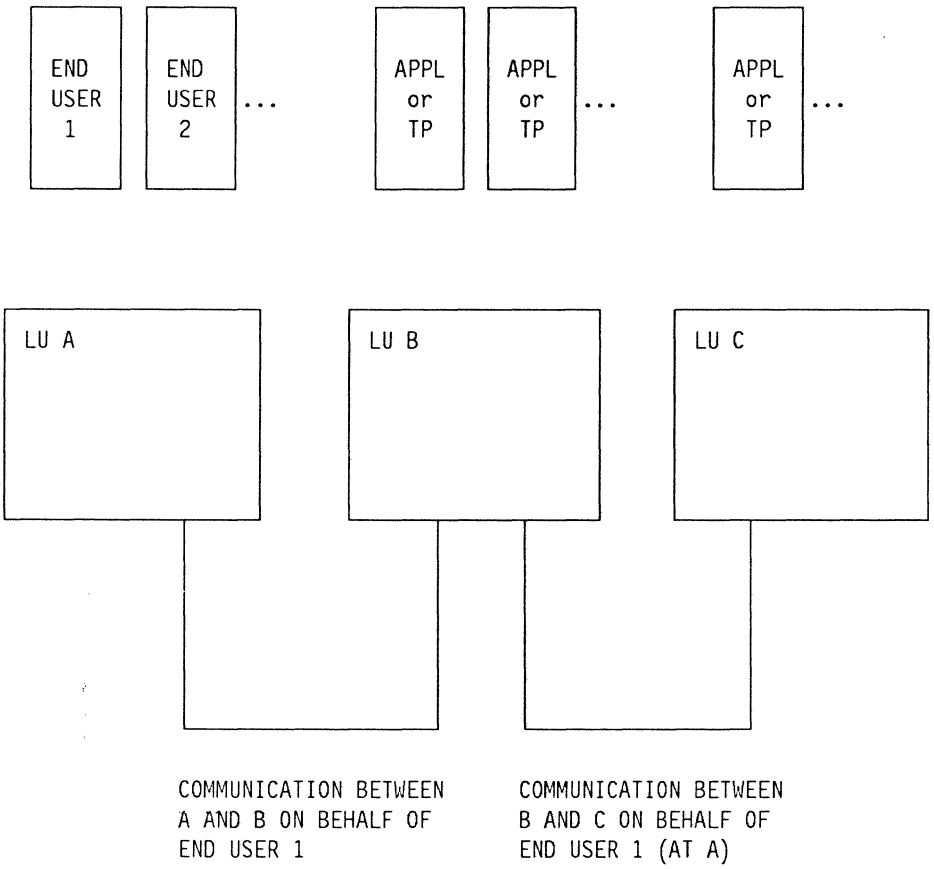


Figure 2-7. Problem Determination Data is Needed

An entirely different problem is presented in a cascaded transaction environment shown in Figure 2-7. Here you may have an end user in LU A communicating with a transaction program (TP) in LU B who in turn communicates with an APPL or another TP in LU C on behalf of the end user in LU A. Suppose the nodes where LU B and LU C resides are unattended, and a problem occur on the session between LU B and LU C affecting the end user in LU A. The network operator must be able to request session PD data from both LU B and LU C related to the problem experienced by the end user in LU A.

An application in an unattended node may or may not detect that a communication problem exists. In this case, the transaction in the unattended node is an intermediate transaction and the problem will most likely be reported by the initiator of the transaction. The network operator would first have to retrieve the relational end user data from the end user's node, e.g. partner LU name, LUW ID, and conversation correlator. The network operator may then use this data to retrieve the corresponding conversation and session PD data for the first leg of the transaction. By using the LUW ID as a key, PD data may be obtained from the other related sessions.

Diagnostic Aids

A number of our existing diagnostic aids have been enhanced to provide support in this environment. Of particular importance, are the enhancements made to:

- VTAM Internal Trace (VIT)
- Session Information Retrieval (SIR)
- Generalized PIU Trace (GPT)
- Trace Analysis Program (ACF/TAP)
- NetView
- S/36 Trace Utility

The major enhancements made to each of these diagnostic aids are are noted in this bulletin but without any great detail.

VTAM Internal Trace (VIT)

The VTAM Internal Trace provides a record of events within VTAM and it is used to diagnose and fix a vast majority of VTAM problems today. Specific APPC enhancements for the VIT include three new trace entries for Message Units (MU1, MU2, and MU3). Message units are used to pass information between APPC components, nodes, or to trigger some event. These entries will contain pertinent information such as the session limit and conversation ID that is associated with an LU-to-LU session.

Other VIT enhancements include the streamlining of the CPWAIT/CPPOST/ CPPURGE entries into two groups, primitive VTAM interface (PVI) and non-PVI. The recording of CCO/CC2 for CPWAIT and CCI/CC2 for CPPURGE/CPPOST will be stopped. The CPWAIT/CPPOST/CPPURGE format has been modified to include more information, some of which was in the CCO/CC2 and CCI/CC2

Session Information Retrieval (SIR)

SIR or session trace is the retrieval of session related data from associated session control blocks. SIR enhancements to satisfy LU 6.2 requirements are two fold:

1. Independent LU Support - Enhancements have been added to use the network management vector transport (NMVT) for the retrieval of boundary session data. The boundary function of SIR has been changed to identify the session (not just the network addressable unit) for which the data is to be retrieved. Included in this data are the sequence numbers of the last ten PIUs that traveled on that session. For a boundary session, the last two outbound and inbound sequence numbers are recorded plus, an indication of whether any of the PIUs were expedited. This sequence number recording is done continuously regardless of whether SIR is enabled or not; all sequence numbers for session activation and deactivation are recorded. If SIR is enabled the data will be forwarded to the Session Monitor function of NetView. Note however, if SIR is not enabled, the data cannot be retrieved.
2. Adaptive Session Pacing Support - The current method of session pacing uses a fixed window which is determined at BIND. A T2.1 node may have limited buffer resources and requires more control over the buffer resources than the fixed window mechanism allows. Adaptive session pacing allows the transmit window to be varied dynamically and to include the window size in the pacing response sent to the partner. SIR enhancements include additional control block data which relates to this new function.

Generalized PIU Trace (GPT)

GPT traces FID4 PIUs inbound from or outbound to a specific resource as they are processed in the NCP. GPT can be activated on a resource basis by link: for all PUs and LUs on that link, by PU: for all LUs on that PU, and by LU. Thus, when NCP receives a command to activate GPT for a resource, the resource is represented by a single network address. For a non LU 6.2, session there is always a one-to-one relationship between LU name and LU address and GPT activation cause all PIUs which flowed to and from that LU to be traced.

Because of the support for T2.1 nodes and multiple sessions, if GPT is activated for an independent LU, all PIUs flowing to and from that LU will be traced for the multiple LU-LU sessions it is capable of having.

GPT will continue to be a reliable service aid with which to record session traffic but will generate a sizeable amount of trace data when used in this environment. The trace analysis program is used later to edit and print this data.

ACF/TAP provides a method for sorting through the output and selecting the desired conversation/session for an LU pair. Although this will increase the amount of data traced at the NCP, it decreases the processing overhead on each PIU at the NCP and shifts the extra processing to ACF/TAP where it is done once for each trace record.

Trace Analysis Program (ACF/TAP)

When using ACF/TAP to print GPT data from this environment you must make two successive runs of ACF/TAP. Two new commands have been added, the first command creates a conversation/session index and the second command will allow a user to selectively print PIUs for a specific conversation/session.

The IXPRT = YES command will create and print the session/conversation index. It causes ACF/TAP to read the trace dataset and create another dataset consisting of:

- New status records for 'dummy BINDs' which ACF/TAP uses to create the session index. When GPT is active, the NCP creates a 'dummy BIND' to denote the successful initiation of an LU-LU session. This means that there will be no entry in the TAP index for unsuccessfully initiated LU-LU sessions. However, the BIND and the negative response BIND or the UNBIND will be captured and appear in the trace.
- The function management header 5 (FMH5) control block may contain up to 255 bytes of conversational PD data for an LU 6.2 session. The FMH5 entries will also have an associated record number that is used in selecting the conversations to be printed. The FMH5 contains the following PD data; the transaction program name, the fully qualified LU name, the logical unit of work ID, and the conversation correlator (if present). It is important to note when GPT is activated for a specific PU, PIUs flowing to and from subordinate LUs for that PU will be captured, but only 40 bytes of the FMH5 will be traced.

After viewing this newly created index, the user may issue the INDEX command for an OAF/DAF pair to print the selected PIUs. The INDEX command will also allow the user to specify multiple address pairs starting with a record number.

Finally, it is important to note again that the TAP session index will only be created for successfully initiated LU-LU sessions and the full FMH5 PIU is captured when GPT was activated for a specific LU.

NetView

In the overall scheme of network management, NetView is the host application designed to collect and process alerts from network resources. It may be thought of as the Focal Point, a concept that allows a customer to centralize the management of the network. NetView provides the following to assist in the problem determination process:

- Command Facility - Provides the ability to enter VTAM commands from the operator terminal to control network resources.
- Hardware Monitor - Provides information about the physical network and system resources
- Session Monitor - Provides information about the logical network resources
- Status Monitor - Displays the status of network resources
- Help Facility - Online help for commands and CLISTs.

S/36 Trace Utility

The S/36 trace utility should prove to be a useful diagnostic aid in the APPN environment. It provides a method of capturing selected events for APPN/APPC related problems. The 32-byte trace entries, containing register and parameter list information, are logged into a variable sized wraparound table in the S/36 main storage. There are five major trace entry types that may be traced in the S/36, they are:

- SVC Main Storage SVC Requests
- CSP Control Storage Event Requests
- XRF Program transfer/Transient Requests
- LD Program Load Requests
- LOG Log Trace Requests.

The APPC/APPN traceable events are Log Trace Requests (LOG). These events are traced whenever one of the system components issues the log trace SVC and that function ID was selected to be traced. There are six major traceable function IDs for APPC/APPN with each having multiple trace entries. The S/36 Program Problem Diagnosis and Diagnostic Aid (SY21-0593-5) contains invaluable reference material in the form of templates that are needed to analyze the individual trace records. The major function IDs associated with APPC/APPN events are:

- SSP-ICF APPC Subsystem
- APPN CP Manager Trace
- APPN CP Send Trace
- APPN CP Target Trace
- APPN CP Route Services Trace
- APPN CP Directory Services Trace.

The S/36 trace utility may be invoked from any display on the system (interactive mode) or from the job queue (batch mode). When invoked interactively, a series of displays allow the user to select the options, profiles, and function IDs to be traced. The A/S400 System has an additional trace facility, the Communication Trace, which is similar to the VTAM Buffer Trace and is capable of capturing the entire PIU.

Summary

SNA Low Entry Networking is an extension to the SNA architecture that will support the S/36 and AS/400 Advance Peer-to-Peer Networking feature. Once again I want to emphasize that the APPN feature is a product implementation and not an SNA LEN Architecture implementation. Throughout this bulletin, I have tried to focus on the new services introduced in an APPN environment. These new networking services are distributed across peers, will allow for the dynamic definition of network resources, and will eventually expand the network user base down to the PC level. Another very important consideration is that the APPN feature is compatible with the multiple SNA protocols and configurations of today.

Advanced Peer-to-Peer Networking will add a new and challenging dimension to the SNA environment. End-users in this environment will have the flexibility to enter and leave the network as desired and have multiple session partners. Serviceability in this arena will be challenging and require a new approach to gathering the PD data needed to isolate problems down to the end-user level.

As stated at the beginning of this bulletin, my intent was to present a high level overview of APPN with enough depth for a basic understanding by those readers with previous SNA knowledge.

Appendix A. READER'S COMMENTS

An Overview

Advanced Peer-to-Peer Networking

NSD Systems Center - Washington Technical Bulletin

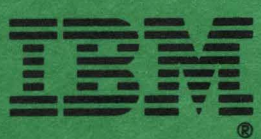
IBM may use or distribute any information you supply in any way it believes appropriate without incurring any obligation whatever. You may of course continue to use the information you supply.

Please state your occupation:_____

Comments:

Please mail to:
Robert J. Hines
IBM Corporation
Building 2 Dept. 76X
18100 Frederick Pike
Gaithersburg, Maryland 20879

GC38-7026-00
Printed in U.S.A.
DAPS Code 0895



GC38-7026-00